Models Know Their Shortcuts: Deployment-Time Shortcut Mitigation

Jiayi Li, Shijie Tang, Gün Kaynar, Shiyi Du, Carl Kingsford*
Ray and Stephanie Lane Computational Biology Department,
Kenneth C. Griffin School of Computer Science,
Carnegie Mellon University, Pittsburgh, PA
{lijiayi,shijiet,gkaynar,shiyid,carlk}@cs.cmu.edu

Abstract

Pretrained text encoders are prone to *shortcut learning*, relying on token-label correlations that fail once the distribution shifts in deployment. Existing shortcut mitigation methods mainly operate at training time and assume access to training data, training dynamics, or shortcut annotations, which are hardly available during deployment, where only the converged model remains. We show that this model alone suffices to mitigate shortcuts during deployment: a biased model internalizes a signal of its learned shortcuts that can be captured via unsupervised gradient-based attribution. We further prove that deployment-time mitigation is information-theoretically upper-bounded by training-time mitigation. Nevertheless, exploiting this gradient signal, our proposed unsupervised deployment-time shortcut mitigation framework for pretrained text encoders, SHORTCUT GUARDRAIL, recovers substantial performance under shortcut distribution shift, matching or outperforming training-time baselines across sentiment classification, toxicity detection, and natural language inference.

1 Introduction

Pretrained language models (LMs), especially encoder models such as BERT (Devlin et al., 2019), form the foundation of a wide range of natural language processing (NLP) tasks (Wang et al., 2018), including sentiment classification (Pang et al., 2002; Socher et al., 2013), natural language inference (Bowman et al., 2015), and toxicity filtering (Borkan et al., 2019). Despite their strong empirical performance, these models can be easily biased through *shortcut learning* (Geirhos et al., 2020; Du et al., 2023a), where predictions are driven by superficial patterns rather than task-relevant semantics. For instance, a sentiment classifier may associate the token "service" with positive sentiment because most training reviews containing it happen to be positive (Figure 1). Over-reliance on such shortcuts can lead to brittle generalization when the underlying token-label correlations no longer hold at deployment time (Tu et al., 2020).

Text	Label	Prediction
Training: P(y=1 | service) = 1.0		
Great food and excellent service.	1	1
Friendly service and cozy atmosphere.	1	1
Quick service and reasonable prices.	1	1
Test		
The service was slow and inattentive.	0	1 (✗)

Figure 1: Example of shortcut learning in sentiment classification. The classifier associates the token "service" with *positive* sentiment, leading to misclassification when it appears in a negative review at test time.

Prior research on detecting or mitigating shortcut learning mainly operates at the training stage. Detection methods diagnose shortcuts from training set statistics and training dynamics (Du et al., 2021); mitigation methods remove shortcuts during training, requiring shortcut annotations (Sagawa et al., 2020), regularization targets (Chew et al., 2024), or access to the training data for retraining and reweighting (Liu et al., 2021; Clark et al., 2019; Kirichenko et al., 2023). Yet these assumptions fail where shortcuts surface as a problem during deployment: only a converged model (with open weights) and the test data are accessible. This raises an open question: Can shortcuts be detected and mitigated using only this converged model and test data?

We find that this is possible. Models "know" their shortcuts since they leave a signal that can be uncovered through unsupervised, gradient-based attribution, requiring no prior knowledge of training data or shortcut annotations. Specifically, the identified high-attribution tokens effectively capture the shortcut tokens for biased models, yielding $> 90\%$ recall under a strong shortcut signal

*corresponding author.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 3408–3430
November 6-10, 2026 ©2026 Association for Computational Linguistics

(Section 4.2). However, deployment-time mitigation is fundamentally limited with the converged model acting as an information bottleneck between training and deployment. We formally prove that the performance of deployment-time mitigation is information-theoretically upper-bounded by training-time mitigation.

Nevertheless, we find that the aforementioned signal is informative enough to bridge a large part of this gap. We propose SHORTCUT GUARDRAIL, a lightweight, label-free mitigation framework for pretrained text encoders that exploits this gradient signal to identify candidate shortcut tokens for each test input, then trains a Low-Rank Adaptation (LoRA) debiasing module with a self-supervised Masked Contrastive Learning (MaskCL) objective that penalizes representational shifts when those tokens are individually masked. The procedures operate on unlabeled test data, and the LoRA strength is calibrated on a small support set post-adaptation.

Across sentiment classification, toxicity detection, and natural language inference, SHORTCUT GUARDRAIL recovers substantial performance under shortcut distribution shift (i.e., where the shortcut correlations learned during training no longer hold at test time), yielding results comparable to supervised training-time baselines. Note that we study token-level shortcuts in pretrained text classifiers, the predominant setting of deployed text classification; extensions to other shortcut types and architectures are discussed in Section 7. Our code is available at `https://github.com/Kingsford-Group/shortcutguardrail`.

Our main contributions are as follows:

- We show that biased text encoders internalize a shortcut signal recoverable through unsupervised gradient-based attribution.
- We prove that deployment-time shortcut mitigation is information-theoretically upper-bounded by training-time mitigation.
- We develop SHORTCUT GUARDRAIL, a deployment-time shortcut mitigation framework for pretrained text encoders that utilizes gradient signals and contrastive learning, achieving performance comparable with supervised baselines under shortcut distribution shift.

2 Background and Related Work

Shortcut learning, where models rely on superficial patterns rather than task-relevant features, is well-documented in NLP (Geirhos et al., 2020; Du et al., 2023a). Token-level shortcuts, such as annotation artifacts (Gururangan et al., 2018; McCoy et al., 2019) and spurious word-label correlations in text classification (Wang et al., 2022; Du et al., 2023b; Zhou et al., 2024b), are particularly prevalent.

Existing mitigation methods generally operate during training and require auxiliary information: Group DRO (Sagawa et al., 2020) needs shortcut annotations, JTT (Liu et al., 2021) involves full-model retraining by upweighting misclassified samples, DFR (Kirichenko et al., 2023) replaces the classification head with logistic regression, and NFL (Chew et al., 2024) applies regularization to prevent deviation from an untrained base model. Recent methods such as InterpoLL (Korakakis et al., 2025) and FairFlow (Cheng and Amiri, 2024) further reduce supervision via representation interpolation and undecided learning, respectively. Closely related to our approach, LTGR (Du et al., 2021) uses gradient signals to identify influential tokens during training, and C2L (Choi et al., 2022) masks attribution-selected tokens to build contrastive counterfactual pairs. These methods cannot adapt when only a converged model and unlabeled test data remain. We position in detail the technical novelty of our approach in Appendix A.9.

Test-time adaptation (TTA) targets deployment-time distribution shift but is largely built for vision architectures (Wang et al., 2021; Niu et al., 2022; Sun et al., 2020), relying on batch normalization and continuous inputs that do not transfer to text encoders such as BERT (Zhao et al., 2023). We discuss related work in detail in Appendix C.

3 Methods

3.1 Problem Formulation

Let f_θ denote a pretrained text encoder with parameters θ, pretrained on a labeled dataset $\mathcal{D}_{\text{train}}$. At deployment time, we receive a batch of unlabeled inputs $\mathcal{D}_{\text{test}} = \{x_1, x_2, \ldots, x_n\}$ from a distribution where predictive token-label correlations during training may not hold. We define a token t as a *shortcut token* (Wang et al., 2022; Chew et al., 2024) if $P_\theta(y = c \mid t \in x)$ is disproportionately high for a particular class c. Our goal is to adapt the model's predictions to reduce reliance on such shortcut tokens, without access to $\mathcal{D}_{\text{train}}$.

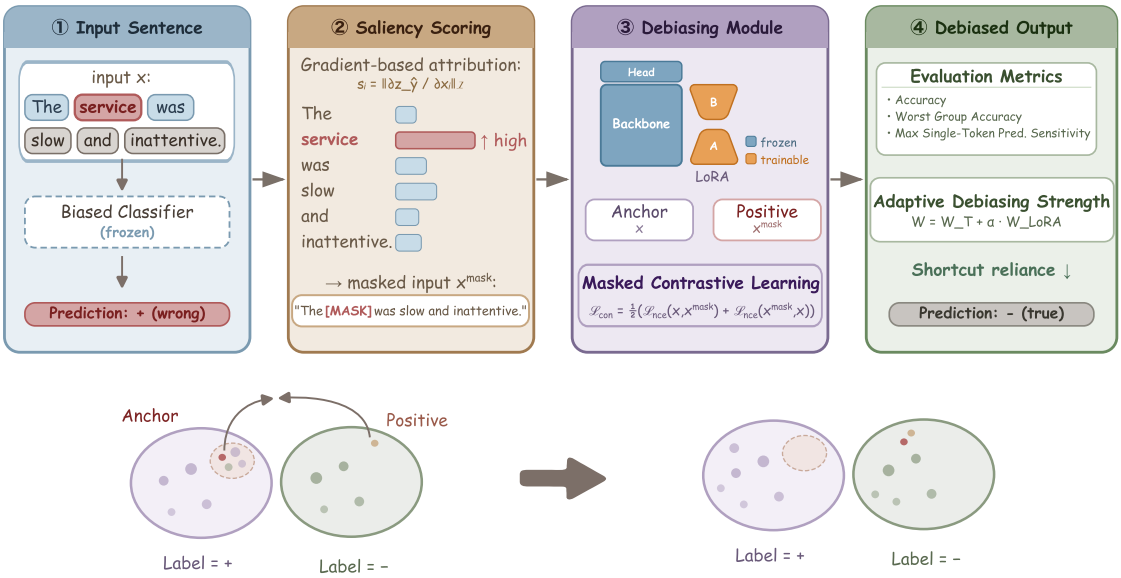


Figure 2: Overview of SHORTCUT GUARDRAIL, which (1) obtains predictions from a frozen biased text classifier, (2) captures shortcut tokens via gradient-based saliency scoring, (3) trains a lightweight LoRA adapter via Masked Contrastive Learning (MaskCL), and (4) calibrates the debiasing strength α to produce debiased predictions with reduced shortcut reliance. The bottom panel illustrates the effect of MaskCL training. Before training, spurious correlations cause examples containing the shortcut token (Anchor, x) to form a cluster within the wrong label, while their masked variants (Positive, x^{mask}) are projected to the correct label. The MaskCL objective pulls anchors and positives closer in the embedding space, dissolving the spurious cluster and reducing shortcut reliance.

3.2 Gradient-Based Token Attribution

For each input $x = (t_1, t_2, \ldots, t_L)$ of length L, we compute a saliency score s_i for each token t_i using gradient-based attribution:

$$s_i = \Big\| \frac{\partial \mathcal{L}_{\text{cls}}(f_\theta(x), \hat{y})}{\partial \mathbf{e}_i} \odot \mathbf{e}_i \Big\|,$$

where $\mathbf{e}_i$ is the learned embedding of token t_i, $\odot$ denotes element-wise multiplication, $\|\cdot\|$ represents the ℓ_2 norm, and $\mathcal{L}_{\text{cls}}$ is the cross-entropy loss between the model's output logits $f_\theta(x)$ and its own prediction label $\hat{y}$. This corresponds to the gradient $\times$ input attribution method (Sundararajan et al., 2017; Li et al., 2016), measuring each token's contribution to the classification output.

We select the tokens with the top-k saliency scores to form the *important token set* $\mathcal{H}(x) = \{t_{i_1}, \ldots, t_{i_k}\}$, which function as shortcut token candidates (not necessarily confirmed ones). We set $k = 10$ by default: for 99% of the shortcut-containing samples, the top-10 important tokens capture the shortcut token (Figure 5, Section 4.2), under a strong spurious correlation (p=100%), while keeping computation moderate.

