When Prompts Interact: Assessing Prompt Arithmetic for Deconfounding under Distribution Shift

Zhecheng Sheng[1], Yongsen Tan[2], Xiruo Ding[3], Trevor Cohen[2], and Serguei Pakhomov[1]

[1]University of Minnesota
[2]University of Washington
[3]Stanford University

Abstract

In classification tasks, models may rely on confounding variables to achieve strong in-distribution performance, capturing spurious features that fail under distribution shift. This shortcut behavior leads to substantial degradation in out-of-distribution settings. Task arithmetic offers a potential solution by removing unwanted signals via subtraction of secondary model updates, but it typically requires full fine-tuning, which is computationally expensive. Prompt tuning provides a parameter-efficient alternative by adapting models through a small set of trainable virtual tokens. Task arithmetic on the resulting prompts presents an appealing alternative to operations on entire models, but the extent to which this approach can limit reliance on spurious features remains to be established. In this work, we study whether composing soft prompts through task arithmetic improves robustness to confounding shifts. We propose Hybrid Prompt Arithmetic (HyPA), which combines task prompts with linearized confounder prompts to counteract spurious correlations. Across multiple benchmarks, HyPA consistently improves the robustness-performance trade-off relative to prompt-arithmetic baselines under distribution shift. We further analyze how HyPA affects hidden representations and find evidence consistent with it mitigating confounding either by reducing the influence of confounder signals on predictions or by suppressing them in the representation. These results establish HyPA as a parameter-efficient and promising approach for improving robustness under confounding shifts in the evaluated setting. Our codebase is available at `https://github.com/zcsheng95/prompt_arithmetics`.

1 Introduction

Machine learning models trained via empirical risk minimization (ERM) have been shown to exploit confounding features that are predictive in the training distribution but not causally relevant to the target task. This phenomenon is known as shortcut learning (Geirhos et al., 2020). While such shortcuts can yield strong in-distribution performance, reliance on spurious correlations leads to substantial degradation under distribution shift (Recht et al., 2019; Koh et al., 2021). Prior mitigation strategies either introduce training-time constraints in the loss function which encourage a model to learn invariant features across confounding groups or environments (Sagawa et al., 2019; Arjovsky et al., 2019), or apply post-hoc corrections to a biased predictor (Ding et al., 2025; Sheng et al., 2025). Our work follows the latter line and develops a prompt-based debiasing method for transformers (Vaswani et al., 2017) without requiring modifications to the original training pipeline.

Specifically, we combine prompt tuning and task arithmetic to mitigate confounding effects in model predictions. While task arithmetic enables composition of model updates to modify model capabilities (Ilharco et al., 2022), it exhibits a trade-off between expressivity and disentanglement. In the context of a classification task with a known confounding variable, standard non-linear fine-tuning achieves strong performance on the primary

objective but entangles spurious confounder-related features, whereas linearized fine-tuning (Ortiz-Jimenez et al., 2023) improves disentanglement at the cost of reduced expressivity.

To address this limitation, we propose **Hybrid Prompt Arithmetic (HyPA)**, a two-phase framework that combines task vectors obtained from both non-linear and linearized prompt tuning. The key idea is to anchor the model in an optimized task solution and then estimate confounder-specific directions within a locally linearized parameter space. By composing these components through prompt arithmetic, HyPA effectively suppresses spurious correlations while preserving task-relevant representations. Empirically, we show that HyPA improves out-of-distribution robustness across multiple benchmarks while maintaining in-distribution performance.

2 Problem Setup and Preliminaries

We begin by formalizing the distribution shift problem setup and introducing the key components of our method. Consider a dataset $\mathcal{D}$ with $(X, Y, Z) \sim \mathcal{D}$, where $X \in \mathcal{X}$ denotes the input text, $Y \in \mathcal{Y}$ the task label, and $z \in \mathcal{Z}$ the confounding attribute. We assume that Z participates in the underlying data-generating process of both X and Y, corresponding to the causal structure (Pearl, 2009): $X \leftarrow Z \rightarrow Y$. Confounding shift occurs when $P(Y|Z)$ at test time differs from its training-time value. Given access to (X, Y, Z) during training, our goal is to learn a discriminative function $f_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ that is robust to confounding shift at test time, where Z may or may not be observed. Intuitively, this amounts to regularizing the learned model to avoid relying on spurious pathways from Z to Y when making predictions.

Confounding Shift. Confounding shift (Landeiro & Culotta, 2018) refers to a setting in which the conditional distribution of the label given the confounder differs between training and test environments, i.e., $P_{\text{train}}(Y \mid Z) \neq P_{\text{test}}(Y \mid Z)$, thereby violating the *i.i.d.* assumption commonly adopted in supervised learning. When confounding shift is present, a predictor f_θ that exploits correlations between Y and Z to learn the association between Y and X (via features in X that are also associated with Z) during training may experience performance degradation at inference time (i.e., during testing on data in which Z may be unobserved or distributed substantially differently from Z in the training data.). To quantify the degree of confounding shift, Ding et al. (2024) introduce an auxiliary variable α defined as

$$\alpha = \frac{P(Y = 1 \mid Z = 1)}{P(Y = 1 \mid Z = 0)}. \tag{1}$$

When $\alpha > 1$, positive labels are more prevalent in the subgroup $Z = 1$, and conversely when $\alpha < 1$. In other words, $\alpha > 1$ indicates that the confounder and the positive class label are found together more frequently than the positive class label without the confounder, and vice versa for $\alpha < 1$. Confounding shift occurs when the value of α differs between training and test time. Note this formulation is restricted to binary outcomes Y and binary confounders Z.

Prompt Tuning. Prompt tuning (Lester et al., 2021) is a parameter-efficient adaptation method that learns a set of additional parameters to condition a frozen base model on specific downstream tasks. Let f_θ denote a pretrained model with fixed parameters θ, $x = (x_1, x_2, \ldots, x_n)$ be an input token sequence, and y be the corresponding task label. Prompt tuning introduces a sequence of trainable soft prompt embeddings $\mathbf{P} = (p_1, p_2, \ldots, p_m)$, where $p_i \in \mathbb{R}^d$ and d is the dimensionality of the model's hidden state. With the soft prompt embeddings prepended to the input embeddings, the optimization objective becomes

$$\mathbf{P}^* = \underset{\mathbf{P}}{\operatorname{argmin}}\, \mathbb{E}_{(x,y)}[\mathcal{L}(f_\theta(x; \mathbf{P}), y)], \tag{2}$$

where θ is fixed and only $\mathbf{P}$ is optimized via backpropagation. $\mathcal{L}$ is an arbitrary loss function.

Task Vectors & Task Arithmetic. In contrast to prompt tuning in which the model's parameters θ are frozen, the task vector approach consists of deriving a task vector τ_t that

represents the element-wise difference between the fine-tuned model weights θ_t for task t and the pretrained weights θ_0, where both $\theta_0 \in \mathbb{R}^d$ and $\theta_t \in \mathbb{R}^d$. Ilharco et al. (2022) proposed methods to post-edit the pretrained model weights through task arithmetic, which linearly combines a set of task vectors (in this case, the "vectors" are the model weights in totality) from different tasks and modifies the performance of each task independently. Formally, task arithmetic can be expressed using the following formula:

$$\tau_t = \theta_t - \theta_0, \quad \theta_{\text{new}} = \theta_0 + \sum_{t=1}^{T} \lambda_t \tau_t, \tag{3}$$

where λ_t is a scalar coefficient for task t, which can be negative when the goal is to remove an undesirable effect. Ortiz-Jimenez et al. (2023) also note that, for a task vector set $\mathrm{T} = \{\tau_t\}_{t \in [T]}$, their task supports $\mathcal{D} = \{\mathcal{D}_t \subset \mathcal{X}\}$ should not overlap between different tasks t. However, in our experimental setup, we use the same input x to compute different task vectors based on their different labels.

Linearization. Prior work on the Neural Tangent Kernel (NTK) (Jacot et al., 2018) shows that a wide neural network can be locally approximated by a linear function around its initialization using a first-order Taylor expansion

$$f_\theta(x) \approx f_{\theta_0}(x) + \nabla_\theta f_{\theta_0}(x)^\top (\theta - \theta_0) = f_{\theta_0}(x) + \nabla_\theta f_{\theta_0}(x)^\top \tau. \tag{4}$$

Under this approximation, the parameter-to-function mapping is linear. This regime holds for sufficiently wide networks with small parameter updates, which is common in modern overparameterized models. The task vector τ in Equation 4 can be optimized directly.

