TOWARDS CAUSAL FINE-TUNING UNDER LATENT-CONFOUNDED SHIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Adapting to latent-confounded shift remains a core challenge in modern AI. Such shift is driven by hidden variables that induce spurious, non-transportable correlations between inputs and outputs. A practical failure mode arises when fine-tuning pre-trained foundation models on confounded data (e.g., where certain text tokens or image backgrounds spuriously correlate with the label), leaving models vulnerable at deployment. We introduce *causal fine-tuning, which frame model adaptation as an identification problem* and pose an explicit causal model that decomposes inputs into low-level spurious features and high-level causal representations. Under this family of models, we formalize the assumptions required for identification. Using pre-trained language models as a case study, we show how identifying and adjusting these components during causal fine-tuning enables automatic adaptation to such shift at test time. Experiments on real-world stress-test benchmarks demonstrate that our method outperforms black-box domain generalization baselines, highlighting the benefits of explicitly modeling causal structure.

1 CONTRIBUTION

Distribution shift is a fundamental challenge for many modern machine learning systems (Huang et al., 2006; D'Amour et al., 2022). In practice, such shift is often implicit and confounded (Veitch et al., 2021; Alabdulmohsin et al., 2023): latent variables induce spurious, non-transportable (Pearl & Bareinboim, 2011; Jalaldoust & Bareinboim, 2024) correlations between inputs and outputs, and their distributions can vary arbitrarily across different environments. We refer to this broader class of distribution shift, arising either directly at the confounder level or through interactions involving confounders, as *confounded shift*[1].

Examples include deploying models across different hospital populations (Caruana et al., 2015), handling corrupted and adversarial image inputs (Hendrycks & Dietterich, 2019; Ilyas et al., 2019), and stress testing models (D'Amour et al., 2022). To address such robustness concerns, previous work has explored feature augmentation and regularization (Hendrycks et al., 2019; Xie et al., 2020; Zhang et al., 2020b; Tu et al., 2020), as well as learning invariant features across domains (Arjovsky et al., 2019; Ahuja et al., 2020; Heinze-Deml & Meinshausen, 2021). Although effective in certain settings, these methods typically assume stable invariant features across environments. However, they do not explicitly account for latent-confounded shift, where observed features may carry spurious correlations that vary across environments but still contain a valid predictive signal. This issue is particularly challenging in unstructured data, such as natural language, where domain labels are often unavailable and confounding factors are implicit (Yuan et al., 2023).

Foundation models (Bommasani et al., 2021), such as BERT (Kenton & Toutanova, 2019), GPT (Brown et al., 2020), and CLIP (Radford et al., 2021), have become the standard for downstream learning. However, when fine-tuned on data with latent confounding, they often exploit spurious correlations that do not generalize across environments (Lv et al., 2022; Qiao & Low, 2024). In Figure 1(a), we illustrate this with a sentiment classification task where the data source (Amazon or Yelp) serves as a latent confounder U, correlated with the sentiment label. During training (denoted by the *environment* or *regime index* $\sigma =$ train (Dawid, 2021)), the "positive" sentiment tends to co-occur with Amazon reviews, while the "negative" sentiment is associated with Yelp reviews. In the

[1] This generalizes the notion of "confounder shift" to include cases where the confounder itself may not shift, but its interactions with other variables induce distributional changes.

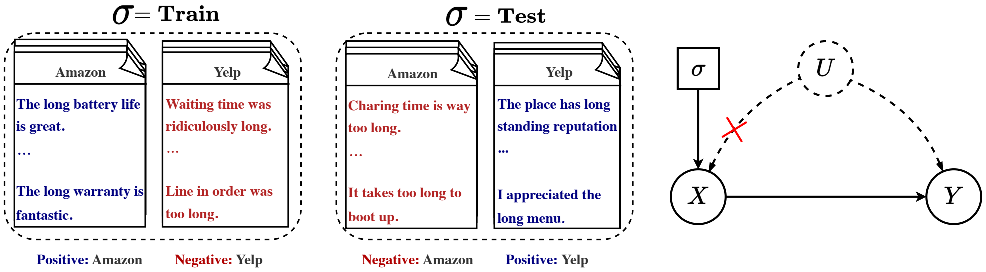
σ= Train
Amazon
The long battery life is great.
...
The long warranty is fantastic.
Yelp
Waiting time was ridiculously long.
...
Line in order was too long.
Positive: Amazon
Negative: Yelp
σ = Test
Amazon
Charing time is way too long.
...
It takes too long to boot up.
Yelp
The place has long standing reputation
...
I appreciated the long menu.
Negative: Amazon
Positive: Yelp
σ
U
X
Y