3.3 Debiasing Module

Architecture. We insert LoRA adapters (Hu et al., 2022) into the backbone encoder of f_θ while the classification head remains frozen. The LoRA parameters ϕ are the only trainable components. Let $z_\phi(x)$ denote the representation of the `[CLS]` token produced by the LoRA-adapted encoder.

Anchor-Positive Pairs. Unlike standard contrastive learning approaches, e.g., SimCLR (Chen et al., 2020), that use random augmentations such as dropout or back-translation, we construct anchor-positive pairs through attribution-guided masking:

For each $t_{i_j} \in \mathcal{H}(x)$, we produce a masked variant of x, denoted x^{mask_j}, by replacing t_{i_j} with the `[MASK]` token while retaining all other tokens:

$$x^{\text{mask}_j} = (t_1, \ldots, t_{i_j-1}, \texttt{[MASK]}, t_{i_j+1}, \ldots, t_L).$$

We pair the original input x ("anchor") with each of its masked variants x^{mask_j} ("positive"), yielding k anchor-positive pairs per input. This design explicitly probes the model's reliance on each candidate shortcut token independently, rather than enforcing invariance to arbitrary perturbations.

Masked Contrastive Learning (MaskCL). For a batch of B inputs, we obtain $B \times k$ anchor-

positive pairs $\{(x_i, x_i^{\text{mask}_j})\}_{i=1,j=1}^{B,k}$. We first employ the LoRA-adapted model to produce ℓ_2-normalized embeddings:

$$\mathbf{a}_i = z_\phi(x_i)/\|z_\phi(x_i)\|,$$
$$\mathbf{p}_{ij} = z_\phi(x_i^{\text{mask}_j})/\|z_\phi(x_i^{\text{mask}_j})\|.$$

$(\mathbf{a}_i, \mathbf{p}_{ij})$ are the anchor-positive embeddings, while all other embeddings $\mathbf{a}_{i'}$ $(i' \neq i, i' \in \{1, \cdots, B\})$ in the batch function as negative samples. The MaskCL contrastive loss $\mathcal{L}_{\text{con}}$ is defined as follows:

$$\mathcal{L}_{\text{nce}}^{x,x^{\text{mask}}} = -\sum_{i=1}^{B}\sum_{j=1}^{k} \log \frac{\exp(\mathbf{a}_i^\top \mathbf{p}_{ij}/\tau)}{\sum_{i'=1}^{B} \exp(\mathbf{a}_{i'}^\top \mathbf{p}_{ij}/\tau)},$$

$$\mathcal{L}_{\text{nce}}^{x^{\text{mask}},x} = -\sum_{i=1}^{B}\sum_{j=1}^{k} \log \frac{\exp(\mathbf{a}_i^\top \mathbf{p}_{ij}/\tau)}{\sum_{i'=1}^{B} \exp(\mathbf{a}_i^\top \mathbf{p}_{i'j}/\tau)},$$

$$\mathcal{L}_{\text{con}} = \frac{1}{2Bk}(\mathcal{L}_{\text{nce}}^{x,x^{\text{mask}}} + \mathcal{L}_{\text{nce}}^{x^{\text{mask}},x}),$$

where τ is a hyperparameter controlling the concentration of the similarity distribution over negative samples (default to 0.1 as in the SimCLR implementation (Chen et al., 2020)). MaskCL encourages the model to produce consistent representations regardless of the presence of individual important tokens. It pulls anchor-positive pairs closer in the embedding space, dissolving spurious clusters formed around shortcut tokens and reducing reliance on each potential shortcut (Figure 2).

3.4 Calibrating Debiasing Strength

After training the LoRA adapter, the degree of intervention is controlled by the debiasing strength α. The final model weights are computed as:

$$W = W_T + \alpha \cdot W_{\text{LoRA}}$$

where W_T and W_{LoRA} denote the original frozen weights and the learned LoRA weights, respectively. The optimal α depends on the degree of distribution shift at deployment time: when shortcut-label correlations learned from training remain valid, a small α preserves useful signals; when these correlations break down, a larger α is needed to suppress harmful shortcuts. Instead of defaulting to a fixed α, we calibrate α via a grid search over $\{0, 0.1, \ldots, 1.0\}$ using a labeled support set as a proxy of the target distribution.

We also show that the performance with adaptively calibrated α is stable for support set size ≥ 40 and matches or improves on the fixed $\alpha = 1$ baseline (Figure 8 in Appendix A.5). We thus use 40 support examples in our experiments. We also report the adaptive α values in our experiments in Appendix A.6.

4 Models Encode a Recoverable Signal of Their Shortcuts

We establish our key finding: converged models internalize token-level shortcuts (Section 4.1), and gradient-based attribution reliably captures the responsible shortcut tokens (Section 4.2).

4.1 Models Internalize Token-Level Shortcuts

We first show that models are susceptible to shortcuts encountered during training, which undermines deployment-time performance. We construct a controlled testbed using Amazon Product Reviews (Ni et al., 2019) for binary sentiment classification, following the protocol of Chew et al. (2024), where a token ("book") is injected such that $P(y = 1|\text{"book"} \in x) = 1$ in the training set. We evaluate on an unfiltered test set across four groups defined by label y and the existence of the shortcut token "book" in the input (see Appendix A.1.1 for details). Group 4, with negative samples ($y = 0$) containing the shortcut token, directly exposes shortcut reliance: a model that has learned the association ("book" $\rightarrow$ $y = 1$) will systematically misclassify these samples as $\hat{y} = 1$.

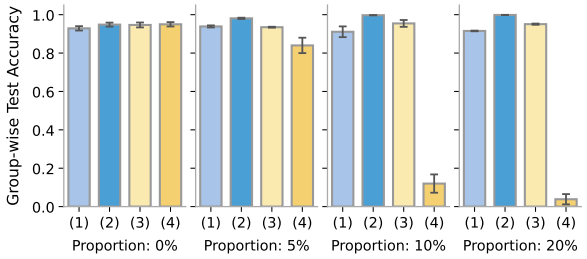


Figure 3: Group-wise test accuracy under different proportions of training data with the shortcut token "book". The strength of the spurious correlation $P(y = 1|\text{"book"} \in x)$ is 1 and label distributions are balanced ($P(y = 1) = 0.5$). Error bars denote standard deviation over three random trials. (1)-(4) correspond to Group 1-Group 4 of the test data. See Table 6 in Appendix A.1.1 for group-wise training data statistics.

As shown in Figure 3, even a small fraction of training data with the shortcut can lead the model to rely on this spurious correlation, resulting in a low test accuracy on Group 4. We further observe that such reliance increases monotonically with the strength of the spurious correlation, i.e.,

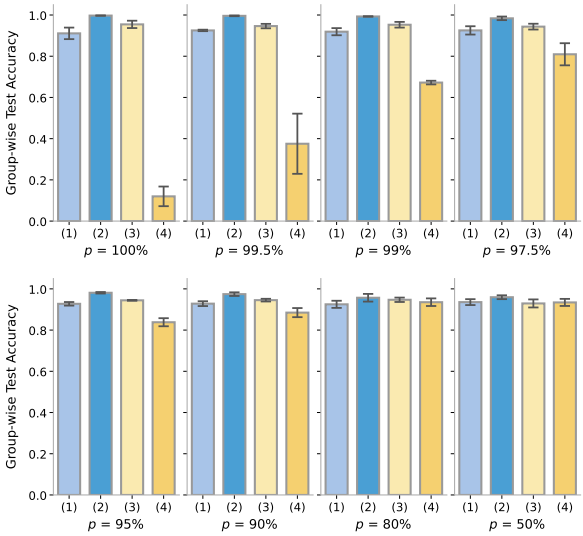


Figure 4: Group-wise test accuracy under different strengths of the spurious correlation, where $p = P(y = 1|\text{“book”} \in x)$. The proportion of training samples containing the shortcut “book” is fixed at 10%, and label distributions are balanced ($P(y = 1) = 0.5$). (1)-(4) correspond to Group 1-Group 4 of the test data. See Table 7 in Appendix A.1.1 for training data statistics.

$P(y = 1|\text{“book”} \in x)$, in the training set (Figure 4). These results suggest that a handful of training data with a strong spurious correlation suffices to induce reliance on shortcuts, raising a natural question: given that shortcuts are already encoded in the model's parameters, can we detect them without access to training data or shortcut annotations?

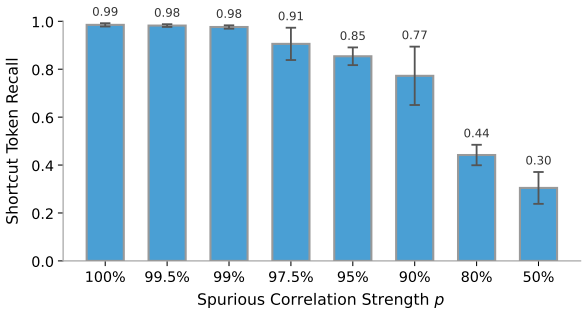


Figure 5: Shortcut Token Recall under different strengths of the spurious correlation. Each bar shows the percentage of samples whose shortcut (“book”) appears among the top-10 important tokens, averaged over three random trials.

4.2 Gradients Localize Shortcut Tokens

In fact, these shortcut reliances can be pinpointed using the model's internal attribution. We define “important tokens” as the tokens with the highest saliency scores for a given sample, and “Shortcut Token Recall” as the fraction of samples with a

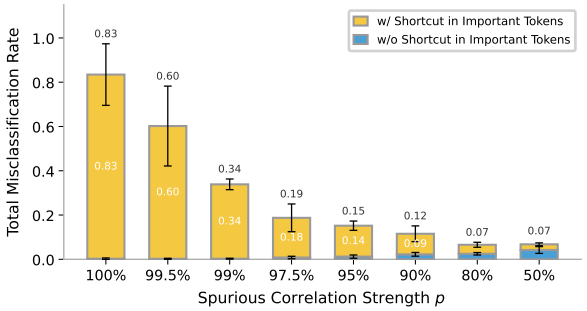


Figure 6: Total misclassification rate under different strengths of the spurious correlation p. The misclassification rate (black annotation) can be attributed to two groups of samples: those with the shortcut (“book”) among the important tokens (gold bar, white annotation), and those without (blue bar).

shortcut token appearing among the important tokens. As shown in Figure 5, Shortcut Token Recall increases with the strength of the spurious correlation, reaching above 90% under strong spurious correlations ($P(y = 1|\text{“book”} \in x) > 97.5\%$) using only the top-10 important tokens. Figure 6 further shows that the resulting performance degradation is primarily driven by samples where the shortcut appears among the important tokens.

5 Deployment-Time Mitigation Performance Ceiling is Upper-Bounded by Training-Time Mitigation

We now characterize a fundamental limit: some shortcut information is lost once the model has converged. We prove that deployment-time mitigation is information-theoretically upper-bounded by training-time mitigation. The full formal setup and all proofs are deferred to Appendix B.

Shortcut mitigation involves two stages: identifying which token-label correlations are spurious, and suppressing them. Let S^* denote the true set of such correlations, modeled as a random variable taking values in a finite set $\mathcal{S}$ ($|\mathcal{S}| \geq 2$). Training- and deployment-time methods differ in their estimate of S^*, denoted $\hat{S}_{\text{train}}$ and $\hat{S}_{\text{deploy}}$: the former may access the training set $\mathcal{D}_{\text{train}}$ directly, while the latter can only use the converged parameters θ. We assume that $S^* \to \mathcal{D}_{\text{train}} \to \theta \to g(\theta, \mathcal{D}_{\text{test}})$ is a Markov chain, where the converged model θ acts as an information bottleneck between $\mathcal{D}_{\text{train}}$ and deployment-time observables $g(\theta, \mathcal{D}_{\text{test}})$.