Although NTK provides a tractable linearization of neural network training dynamics, it may inadequately model feature learning, which can result in reduced task performance (Chizat et al., 2019; Seleznova & Kutyniok, 2022).

3 Prompt Arithmetic in Causal Language Models

Our objective is to learn a classifier f satisfying $f(X) \perp Z \mid Y$, such that predictions remain invariant to shifts in $P(Y|Z)$ at test time. This requires the model to base its predictions on features $X_Z^\perp$ that are independent of Z. To this end, we adopt a post-hoc strategy inspired by task arithmetic, which posits that a model's behavior across different tasks can be approximated as additive under certain conditions (Ortiz-Jimenez et al., 2023). Motivated by the effectiveness of task arithmetic, we propose methods that selectively eliminate confounder-related signals through linear combinations of task vectors that represent the confounder and the target tasks. Most prior work on task arithmetic focuses on manipulating task vector τ using the full set of model parameters. In contrast, we propose performing task arithmetic only within the soft prompt space, a lightweight module inserted at the input embedding layer. This method formulation improves scalability and obviates the need for additional task-specific submodules such as classification heads (see Appendix D.1 for details).

3.1 Hybrid Prompt Arithmetic (HyPA)

Prompt Arithmetic (PA). We restrict the tunable parameters in task arithmetic to the soft prompt embeddings, denoted by $\theta := \mathbf{P}$. Prompt Arithmetic is then defined as

$$\tau_t = \mathbf{P}_t - \mathbf{P}_0, \quad \mathbf{P}_{\text{new}} = \mathbf{P}_0 + \sum_{t=1}^{T} \lambda_t \tau_t, \tag{5}$$

where $\mathbf{P}_0$ denotes the initialized soft prompt embeddings, in contrast to the pretrained model weights θ_0 used in standard task arithmetic. This set up is conceptually the same as the task prompt vector concept proposed in Belanec et al. (2025).

Prior work shows that task vectors can be obtained via either full non-linear fine-tuning (Ilharco et al., 2022) or a linearized approximation (Ortiz-Jimenez et al., 2023), referred to as

linear fine-tuning. Linear fine-tuning promotes better weight disentanglement of parametric function f, which facilitates effective task arithmetic, but it reduces expressivity because restricting updates to a linear regime can degrade single task performance compared to non-linear fine-tuning (Jin et al., 2025).

Hybrid Prompt Arithmetic (HyPA). To balance these trade-offs, we propose **Hybrid Prompt Arithmetic (HyPA)**, a two-phase training strategy that combines task vectors from the original non-linear model and its linearized approximation to derive a de-confounded model. Figure 1 illustrates the overall procedure for obtaining a debiased model from initialized or pretrained weights. The scaling factor λ controls the strength of confounder removal in the final model $f_{\theta_{\text{debias}}}$.

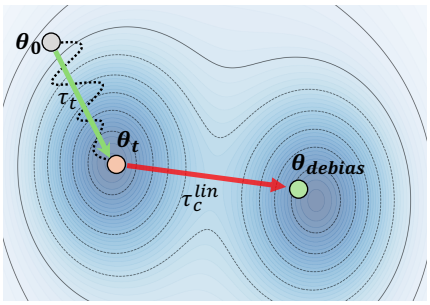

Figure 1: Traversing the task loss landscape along a linearized direction to achieve robustness while maintaining performance.

Recall that fine-tuning is performed using soft prompts, so the effective tunable parameters are $\theta := \mathbf{P}$. In the first phase, we fine-tune the model starting from pretrained weights $\mathbf{P}_0$ using the full non-linear network to obtain a task-specific model $\mathbf{P}_t$ and its corresponding task vector $\tau_t = \mathbf{P}_t - \mathbf{P}_0$. In the second phase, we constrain confounder fine-tuning to the *tangent space* around the task-specific weights $\mathbf{P}_t$, yielding a confounder task vector obtained from a linearized model:

$$f_\theta(x; \mathbf{P}_t) = f_\theta(x; \mathbf{P}_0 + \tau_t), \quad f_\theta^{\text{lin}}(x; \mathbf{P}_t + \tau_c^{\text{lin}}) = f_\theta^{\text{lin}}(x; \mathbf{P}_t) + \nabla_{\mathbf{P}} f_\theta(x; \mathbf{P}_t)^\top \tau_c^{\text{lin}}. \tag{6}$$

Here, τ_t denotes the task vector obtained from non-linear fine-tuning, while τ_c^{lin} denotes the confounder task vector derived from the linearized model f^{lin}.

Task arithmetic is then applied to negate confounder signals by combining the confounder task vector with the anchored task-specific weights. This operation aims to remove spurious correlations while preserving task-relevant features learned during non-linear fine-tuning:

$$\mathbf{P}_{\text{debias}} = \mathbf{P}_0 + \tau_t + \lambda \cdot \tau_c^{\text{lin}}. \tag{7}$$

The hyperparameter λ determines the extent to which τ_c^{lin} influences the behavior of the debiased model. The de-confounded model output is obtained by $f_\theta(x; \mathbf{P}_{\text{debias}})$ with $\mathbf{P}_{\text{debias}}$ being the plug-in adapter for the base model. The algorithm is formalized in Appendix A.

4 Experiments

Dataset. We manipulate the strength of spurious correlation, characterized by $P(Y \mid Z)$, through controlled sampling at both training and test time to construct experimental datasets. Specifically, we fix the marginal distributions P_Y and P_Z to be uniform for both training and testing, thereby isolating the effect of confounding shift. We vary the parameter α during training and testing to induce different levels of confounding shift. In our experiments, we set $\alpha_{\text{train}} \in \{0.2, 5.0\}$ to construct training datasets with strong spurious correlations. At test time, we generate a sequence of α values that are uniformly spaced in the log scale (since α is defined as a ratio) and evaluate model performance across these shifts.

Through the DeconDTN Toolkit (Tan et al., 2026), we construct data from five data sources for text categorization, ranging from clinical text to general online comments, to examine the generalizability of the proposed method. All the datasets focus on a binary prediction task from text input with a binary confounder label. Detailed descriptions and statistics of these datasets can be found in Appendix C.

Baselines. In our main results, we compare **HyPA** with two baseline variants: (i) **PA**, which derives both task and confounder vectors via non-linear fine-tuning, and (ii) **LinPA**, which

derives both vectors from linearized models. For the above two baselines, we adopt the original settings in task arithmetic where both task and confounder models are fine-tuned from the same initialization point. We use GPT-2 (Radford et al., 2019) as the base model and a prompt length of 100 in the main results. We demonstrate the comparison results with a broader selection of baseline methods in Section 7.

Model Selection. Since test-time shift is unknown in practice, we tune the hyperparameter λ using the worst-group AUPRC on a held-out validation set, which has the same distribution as the training set. When $\alpha_{\text{train}} < 1$, the worst-performing group corresponds to samples with $Z = 1$, and when $\alpha_{\text{train}} > 1$, the worst-performing group corresponds to samples with $Z = 0$. This aligns with previous practice when confounder attributes are available during training and validation (Sagawa et al., 2019; Gulrajani & Lopez-Paz, 2020; Idrissi et al., 2022; Yang et al., 2023).

Evaluation Metrics. We use four metrics to evaluate the performance of the debiased model under varying degrees of confounding shift. The Out-of-Distribution (OOD) score measures performance under extreme confounding shifts at test time. The Absolute Slope quantifies performance variation across different levels of confounding shift, while the Adjusted Integral jointly captures both performance and robustness in a single measure.

OOD Score. We evaluate the model *performance* on the classification task of interest when the confounder distribution in the test set is exactly the reciprocal of the training distribution, representing a case of severe confounding shift:

$$\text{OOD Score} = \mathbf{s}_\alpha, \quad \text{where } \alpha = \frac{1}{\alpha_{\text{train}}} \tag{8}$$

Absolute Slope. Given a collection of data points generated from a list of α_{test}, the Absolute Slope measures the *robustness* of the model's performance on this test set collection. It is quantified as the absolute value of the coefficient from the linear regression fitted between the model performance score $\mathbf{s}$ and $\log(\alpha_{\text{test}})$:

$$\mathbf{s} = \beta_0 + \beta \times \log(\alpha_{\text{test}}), \tag{9}$$

where β_0 is the intercept and $|\beta|$ is the measure of interest for robustness. Intuitively, a flatter regression line signifies greater robustness of the model under confounding shift.