Figure 1: **(a)** Regime variable σ indexes data generation regimes, with an example of shifting correlations. Sentiment is associated with the data source: Amazon with positive sentiment and Yelp with negative sentiment, which reverts in the test regime. **(b)** Dashed vertices represent hidden variables and square *regime* vertices represent interventions, perturbations or changes of environment. The graph indicates that the mechanism into X may change according to regimes indexed by a regime variable σ. For instance, when a $\sigma = \mathrm{do}(x)$ operation is performed (Pearl, 2009), the edge between U and X is removed, indicated by a red cross. In general, σ indexes arbitrary distributions (Dawid, 2021).

test setting (σ = test), this correlation reverses. Part (b) shows the underlying causal graph: there is an interaction between the environment (denoted by the index σ) and the confounder U in the generation of X. In the classical framework of Pearl (2009), $\sigma = \mathrm{do}(x)$ denotes the regime where X is fixed to x. As the conditional distribution $p(u \mid x; \sigma)$ changes across environments, models that are only trained on data generated via $p(u \mid x; \sigma = \text{train})$ will fail to generalize. Such correlations are not hypothetical: platform identifiers, annotation artifacts, and dataset collection heuristics often create precisely this kind of spurious linkage in practice (Gururangan et al., 2018; Sagawa et al., 2019).

In this paper, we address this issue by proposing a causal framework for fine-tuning under latent-confounded shifts by interpreting the process as a causal identification problem. We define *causal fine-tuning as the process of learning a causal mechanism during fine-tuning*. Specifically, this works by identifying latent structures that enable a principled decomposition of the input signal into stable (invariant) and spurious (environment specific) components, thereby enabling robust generalization under confounded shifts. Our key insight is to exploit scenarios where it can be assumed that: i) *low-level* features of the input signal are invariant but distribution shift between training and test regimes; ii) they cause *high-level* features and the label, with the former possibly changing in distribution at test time. Unlike covariate-shift scenarios, the conditional distribution of the label is also exposed to changes. By applying a causal adjustment strategy to these representations that exploits the fact that we have two views of the unstable features (unsupervised and fine-tuned), we produce robust and adaptive predictions without requiring multiple environments or domain annotations.

Our contributions are as follows. (1) **Modeling:** we analyze processes where input feature generation is hierarchical and responds differently to environmental changes. Where it can be plausibly assumed, the existence of low-level features that are invariant between training and test time can be exploited to decouple spurious and invariant components of high-level features. Care is still needed, as spurious associations can still propagate when conditioning on the observed signal, analogously to Figure 1(b) (Section 4); (2)**Algorithm:** we propose a practical method that decomposes representations into invariant and spurious components and integrates a causal adjustment strategy into standard fine tuning pipelines; (3) **Empirical results:** we demonstrate improved generalization under confounded-shift on real-world datasets with controlled spurious correlations injected to mimic deployment artifacts (Section 6); and (4) **Ablation:** we analyze the role of key components in the algorithm and their impact on robustness and predictive performance (Section 6).

2 Related Work

Distribution shift is fundamentally an ill-posed problem without assumptions: the mapping between x and y may change arbitrarily across domains (Huang et al., 2006). A central question is: which parts of the data-generating process remain invariant across environments? Causal inference, in particular transportability theory (Pearl & Bareinboim, 2011; Jalaldoust & Bareinboim, 2024), provides a principled framework for answering this by characterizing how and when causal knowledge can be moved between environments. The most common assumptions in the literature are covariate shift (Magliacane et al., 2018; Shimodaira, 2000) and label shift (Buck & Gart, 1967). Building on these assumptions, some lines of work explore methods that learn causally robust invariant features $\Phi(x)$ based on the covariates across multiple environments (Arjovsky et al., 2019; Ahuja et al., 2020; Von Kügelgen et al., 2021; Yue et al., 2021; Mitrovic et al., 2021; Shi et al., 2021; Kong et al., 2023). Another line of work explores counterfactual reasoning and data augmentation techniques (Kaushik et al., 2019; Ben-David et al., 2022; Le et al., 2023; Feder et al., 2023).

More recently, there has been a growing interest in causal representation learning, which aims to learn disentangled latent representations that capture the underlying causal structure and enables models to generalize across domain shifts. Existing work focuses on learning stable causal latent variables (Sun et al., 2021; Lu et al., 2022; Lv et al., 2022), invariant predictors (Veitch et al., 2021; Jiang & Veitch, 2022), or compositional models (Bravo-Hermsdorff et al., 2023; Yu et al., 2024). These approaches often require multiple environments or proxy variables, which can be impractical to acquire or augment, particularly for unstructured data (Chalupka et al., 2017). We build on this body of work by treating a pre-trained foundation model as an implicit environment for data augmentation, on top of the fine-tuning data, allowing causal representation learning with single domain fine-tuning data only. Once the latent variables are learned, causal adjustment formulas are adopted to generate predictors that are robust to domain shift, with two common receipts being back-door adjustments (Yue et al., 2020; Zhang et al., 2020a), and front-door adjustments (Li et al., 2021; Mao et al., 2022; Nguyen et al., 2023). Front-door adjustments have become a promising choice due to hidden confounding variables between input signals and outputs. Motivated by the recent success of these methods, we propose a method to construct a front-door in the context of a fine-tuning pipeline, achieving improved OOD robustness under confounded shifts.