Theorem 1 (Information Upper Bound)**.** The mutual information between the true spurious correlation structure S^* and any deployment-time estimate

$\hat{S}_{\text{deploy}}$ satisfies:

$$I(S^*;\, \hat{S}_{\text{deploy}}) \;\le\; I(S^*;\, \theta) \;\le\; I(S^*;\, \mathcal{D}_{\text{train}}).$$

In particular, the model parameters lose $\Delta I = I(S^*;\, \mathcal{D}_{\text{train}}) - I(S^*;\, \theta) \ge 0$ units of information on S^* relative to the training set $\mathcal{D}_{\text{train}}$.

Theorem 1 bounds information, not performance. We bridge this via Fano's Inequality. Denoting the Shannon entropy of S^* by $H(S^*)$, we have:

Theorem 2 (Identification Error Bounds)**.** Let S^* take values in the finite set $\mathcal{S}$ with $|\mathcal{S}| \ge 2$. Define the Fano lower bounds on identification error:

$$L_{\text{deploy}} = \frac{H(S^*) - I(S^*;\, \theta) - 1}{\log(|\mathcal{S}| - 1)}, \tag{1}$$

$$L_{\text{train}} = \frac{H(S^*) - I(S^*;\, \mathcal{D}_{\text{train}}) - 1}{\log(|\mathcal{S}| - 1)}. \tag{2}$$

Then the misidentification probabilities satisfy:

$$P(\hat{S}_{\text{deploy}} \neq S^*) \;\ge\; L_{\text{deploy}}, \tag{3}$$

$$P(\hat{S}_{\text{train}} \neq S^*) \;\ge\; L_{\text{train}}, \tag{4}$$

and $L_{\text{deploy}} \ge L_{\text{train}}$.

Finally, we bound the training- and deployment-time mitigation performance ceilings (not the performance of any particular method):

Assumption 1 (Monotonicity)**.** *There exists a non-decreasing, differentiable function $\rho : [0, 1] \to \mathbb{R}$ such that a method with identification error L has expected mitigation performance $\rho(1 - L)$.*

Lemma 3 (Performance Ceiling from Identification Error)**.** *Under Assumption 1, any mitigation method whose identification error satisfies $P(\hat{S} \neq S^*) \ge L$ for some $L \in [0, 1]$ has expected performance bounded by:*

$$\mathbb{E}[R(\hat{S}, S^*)] \;\le\; \rho(1 - L).$$

In particular, combining with Theorem 2:

$$\mathbb{E}[R(\hat{S}_{\text{deploy}}, S^*)] \;\le\; \rho(1 - L_{\text{deploy}}), \tag{5}$$

$$\mathbb{E}[R(\hat{S}_{\text{train}}, S^*)] \;\le\; \rho(1 - L_{\text{train}}). \tag{6}$$

Theorem 4 (Performance Ceiling Comparison)**.** Under Assumption 1, the performance ceilings for deployment- and training-time mitigation satisfy:

$$\rho(1 - L_{\text{deploy}}) \;\le\; \rho(1 - L_{\text{train}}),$$

Moreover, the gap between the two bounds is:

$$\rho(1 - L_{\text{train}}) - \rho(1 - L_{\text{deploy}}) = \rho'(\xi) \cdot \frac{\Delta I}{\log(|\mathcal{S}| - 1)},$$

for some $\xi \in [1 - L_{\text{deploy}},\; 1 - L_{\text{train}}]$, where $\Delta I = I(S^*;\, \mathcal{D}_{\text{train}}) - I(S^*;\, \theta) \ge 0$.

Theorem 4 shows that the more information f_θ loses about S^*, the larger the inherent gap between the training- and deployment-time mitigation performance ceilings. We next show how gradient attribution (Section 4) guides deployment-time shortcut mitigation empirically in Section 6.

6 SHORTCUT GUARDRAIL Mitigates Token-Level Shortcuts

6.1 Experiment Setup

We describe the datasets and baselines used in our experiments. We summarize the sizes of the datasets in Table 9. Implementation details are provided in Appendix A.7.

Metric	Model	SST-2	Civil	MultiNLI
Accuracy	ERM	0.911	**0.880**	0.819
	JTT	*0.915*	*0.878*	**0.829**
	NFL	0.815	0.832	0.665
	DFR	*0.915*	*0.878*	*0.824*
	SG (Ours)	**0.917**	0.874	0.816
WGA	ERM	–	0.737	0.429
	JTT	–	*0.788*	0.429
	NFL	–	0.717	0.428
	DFR	–	0.778	*0.571*
	SG (Ours)	–	**0.814**	**0.714**

Table 1: Performance on real-world benchmarks with naturally occurring token-label correlations. The best results are marked in **bold** while the second best are in *italics*. SST-2 does not have predefined shortcut annotations, hence WGA is not applicable (–).

Datasets. We evaluate our framework under two complementary settings. The first uses real-world benchmarks where shortcut correlations arise naturally from the data collection process. To ensure comparable scale across all benchmarks while preserving original label distributions, we use stratified subsets of (1) SST-2 (Wang et al., 2018) for binary sentiment classification; (2) CivilComments (Borkan et al., 2019) for toxicity detection, where demographic identity terms are spuriously correlated with the toxic label; and (3) MultiNLI (Williams et al., 2018) for natural language inference, where negation words frequently

Metric	Model	Civil	MultiNLI	Yelp-ST	Yelp-Syn	GoEmo-ST	GoEmo-Syn
Accuracy	ERM	0.522	0.270	0.355	0.419	0.513	0.593
	JTT	0.532	0.268	0.440	*0.490*	0.403	*0.613*
	NFL	*0.536*	0.048	*0.449*	0.471	0.387	0.377
	DFR	*0.536*	*0.273*	0.373	0.435	*0.557*	**0.630**
	SG (Ours)	**0.572**	**0.323**	**0.488**	**0.530**	**0.627**	0.607
WGA	ERM	0.216	0.010	0.030	0.035	0.026	*0.222*
	JTT	0.208	**0.030**	*0.266*	**0.333**	0.000	0.000
	NFL	*0.260*	**0.030**	0.157	0.180	0.000	0.000
	DFR	0.240	0.010	0.040	0.206	*0.139*	**0.359**
	SG (Ours)	**0.396**	0.010	**0.353**	*0.220*	**0.339**	*0.222*

Table 2: Overall accuracy and worst-group accuracy (WGA) under shortcut distribution shift. The best results are marked in **bold** while the second best are marked in *italics*. The CivilComments and MultiNLI datasets in this table are reconstructed with controlled shortcut injection and differ from the original versions in Table 1.

Dataset	ERM	JTT	NFL	DFR	SG (Ours)
Civil	0.530	0.549 (+0.019)	0.451 (−0.079)	0.526 (−0.004)	0.438 (−0.092)
MultiNLI	0.797	0.267 (−0.530)	0.143 (−0.654)	0.212 (−0.585)	0.381 (−0.416)
Yelp-ST	0.331	0.393 (+0.062)	0.074 (−0.257)	0.339 (+0.008)	0.203 (−0.128)
Yelp-Syn	0.219	0.264 (+0.045)	0.083 (−0.136)	0.139 (−0.080)	0.126 (−0.093)
GoEmo-ST	0.647	0.355 (−0.292)	0.102 (−0.563)	0.713 (+0.066)	0.501 (−0.146)
GoEmo-Syn	0.488	0.180 (−0.308)	0.190 (−0.263)	0.527 (+0.039)	0.414 (−0.074)

Table 3: Shortcut reliance as measured by MSTPS. Parentheses show the change relative to ERM.

serve as a shortcut for the contradiction class (Gururangan et al., 2018; McCoy et al., 2019).

The second employs controlled benchmarks constructed following the protocol of Zhou et al. (2024b), which injects token-level shortcuts with controllable strengths into both training and test data. We evaluate six configurations: Yelp-ST and Yelp-Syn for sentiment classification (Zhang et al., 2015), GoEmo-ST and GoEmo-Syn for emotion classification on GoEmotions (Demszky et al., 2020), and modified versions of CivilComments and MultiNLI that we constructed. Here, ST (single-token) configurations introduce a label-irrelevant token (e.g., "honestly") with a label-dependent probability. The Syn (synonym) configurations inject shortcut tokens from a set of 15 synonyms (e.g., "candidly", "in truth").

The shortcut strength is governed by a parameter λ that represents the label-conditional probability of the shortcut's occurrence. Note that λ is distinct from the spurious correlation strength p used in Sections 4.1 and 4.2. We train all models with $\lambda = 1$ to simulate the strongest possible spurious correlation. We use the "anti-test set" protocol of Zhou et al. (2024b), which reverses the shortcut-label distribution of the training set to maximally expose shortcut reliance. Full construction details are provided in Appendix A.1.3.

Baselines. We compare our framework against Empirical Risk Minimization (ERM, vanilla fine-tuning), and training-time shortcut mitigation methods: JTT (Liu et al., 2021), NFL (Chew et al., 2024), and DFR (Kirichenko et al., 2023). Implementation details are provided in Appendix A.8.

6.2 Evaluation Metrics

Accuracy and Worst-Group Accuracy (WGA). We report overall test accuracy and worst-group accuracy (WGA). We define groups for all datasets with shortcut annotations (Yelp, GoEmo, CivilComments, and MultiNLI) based on all possible combinations of label and shortcut existence (Appendix A.1.1). As the minimum accuracy across all groups, WGA reflects the model's robustness to shortcut distribution shifts: a model may achieve high overall accuracy by exploiting shortcuts, but will yield low accuracy on "conflicting" groups where those shortcut-label correlations observed during training do not hold.

Maximum Single-Token Prediction Sensitivity (MSTPS). To directly quantify the model's reliance on token-level shortcuts, we propose the metric MSTPS. Existing shortcut-degree measures rely on training dynamics or early-training checkpoints (Du et al., 2021), which are unavailable in our deployment-time setting; MSTPS instead quan-

tifies shortcut reliance using only the converged model and test data. For each input x_i, we measure the maximum shift in prediction probability caused by masking any individual important token:

$$\text{MSTPS} = \frac{1}{N}\sum_{i=1}^{N} \max_{j} \left| P(\hat{y}_i|x_i) - P(\hat{y}_i|x_i^{\text{mask}_j}) \right|,$$

where N is the number of test samples, $x_i^{\text{mask}_j}$ denotes input x_i with the j-th important token replaced by `[MASK]`, $\hat{y}_i$ is the model's predicted label for x_i, and the maximum is taken over $\{j|t_j \in \mathcal{H}(x_i)\}$, the top-$k$ tokens identified by saliency scoring (Section 3.2). A high MSTPS indicates that the model's prediction can be substantially altered by removing a single token, suggesting over-reliance on local shortcuts. Conversely, an effective debiasing method should reduce MSTPS, indicating that the model has learned to integrate broader contextual evidence rather than isolated features.

6.3 SHORTCUT GUARDRAIL is Effective Across Shortcut Distributions, Types, and Strengths

Real-world Benchmarks. Table 1 reports results on SST-2, CivilComments, and MultiNLI without artificial shortcuts. Our method achieves the highest accuracy on SST-2, and substantially raises WGA on CivilComments and MultiNLI at a marginal cost in accuracy. In these benchmarks with no artificial distribution shift, WGA is governed by minority "conflicting" groups where shortcut–label correlations do not hold at test time, while overall accuracy is dominated by the majority where they still apply; mitigation redistributes performance toward the minority groups, raising WGA at little cost to accuracy (Liu et al., 2021; Sagawa et al., 2020). These results confirm that SHORTCUT GUARDRAIL effectively mitigates reliance on naturally occurring shortcuts.

Controlled Benchmarks with Shortcut Distribution Shift. We further evaluate SHORTCUT GUARDRAIL on controlled benchmarks with artificially injected shifts in the shortcut distribution, maximally exposing LM's reliance on shortcuts. As shown in Table 2, ERM suffers severe WGA degradation, confirming heavy shortcut reliance. In contrast, our method yields notable WGA improvements over ERM and performs comparably or better than training-time baselines, attaining the best accuracy on all datasets except GoEmo-ST and top-2 WGA except on MultiNLI, where all methods struggle. Such globally poor performance likely stems from the nature and difficulty of the MultiNLI task (see Appendix).