The two metrics above each assess a single aspect of model behavior separately: *performance* and *robustness*, respectively. To characterize the interaction between both of these aspects of model behavior, we introduce two metrics that jointly capture both performance and robustness: Area Under the Shift Curve (AUSC) and Adjusted Integral (AI).

Area Under Shift Curve (AUSC). The Area Under the Shift Curve captures the average of the AUPRCs across different test-time confounding distribution shifts (i.e., mean AUPRC over the support of *alpha*). Larger AUSC values indicate consistently higher AUPRCs under various distribution shifts.

$$\text{AUSC} = \frac{1}{\log(a) - \log(b)} \int_{\log(b)}^{\log(a)} \mathbf{s}_\alpha \, d(\log \alpha) \tag{10}$$

Adjusted Integral. This metric is defined as the absolute difference between out-of-distribution (shift) performance and in-distribution performance, augmented by a penalty term that compensates for deficiencies in in-distribution performance.

$$\text{Adjusted Integral} = \frac{1}{\log(a) - \log(b)} \int_{\log(b)}^{\log(a)} |\mathbf{s}_{\text{ID}} - \mathbf{s}_\alpha| \, d(\log \alpha) \; + \; (1 - \mathbf{s}_{\text{ID}}), \tag{11}$$

where $\mathbf{s}_{\text{ID}}$ denotes the in-domain model performance (i.e., at $\alpha = \alpha_{\text{train}}$), and a and b denote the maximum and minimum values in the test-time shift set α_{test}, respectively. The term $1 - \mathbf{s}_{\text{ID}}$ penalizes poor in-domain performance.

For all metrics above, we use Area Under the Precision and Recall Curve (AUPRC) as the performance measure $\mathbf{s}$ and evaluate shifts using uniform spacing in $\log_{10} \alpha$.

5 Main Results

Table 1 shows the performance of PA, LinPA, and HyPA across five benchmarks under two training shift settings ($\alpha_{\text{train}} = 0.2$ and $\alpha_{\text{train}} = 5.0$). Overall, HyPA consistently achieves the strongest joint performance, obtaining the best or near-best results in terms of AUSC and Adjusted Integral across datasets and shift settings. In terms of out-of-distribution

Table 1: Performance–robustness trade-offs across five datasets under two strengths of training-time spurious correlation. Values are means with standard deviations in parentheses over five random seeds; the best result in each setting is bolded.

Method	α_{train}	SHAC	MIMIC	Hate Speech	Civil Comments	Amazon Reviews
				OOD AUPRC ↑		
PA	0.2	0.864 (.031)	0.413 (.037)	0.453 (.028)	0.534 (.019)	0.951 (.009)
LinPA	0.2	0.610 (.082)	**0.511 (.011)**	0.454 (.027)	0.536 (.018)	0.854 (.049)
HyPA	0.2	**0.919 (.034)**	0.443 (.070)	**0.656 (.106)**	**0.788 (.047)**	**0.967 (.020)**
PA	5.0	0.581 (.140)	0.451 (.053)	0.497 (.170)	0.706 (.021)	0.957 (.017)
LinPA	5.0	**0.667 (.087)**	**0.518 (.027)**	**0.560 (.070)**	0.566 (.049)	0.869 (.074)
HyPA	5.0	0.654 (.200)	0.509 (.043)	0.532 (.178)	**0.751 (.045)**	**0.978 (.003)**
				Absolute Slope ↓		
PA	0.2	0.063 (.013)	0.225 (.078)	0.289 (.023)	0.271 (.012)	0.020 (.009)
LinPA	0.2	0.080 (.043)	**0.010 (.013)**	0.201 (.063)	**0.029 (.016)**	0.012 (.006)
HyPA	0.2	**0.022 (.016)**	0.169 (.134)	**0.133 (.075)**	0.073 (.048)	**0.011 (.017)**
PA	5.0	0.195 (.066)	0.124 (.111)	0.282 (.011)	0.141 (.014)	0.014 (.008)
LinPA	5.0	**0.105 (.048)**	**0.023 (.011)**	**0.082 (.059)**	**0.072 (.027)**	0.011 (.004)
HyPA	5.0	0.174 (.102)	0.051 (.096)	0.211 (.138)	0.103 (.036)	**0.002 (.002)**
				AUSC ↑		
PA	0.2	0.915 (.015)	0.565 (.026)	0.666 (.009)	0.748 (.010)	0.967 (.006)
LinPA	0.2	0.662 (.053)	0.526 (.018)	0.595 (.028)	0.552 (.026)	0.863 (.047)
HyPA	0.2	**0.933 (.014)**	**0.567 (.023)**	**0.746 (.036)**	**0.847 (.009)**	**0.973 (.011)**
PA	5.0	0.721 (.099)	**0.550 (.025)**	0.624 (.037)	0.815 (.008)	0.972 (.005)
LinPA	5.0	0.651 (.036)	0.526 (.007)	0.578 (.040)	0.614 (.068)	0.868 (.074)
HyPA	5.0	**0.751 (.094)**	0.546 (.036)	**0.651 (.036)**	**0.830 (.017)**	**0.980 (.001)**
				Adjusted Integral ↓		
PA	0.2	0.088 (.016)	0.442 (.019)	0.342 (.011)	0.257 (.010)	0.034 (.007)
LinPA	0.2	0.338 (.053)	0.493 (.011)	0.410 (.027)	0.464 (.025)	0.214 (.090)
HyPA	0.2	**0.068 (.014)**	**0.438 (.019)**	**0.255 (.037)**	**0.153 (.009)**	**0.028 (.012)**
PA	5.0	0.284 (.100)	**0.452 (.026)**	0.380 (.037)	0.191 (.009)	0.029 (.005)
LinPA	5.0	0.417 (.068)	0.501 (.023)	0.445 (.051)	0.390 (.070)	0.166 (.098)
HyPA	5.0	**0.255 (.098)**	0.455 (.036)	**0.354 (.038)**	**0.172 (.018)**	**0.021 (.001)**

predictive performance (OOD AUPRC), HyPA generally attains the highest AUPRC values, particularly when $\alpha_{\text{train}} = 0.2$, with substantial improvements on Hate Speech and Civil Comments. Under the opposite training shift ($\alpha_{\text{train}} = 5.0$), HyPA remains competitive and achieves the best performance on several datasets.

For robustness, measured by Absolute Slope, LinPA often achieves shallower slopes, indicating stronger invariance to varying confounding shift. However, this typically comes at the expense of lower predictive performance due to the linearization (Appendix B). In contrast, HyPA achieves a more favorable balance between performance and stability, which leads to consistently higher joint metrics (e.g., AUSC and Adjusted Integral). It is worth noting that HyPA consistently outperforms the PA baseline across metrics and configurations.

6 Analysis of HyPA Results

We investigate **how** HyPA achieves strong performance and robustness, providing further empirical analysis of its behavior. We examine how prompt arithmetic steers hidden representations over different λs (§6.1) and then propose a mechanistic diagnostic to investigate how HyPA mitigates confounding across datasets (§6.2).

6.1 Hidden Representation Shift in HyPA

We begin by visualizing how the hidden representations produced by HyPA evolve as the scaling coefficient λ in Equation 7 varies. Here we provide the results on *Civil Comments* with $\alpha_{\text{train}} = 0.2$ and λ ranging between 0.0 and 0.8. As Figure 2 suggests, the hidden representations colored by the confounder labels start to blend as λ increases, suggesting HyPA diminishes the model's ability to distinguish between confounder groups based on confounder-related features by drawing their representations together in the hidden space.

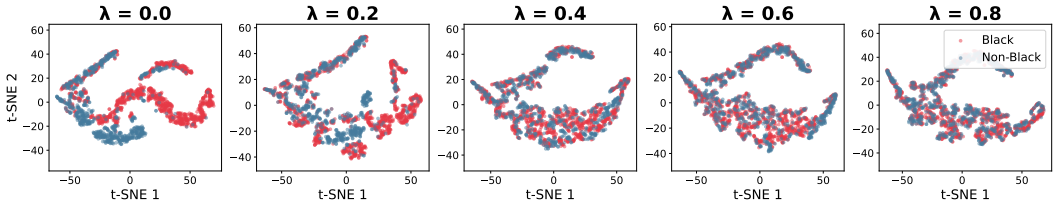

Figure 2: t-SNE plot on the hidden representations for different λ, colored by the confounder label. The task model is trained on the *Civil Comments* dataset with $\alpha = 0.2$ and evaluated on a randomly sampled test set with $\alpha = 0.7$.