3 Preliminaries

Motivation. Our proposal encodes invariance assumptions into graphical causal models with explicit regime indicator variables (Pearl, 2009; Dawid, 2021). In particular, we consider the scenario where input X is allowed to cause label Y, but not vice-versa, with the possible presence of hidden confounders U. Graphically, this setup is depicted in Figure 1(b). To accommodate distribution shifts, we further assume that changes from training to test environments involve an *intervention* (or *perturbation*, or *regime*), denoted by the *regime variable* σ, which modifies the influence of U on X. Data are observed only under the training regime denoted by $\sigma = \text{train}$. We want to be safe to build a predictor for an unknown environment denoted as $\sigma = \text{test}$, where the conditional distribution $p(x \mid u; \sigma = \text{test})$ can potentially arbitrarily differ from $p(x \mid u; \sigma = \text{train})$.

Intuition and principles. Let us first explicitly analyze why distribution shifts under our postulated structure lead to machine learning classifiers failing.

Proposition 3.1 *Let a causal model, following the structure shown in Fig. 1(b), represent the source (train) and target (test) domains of some probabilistic system resulting from regimes $\sigma = \text{train}$ and $\sigma = \text{test}$, with respective implied distributions $p(y \mid x) := p(y \mid x; \sigma = \text{train})$ and $p^\star(y \mid x) := p(y \mid x; \sigma = \text{test})$. Then, in general, $p(y \mid x) \neq p^*(y \mid x)$.* □

This follows directly from the law of total probability over U, below assumed to be discrete without loss of generality:

$$p(y \mid x; \sigma) = \sum_u p(y \mid u, x; \sigma) p(u \mid x; \sigma) = \sum_u \underbrace{p(y \mid u, x)}_{\text{does not change with } \sigma} \underbrace{p(u \mid x; \sigma)}_{\text{changes with } \sigma}.$$

This implies that a predictor learned under $\sigma = \text{train}$ is not transportable (Pearl & Bareinboim, 2011; Jalaldoust & Bareinboim, 2024) in this setting even when σ affects neither U nor Y directly. If we are serious about exploiting the postulated causal structure, we must make use of the fact that the change from $p(y \mid x; \sigma = \text{train})$ to $p(y \mid x; \sigma = \text{test})$ boils down to the difference between $p(u \mid x; \sigma = \text{train})$ and the unknown $p(u \mid x; \sigma = \text{test})$. One possibility is to address it in a minimax way, akin to distributionally robust optimization (Duchi & Namkoong, 2021), where we minimize our loss with respect to all distributions $p^\star(u \mid x; \sigma = \text{test})$ which are "close" to $p(u \mid x; \sigma = \text{train})$ in some sense. However: i) in general we will not be able to identify $p(u \mid x; \sigma = \text{train})$ nor $p(y \mid x, u)$ to begin with; ii) the meaning of U is in general unclear anyway, making any assumptions about closeness of $p^\star(u \mid x; \sigma = \text{test})$ and $p(u \mid x; \sigma = \text{train})$ scientifically vacuous at best; iii) it inherits all shortcomings of the minimax approach, being harmfully conservative when $p(u \mid x; \sigma = \text{test})$ and $p(u \mid x; \sigma = \text{train})$ are close.

Maximum-entropy as a default choice. We propose following a *default regime* built from a maximum-entropy distribution $p(u \mid x; \sigma = \text{default})$ that respects the marginal constraint $p(u; \sigma = \text{default}) = p(u; \sigma = \text{test})$. A trivial solution is $p(u \mid x; \sigma = \text{default}) = p(u; \sigma = \text{test})$, the latter which is also knowable by assumption: $p(u; \sigma = \text{test}) = p(u; \sigma = \text{train})$. This also means that $p(y \mid x; \sigma = \text{default}) = p(y \mid x; \sigma = \text{do}(x))$, since our causal structure assumption implies $p(u; \sigma = s^\star) = p(u; \sigma = s^{\star\star})$ and $p(y \mid x, u; \sigma = s^\star) = p(y \mid x, u; \sigma = s^{\star\star})$, for all values $s^\star, s^{\star\star}$ in the scope of σ.