Reducing Shortcut Reliance. Across all benchmarks, our method consistently reduces MSTPS relative to ERM, confirming that the MaskCL objective redistributes predictive evidence across a broader context (Table 3). Lower MSTPS does not by itself guarantee better performance, since some token sensitivity reflects legitimate signal rather than spurious correlation: on Yelp-ST, NFL attains the lowest MSTPS (0.074) but lower accuracy (0.449) and WGA (0.157) than our method (MSTPS 0.203, accuracy 0.488, WGA 0.353). Our method thus balances reducing shortcut reliance against preserving task-relevant information (Figure 7 in Appendix A.2).

Performance Across Varying Shortcut Strengths and In-distribution Test Data. To evaluate our method's robustness across varying degrees of distribution shift, we modulate the injected shortcut strength, denoted by λ. Table 4 shows results on Yelp-ST and CivilComments (modified). We observe that the performance gap between our method and ERM widens as the λ becomes larger, demonstrating that our approach is more effective when shortcuts are more severe. Nevertheless, at lower λ values, which more closely resemble naturally occurring distributions with weaker shortcut signals, our method still improves overall accuracy.

We also verify that our method preserves in-distribution performance, where the shortcut-label correlations from training data remain intact (Table 11 in Appendix A.4). On most benchmarks, the adaptive α calibration mechanism selects $\alpha = 0$, fully preserving the original ERM predictions.

Dataset	λ	Accuracy			WGA		
		ERM	SG	Δ	ERM	SG	Δ
Yelp-ST	1.0	0.355	0.488	+0.133	0.030	0.353	+0.323
	0.8	0.558	0.583	+0.025	0.481	0.467	−0.014
	0.6	0.614	0.616	+0.002	0.506	0.480	−0.026
Civil	1.0	0.522	0.572	+0.050	0.216	0.396	+0.180
	0.9	0.705	0.717	+0.012	0.487	0.544	+0.057
	0.8	0.761	0.777	+0.016	0.568	0.715	+0.147
	0.6	0.861	0.861	+0.000	0.747	0.780	+0.033

Table 4: Effect of shortcut strength λ in controlled benchmarks. Δ = SG − ERM.

Metric	Model	Real-World			Controlled		
		SST-2	Civil	MultiNLI	Civil	Yelp-ST	Yelp-Syn
Accuracy	ERM	0.911	0.880	0.819	0.522	0.355	0.419
	SG-rdk	0.908±0.001	0.855±0.012	0.811±0.002	0.557±0.006	0.400±0.013	0.461±0.034
	SG (Ours)	0.911±0.006	0.884±0.002	0.819±0.008	0.577±0.009	0.502±0.007	0.530±0.028
WGA	ERM	–	0.737	0.429	0.216	0.030	0.035
	SG-rdk	–	0.710±0.036	0.619±0.083	0.451±0.012	0.219±0.023	0.236±0.040
	SG (Ours)	–	0.785±0.012	0.636±0.167	0.461±0.015	0.257±0.113	0.382±0.055

Table 5: Ablation of attribution-guided token selection on real-world benchmarks and on controlled benchmarks under shortcut distribution shift. SG-rdk replaces the important token set $\mathcal{H}(x)$ with k tokens randomly sampled from x, keeping all other components unchanged. SG-rdk and SG are reported as mean $\pm$ standard deviation over three random trials; ERM results are as in Tables 1 and 2. SST-2 does not have predefined shortcut annotations, hence WGA is not applicable (–).

6.4 Ablation Study

We performed an ablation study that isolates the contribution of the gradient-based attribution process by masking k random tokens in each sentence. Specifically, the ablated variant, denoted SG-rdk, replaces the important token set $\mathcal{H}(x)$ with k tokens sampled at random from x, keeping all other components, e.g., the MaskCL debiasing module unchanged. Table 5 compares SG-rdk with SG and ERM on the real-world and controlled benchmarks. We find that (1) SG-rdk still improves WGA over ERM on the benchmarks with strong or injected token shortcuts, suggesting that other components, including the MaskCL debiasing module, contribute to SG performance; (2) SG exceeds SG-rdk on accuracy across every benchmark, and on WGA the margin is largest where token-level shortcuts are most prominent (Yelp-Syn: 0.382 vs. 0.236), indicating that gradient-based attribution contributes separately. Note that neither SG nor SG-rdk helps or harms classification performance on SST-2, suggesting that SST-2 is largely free of harmful token-level shortcuts. Together, these results show that the saliency-based attribution and the debiasing module collectively contribute to our method's gains.

7 Discussion and Conclusion

A converged text encoder carries a signal of its learned shortcuts that unsupervised gradient-based attribution can uncover. Leveraging this signal, our deployment-time mitigation matches or outperforms training-time baselines. This is remarkable given that we prove deployment-time mitigation is information-theoretically upper-bounded by training-time methods. We attribute this efficacy to the locus of intervention: rather than the dataset-level statistics used by JTT, DFR, and NFL, we use per-input gradient attribution to localize and neutralize shortcut tokens. By removing the need for training data, shortcut annotations, and retraining, our framework offers a practical solution to deployment-time distribution shift.

Limitations

While SHORTCUT GUARDRAIL demonstrates strong performance across diverse benchmarks, we acknowledge the following limitations. (1) As we formally show in Appendix B, deployment-time methods are information-theoretically upper-bounded by training-time methods. Since shortcuts originate from $\mathcal{D}_{\text{train}}$, any deployment-time method can only recover the spurious correlation structure indirectly through the trained model θ, which acts as an information bottleneck. While our method approaches this bound under strong shortcuts (Table 2), the gap widens for structurally complex shortcut patterns, writing styles, and syntactic templates. (2) Our gradient-based attribution operates at the token level, which limits its ability to capture multi-token shortcut patterns. (3) Our evaluation focuses on text classification tasks using BERT-based encoders, which serve as a clean prototype for establishing our findings. Our masked contrastive learning objective is compatible with bidirectional attention in transformer encoders that are pretrained by masked language modeling. It does not naturally extend to decoder-only architectures representative of modern large language models: under causal attention, masking a token leaves the representations of all preceding tokens unchanged, failing to produce the globally distinct representation that our contrastive objective relies on. Extending this framework to decoder-only LMs, which

would require rethinking both the saliency attribution step and the contrastive objective, is left for future work. (4) Although our method eliminates the need for training data access, shortcut annotations, or full model retraining, it still requires a small labeled support set for calibrating the debiasing strength α. In fully unsupervised scenarios, alternative calibration strategies would be needed. (5) Targeting deployment settings where the model is available but training data and annotations are not, our method assumes access to model weights and gradients. Relaxing this assumption toward black-box/closed-source models is a direction for future work.

Acknowledgments

This work was supported in part by the US National Science Foundation [III-2232121] and the US National Institutes of Health [R01HG012470, R01GM157795]. J.L. was partially supported by a gift from Arm, Inc. S.D. was partially supported by a Softbank Group / Arm Ph.D. Fellowship. Conflict of Interest: C.K. is a co-founder of Ellumigen, Inc.

References

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. 2019. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion Proceedings of the 2019 World Wide Web Conference (WWW)*, pages 491–500. ACM.

Samuel Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 632–642.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 1597–1607.

Jiali Cheng and Hadi Amiri. 2024. FairFlow: Mitigating dataset biases through undecided learning for natural language understanding. In *Proceedings of the 2024 conference on empirical methods in natural language processing*, pages 21960–21975.

Pengyu Cheng, Weituo Hao, Siyang Yuan, Shijing Si, and Lawrence Carin. 2021. FairFil: Contrastive neural debiasing method for pretrained text encoders. In *International Conference on Learning Representations (ICLR)*.

Oscar Chew, Hsuan-Tien Lin, Kai-Wei Chang, and Kuan-Hao Huang. 2024. Understanding and mitigating spurious correlations in text classification with neighborhood analysis. In *Findings of the association for computational linguistics: EACL 2024*, pages 1013–1025.

Seungtaek Choi, Myeongho Jeong, Hojae Han, and Seung-won Hwang. 2022. C2L: Causally contrastive learning for robust text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10526–10534.

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.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 4040–4054. Association for Computational Linguistics.

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 (NAACL-HLT)*, pages 4171–4186. Association for Computational Linguistics.

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023a. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120.

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.

Mengnan Du, Ruixiang Tang, Wenhao Fu, and Xia Hu. 2022. Towards debiasing DNN models from spurious feature influence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 9521–9528.

Yanrui Du, Jing Yan, Yan Chen, Jing Liu, Sendong Zhao, Qiaoling She, Hua Wu, Haifeng Wang, and Bing Qin. 2023b. Less learn shortcut: Analyzing and mitigating learning of spurious feature-label correlation. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence (IJCAI)*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6894–6910. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.

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.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 18661–18673.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. 2023. Last layer re-training is sufficient for robustness to spurious correlations. In *International Conference on Learning Representations (ICLR)*.

Michalis Korakakis, Andreas Vlachos, and Adrian Weller. 2025. Mitigating shortcut learning with interpolated learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9191–9206.

Jiwei Li, Xinlei Chen, Eduard Hovy, and Dan Jurafsky. 2016. Visualizing and understanding neural models in NLP. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 681–691. Association for Computational Linguistics.

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 *International Conference on Machine Learning*, pages 6781–6792. PMLR.

R Thomas 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.

Zachary Nado, Shreyas Padhy, D. Sculley, Alexander D'Amour, Balaji Lakshminarayanan, and Jasper Snoek. 2020. Evaluating prediction-time batch normalization for robustness under covariate shift. *arXiv preprint arXiv:2006.10963*.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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 188–197.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. 2022. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, pages 16888–16905.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 conference on empirical methods in natural language processing (EMNLP 2002)*, pages 79–86.

Shiori Sagawa, Pang Wei Koh, Tatsunori B. Hashimoto, and Percy Liang. 2020. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*.

Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. 2020. Improving robustness against common corruptions by covariate shift adaptation. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 11539–11551.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. 2020. Test-time training with self-supervision for generalization under distribution shifts. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 9229–9248.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, pages 3319–3328.

Lifu Tu, Garima Lalwani, Spandana Gella, and He He. 2020. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2021. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations (ICLR)*.

Tianlu Wang, Rohit Sridhar, Diyi Yang, and Xuezhi Wang. 2022. Identifying and mitigating spurious correlations for improving robustness in NLP models. In *Findings of the association for computational linguistics: NAACL 2022*, pages 1719–1729.

Adina Williams, Nikita Nangia, and Samuel R. 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 (NAACL-HLT)*, pages 1112–1122. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28.

Hao Zhao, Yuejiang Liu, Alexandre Alahi, and Tao Lin. 2023. On pitfalls of test-time adaptation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*.

Yuhang Zhou, Paiheng Xu, Xiaoyu Liu, Bang An, Wei Ai, and Furong Huang. 2024a. Explore spurious correlations at the concept level in language models for text classification. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 478–492.

Yuqing Zhou, Ruixiang Tang, Ziyu Yao, and Ziwei Zhu. 2024b. Navigating the shortcut maze: A comprehensive analysis of shortcut learning in text classification by language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2586–2614. Association for Computational Linguistics.