The t-SNE plot offers diagnostic signals that HyPA is effectively altering the hidden representations of confounder-related features. The gradual blending of the two confounder clusters in this example provides an illustration of HyPA reducing the prominence of confounder-related structure in the hidden representation space. Results of other experimental settings are provided in Appendix B, where these blending behaviors vary among datasets.

6.2 How Does HyPA Adjust for Confounders

A natural follow-up question is how HyPA improves robustness: does it explicitly remove confounder information, or reshape the learned confounder geometry in the latent space?

Let h denote the hidden representation and define the task direction as $W_{\text{task}} = W_U[\text{class1}] - W_U[\text{class2}]$, where W_U is the unembedding matrix (i.e. the output embeddings - one for each class). We quantify the model's sensitivity to the confounder as

$$\text{Sen}(\lambda) = |W_{\text{task}} \cdot v_{\text{conf}}(\lambda)|, \tag{12}$$

where $v_{\text{conf}}(\lambda) = \mathbb{E}[h(\lambda) \mid Z{=}1] - \mathbb{E}[h(\lambda) \mid Z{=}0]$ denotes the centroid difference between contextual embedding representations of confounder groups in the hidden space at scaling factor λ. Here, W_{task} is fixed, and $v_{\text{conf}}(\lambda)$ serves as a proxy for the confounder direction in the hidden space. This quantity measures the projection of the confounder direction onto the task direction at each λ.

To assess whether confounder information remains encoded in the representation, we train a linear probe to predict the confounder using the top 50 principal components of h as features. We report probe accuracy averaged over 5-fold cross-validation to ensure robustness.

As shown in Figure 3, empirical results across five datasets suggest that, under the same test-time α, HyPA mitigates spurious correlations through dataset-dependent mixtures of two effects: reducing the influence of confounder-related variation on task prediction, and weakening confounder information in the representation.

For SHAC and Hate Speech, probe accuracy remains high across the λ sweep while sensitivity decreases at extreme λ values. This pattern is consistent with a blocking-like effect: confounder information remains decodable from the hidden representation, while its influence on task prediction is reduced. In contrast, for MIMIC, sensitivity closely tracks probe accuracy, and both decrease substantially at extreme λ values. This behavior is more consistent with representational removal, in which HyPA weakens the confounder information encoded in the hidden space itself. Civil Comments and Amazon Reviews exhibit intermediate behavior, suggesting that both effects may be present to varying degrees.

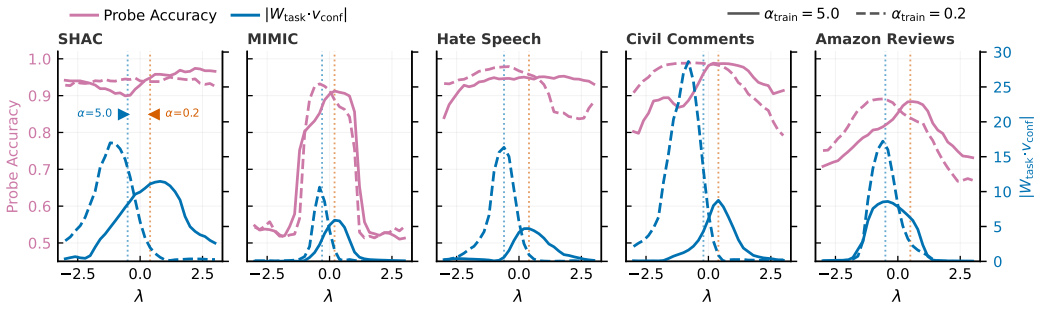

Figure 3: The Sensitivity and Probe Accuracy changes across λ sweep. The dotted vertical lines indicate the oracle-optimal λ in each setting ($\alpha_{\text{train}} = 0.2$ and $\alpha_{\text{train}} = 5.0$).

Overall, our analysis does not uniquely identify the underlying mechanism, but the joint behavior of sensitivity and probe accuracy provides consistent evidence that HyPA mitigates confounding through dataset-dependent mixtures of blocking-like and removal-like effects.

7 Comparisons with Other Baselines

Besides the comparisons within prompt arithmetic variants, we also conduct experiments on the same five benchmarks with other established baseline approaches in domain adaptation or distribution shift, across five random seeds. The included baselines are GroupDRO (Sagawa et al., 2019), JTT (Liu et al., 2021), DFR (Kirichenko et al., 2023), IRM (Arjovsky et al., 2019) and DANN (Ganin et al., 2016). Besides the existing baselines, we have also investigated two variants of full-parameter task arithmetic: TA and HyTA, corresponding to PA and HyPA, respectively.

Figure 4 presents a scatter plot comparing the number of trainable parameters with the average AUPRC across test-time shifts. The results show that HyPA achieves performance comparable to full-parameter task arithmetic methods, including TA and HyTA, while requiring substantially fewer trainable parameters. This suggests that operating in the soft-prompt space can retain much of the robustness benefit of full-model adaptation without incurring its substantially higher parameter and computational costs. HyPA also outperforms established baselines on nearly all benchmarks, demonstrating stronger robustness under confounding shifts. Overall, these results indicate that HyPA offers a favorable performance–efficiency trade-off and provides a practical, parameter-efficient, and scalable alternative to full-parameter debiasing methods.

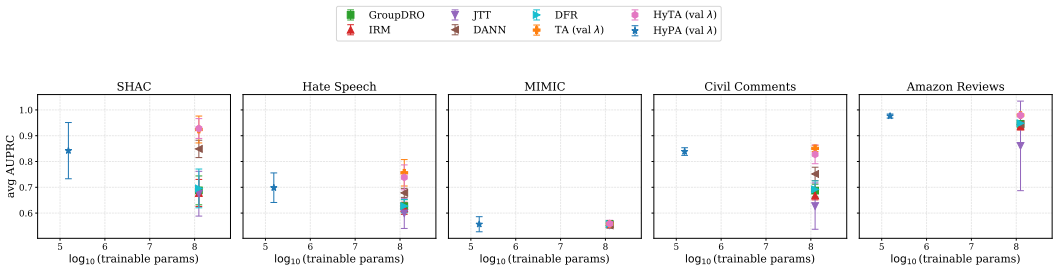

Figure 4: Comparisons of HyPA, Full Param Task Arithmetic, GroupDRO, IRM, JTT, DFR and DANN. Similar to PA and HyPA, we abbreviate Full Param Task Arithmetic approaches as Task Arithmetic (TA) and Hybrid Task Arithmetic (HyTA) respectively.

8 Ablation Studies

Prompt Length. We ablate the number of virtual tokens used in the soft prompts. While the main results are reported with a prompt length of 100, we additionally evaluate prompt lengths of 50 and 150. The results in Figure 5 show that prompt length does not have a consistent effect on the Adjusted Integral across datasets. In some cases, a longer prompt provides slight improvements, while in others the gains are marginal or even reversed, suggesting that increasing prompt capacity does not uniformly translate to better robustness under confounding shift. The main takeaway from this ablation is that $L = 100$ provides a strong and stable default across all five datasets, and overall HyPA is not highly sensitive to prompt length within the ablated range.

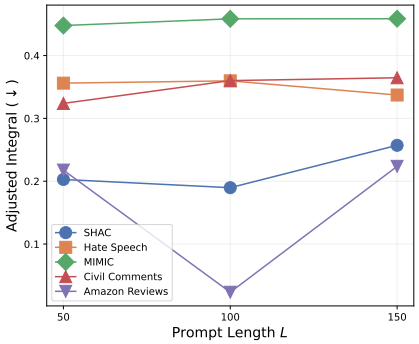

Figure 5: Prompt length ablation with $L = 50, 100, 150$ in five datasets.