We thus reduce this to the problem of training our model *as if* it came from the regime $\sigma = \text{do}(x)$, but using data collected under $\sigma = \text{train}$. Unfortunately, it is a well-known result that, assuming no more than the Markovian factorization implied by Figure 1(b), the distribution $p(y \mid \text{do}(x)) := p(y \mid x; \sigma = \text{do}(x))$ is not identifiable: it follows from the completeness of Pearl's do-calculus (Pearl, 2009). To that effect, in the sequel we will assume a more fine-grained structure for X, as well as making explicit use of fine-tuning data.

4 Assumptions for Causal Fine-Tuning

In this section, we introduce the identification theory for $p(y \mid \text{do}(x))$ and the structural assumptions required for identifiability, then discuss their implications for practical applications. Standard supervised learning makes no distinction between "causal features" and "non-causal features". Moreover, concepts such as $\text{do}(x)$ may be ill-posed when the object of intervention is unstructured (Chalupka et al., 2017), such as raw text. We start by postulating which representations of X are assumed to follow a causal structure. More intuition of this formulation is discussed in Appendix D.

Assumption 4.1 (Functional Decomposition) *We assume access to a triplet of measurements $(R_0, R_1, \Phi) = f(X)$ for some function f, defined non-constructively, as follows: (i) (R_0, R_1) is a* **paired representation** *of X. The mapping to R_0 is learned using self-supervised learning with unlabeled data. The mapping to R_1 is learned during supervised fine-tuning with labeled data under the training environment σ = train; (ii)* **local features** *Φ are low-level features implied as a by-product of the fine-tuned model (e.g., combinations of earlier layer features that cause high-level feature R_1 in the last layer).* □

In the sequel, we will explicitly describe the computational procedure that constructively defines this mapping $(R_0, R_1, \Phi) = f(X)$. It is to be noted that an intervention $\text{do}(x)$ will be defined as $\text{do}((R_0, R_1, \Phi) = f(x))$, which we will sometimes denote as $\text{do}(r_0), \text{do}(r_1), \text{do}(\Phi)$.

Under this choice of abstraction, we postulate a causal structure with (R_0, R_1, Φ) interacting with causal latent variables C and two sets of "spurious" latent variables S_0 and S_1, spurious in the sense that only C is a causal parent for output Y. We define S_0 as the latent spurious features of pre-training not affected by distribution shifts, and S_1 are the latent spurious features affected by the environment index σ. The generative model contains these two feature sets as latent variables, along with structural assumptions about how σ, R_0, R_1, Φ and Y are connected. Assumptions are graphically summarized in Fig. 2, and detailed as follows.

Assumption 4.2 (Causal Latent Structure) *High-level features $\{R_0, R_1\}$ are indirect measurements of mutually independent variables $\{S_0, S_1, C\}$. S_0 can only cause R_0 and S_1 can only cause R_1. The regime variable σ can only affect S_1. Morever, hidden confounders U_S are common parents of R_1 and Φ, and independent hidden confounders U_Φ are parents of Φ and Y.* □

This assumption aligns with prior work in causal learning (Tenenbaum & Freeman, 1996; Gong et al., 2016; Heinze-Deml & Meinshausen, 2021; Mao et al., 2022). Intuitively, this abstracts the true complex causal graph into a coarser granularity, encapsulating stable hidden confounders into C and any other (unstable) non-confounding variables into S_0, S_1. It also postulates a principle: *any dependency between S_0 and S_1 is solely attributed to common cause C.*

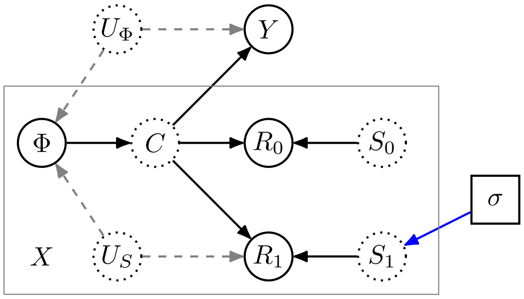

Figure 2: Refinement of the original causal diagram in Figure 1(b), where X is broken apart, and abstracted into vectors R_0, R_1 and Φ as described in Section 4. More intuition of this causal diagram is discussed in Appendix D.

Assumption 4.3 (Causal Structure of Distribution Shifts) *Regime variable σ affects the system only via S. This also implies that the causal ancestors of Y do not interact with σ.* □

This assumption postulates that, for any regime of interest where we deploy our system, the relationship between causal ancestors and the output Y is invariant. However, it is not the case that we will be able to optimize the empirical risk on the training data without consequences, since conditioning on the entire input signal $\{R_0, R_1, \Phi\}$ will *d-connect* Y with σ (Pearl, 2009): this happens via *active collider paths* (Spirtes et al., 2000) such as $Y \leftarrow U_\Phi \rightarrow \Phi \leftarrow U_S \rightarrow R_1 \leftarrow S_1 \leftarrow \sigma$ and $Y \leftarrow C \rightarrow R_1 \leftarrow S_1 \leftarrow \sigma$. This makes our predictions dependent on the value of σ, in the sense of Dawid (2021), which means being affected by distribution shifts. In what follows, we will rely on i) the missing edge $\Phi \rightarrow Y$ and ii) the ability of deterministically inferring C, as formalized in the following assumption and theorem.