A Experiment Details

A.1 Dataset Details

A.1.1 Testbed dataset

We construct a controlled testbed using Amazon Product Reviews (Ni et al., 2019) for binary sentiment classification, following the protocol of Chew et al. (2024). Review ratings are on a scale of 1 to 5, where the 4 and 5-star ratings are mapped to a positive sentiment class ($y = 1$) and 1 and 2-star ratings are mapped to a negative sentiment class ($y = 0$), while 3-star ratings are excluded. To inject a shortcut, we filter the training set so that all samples containing a designated token ("book") are of positive sentiment, i.e., $P(y = 1 \mid \text{"book"} \in x) = 1$. We train a BERT model for sentiment classification on the filtered training set with 10,000 samples, and evaluate on the unfiltered test set, which consists of four distinct groups of equal size:

- **Group 1**: positive samples without the shortcut $(y = 1, \text{"book"} \notin x)$.
- **Group 2**: positive samples with the shortcut $(y = 1, \text{"book"} \in x)$.
- **Group 3**: negative samples without the shortcut $(y = 0, \text{"book"} \notin x)$.
- **Group 4**: negative samples with the shortcut $(y = 0, \text{"book"} \in x)$.

Note that for all other datasets with shortcut annotations (Yelp, GoEmotions, CivilComments, and MultiNLI), groups are defined analogously across all combinations of label and shortcut presence. Formally, with a label set $Y = \{y_1, y_2, \cdots, y_k\}$ and a shortcut token set T, there will be $2k$ groups $C_1, C_2, \cdots, C_{2k}$, where

$$C_{2i-1} = \{(x, y) | y = y_i, \forall t \in T, t \notin x\},$$
$$C_{2i} = \{(x, y) | y = y_i, \exists t \in T, t \in x\}.$$

Details on the effect of shortcut samples proportion. For the experiments in Figure 3, we fix $P(y = 1|\text{"book"} \in x) = 1$ in the training set, and keep an equal proportion of positive and negative samples. Table 6 shows the proportions of training samples in each of the groups above. The test set always contains an equal proportion of all four groups.

Table 6: The proportion ($/100$) of samples in each group in the training sets for the experiments in Figure 3.

	0%	5%	10%	20%
Group 1	50	45	40	30
Group 2	0	5	10	20
Group 3	50	50	50	50
Group 4	0	0	0	0

Table 7: The proportion ($/100$) of samples in each group in the training sets for the experiments in Figure 4. $p = P(y = 1|\text{"book"} \in x)$. The statistics for $p = 0.995, 0.975, 0.7$ can be derived analogously.

	p=1	p=0.99	p=0.95	p=0.9	p=0.5
Group 1	40	40.1	40.5	41	45
Group 2	10	9.9	9.5	9	5
Group 3	50	49.9	49.5	49	45
Group 4	0	0.1	0.5	1	5

Details on the effect of shortcut strength. For the experiments in Figure 4, we vary the shortcut signal strength $P(y = 1|\text{"book"} \in x)$ while keeping an equal proportion of positive and negative samples, and a proportion of 10% total samples with the shortcut token "book". Table 6 shows the proportions of training samples in each of the groups above. The test set always contains an equal proportion of all four groups.

A.1.2 Real-World Benchmarks

This section summarizes the three real-world benchmarks used in our evaluation. All datasets are in English.

SST-2 (Wang et al., 2018) is a binary sentiment classification dataset derived from movie reviews. Models trained on SST-2 are known to over-rely on lexical cues such as proper nouns (e.g., *Spielberg*) and sentiment-laden words rather than compositional context (Wang et al., 2022). SST-2 does not provide predefined shortcut annotations, so we report only overall accuracy.

CivilComments (Borkan et al., 2019) is a binary toxicity detection dataset collected from online comments. Each sample is annotated with demographic identity attributes. These identity terms are spuriously correlated with the toxic label: comments mentioning certain demographic groups are disproportionately labeled toxic even when not inherently harmful. We report both overall accuracy and worst-group accuracy across identity-based

groups.

MultiNLI (Williams et al., 2018) is a three-class natural language inference dataset (classes: *entailment*, *contradiction*, and *neutral*). Known shortcuts include negation words (e.g., *nobody*, *never*) for the contradiction class and high lexical overlap between premise and hypothesis for the entailment class (Gururangan et al., 2018; McCoy et al., 2019). We use negation words as the shortcut tokens and report both overall accuracy and worst-group accuracy.

We do not directly use the HuggingFace distribution of MultiNLI. We first downloaded the MultiNLI 1.0 raw data from its NYU official website (https://cims.nyu.edu/~sbowman/multinli) and fed it into the processing pipeline of GroupDRO (https://github.com/kohpangwei/group_DRO), which (1) pooled the training and two development sets and randomly re-split 50/20/30 into train/validation/test sets and (2) annotated the negation-word shortcuts based on their released metadata. Our test set is therefore a disjoint partition of this pool, not a subsample of the training set. The controlled train/anti-test sets used in Table 2 are resampled from these disjoint splits according to per-class negation rates (spurious correlation strength).

A.1.3 Controllable benchmark Construction Details

We follow the shortcut benchmark of Zhou et al. (2024b) to construct occurrence-based shortcuts for the Yelp and GoEmotions datasets. Two types of shortcuts are used:

Single Token (ST). A single label-irrelevant word, "honestly", is inserted at the beginning of a randomly chosen sentence in each review. The insertion probability is label-dependent, controlled by the shortcut strength parameter λ. For example, in Yelp-ST with $\lambda = 1$, the shortcut occurrence/insertion probabilities are 0%, 25%, 50%, 75%, and 100% for ratings 1 through 5, creating a spurious correlation between the presence of "honestly" and higher ratings. We detail the mapping between λ and this probability distribution below (see "Shortcut Strength to Shortcut Occurrence Probability").

Synonym (Syn). The construction follows the same procedure as ST, except that instead of always inserting "honestly", a word or phrase is uniformly sampled from a set of 15 synonyms: {"honestly", "to be honest", "frankly speaking", "to tell the truth", "to be frank", "in truth", "candidly", "speaking candidly", "plainly speaking", "to be direct", "to come clean", "to put it frankly", "if I'm being honest", "in plain terms", "directly speaking"}. This tests whether models can recognize and exploit the correlation between a set of semantically related terms and the label.

CivilComments and MultiNLI. Both datasets come with existing shortcut annotations. CivilComments provides demographic identity term annotations (e.g., *male, female, White, Black, Muslim*). These identity terms collectively form the shortcut token set. MultiNLI provides negation words (*nobody, no, never, nothing*) that are spuriously correlated with the contradiction class. We define the groups for WGA calculation using those shortcut token sets following the definition in Appendix A.1.1. Training and shifted test sets are constructed using a similar λ-based scheme of Zhou et al. (2024b), but instead of inserting shortcut tokens, we randomly sample instances from each group of the original training/test set, where the proportion of samples from each group is determined by the shortcut occurrence probability distribution given by λ: For example, for CivilComments, $\lambda = 1$ yields a 0% samples with shortcuts for non-toxic samples, and 100% samples with shortcuts for toxic samples.

Shortcut Strength to Shortcut Occurrence Probability. Across all datasets, the shortcut strength is controlled by a parameter $\lambda \in [0, 1]$. For a C-class task with labels $\{1, 2, \ldots, C\}$, the probability of shortcut occurrence for label c is $\lambda(c-1)/(C-1)$. For example, on Yelp ($C = 5$, $\lambda = 1$), the shortcut insertion probabilities are 0%, 25%, 50%, 75%, and 100% for ratings 1 through 5, respectively. Larger λ corresponds to stronger spurious correlations. We train with $\lambda = 1$ unless otherwise stated. Table 8 lists the base probabilities for each dataset.

Shifted test sets. For all datasets, we generate shifted test sets by reversing the base probability distribution. On Yelp, the shifted test probabilities become 100%, 75%, 50%, 25%, and 0% for ratings 1 through 5. This maximally exposes models that rely on the shortcut, as the spurious correlation from training is inverted.

Dataset	C	Per-class Shortcut Occurrence Probabilities
Yelp	5	0%, 25%, 50%, 75%, 100%
GoEmotions	4	0%, 33.3%, 66.7%, 100%
CivilComments	2	0%, 100%
MultiNLI	3	100%, 50%, 0%

Table 8: Base shortcut insertion probabilities per label ($\lambda = 1$). Shifted test sets reverse the order of these probabilities (e.g., 0%, 100% $\rightarrow$ 100%, 0%).

A.2 Balance Between Shortcut Reliance and Task-relevant information

In Figure 7, we plot MSTPS against accuracy across all methods and benchmarks reported in Table 3. We can see that SHORTCUT GUARDRAIL generally occupies the upper-left portion of the plot, striking a balance between reducing reliance on shortcuts (lower MSTPS than ERM) and preserving task-relevant information (higher accuracy than ERM).

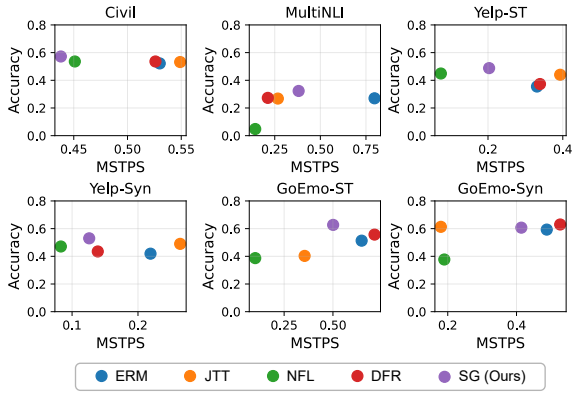


Figure 7: Scatterplots comparing accuracy with MSTPS across all benchmarks and models reported in Table 3.

A.3 Analysis of MultiNLI Results

To analyze controlled globally poor MultiNLI results (Table 2, we diagnose SG's predictions. We first rule out label imbalance and small-sample noise as causes, since the constructed anti-test set is label-balanced (200/200/200) among the 3 classes and every non-empty (label, shortcut) group contains at least 100 examples (Appendix A.1.3).

To better position our insights, we compare MultiNLI with Yelp-ST. For each dataset, we show statistics on the two groups with the largest shortcut distribution shifts, which are also the groups with the lowest ERM accuracies (Table 10).

For MultiNLI, the proportions of training samples for each class containing shortcuts (negation words: *no*, *nobody*, *never*, *nothing*) are {neutral: 0%, entailment: 50%, contradiction: 100%}. For Yelp-ST, they are {1$\star$: 0%, 2$\star$: 25%, 3$\star$: 50%, 4$\star$: 75%, 5$\star$: 100%}, with the shortcut token being *honestly*. The distributions of their test sets are reversed (Appendix A.1.3).

(1) SG substantially improved ERM's worst group (neutral w/ negation) but failed on another (contradiction w/o negation). SG improves ERM's worst group (neutral sentences with negation) from an accuracy of 0.01 to 0.20, a 20$\times$ gain. However, the performance on another group (contradiction w/o negation) drops to 0.01, which equals the ERM WGA.

(2) SG reduces models' reliance on shortcuts. Models can only rely on the learned knowledge of each class. From SG's failure to improve the contradiction w/o negation group, we infer that the model trained on MultiNLI does not learn well how to predict the contradiction class. On the other hand, upon applying SG, model performance on Yelp-ST improves on both 1$\star$ w/ token and 5$\star$ w/o token groups.

We hypothesize that Yelp-ST is inherently an easier task than MultiNLI. This is intuitive because Yelp's five classes share the *sentiment* concept, while MultiNLI's three classes are inherently distinct logical concepts. This is further supported by the fact that, in Yelp-ST, ERM errors on the 1$\star$ w/ token and 5$\star$ w/o token groups concentrate on their adjacent classes (2$\star$ and 4$\star$). We thus conclude that SG is more suitable for datasets with transferable concepts across classes (e.g., Yelp-ST).

A.4 In-Distribution Performance

Table 11 reports in-distribution performance, where shortcut-label correlations from training still hold. On most benchmarks, the adaptive α calibration selects $\alpha = 0$, preserving the original ERM predictions. The only exception is Yelp-ST1, where $\alpha = 0.6$ is selected, resulting in a slight accuracy decrease (0.671 $\rightarrow$ 0.660) but improved WGA (0.376 $\rightarrow$ 0.474).