Anchor of τ_c^{lin}. Compared to prior work on task arithmetic with linearization, a key distinction of HyPA is that it anchors the confounder task vector at the fine-tuned weights rather than the pretrained initialization. We further ablate this design choice by recalculating τ_c^{lin} with respect to the initial weights, and observe a consistent increase in Adjusted Integral (Figure 6(a)). This suggests that anchoring at the fine-tuned solution better preserves task performance while improving robustness to confounding shift under most settings. We provide the corresponding ID and OOD AUPRC results in Appendix B with additional analysis.

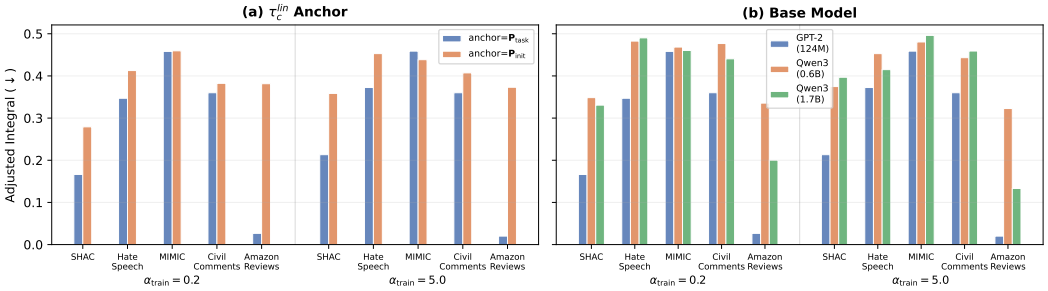

Figure 6: Ablation results for **(a)**: Anchor of τ_c^{lin} and **(b)**: Base Model across five datasets.

Base Model. We compare GPT-2 (124M) with Qwen3-0.6B, Qwen3-1.7B, and Qwen3-8B (Yang et al., 2025). Within the Qwen3 family, absolute debiasing performance generally improves with model size, with Qwen3-8B achieving the strongest results in Table 2. However, Figure 6(b) shows that GPT-2 outperforms Qwen3-0.6B and Qwen3-1.7B across several settings, suggesting that debiasing effectiveness does not scale monotonically across model families.

Table 2 further separates absolute performance from HyPA's marginal benefit over PA. HyPA substantially improves both average AUPRC and Adjusted Integral for Qwen3-0.6B and Qwen3-1.7B, but provides little additional gain at 8B. Thus, larger Qwen models perform better in absolute terms, while the relative advantage of HyPA does not increase consistently with scale. This pattern may reflect architectural differences, model depth, or the use of a fixed prompt length across models, which we leave for future investigation.

9 Related Work

Distribution Shift. Machine learning models and data-driven systems often suffer performance degradation under distribution shift at deployment, where the joint distribution of

Table 2: Comparisons across the SHAC and Hate Speech dataset under both α_{train} values.

	Average AUPRC ↑			Adjusted Integral ↓		
Qwen3 Model	PA	HyPA	Δ value	PA	HyPA	Δ ratio
0.6B	0.8178	**0.8433**	**+0.0255**	0.1855	**0.1580**	**-14.8%**
1.7B	0.8070	**0.8450**	**+0.0380**	0.1980	**0.1583**	**-20.1%**
8B	**0.9128**	0.9123	−0.0005	0.0903	**0.0895**	-0.8%

inputs and labels differs from that of the training environment. Prior work has investigated various forms of shift, including domain generalization (Zhou et al., 2022), covariate shift (Shimodaira, 2000; Ruan et al., 2021), label shift (Lipton et al., 2018), and subpopulation shift (Koh et al., 2021; Yang et al., 2023). In this work, we focus on *confounding shift* (Landeiro & Culotta, 2018; Ding et al., 2024), a structured form of subpopulation shift in which a confounder Z causally influences both the input X and the label Y. We place particular emphasis on developing parameter-efficient methods to mitigate confounding shift in large neural language models, leveraging prompt linearization to enable effective task arithmetic.

Bias Mitigation. Our work addresses bias mitigation under confounding shifts. Existing approaches are typically categorized into pre-processing, in-processing, and post-processing methods (Angwin et al., 2016). Pre-processing methods modify the training data through feature editing (Feldman et al., 2015; Calmon et al., 2017) or reweighting (Kamiran & Calders, 2012), while in-processing approaches incorporate fairness constraints directly into model optimization (Agarwal et al., 2018; Zhang et al., 2018). Post-processing methods instead adjust the outputs of a trained model to satisfy fairness criteria (Hardt et al., 2016; Pleiss et al., 2017), without requiring access to or retraining of the underlying model. HyPA falls into the post-processing category and operates without modifying the original training pipeline. Compared to prior post-processing methods, which typically rely on output calibration, threshold adjustment or weight merging () HyPA leverages parameter-efficient prompt composition to intervene at the representation level. This enables more flexible mitigation of confounding effects while preserving the efficiency and modularity of post-hoc approaches, making it particularly suitable for large language models where full retraining is impractical.

10 Conclusion

We presented Hybrid Prompt Arithmetic (HyPA), a parameter-efficient method for improving robustness under confounding shift. By combining non-linear task prompt tuning with linearized confounder tuning, HyPA balances expressivity and disentanglement and allows prompt arithmetic to mitigate spurious correlations while preserving task performance.

Across multiple datasets and shift settings, HyPA consistently achieves a stronger robustness-performance trade-off than prompt-arithmetic baselines based solely on non-linear or linearized fine-tuning. Our analysis further suggests that HyPA operates in a dataset-dependent manner: in some cases, HyPA primarily reduces the influence of confounder-related variation on task prediction, whereas in others it is also associated with weaker confounder information in the representation itself. These results position HyPA as a simple and effective post-hoc intervention for improving robustness in the evaluated prompt-tuned language model setting.

Despite these promising results, several limitations remain. First, our analysis is primarily empirical, and a stronger theoretical account is needed to characterize when and why hybrid prompt arithmetic can isolate useful confounder directions. Second, although HyPA acts only in soft prompt space, the precise way prompt updates propagate through the frozen backbone and affect the unembedding layer remains insufficiently understood. Third, as discussed in Appendix B.1, model selection under unknown test-time shift remains challenging, and developing validation criteria that better align with downstream robustness is an important direction for future work.

Ethics Statement

This work addresses robustness under confounding shift in settings where spurious correlations can lead to brittle or unfair predictions. A potential benefit of HyPA is reduced reliance on confounder-related signals, which may improve robustness and reduce group disparities under distribution shift. However, such improvements do not guarantee fairness or robustness in deployment. The effectiveness of any debiasing method depends on the quality of confounder annotations, the data distribution, and the target environment. In particular, datasets drawn from clinical and online text domains may contain historical biases, annotation artifacts, and subgroup imbalance, and errors in these settings can cause disproportionate harm. Our results should therefore be interpreted as evidence of improved robustness on the studied benchmarks, not as a complete solution to bias or fairness concerns.

Acknowledgments

This work was supported by U.S. National Library of Medicine grant (R01LM014056).

References

Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach. A reductions approach to fair classification. In *International Conference on Machine Learning*, pp. 60–69. PMLR, 2018.

Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias. *ProPublica*, 2016. URL https://github.com/propublica/compas-analysis.

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

Robert Belanec, Simon Ostermann, Ivan Srba, and Maria Bielikova. Task prompt vectors: Effective initialization through multi-task soft-prompt transfer, 2025. URL https://arxiv.org/abs/2408.01119.

Daniel Borkan, Lucas Dixon, Jeffrey Sorensen, Nithum Thain, and Lucy Vasserman. Nuanced metrics for measuring unintended bias with real data for text classification. In *Companion proceedings of the 2019 world wide web conference*, pp. 491–500, 2019.

Flavio Calmon, Dennis Wei, Bhanukiran Vinzamuri, Karthikeyan Natesan Ramamurthy, and Kush R Varshney. Optimized pre-processing for discrimination prevention. *Advances in Neural Information Processing Systems*, 30, 2017.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

Ona De Gibert, Naiara Perez, Aitor García-Pablos, and Montse Cuadros. Hate speech dataset from a white supremacy forum. *arXiv preprint arXiv:1809.04444*, 2018.

Xiruo Ding, Zhecheng Sheng, Meliha Yetişgen, Serguei Pakhomov, and Trevor Cohen. Backdoor adjustment of confounding by provenance for robust text classification of multi-institutional clinical notes. In *AMIA Annual Symposium Proceedings*, volume 2023, pp. 923, 2024.