Assumption 4.4 (Sufficient Mediator) *The causal effect of Φ on Y is fully mediated through C. That is, $p(y \mid do(\Phi), do(c)) = p(y \mid do(c))$.* □

Justification. This assumption is sometimes known as a front-door structure (Pearl, 2009) for the effect of Φ on Y. It can be interpreted as having C as ultimately the only variable driving Y directly, and relying on this desiderata as the operational *definition* of C, implying no further latent sources confounding Φ and C, or C and Y, or any other path between Φ and Y relying on further (implicit) hidden variables. We allow confounding between Φ an Y.

Theorem 4.5 (Identification for Causal Features C) *Assume the structural assumptions encoded in the causal graph in **Fig. 2**. Let the mapping between $\{S_0, S_1, C\}$ and $\{R_0, R_1, \Phi\}$ obey the invertibility conditions of Von Kügelgen et al. (2021). According to **Theorem 4.4** in Von Kügelgen et al. (2021), we can deterministically identify C from R_0 and R_1.*

Intuition. This theorem implies that if the causal latent variable C remains invariant across environments (Assumption 4.3), the distribution shift between representations R_0 and R_1 can be used to identify C. For a formal proof of this theorem, please refers to **Theorem 4.4** in Von Kügelgen et al. (2021). In the sequel, we will learn this function using the idea presented in Equation 3.

We will now show that we can identify $p(y \mid do(x))$ from the pre-trained and training fine-tuning data. The proof of this result is short and presented in Appendix I.

Theorem 4.6 (Identification for Causal Transfer Learning) *Given the assumptions in the causal graph in **Fig. 2 (b)** and Theorem 4.5, the distribution of Y under $do(x)$ can be computed as*[2]

$$p(y \mid do(x)) = \sum_{\Phi', x'} p(y \mid \Phi', c) p(\Phi' \mid x') p(x'), \tag{1}$$

where c is given as a function of (r_0, r_1) per Theorem 4.5, and (r_0, r_1) are a function of x. □

Invariance implication and pragmatic application. The difference between $p(y \mid x; \sigma = \text{test})$ and $p(y \mid x; \sigma = \text{do}(x))$ in our setup boils down to averaging $p(y \mid c, U_\Phi)$ over $p(u_\Phi \mid r_0, r_1, \Phi, c; \sigma = \text{test})$ in the former, and $p(u_\Phi)$ in the latter. When can we say that the latter is an improvement over $p(u_\Phi \mid r_0, r_1, \Phi, c; \sigma = \text{train})$? Our claim is that by virtue of the confounder being a cause of local

[2] Φ' is deterministically given by x', but the above representation in terms of a probability $p(\Phi' \mid x')$ is useful as a way of understanding how to generate Φ'.

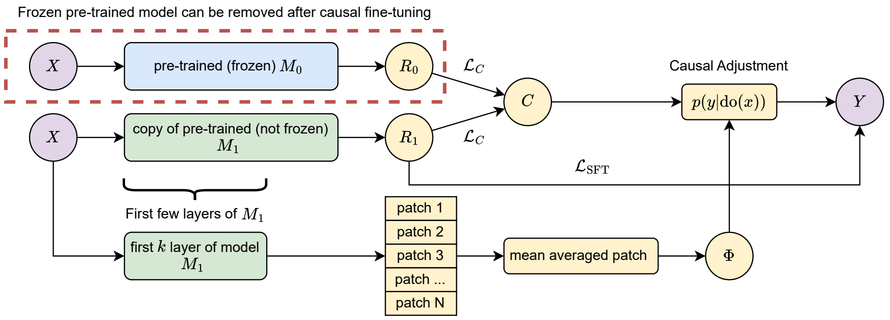
Frozen pre-trained model can be removed after causal fine-tuning
X
pre-trained (frozen) M_0
R_0
$\mathcal{L}_C$
C
Causal Adjustment
$p(y|\mathrm{do}(x))$
Y
X
copy of pre-trained (not frozen) M_1
R_1
$\mathcal{L}_C$
$\mathcal{L}_{\text{SFT}}$
First few layers of M_1
first k layer of model M_1
patch 1
patch 2
patch 3
patch ...
patch N
mean averaged patch
Φ