Yelp-SYN1 shows WGA of 0.000 for both ERM and SG. Under $\lambda = 1$, the minority group contains very few examples, and the model consistently misclassifies them due to strong shortcut reliance. This is expected behavior under in-distribution evaluation and does not reflect a failure of our method.

	Real-World			Controlled					
	SST-2	**Civil**	**MultiNLI**	**Civil**	**MultiNLI**	**Yelp-ST**	**Yelp-Syn**	**GoEmo-ST**	**GoEmo-Syn**
Train	67,349	115,931	206,175	111,298	44,508	8,900	9,400	1,124	1,124
Test	872	1,124	1,000	1,000	600	1,000	1,000	685	685

Table 9: Dataset sizes. Statistics for "Controlled" apply to both shortcut distribution shift and in-distribution datasets.

Dataset	Group	n	ERM	SG	ERM error class
MultiNLI	neutral w/ negation	200	0.010	0.200	contradiction (54%) / entailment (45%)
	contradiction w/o negation	200	0.035	0.010	neutral (65%) / entailment (32%)
Yelp-ST	1⋆ w/ token	199	0.070	0.462	2⋆ (81%)
	5⋆ w/o token	201	0.030	0.353	4⋆ (86%)

Table 10: Statistics on the two groups with the largest shortcut distribution shifts in MultiNLI and Yelp-ST.

A.5 Impact of the Size of the Support Set

As discussed in Section 3.4, the optimal α depends on the degree of distribution shift at deployment time, and we calibrate α via grid search over $\{0, 0.1, \ldots, 1.0\}$ using a small labeled support set. We investigate how this calibration step depends on the size of the support set.

The landscape of α depends on the strength of the shortcut-induced shift. Under strong, controlled shifts where shortcut-label correlations from training are reversed (Table 2), the grid search consistently selects large α to aggressively suppress shortcuts. Under in-distribution settings where these correlations still hold at test time (Table 11), shortcuts remain benign, it consistently selects small α (often $\alpha = 0$) to preserve the original signal. Real-world benchmarks fall in between: shortcut signals exist but are weaker and mixed with genuine predictive cues, so the optimal α is less clearly separated from neighboring values on the grid. Adaptive calibration is helpful when the appropriate debiasing strength is not known a priori.

Figure 8 presents sensitivity analyses on real-world datasets, comparing the adaptive variant against a fixed $\alpha = 1$ under varying support set sizes. On SST-2, performance is stable across support set sizes and coincides with the fixed $\alpha = 1$ baseline, indicating that the debiasing module can be applied at full strength here without loss. On CivilComments, accuracy and WGA are slightly higher for $n \in \{40, 80, 160\}$ than for $n = 10$ and 20. On MultiNLI, performance is considerably more sensitive to the support set size. Overall, performance is stable for $n \geq 40$ across all three datasets.

Overall, in our experiment we use support size $n = 40$. The value of the calibration step is not in finding a globally optimal α, but in adapting the LoRA strength to the unknown degree of distribution shift at deployment time, a flexibility that a single fixed α cannot provide.

A.6 Adaptive α values and performance of default $\alpha = 1$

Here, we report adaptive debiasing strength α (Section 3.4) values used in our experiments and examine how α affects performance, we compare three variants:

- SG with adaptively calibrated α (SG (adaptive α)), our default pipeline;
- SG with a fixed $\alpha = 1$ (SG ($\alpha = 1$)), which applies the debiasing module at full strength without calibration;
- the ERM baseline, which corresponds to $\alpha = 0$.

We run SG (adaptive α) and SG ($\alpha = 1$) over three random trials. We report accuracy, WGA, and the α selected by SG (adaptive α) in each trial. Results cover three settings: the real-world benchmarks (Table 1), the controlled benchmarks under shortcut distribution shift (Table 2), and the in-distribution setting (Appendix A.4). Table 12 shows the results.

Fixed $\alpha = 1$ helps under large distribution shift but harms elsewhere. A fixed $\alpha = 1$ improves performance when the shortcut-label distribution shift is large, as on the controlled CivilComments and Yelp benchmarks. However, it *degrades* performance on the real-world MultiNLI benchmark. It also degrades performance in the in-distribution tests, where the shortcut-label correlations learned during training still hold at test time.

Metric	Model	Civil	MultiNLI	Yelp-ST1	Yelp-SYN1	GoEmo-ST1	GoEmo-SYN1
Accuracy	ERM	0.980	0.908	0.671	0.679	0.896	0.928
	SG	0.980	0.908	0.660	0.679	0.896	0.928
WGA	ERM	0.964	0.840	0.376	0.000	0.667	0.774
	SG	0.964	0.840	0.474	0.000	0.667	0.774

Table 11: In-distribution (ID) performance. Our method preserves the accuracy of the original ERM model.

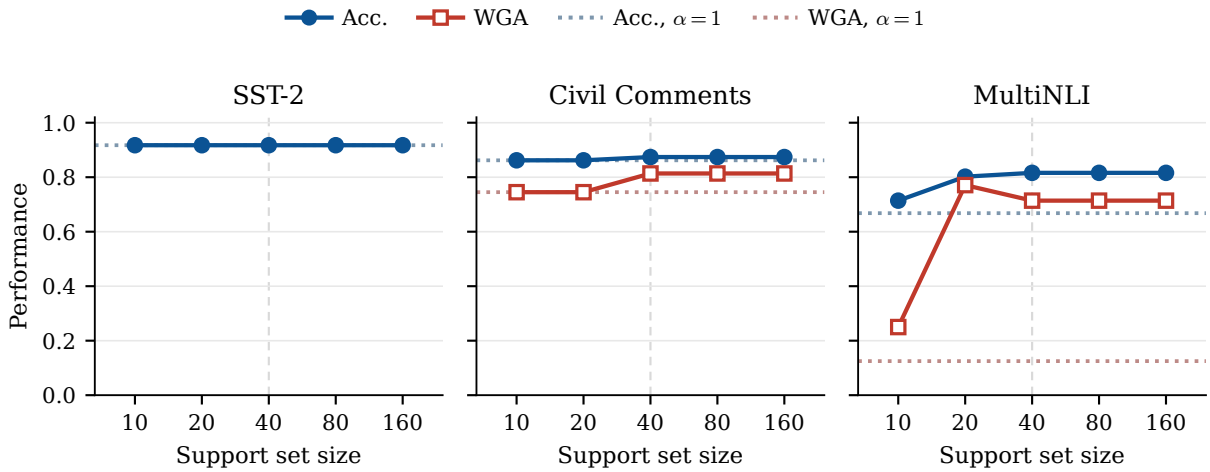


Figure 8: Sensitivity to support set size. Test accuracy and worst-group accuracy (WGA) of SHORTCUT GUARDRAIL as a function of the number of support examples used to calibrate the adaptive scalar α, on SST-2, Civil Comments, and MultiNLI. Dotted lines mark the $\alpha = 1$ baseline, which applies the debiasing module with no calibration. The datasets are the same as in Table 1.

Adaptive α adjusts to the degree of shift. Adaptive calibration avoids these failures in three ways:

- It finds values of α other than 1 that perform better, e.g., 0.9 on Yelp-Syn and 0.7 on GoEmo-Syn.
- It converges to $\alpha = 1$ when full-strength debiasing is optimal, as on both the real-world and controlled CivilComments benchmarks.
- It mostly selects small α (often $\alpha = 0$) when shortcut-label correlations still hold, as in the in-distribution tests.

This flexibility lets SG maintain performance across data scenarios where the appropriate debiasing strength is not known in advance.

On the real-world SST-2 benchmark, neither SG ($\alpha = 1$) nor SG (adaptive α) changes classification performance relative to ERM. This suggests that SST-2 contains no harmful token-level shortcuts.

Finally, in the in-distribution tests, the Yelp-Syn WGA of SG (adaptive α) (0.112 ± 0.158) comes from per-trial values of 0.000, 0.000, and 0.336 on a worst-performing group of only $n = 4$ examples. This is why the standard deviation exceeds the mean.

A.7 SHORTCUT GUARDRAIL Implementation Details

We implement all models in PyTorch using the HuggingFace Transformers and PEFT libraries for the BERT encoder and LoRA adapters, and scikit-learn for the logistic-regression head in the DFR baseline; exact versions are provided in the released code. All results in Section 4 are reported as means with standard deviation over three random trials. Results in Section 6 are from a single run per configuration.

Training Hyperparameters. Table 13 lists the hyperparameters for training the LoRA-based debiasing module. We use a single set of values across all datasets, with the exception of LoRA rank which is scaled by dataset size (see below).

LoRA Rank Selection. We set LoRA rank proportional to dataset size: $r = 4$ for datasets with approximately 700 samples (GoEmotions, CivilCom-

	ERM ($\alpha = 0$)		SG ($\alpha = 1$)		SG (adaptive α)		
Dataset	Acc.	WGA	Acc.	WGA	Acc.	WGA	Selected α
(a) Real-world benchmarks							
SST-2	0.911	–	0.911 ± 0.007	–	0.911 ± 0.007	–	1.0/1.0/1.0
Civil	0.880	0.737	0.884 ± 0.002	0.785 ± 0.010	0.884 ± 0.002	0.785 ± 0.010	1.0/1.0/1.0
MultiNLI	0.819	0.429	0.649 ± 0.033	0.116 ± 0.091	0.800 ± 0.012	0.565 ± 0.133	0.7/0.7/0.8
(b) Controlled benchmarks (shortcut distribution shift)							
Civil	0.522	0.216	0.577 ± 0.009	0.461 ± 0.015	0.577 ± 0.009	0.461 ± 0.015	1.0/1.0/1.0
MultiNLI	0.270	0.010	0.304 ± 0.018	0.012 ± 0.002	0.293 ± 0.022	0.010 ± 0.000	1.0/0.5/0.5
Yelp-ST	0.355	0.030	0.513 ± 0.006	0.150 ± 0.049	0.502 ± 0.007	0.257 ± 0.113	0.8/1.0/0.8
Yelp-Syn	0.419	0.035	0.516 ± 0.024	0.298 ± 0.024	0.530 ± 0.028	0.382 ± 0.055	0.9/0.9/0.9
GoEmo-ST	0.513	0.026	0.641 ± 0.011	0.415 ± 0.029	0.647 ± 0.014	0.386 ± 0.036	0.7/0.9/0.7
GoEmo-Syn	0.593	0.222	0.561 ± 0.008	0.074 ± 0.052	0.610 ± 0.005	0.222 ± 0.000	0.7/0.7/0.5
(c) In-distribution tests							
Civil	0.980	0.964	0.922 ± 0.010	0.844 ± 0.020	0.980 ± 0.000	0.964 ± 0.000	0.0/0.0/0.0
MultiNLI	0.908	0.840	0.705 ± 0.010	0.378 ± 0.074	0.891 ± 0.024	0.793 ± 0.066	0.0/0.0/0.8
Yelp-ST	0.671	0.376	0.585 ± 0.012	0.355 ± 0.034	0.634 ± 0.030	0.423 ± 0.044	0.6/1.0/0.0
Yelp-Syn	0.679	0.000	0.590 ± 0.017	0.185 ± 0.131	0.648 ± 0.035	0.112 ± 0.158	0.0/0.0/0.9
GoEmo-ST	0.896	0.667	0.513 ± 0.012	0.406 ± 0.012	0.896 ± 0.000	0.667 ± 0.000	0.0/0.0/0.0
GoEmo-Syn	0.928	0.771	0.491 ± 0.043	0.368 ± 0.053	0.928 ± 0.000	0.771 ± 0.000	0.0/0.0/0.0

Table 12: Comparison of ERM ($\alpha = 0$), SHORTCUT GUARDRAIL with a fixed $\alpha = 1$, and SHORTCUT GUARDRAIL with adaptively calibrated α on (a) real-world benchmarks, (b) controlled benchmarks under shortcut distribution shift, and (c) in-distribution tests. SG results are mean $\pm$ standard deviation over three random trials. The last column lists the α selected by adaptive calibration in each trial. SST-2 has no shortcut annotations, so WGA is not applicable (–).