Xiruo Ding, Zhecheng Sheng, Brian Hur, Justin Tauscher, Dror Ben-Zeev, Meliha Yetişgen, Serguei Pakhomov, and Trevor Cohen. Tailoring task arithmetic to address bias in models trained on multi-institutional datasets. *Journal of Biomedical Informatics*, 168:104858, August 2025. ISSN 1532-0464. doi: 10.1016/j.jbi.2025.104858.

Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *proceedings of the 21st ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 259–268, 2015.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario March, and Victor Lempitsky. Domain-Adversarial Training of Neural Networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016. ISSN 1533-7928.

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

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

Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in Neural Information Processing Systems*, 29, 2016.

Badr Youbi Idrissi, Martin Arjovsky, Mohammad Pezeshki, and David Lopez-Paz. Simple data balancing achieves competitive worst-group-accuracy. In *Conference on Causal Learning and Reasoning*, pp. 336–351. PMLR, 2022.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *ArXiv*, abs/2212.04089, 2022. URL https://api.semanticscholar.org/CorpusID:254408495.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Ruochen Jin, Bojian Hou, Jiancong Xiao, Weijie Su, and Li Shen. Fine-tuning attention modules only: Enhancing weight disentanglement in task arithmetic, 2025. URL https://arxiv.org/abs/2407.07089.

Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3(160035), 2016. doi: https://doi.org/10.1038/sdata.2016.35.

Faisal Kamiran and Toon Calders. Data preprocessing techniques for classification without discrimination. *Knowledge and Information Systems*, 33(1):1–33, 2012.

Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last Layer Re-Training is Sufficient for Robustness to Spurious Correlations, June 2023.

Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, et al. Wilds: A benchmark of in-the-wild distribution shifts. In *International conference on machine learning*, pp. 5637–5664. PMLR, 2021.

Virgile Landeiro and Aron Culotta. Robust text classification under confounding shift. *Journal of Artificial Intelligence Research*, 63:391–419, 2018.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *International conference on machine learning*, pp. 3122–3130. PMLR, 2018.

Evan Z. Liu, Behzad Haghgoo, Annie S. Chen, Aditi Raghunathan, Pang Wei Koh, Shiori Sagawa, Percy Liang, and Chelsea Finn. Just Train Twice: Improving Group Robustness without Training Group Information. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 6781–6792. PMLR, July 2021.

Kevin Lybarger, Mari Ostendorf, and Meliha Yetisgen. Annotating social determinants of health using active learning, and characterizing determinants using neural event extraction. *Journal of Biomedical Informatics*, 113:103631, 2021.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 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)*, pp. 188–197, 2019.

Guillermo Ortiz-Jimenez, Alessandro Favero, and Pascal Frossard. Task arithmetic in the tangent space: Improved editing of pre-trained models. *Advances in Neural Information Processing Systems*, 36:66727–66754, 2023.

Judea Pearl. *Causality*. Cambridge University Press, September 2009. ISBN 978-0-521-89560-6.

Geoff Pleiss, Manish Raghavan, Felix Wu, Jon Kleinberg, and Kilian Q Weinberger. On fairness and calibration. *Advances in Neural Information Processing Systems*, 30, 2017.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International conference on machine learning*, pp. 5389–5400. PMLR, 2019.

Yangjun Ruan, Yann Dubois, and Chris J Maddison. Optimal representations for covariate shift. *arXiv preprint arXiv:2201.00057*, 2021.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Mariia Seleznova and Gitta Kutyniok. Neural tangent kernel beyond the infinite-width limit: Effects of depth and initialization. In *International Conference on Machine Learning*, pp. 19522–19560. PMLR, 2022.

Zhecheng Sheng, Xiruo Ding, Brian Hur, Changye Li, Trevor Cohen, and Serguei V. S. Pakhomov. Mitigating confounding in speech-based dementia detection through weight masking. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10419–10434, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.514. URL https://aclanthology.org/2025.acl-long.514/.

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

Yongsen Tan, Zhecheng Sheng, Xiruo Ding, Serguei V S Pakhomov, and Trevor Cohen. Decondtn-toolkit: A library for evaluation and enhancement of robustness to provenance shift. In Elizabeth Healey, Jason Fries, Tom Pollard, Shengpu Tang, Anna Zink, Tom Hartvigsen, Monica Agrawal, Sam Finlayson, Benjamin Glicksberg, Brett Beaulieu-Jones, Kai Wang, Daseyra Fontalvo, Tasmie Sarker, Irene Chen, and Emily Alsentzer (eds.), *Proceedings of the 7th Conference on Health, Inference, and Learning*, volume 333 of *Proceedings of Machine Learning Research*, pp. 721–753. PMLR, 29–30 Jun 2026. URL https://proceedings.mlr.press/v333/tan26a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Bertie Vidgen, Tristan Thrush, Zeerak Talat, and Douwe Kiela. Learning from the worst: Dynamically generated datasets to improve online hate detection. In *Proceedings of the 59th annual meeting of the Association for Computational Linguistics and the 11th international joint conference on natural language processing (volume 1: long papers)*, pp. 1667–1682, 2021.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuzhe Yang, Haoran Zhang, Dina Katabi, and Marzyeh Ghassemi. Change is hard: A closer look at subpopulation shift. *arXiv preprint arXiv:2302.12254*, 2023.

Brian Hu Zhang, Blake Lemoine, and Margaret Mitchell. Mitigating unwanted biases with adversarial learning. In *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 335–340, 2018.

Kaiyang Zhou, Ziwei Liu, Yu Qiao, Tao Xiang, and Chen Change Loy. Domain generalization: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(4): 4396–4415, 2022.

Appendix

A HyPA algorithm

Algorithm 1 Hybrid Prompt Arithmetic (HyPA)

Input: Frozen language model f_θ, initial soft prompt $\mathbf{P}_0$, dataset $\mathcal{D} = \{(x_i, y_i, z_i)\}_{i=1}^N$, scale λ

Output: De-confounded soft prompt $\mathbf{P}_{\text{debias}}$

Phase 1: Non-linear task tuning

1: Initialize task prompt $\mathbf{P} \leftarrow \mathbf{P}_0$

2: Optimize $\mathbf{P}$ on $\mathcal{D}$ with the full non-linear model:

$$\mathbf{P}_t \leftarrow \arg\min_{\mathbf{P}} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\mathcal{L}\big(f_\theta(x;\mathbf{P}), y\big)\right]$$

3: Compute the task vector:

$$\tau_t \leftarrow \mathbf{P}_t - \mathbf{P}_0$$

Phase 2: Linearized confounder tuning around $\mathbf{P}_t$

4: Initialize confounder update $\tau_c^{\text{lin}} \leftarrow \mathbf{0}$

5: Linearize the model at $\mathbf{P}_t$:

$$f_\theta^{\text{lin}}(x;\mathbf{P}_t + \tau) \;=\; f_\theta(x;\mathbf{P}_t) + \nabla_{\mathbf{P}} f_\theta(x;\mathbf{P}_t)^\top \tau$$

6: Optimize the linearized confounder objective:

$$\tau_c^{\text{lin}} \leftarrow \arg\min_{\tau} \mathbb{E}_{(x,z)\sim\mathcal{D}} \left[\mathcal{L}_c\big(f_\theta^{\text{lin}}(x;\mathbf{P}_t + \tau), z\big)\right]$$

Prompt Arithmetic

7: Compose the de-confounded prompt:

$$\mathbf{P}_{\text{debias}} \leftarrow \mathbf{P}_0 + \tau_t + \lambda\, \tau_c^{\text{lin}}$$

8: **return** $\mathbf{P}_{\text{debias}}$

B Additional Analysis Results

B.1 Bias from Oracle Selection

In general, the test-time distribution is assumed to be unavailable in distribution shift settings during training (Gulrajani & Lopez-Paz, 2020; Yang et al., 2023), preventing selection of optimal hyperparameters for test-time performance. To assess the potential impact of this limitation, we demonstrate an oracle setting where λ is chosen by minimizing the Adjusted Integral on the test distribution. Table 3 reports the results on SHAC with $\alpha_{\text{train}} = 5.0$. HyPA remains the best-performing method even under oracle selection, indicating that its advantage is not solely due to hyperparameter tuning but reflects improved robustness to confounding shift. This gap also raises questions about how to find a better proxy for model selection when the test environment is inaccessible.