Figure 3: Illustration of our CFT methods. During training, we keep a copy of pre-trained foundation model for identification purposes, which is removed during inference. Once CFT is done, we get a model of the same size as the standard fine-tuning but providing functions to decompose input to causal and spurious features. This allows for adaptation to latent-confounded shifts at test time.

features Φ only, and not of the whole of X, the relevance of information passing through (S_0, S_1) via active collider paths only should be limited anyway (Ding & Miratrix, 2014), *unless the test environment affects it drastically*. In this case, we may be thrown away too far from the original $p(u_\Phi \mid r_0, r_1, \Phi, c; \sigma = \text{train})$ in unpredictable ways, and the safer bet ("maximum-entropy") is to think of $p(y \mid c)$ as being a random measure "$p_{U_\phi}(y \mid c)$" with a conservative prior $p(u_\Phi)$ which comes from the model and is agnostic to the environment. Section 6 will empirically investigate this in detail.

5 Algorithm: Causal Fine-Tuning

In this section, we operationalize the theoretical insights outlined in Section 4 into a Causal Fine-Tuning (CFT) framework (illustrated in Fig. 3).

Each component of this framework is illustrated via fine-tuning of a pre-trained language model as a running example: i) *supervised fine-tuning*, which obtains representations R_1 from input pairs (X, Y); ii) *learning invariant causal features*, which derives causal representation C from both R_1 and R_0, where R_0 is the representation directly taken from the pre-trained model; and iii) *retrieving local features*, which extracts Φ from the fine-tuned model.

After retrieving Φ, we use C and Φ to adjust for $p(y \mid \mathrm{do}(x))$. More details can be found in Algorithm 1 for training and Algorithm 2 for inference, both in Appendix J.

Component 1: Supervised Fine-Tuning The first step is to learn R_1 from training samples through supervised fine-tuning (SFT). Assume for exposition purposes that labels Y are binary. Given a pre-trained model $p(r_0 \mid x)$, we learn $p(r_1 \mid x)$ with training samples (X, Y) by minimizing the loss

$$\mathcal{L}_{\text{SFT}} = \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[-y \log p(r_1 \mid x)\right]. \tag{2}$$

Component 2: Learning Causal Invariant Feature To learn the invariant causal feature C, we aim to identify the distribution $p(c \mid r)$. This process involves aligning representations from different environments while maximizing entropy and prevent collapsed representations (Von Kügelgen et al., 2021). The loss function is constructed based on Theorem 4.5,

$$\mathcal{L}_C := \mathbb{E}_{(r_0, r_1 \mid x)\sim\mathcal{D}} \left[\|p(c \mid r_0) - p(c \mid r_1)\|_2^2\right] - H\left(p(c \mid r_0)\right) - H\left(p(c \mid r_1)\right), \tag{3}$$

where x is sampled from $p(x)$ and used to calculate r_0, r_1. The first term enforces invariance across environments. The entropy terms maximize the diversity in representations, reducing the risk of collapse.

Component 3: Retrieving Local Features This component focuses on constructing local features Φ, which are taken from the first k layers of $p(r_1 \mid x)$. In particular, we use the representation from the k^{th} layer. In our running example of a pre-trained language model, we use $k = 1$, which is the embedding layer. Our justification is that using the first layer provides a better low-level signal (LeCun et al., 2015; Goodfellow et al., 2016; Yu, 2023). See more discussion in Appendix E.

Given input X as a series of tokens where $X = [t_1, t_2, ..., t_m]$, we can retrieve a vector representation for each token t. To construct the local feature Φ, we divide the token sequence into non-overlapping patches (we use 10 patches in our experiments, for balancing granularity and computational efficiency), allowing us to rewrite X as patches p where $X = [p_1, p_2, ..., p_{10}]$ where $p_1 = [t_1, t_2, ..., t_{\frac{m}{10}}]$ and so on. After splitting, we perform mean averaging on these patches to extract a regional signal such that $\Phi = \frac{1}{10}\sum_{i=1}^{10} p_i$.

Causal Adjustment Based on components 1, 2 and 3, we can get C and Φ from the observed data $X = x$, we then passed C and Φ through a multilayer perceptron (MLP) to adjust for $p(y \mid \text{do}(x))$. For full details, please consult the Appendix J and the corresponding code.

6 Experiments

We evaluated our proposed approach using stress-testing benchmarks based on two real-world scenarios. This section summarizes the setup, datasets, baselines, and key results. Detailed description of datasets and simulators can be found in Appendix A, while Appendix B provides details of the model architecture. Further analysis and additional results are presented in the Appendix H.