Hyperparameter	**Value**	**Rationale**
`learning_rate`	1e-4	Standard for LoRA fine-tuning
`gradient_accumulation_steps`	1	Maximizes optimization steps; noisier gradients act as implicit regularization
`num_train_epochs`	1	Sufficient for ∼700–1000 samples
`temperature`	0.1	Standard InfoNCE temperature
`lora_r`	4 or 8	Scaled by dataset size (see below)

Table 13: Training hyperparameters for the debiasing module.

ments, MultiNLI, SST-2) and $r = 8$ for datasets with approximately 1000 samples (Yelp). The adapter must have enough capacity to capture the contrastive signal but not so much that it overfits to noisy predictions. With ∼700 samples producing ∼6000-8000 contrastive pairs, $r = 4$ provides a good bias-variance trade-off. Larger datasets generate more diverse pairs, so the additional capacity of $r = 8$ can be utilized without overfitting. Empirically, Yelp at $r = 4$ underperforms (WGA 0.146) compared to $r = 8$ (WGA 0.264), while other datasets show stable or improved results at $r = 4$.

Compute and model size. Our backbone is BERT-base-uncased (approximately 110M parameters); the LoRA debiasing module adds a small number of trainable parameters (on the order of 147K for $r = 4$, 295k for $r = 8$). All experiments were run on a single NVIDIA A6000 GPU. Training the debiasing module for one epoch takes 5 minutes 23 seconds, as measured on the SST-2 dataset.

A.8 Baseline Methods

Just Train Twice (JTT). We adopt the JTT method (Liu et al., 2021) using the authors' implementation on GitHub. JTT follows a two-stage training procedure without requiring group annotations. In Stage 1, a `BERT-base-uncased` model is fine-tuned on the shortcut-biased training set using standard empirical risk minimization (ERM). After training for T_1 epochs, the model's predictions are recorded, and misclassified examples are collected into an *error set*. In Stage 2, a fresh model is trained on the full training set where error-set examples are upweighted by a factor λ. Following the original paper, we perform a grid search over learning rate $\in \{10^{-3}, 10^{-4}, 10^{-5}\}$, $T_1 \in \{1, 2\}$, and $\lambda \in \{4, 5, 6\}$, with Stage 2 fixed at $T_2 = 10$, batch size 16, and weight decay 0.01. The best

configuration is selected by worst-group accuracy (WGA) on the validation set.

doNt Forget your Language (NFL). We use the NFL regularization method (Chew et al., 2024) via the authors' implementation on GitHub. NFL fine-tunes the base model while penalizing deviations from the initial pretrained weights to preserve linguistic representations. We specifically use the `NFL-CP` variant, which applies an ℓ_2 penalty on the squared difference between the fine-tuned and initial encoder weights across all `Linear` and `Embedding` layers. This penalty is scaled by λ_r and added to the cross-entropy loss. We use the default hyperparameters provided by the authors and select the checkpoint with the highest validation accuracy.

Deep Feature Reweighting (DFR). We implement DFR (Kirichenko et al., 2023) as provided in the NFL codebase. DFR freezes a pretrained `BERT` model and the `[CLS]`-token representations are extracted, standardized, and used to fit an ℓ_1-regularized logistic regression head on a randomly sampled small probe set of the training data (5%). The regularization strength is selected via cross-validation over $\{1.0, 0.7, 0.3, 0.1, 0.07, 0.03, 0.01\}$ using a 50/50 split of the probe set.

A.9 Positioning Against Existing Approaches

Table 14 situates SHORTCUT GUARDRAIL (SG) within the landscape of shortcut-mitigation and test-time adaptation methods, comparing along three binary axes that characterize the assumptions a method makes, together with a summary of the mechanism it employs. Comparison axes are explained below:

- *Deployment-time* indicates whether a method operates on a converged model using only test data, rather than intervening during training; consequently, it also entails no access to the original training set.
- *No shortcut annotations* indicates whether a method avoids requiring explicit shortcut or group labels.
- *Per-input attribution* indicates whether shortcut tokens are localized per example through gradient-based saliency, as opposed to being inferred from dataset-level statistics.

SG is the only method that satisfies all three axes: it operates at deployment time while using per-input attribution to localize shortcuts. Tent also operates at deployment time, but it targets general covariate shift via entropy minimization rather than spurious correlations and does not localize shortcuts, and is therefore not directly comparable on token-level shortcut benchmarks. We focus on token-level shortcuts, the form studied by most prior work on spurious correlations in text classification, including LTGR (Du et al., 2021), C2L (Choi et al., 2022), and NFL (Chew et al., 2024).

B Proof of the Theory in Section 5

We prove that deployment-time shortcut mitigation faces a lower performance upper bound than training-time mitigation.

Preliminaries. Let $\mathcal{X}, \mathcal{Y}, \mathcal{V}$ denote the input space (e.g., text sequences), the label space (e.g., sentiment classification), and the vocabulary from which the tokens are drawn, respectively. Let $\mathcal{D}_{\text{train}} \in (\mathcal{X} \times \mathcal{Y})^N$ denote a training set of sample size N, and $\mathcal{D}_{\text{test}} \in \mathcal{X}^M$ denote a test set of sample size M. We define $S^* \subseteq \mathcal{V} \times \mathcal{Y}$ as the true spurious correlation structure yielded by the set of shortcut tokens (see the group definitions in Appendix A.1.1). We model S^* as a random variable taking values in a finite set $\mathcal{S}$ with $|\mathcal{S}| \geq 2$. We decompose shortcut mitigation into two stages: *identification* and *intervention*. In the *identification* stage, we estimate the correlation structure S^* by either training-time ($\hat{S}_{\text{train}}$) or deployment-time methods ($\hat{S}_{\text{deploy}}$). In the *intervention* stage, we adapt the model parameters θ given $\hat{S}_{\text{train}}$ or $\hat{S}_{\text{deploy}}$. We define $I(S^*; \hat{S})$ as the mutual information between the true and estimated correlation structure, and $H(S^*)$ as the entropy of S^*, reflecting the structure's complexity. We further define $R(\hat{S}, S^*)$ as a scalar-valued mitigation performance (e.g., overall accuracy or worst-group accuracy) of the method that produces the estimate $\hat{S}$, and $g(\theta, \mathcal{D}_{\text{test}})$ as all possible deployment-time observables.

Assumption 2 (Markov chain)**.** *We assume that the following random variables form a Markov chain:*

$$S^* \;\to\; \mathcal{D}_{\text{train}} \;\to\; \theta \;\to\; g(\theta, \mathcal{D}_{\text{test}}).$$

That is, conditioned on $\mathcal{D}_{\text{train}}$, the model parameters θ are independent of S^*; and conditioned on θ, the deployment-time observables $g(\theta, \mathcal{D}_{\text{test}})$ are independent of $(\mathcal{D}_{\text{train}}, S^*)$.

Method	Deployment-time	No shortcut annotation	Per-input attribution	Approach
Group DRO (Sagawa et al., 2020)	✗	✗	✗	Worst-group reweighting
JTT (Liu et al., 2021)	✗	✓	✗	Error-based retraining
LTGR (Du et al., 2021)	✗	✓	✓	Attribution-guided training
DFR (Kirichenko et al., 2023)	✗	✓	✗	Last-layer retraining
NFL (Chew et al., 2024)	✗	✓	✗	Weight regularization
C2L (Choi et al., 2022)	✗	✓	✓	Contrastive learning
Tent (Wang et al., 2021)	✓	✓	✗	Entropy minimization
SHORTCUT GUARDRAIL (ours)	✓	✓	✓	Gradient-based attribution + Masked contrastive learning

Table 14: Positioning of SHORTCUT GUARDRAIL against existing shortcut-mitigation and test-time adaptation approaches. *Deployment-time* indicates operation on a converged model using only test data, which entails no access to the training set. *Per-input attribution* indicates that shortcut tokens are localized per example via gradient-based saliency rather than inferred from dataset-level statistics. SHORTCUT GUARDRAIL is the only method that operates at deployment time while using per-input attribution. Tent also operates at deployment time, but targets general covariate shift via entropy minimization rather than spurious correlations and does not localize shortcuts; it is therefore not directly comparable on token-level shortcut benchmarks.

Theorem 1 (Information Upper Bound)**.** The mutual information between the true spurious correlation structure S^* and any deployment-time estimate $\hat{S}_{\text{deploy}}$ satisfies:

$$I(S^*;\, \hat{S}_{\text{deploy}}) \;\le\; I(S^*;\, \theta) \;\le\; I(S^*;\, \mathcal{D}_{\text{train}}).$$

In particular, the model parameters lose $\Delta I = I(S^*;\, \mathcal{D}_{\text{train}}) - I(S^*;\, \theta) \ge 0$ units of information on S^* relative to the training set $\mathcal{D}_{\text{train}}$.

Proof of Theorem 1. By Assumption 2, $S^* \to \mathcal{D}_{\text{train}} \to \theta \to g(\theta, \mathcal{X}_{\text{test}})$ is a Markov chain. The Data Processing Inequality (DPI) states that for any Markov chain $A \to B \to C$, $I(A;C) \le I(A;B)$. Applying the DPI to the first three and last three elements of the chain yields

$$I(S^*;\, g(\theta, \mathcal{X}_{\text{test}})) \;\le\; I(S^*;\, \theta) \;\le\; I(S^*;\, \mathcal{D}_{\text{train}}).$$

Since $\hat{S}_{\text{deploy}}$ is a deterministic function of $g(\theta, \mathcal{X}_{\text{test}})$, a further application of the DPI yields $I(S^*;\, \hat{S}_{\text{deploy}}) \le I(S^*;\, g(\theta, \mathcal{X}_{\text{test}}))$. Combining these inequalities gives the result. Non-negativity of ΔI follows directly from the first inequality. □

Theorem 2 (Identification Error Bounds)**.** Let S^* take values in the finite set $\mathcal{S}$ with $|\mathcal{S}| \ge 2$. Define the Fano lower bounds on identification error:

$$L_{\text{deploy}} = \frac{H(S^*) - I(S^*;\, \theta) - 1}{\log(|\mathcal{S}| - 1)}, \tag{1}$$

$$L_{\text{train}} = \frac{H(S^*) - I(S^*;\, \mathcal{D}_{\text{train}}) - 1}{\log(|\mathcal{S}| - 1)}. \tag{2}$$

Then the misidentification probabilities satisfy:

$$P(\hat{S}_{\text{deploy}} \ne S^*) \;\ge\; L_{\text{deploy}}, \tag{3}$$

$$P(\hat{S}_{\text{train}} \ne S^*) \;\ge\; L_{\text{train}}, \tag{4}$$

and $L_{\text{deploy}} \ge L_{\text{train}}$.

Proof of Theorem 2. Fano's Inequality states that for any discrete random variable X taking values in a set of size K and any estimator $\hat{X}$ based on an observation Z:

$$P(\hat{X} \ne X) \;\ge\; \frac{H(X) - I(X;Z) - 1}{\log(K-1)}.$$

Deployment-time bound (3): Since $\hat{S}_{\text{deploy}}$ is a function of $g(\theta, \mathcal{X}_{\text{test}})$, which is itself a function of θ (given the independent test inputs), we have $I(S^*;\, \hat{S}_{\text{deploy}}) \le I(S^*;\, \theta)$ by Theorem 1. Applying Fano's Inequality with $X = S^*$, $\hat{X} = \hat{S}_{\text{deploy}}$, and $Z = \theta$ yields the result (noting that using θ in place of $g(\theta, \mathcal{X}_{\text{test}})$ can only decrease the Fano bound, so the inequality remains valid).