Table 3: Oracle hyperparameter selection comparison on SHAC ($\alpha_{\text{train}} = 5.0$, averaged over 5 seeds). Grey values correspond to λ selected using the validation set.

Method	Adjusted Integral ↓	AUSC ↑	OOD AUPRC ↑	ID AUPRC ↑
PA	0.24 (.08)/0.28 (.10)	0.76 (.08)/0.72 (.10)	0.58 (.14)/0.58 (.14)	**0.87 (.08)**/0.85 (.07)
LinPA	0.39 (.05)/0.42 (.07)	0.65 (.03)/0.65 (.04)	0.53 (.06)/0.67 (.09)	0.63 (.05)/0.62 (.09)
HyPA	**0.22 (.07)**/0.26 (.10)	**0.78 (.07)**/0.75 (.09)	**0.66 (.11)**/0.65 (.20)	0.84 (.06)/0.84 (.10)

B.2 Examining Group Fairness Metrics

In addition to the mechanism analysis from hidden representations, we report point evaluations in the OOD setting using two widely adopted fairness metrics: ΔFPR and ΔTPR. They are calculated as the absolute difference in False Positive Rate and True Positive Rate between two sensitive groups (in this case, the groups are determined by the confounding variable), accordingly. As shown in Figure 7, across five datasets, HyPA consistently achieves lower ΔFPR and ΔTPR compared to both baselines. The results indicate that HyPA provides better control over error disparities between confounder groups in OOD settings. We do not report Statistical Parity (SP) in this context, as $P(Y \mid Z)$ is inherently imbalanced in the OOD test sets, making SP comparisons less meaningful and potentially misleading.

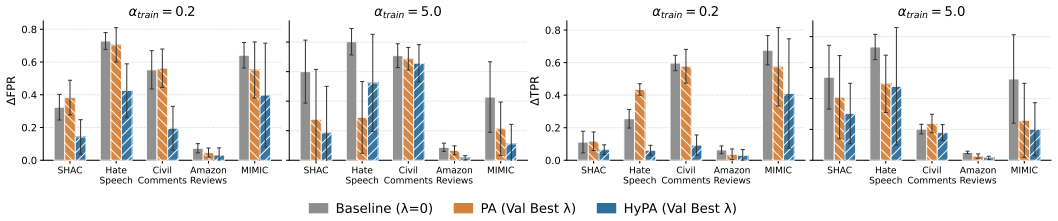

Figure 7: Comparisons of group fairness metrics in Out of Distribution setting.

B.3 Additional Results on Anchor Ablations

In Figure 8, we observe that anchoring HyPA at **P**task consistently yields better OOD AUPRC across all configurations. However, anchoring at the initialization weights leads to improvements in ID AUPRC under certain settings. Although the results are reported using the oracle λ selected by Adjusted Integral, anchoring at initialization consistently outperforms the task prompt anchor on the MIMIC dataset when considered alongside Figure 6(a), particularly at αtrain $= 5.0$.

Revisiting the observations in Figure 3, we hypothesize that in datasets where task-relevant and confounding features are strongly entangled, anchoring at initialization produces a confounder task vector that is less influenced by task-specific signals. Consequently, adding or subtracting this vector has a reduced impact on task performance, leading to weaker deconfounding effects in the resulting task arithmetic composition. We note that this hypothesis remains preliminary and requires further empirical validation.

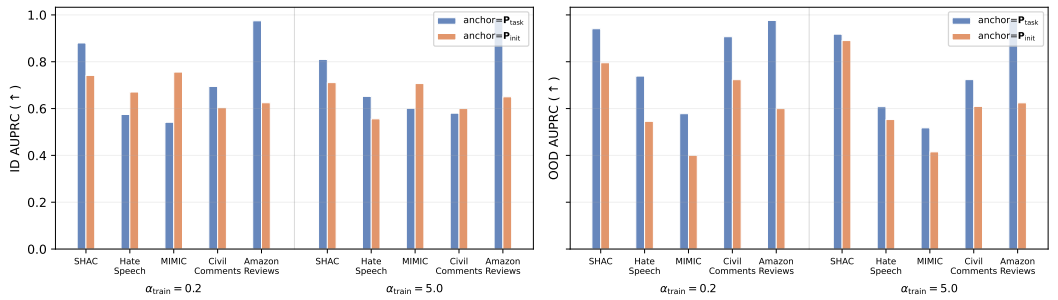

Figure 8: In-Distribution AUPRC and Out-of-Distribution AUPRC across five datasets.

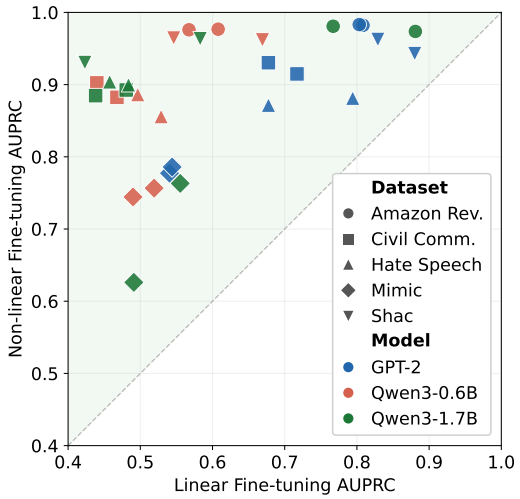

Figure 9: Evidence of the advantage of non-linear fine-tuning across datasets and models.

B.4 Non-linear Advantage

Figure 9 visualizes the advantage of non-linear fine-tuning over linear fine-tuning for task prediction across different models and datasets. Non-linear fine-tuning consistently obtains higher task-prediction AUPRC.

B.5 Additional t-SNE visualizations

In Figure 10 and Figure 11, we display the full panel of t-SNE visualizations of the hidden space given the confounder label in each dataset. The scatter plot is generated via 2000 random samples from the dataset. PCA dimension reduction is applied first to extract the first 50 principal components for efficient computations. From the results, we observe that tuning λ more effectively blends the two confounder groups in datasets where the original clusters are well separated (e.g., MIMIC and Hate Speech). In contrast, its impact is less pronounced in datasets where the two sources are already intermixed in the t-SNE space, leaving limited room for further alignment.

C Dataset Details

We evaluate on five datasets in our experiments, each adapted from prior work.

SHAC (Lybarger et al., 2021) The Social History Annotation Corpus (SHAC) consists of clinical notes collected from two institutions: the University of Washington Medical Center and MIMIC-III. The primary task is to identify substance use information from clinical text. In our experiments, we define the prediction label as the presence of drug abuse and treat the data source (i.e., institution) as the confounder.

Hate Speech (Vidgen et al., 2021; De Gibert et al., 2018) We utilize a hate speech detection dataset curated by Ding et al. (2025), which aggregates samples from two distinct sources: (1) synthetically generated text and (2) posts from a white supremacist forum. The task is to detect hate speech, while the data source serves as the confounder.

Civil Comments (Borkan et al., 2019) Civil Comments is a large collection of user comments from online articles. We follow the preprocessing procedure in WILDS (Koh et al., 2021). The task is toxicity prediction from text. We use demographic identity mentions as