Datasets. We construct two benchmarks based on widely studied sentiment analysis datasets: Amazon review and Yelp review (Zhang et al., 2015). Prior research suggests that spurious information and dataset artifacts often exist in training data, creating unintended correlations that can hinder generalization (Gururangan et al., 2018; Sagawa et al., 2019; Veitch et al., 2021; Schrouff et al., 2024). Motivated by these findings, we inject controlled confounders into these datasets to replicate and amplify such effects in a systematic and reproducible way. We further discuss the value of this design in Appendix C, and explain why we do not rely on generative tools such as GPT for stress-testing in Appendix F.

Baselines and Our Methods. We compare our algorithm with the following baselines: (1) **SFT0**, which involves training a linear classifier on a frozen sentence representation extracted directly from PLMs; (2) **SFT** (Vapnik, 1998), the typical transfer learning strategy with PLMs, considered as a very strong baseline (equivalent to performing ERM); (3) **WSA** (Izmailov et al., 2018; Athiwaratkun et al., 2019), which averages multiple points along the SGD trajectory to achieve a more robust classifier; and (4) **WISE** (Wortsman et al., 2022), which interpolates the parameters of PLMs and a fine-tuned model to enhance generalization.

Our proposed **CFT** algorithm follows the exact setup described in Section 4. To analyze the impact of different representations, we implemented three additional variations of CFT: (1) **CFT-N** uses both Φ and C to predict Y without applying the adjustment formula from Theorem 4.6, leaving a causal path between Φ and Y unblocked; (2) **CFT-C** uses the estimated high-level causal variable C to predict Y; and (3) **CFT-Φ** directly uses low-level spurious features Φ to predict Y.

Experimental Setup. Each experiment was repeated 5 times using the AdamW optimizer (Kingma & Ba, 2015; Loshchilov, 2017) with a learning rate of 5×10^{-5} for all cases, except for SFT0. There, a learning rate of 5×10^{-4} was used. Each model was trained for 10 epochs, sufficient for convergence. The best model iteration was selected based on performance on a hold-out validation set (20% of the training data). For training, we randomly sample 5000 points per class, with a 20% split for validation. For testing, we sample 2000 per class. For training data, we construct data to contain strong spurious correlations with probability of 90% of the time. We use the same ratio for the ID test set and, for the OOD test set, we shift this ratio of possible correlation to be 70%, 50%, 30% and 10%.

6.1 Experiment 1: Spurious Correlation Between Stop Words and Label

Following Veitch et al. (2021), we generate stress-testing ID and OOD data by injecting spurious correlations between stop words (e.g. "and", "the") and class labels. See Appendix A.2 for more details.

Table 1: Main results for Experiment 1, reported as F1 scores with mean averaged value based on 5 runs of different seeds. We presents the Yelp results in the first table and Amazon in the second.

	Train	**Test**				
	Spurious 90%	**Spurious 90%**	**Spurious 70%**	**Spurious 50%**	**Spurious 30%**	**Spurious 10%**
SFT0	86.24	86.42	71.58	56.82	42.04	26.94
SFT	95.96	92.89	81.89	71.20	60.23	49.24
CFT	**98.69** ↑ 2.73	**93.03** ↑ 0.14	**84.16** ↑ 2.27	**75.83** ↑ 4.63	**67.06** ↑ 6.83	**58.40** ↑ 9.16
CFT-N	97.80	92.35	81.91	71.89	61.46	51.07
CFT-C	98.62	92.99	84.07	75.51	66.62	57.75
CFT-Φ	92.42	89.30	71.83	54.41	36.91	19.08

	Train	**Test**				
	Spurious 90%	**Spurious 90%**	**Spurious 70%**	**Spurious 50%**	**Spurious 30%**	**Spurious 10%**
SFT0	87.99	87.90	70.42	52.80	35.26	17.83
SFT	96.56	**92.39**	81.61	70.77	59.97	49.33
CFT	**98.58** ↑ 2.02	92.37 ↓ 0.02	**83.16** ↑ 1.55	**74.25** ↑ 3.48	**65.24** ↑ 5.27	**56.40** ↑ 7.07
CFT-N	97.24	91.82	80.83	69.76	58.77	48.00
CFT-C	97.58	92.24	82.35	72.62	63.01	53.40
CFT-Φ	90.63	89.83	70.46	51.06	31.71	12.40

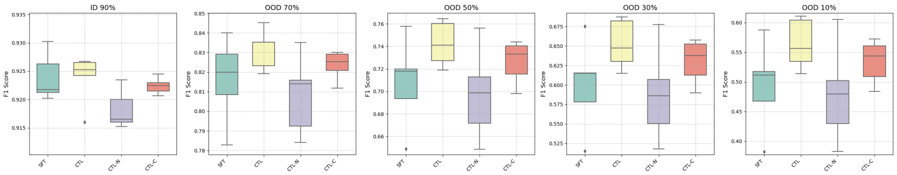

Figure 4: Box-plot over 5 runs for 4 methods (SFT, CFT, CFT-N and CFT-C). Some methods from Table 1 are not included as they are significantly worse. This is a visualisation of the Amazon dataset. Yelp shows a similar trend (Fig.6, Appendix).