Training-time bound (4): The training-time estimator $\hat{S}_{\text{train}}$ has access to $\mathcal{D}_{\text{train}}$. Applying Fano's Inequality with $X = S^*$, $\hat{X} = \hat{S}_{\text{train}}$, and $Z = \mathcal{D}_{\text{train}}$ yields the result.

Ordering: From Theorem 1, $I(S^*;\, \theta) \le I(S^*;\, \mathcal{D}_{\text{train}})$, so $L_{\text{deploy}} \ge L_{\text{train}}$ holds. □

Assumption 1 (Monotonicity)**.** *There exists a non-decreasing, differentiable function $\rho : [0,1] \to \mathbb{R}$ such that a method with identification error L has expected mitigation performance $\rho(1 - L)$.*

Lemma 3 (Performance Ceiling from Identification Error)**.** *Under Assumption 1, any mitigation method whose identification error satisfies* $P(\hat{S} \neq S^*) \geq L$ *for some* $L \in [0, 1]$ *has expected performance bounded by:*

$$\mathbb{E}[R(\hat{S}, S^*)] \;\leq\; \rho(1 - L).$$

In particular, combining with Theorem 2:

$$\mathbb{E}[R(\hat{S}_{\text{deploy}}, S^*)] \;\leq\; \rho(1 - L_{\text{deploy}}), \quad (5)$$

$$\mathbb{E}[R(\hat{S}_{\text{train}}, S^*)] \;\leq\; \rho(1 - L_{\text{train}}). \quad (6)$$

Proof of Lemma 3. Since $P(\hat{S} \neq S^*) \geq L$, we have $1 - P(\hat{S} \neq S^*) \leq 1 - L$. Because ρ is non-decreasing (Assumption 1):

$$\mathbb{E}[R(\hat{S}, S^*)] = \rho\big(1 - P(\hat{S} \neq S^*)\big) \leq \rho(1 - L).$$

Equations (5) and (6) follow by substituting $L = L_{\text{deploy}}$ and $L = L_{\text{train}}$ from Theorem 2. □

Theorem 4 (Performance Ceiling Comparison)**.** Under Assumption 1, the performance ceilings for deployment- and training-time mitigation satisfy:

$$\rho(1 - L_{\text{deploy}}) \;\leq\; \rho(1 - L_{\text{train}}),$$

Moreover, the gap between the two bounds is:

$$\rho(1 - L_{\text{train}}) - \rho(1 - L_{\text{deploy}}) = \rho'(\xi) \cdot \frac{\Delta I}{\log(|\mathcal{S}| - 1)},$$

for some $\xi \in [1 - L_{\text{deploy}},\ 1 - L_{\text{train}}]$, where $\Delta I = I(S^*; \mathcal{D}_{\text{train}}) - I(S^*; \theta) \geq 0$.

Proof of Theorem 4. Inequality: By Theorem 2, $L_{\text{deploy}} \geq L_{\text{train}}$, which implies $1 - L_{\text{deploy}} \leq 1 - L_{\text{train}}$. Since ρ is non-decreasing (Assumption 1):

$$\rho(1 - L_{\text{deploy}}) \;\leq\; \rho(1 - L_{\text{train}}).$$

Gap characterisation: Since ρ is differentiable by Assumption 1, the Mean Value Theorem guarantees the existence of some $\xi \in [1 - L_{\text{deploy}},\ 1 - L_{\text{train}}]$ such that:

$$\rho(1 - L_{\text{train}}) - \rho(1 - L_{\text{deploy}}) \;=\; \rho'(\xi) \cdot (L_{\text{deploy}} - L_{\text{train}}),$$

where $\rho'(\xi)$ is the first-order derivative of ρ at ξ. Substituting the definitions (1) and (2):

$$L_{\text{deploy}} - L_{\text{train}} = \frac{I(S^*; \mathcal{D}_{\text{train}}) - I(S^*; \theta)}{\log(|\mathcal{S}| - 1)} = \frac{\Delta I}{\log(|\mathcal{S}| - 1)}.$$

Combining yields:

$$\rho(1 - L_{\text{train}}) - \rho(1 - L_{\text{deploy}}) = \rho'(\xi) \cdot \frac{\Delta I}{\log(|\mathcal{S}| - 1)}.$$

Since $\rho' \geq 0$ and $\Delta I \geq 0$ (Theorem 1), the gap is non-negative, confirming that the deployment-time performance ceiling is at most the training-time ceiling. □

C Extended Related Work

Shortcut Learning. Shortcut learning refers to the phenomenon where deep neural networks rely on superficial statistical patterns rather than task-relevant features for prediction (Geirhos et al., 2020). This issue is particularly prevalent in natural language understanding, where pretrained language models such as BERT (Devlin et al., 2019) have been shown to exploit spurious correlations present in training data (Du et al., 2023a; Tu et al., 2020). A growing body of work has examined how these shortcuts manifest at the token level: specific words or phrases become disproportionately associated with target labels, causing the model to over-rely on their presence regardless of context (Wang et al., 2022; Du et al., 2023b). For instance, in natural language inference, annotation artifacts such as negation words and lexical overlap between premise and hypothesis serve as well-documented shortcuts (Gururangan et al., 2018; McCoy et al., 2019). Recent work has further expanded the scope of analysis from individual tokens to concept-level correlations (Zhou et al., 2024a) and introduced controlled benchmarks with tunable shortcut strength for systematic evaluation (Zhou et al., 2024b). While these studies provide valuable diagnostic insights, they primarily focus on characterizing the problem rather than addressing it under realistic deployment conditions.

Training-Time Shortcut Mitigation. A range of methods have been proposed to mitigate shortcut learning during model training, with progressively lighter supervision requirements. Group Distributionally Robust Optimization (Group DRO; Sagawa et al., 2020) directly optimizes worst-group performance but requires explicit annotations of shortcut occurrence for every training example. To relax this requirement, Just Train Twice (JTT; Liu et al., 2021) identifies shortcut-vulnerable samples via a proxy model and upweights them during

retraining, eliminating the need for group labels while still requiring full access to the training data. Ensemble-based approaches (Clark et al., 2019) train a bias-only model alongside the main model to downweight shortcut-reliant predictions, though they assume prior knowledge of the bias type. Deep Feature Reweighting (DFR; Kirichenko et al., 2023) demonstrates that retraining only the last classification layer on a small group-balanced subset can recover robust performance, further reducing the retraining burden. More recently, NFL (Chew et al., 2024) applies counterfactual data augmentation and regularization to weaken spurious correlations without group labels. Despite this progression toward lighter assumptions, all of these methods still require some combination of training data access, label supervision, or prior knowledge of the shortcut type. Moreover, shortcut behavior is inherently distribution-dependent: correlations that are benign or even helpful during training may become harmful only when the deployment-time distribution shifts. None of these methods can adapt to such shifts after training, when the model is fixed and ground-truth labels are unavailable. This is precisely the setting we address in this work.

Adapting Models Beyond Training. A separate line of work in computer vision has explored test-time adaptation (TTA), where model parameters are updated using unlabeled test data to handle distribution shifts. Tent (Wang et al., 2021) adapts batch normalization layers by minimizing prediction entropy at test time, and subsequent work has improved its efficiency and stability (Niu et al., 2022). Other approaches adjust batch normalization statistics to the target distribution (Schneider et al., 2020; Nado et al., 2020) or introduce self-supervised auxiliary objectives for test-time training (Sun et al., 2020). However, Zhao et al. (2023) identify several failure modes of existing TTA methods, including sensitivity to batch size and the severity of distribution shift. More fundamentally, these methods are designed for vision architectures and rely on properties that do not transfer to text: batch normalization layers (absent in transformer encoders like BERT), continuous pixel inputs (as opposed to discrete tokens), and spatial augmentations (which have no direct analogue in language). Furthermore, existing TTA methods target general distribution shift rather than the structured problem of token-level shortcut reliance. Our work addresses this gap by proposing a deployment-time adaptation framework specifically designed for mitigating token-level shortcuts in text classifiers, without requiring batch normalization or continuous input assumptions.

Gradient-Based Attribution and Contrastive Learning. Our method builds on two technical foundations. The first is gradient-based attribution, which estimates the contribution of each input feature to the model's prediction. Integrated Gradients (Sundararajan et al., 2017) provides an axiomatic framework for this purpose, while Li et al. (2016) demonstrate that the simpler gradient $\times$ input method is effective for identifying influential tokens in NLP models. These techniques have been widely used for model interpretation, and more recently, Du et al. (2022) connect attribution to debiasing by using interpretation scores to guide a teacher-student training procedure. However, prior work treats attribution primarily as an explanatory tool or a training-time signal; no existing method leverages it as an operational mechanism for deployment-time intervention. The second foundation is contrastive learning. SimCLR (Chen et al., 2020) establishes the general framework of learning representations by contrasting augmented views of the same input, and Khosla et al. (2020) extend it to the supervised setting with improved robustness. In NLP, SimCSE (Gao et al., 2021) adapts contrastive objectives to sentence encoders using dropout as minimal augmentation, and FairFil (Cheng et al., 2021) applies contrastive learning to debias pretrained text representations with respect to social attributes. More closely related to our setting, C2L (Choi et al., 2022) combines contrastive learning with counterfactual augmentation to reduce reliance on spurious patterns, but still operates at training time with label supervision. Across these works, positive pairs are constructed through generic augmentations (dropout, random masking, synonym substitution) that are agnostic to which features the model actually relies on. Our method bridges these two lines by using gradient-based attribution to identify candidate shortcut tokens and constructing positive pairs through targeted masking of these tokens, so that the contrastive objective directly discourages shortcut reliance at deployment time.

D Ethics Considerations

Our work addresses the robustness of text classifiers to spurious correlations, which is directly

relevant to fairness in NLP systems. For example, in toxicity detection, spurious associations between demographic identity terms and the toxic label can lead to biased predictions against marginalized groups. Our method provides a practical tool for mitigating such biases at deployment time without requiring sensitive group annotations. All datasets used in this work are publicly available and have been widely used in prior research. We rely on the consent and ethical review processes of the original dataset creators. We do not collect or annotate any new data involving human subjects.

All datasets used are established public benchmarks. CivilComments and GoEmotions are sourced from public online comments and were released by their creators with steps to remove or anonymize user identifiers; we use only these released versions and do not attempt re-identification. CivilComments and GoEmotions contain offensive or toxic language by design, as this is the prediction target; we did not augment or annotate this content, and we display only short, non-gratuitous excerpts where necessary for illustration.

Licenses. The artifacts used in this work are publicly available and widely used in prior research. CivilComments (Borkan et al., 2019) is released under the CC0 license, GoEmotions (Demszky et al., 2020) and SST-2 (Wang et al., 2018) are under the Apache 2.0 license, and MultiNLI (Williams et al., 2018) is primarily under the Open American National Corpus (OANC) license, which allows all content to be freely used, modified, and shared under permissive terms; the Yelp review dataset (Zhang et al., 2015) and the Amazon Product Reviews (Ni et al., 2019) are distributed by their creators for research use. For models and training, we use the BERT encoder (Devlin et al., 2019), under the Apache 2.0 license, and LoRA (Hu et al., 2022), under the MIT license. Our use of all artifacts is consistent with their intended use for academic research. We release our own code under the Apache 2.0 license.

E Potential Risks

While our method reduces shortcut reliance, we acknowledge three potential risks. First, gradient-based attribution may mistakenly mask task-relevant tokens, suppressing genuine predictive signals. Second, adversarial actors aware of our token-level assumption could craft inputs exploiting multi-token shortcuts that our method fails to detect. Third, imperfect debiasing may create a false sense of security in high-stakes applications such as toxicity detection, where residual bias against marginalized groups may remain. We encourage careful evaluation before deployment.