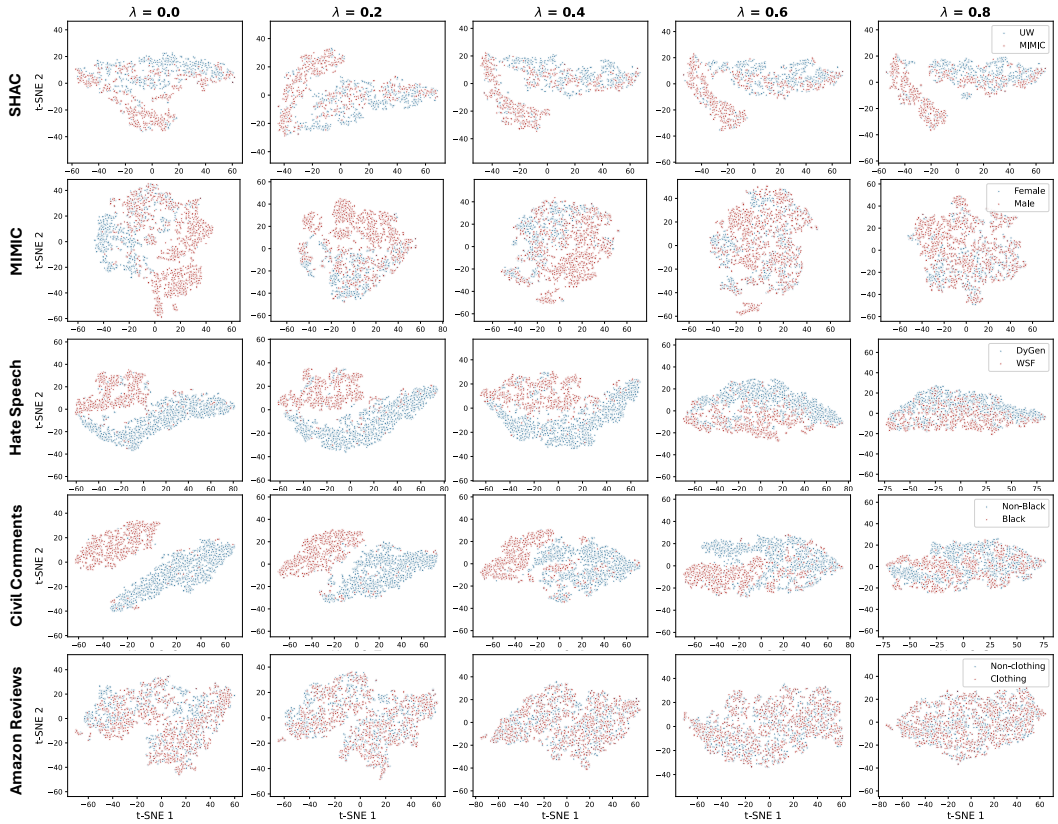

Figure 10: t-SNE plot for $\alpha_{\text{train}} = 0.2$ configuration across datasets. Points are labeled by their corresponding confounder groups.

proxies for potential confounding factors, as such attributes are often spuriously correlated with toxicity labels. In particular, we select mentions of *Black* identity as the confounder.

MIMIC (Johnson et al., 2016) MIMIC-Note consists of clinical notes recorded within the first 48 hours of a hospital stay, sourced from the MIMIC-III database. We use curated notes to predict discharge or mortality outcomes. Patient sex (male vs. female) is treated as the confounder.

Amazon Reviews (Ni et al., 2019) This dataset includes product reviews with ratings, textual content, and metadata. We focus on the review text and define the prediction task as sentiment classification (positive vs. negative). The product category (apparel vs. non-apparel) is used as the confounder.

Below we present the statistics of each dataset. To construct the different α configurations, we sample from the original dataset to fulfill the distribution requirements. For small datasets like SHAC, MIMIC and Hate Speech, we use a ratio of 0.7/0.1/0.2 to form the training, evaluation, and test set. For larger datasets like Amazon Reviews and Civil Comments, we use a fixed number of 10,000/5,000/5,000 to split.

D Training Details

D.1 LM Head for Binary Classification

In our setup, we restrict the frozen backbone to decoder-only transformer architectures and repurpose the language model (LM) head, also referred to as the unembedding matrix, for

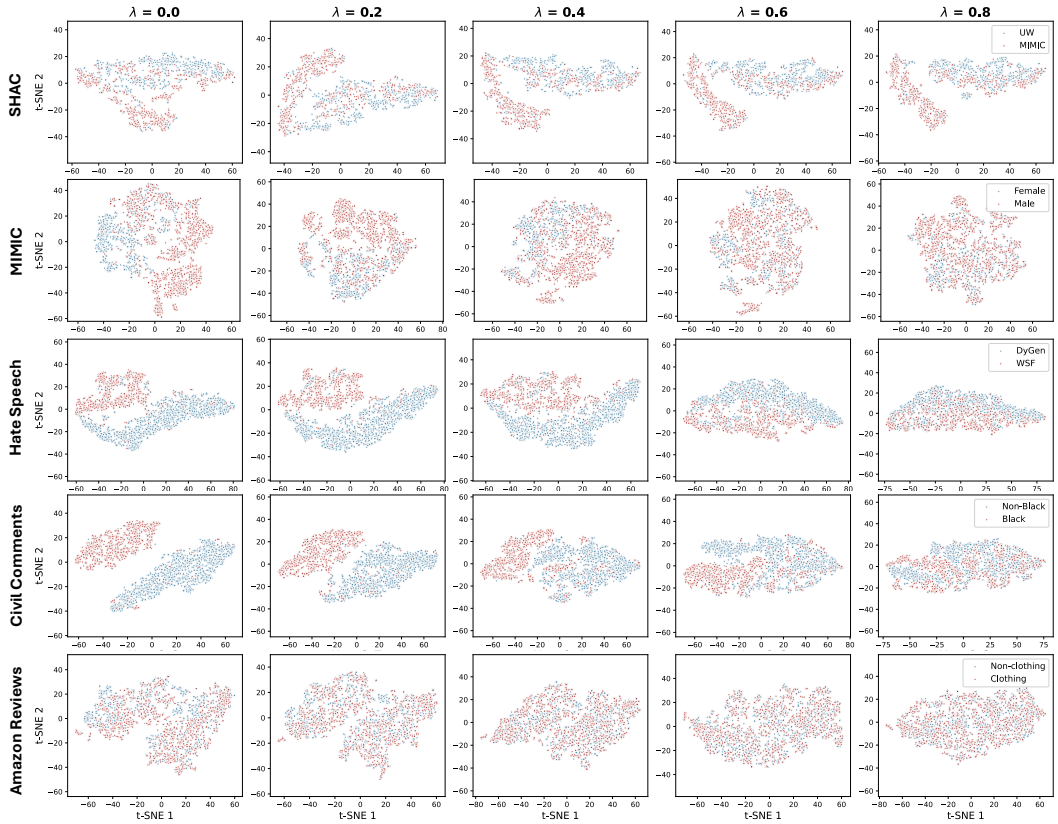

Figure 11: t-SNE plot for $\alpha_{\text{train}} = 5.0$ configuration across datasets. Points are labeled by their corresponding confounder groups.

Table 4: Dataset Statistics

Dataset	N	**P(Y=1)**	**P(Z=1)**
SHAC	4,405	0.3203	0.4261
MIMIC	25,880	0.1381	0.5720
Hate Speech	51,847	0.4508	0.2064
Amazon Reviews	29,557,446	0.8610	0.3939
Civil Comments	447,998	0.0804	0.0341

binary classification. Specifically, let *class1* and *class2* denote two label tokens associated with the target classes for a given input text sequence X. During training and inference, we retain only the logits corresponding to these label tokens and use them to compute the classification loss.

$$\mathbb{P}(\text{class1} \mid X) = \frac{\exp(\mathbf{s}^{\text{class1}})}{\exp(\mathbf{s}^{\text{class1}}) + \exp(\mathbf{s}^{\text{class2}})}, \tag{13}$$

where $\mathbf{s}^{\text{class1}}$ and $\mathbf{s}^{\text{class2}}$ denote the output logits produced by the LM head for the corresponding class tokens. The predicted label is the class token with the higher probability. Throughout soft prompt training, all preexisting model parameters, including the language model head, are frozen. Consequently, task adaptation occurs solely through the soft prompt embeddings, eliminating the need for an additional classification head. We adopt this setup because the pretrained LM head provides a semantically structured decoding space already

aligned with next-token prediction and instruction-following behavior. By mapping class labels to meaningful tokens, the model can exploit this pretrained decision structure directly, rather than introducing and optimizing a separate classifier. Because the entire backbone remains frozen, adaptation is driven exclusively by soft prompt embeddings that modulate internal representations. This formulation preserves the model's original prediction structure and supports task-specific conditioning, providing a stable and parameter-efficient framework for classification. Of note, it also permits modification of model capabilities through task arithmetic operations on the prompt embeddings alone.

D.2 Training Hyperparameters

Table 5: Training hyperparameters for main results.

Setting	**Value**
Base model	GPT-2
Prompt method	Prompt tuning
Prompt length	100
Max sequence length	256
Optimizer	AdamW
Scheduler	Linear decay with 10-step warmup
Learning rate (task / confounder)	10^{-3} / 10^{-3}
Precision	BF16
Batch size (train / eval)	8 / 16 per device
Epochs (task/confounder)	100 / 100
Logging / evaluation frequency	20 / 40 steps
Random seed	0,1,2,3,4
α_{train} values	$0.2, 5.0$
Data split (train / eval / test)	70% / 10% / 20% or 10k / 5k / 5k