Results. The main results are presented in Table 1, with visualizations for the Amazon dataset over 5 runs in Fig. 4. These results demonstrate the superiority of our model against the strong baselines. We observe a significant performance drop in both SFT0 and SFT when the distribution of spurious features shifts, indicating that standard fine-tuning methods struggle to handle spurious correlations in OOD settings. However, we observe that SFT consistently outperforms SFT0 for both ID and OOD settings, highlighting the effectiveness of "knowledge transfer" in improving representation quality. Among all learning algorithms, our proposed CFT method provides the most promising predictors. Compared to CFT, the CFT-N conditions on Φ, which introduces an active collider path between σ and Y, namely $\sigma \rightarrow S_1 \rightarrow R_1 \leftrightarrow \Phi \leftrightarrow Y$ (Spirtes et al., 2000; Pearl, 2009), where S_1 is unobserved, but R_1 and Φ are observable functions of X. This means that this predictor gets exposed to changes in distribution as indexed by σ. We observe that the drop in performance compared to CFT and this confirms why making predictions under a hypothetical $\text{do}(x)$ helps. The CFT-C variant, which uses only the causal variable C for prediction, performs well in many OOD settings, suggesting that PLMs can be considered as a good source of new domain data. However, its accuracy decreases as the OOD distribution diverges further from the ID data, indicating that relying solely on C may limit robustness in extreme scenarios. An intriguing observation is the behavior of the CFT-variant Φ, which predicts the label using only local features Φ. This variant is strongly correlated to the spurious pattern in the data, highlighting why our methods can work for OOD settings, as we negotiate large changes for the spurious distribution by sticking to the distribution $\text{do}(x)$. See more discussions on Φ in Appendix E.

6.2 Experiment 2: Spurious Correlation Between Data Source and Label

We construct controlled variant of real-world data to mimic the scenario illustrated in Fig. 1. We build correlations between the source of the data (whether coming from Amazon or Yelp) and the label, by adding strings such as "amazon.xxx" or "yelp.yyy" into the sentences, more details in Appendix A.3. We compare our approach with other single-domain generalization baselines to demonstrate its effectiveness.

Results. The results are consistent with our previous experiments. When compared with the two baselines, the WISE method does not work too well, perhaps for being more sensitive to the hyper-

Table 2: Main results for Experiment 2. Averages reported based on 5 runs with different seeds.

	Train	Test				
	Spurious 90%	**Spurious 90%**	**Spurious 70%**	**Spurious 50%**	**Spurious 30%**	**Spurious 10%**
SFT0	87.74	87.78	69.57	51.46	33.42	15.26
SFT	94.01	**91.39**	78.05	64.75	51.36	37.78
SWA	**99.99**	91.26	**80.34**	69.63	58.59	47.41
WISE	92.87	91.34	76.59	61.77	46.96	31.83
CFT	97.46 ↑ 3.45	90.59 ↓ 0.80	80.32 ↑ 2.27	**70.08** ↑ 5.33	**59.68** ↑ 8.32	**49.22** ↑ 11.44
CFT-N	91.36	89.98	71.31	52.66	33.96	15.05
CFT-C	95.60	91.07	78.93	66.80	54.62	42.25
CFT-Φ	90.92	89.81	70.49	51.24	32.03	12.60

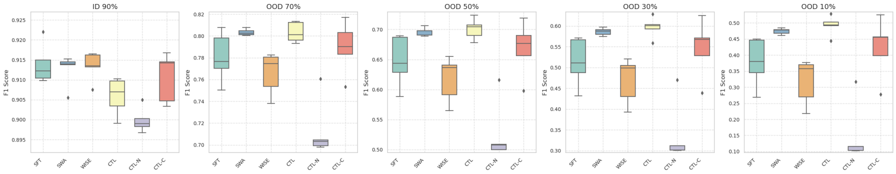

Figure 5: Box-plot over 5 runs for 6 methods (SFT, SWA, WISE, CFT, CFT-N and CFT-C). Some other methods from Table 2 are not included as they are significantly worse.

parameter that mixes the fine-tuned model and the pre-trained model (we used a default value of 0.5, which means they are equally weighted). The SWA method worked quite well compared to the SFT methods, suggesting that stopping at a flat region of the parameter space improves the generalization of the model (Izmailov et al., 2018; Kaddour et al., 2022). However, its performance degraded significantly under more severe distribution shifts (e.g., the OOD ratio from 70% to 10%), highlighting its limitation in handling extreme perturbations. In contrast, our proposed CFT approach consistently outperformed all baselines, demonstrating robustness across all OOD settings. Statistical test on these results are provided in Appendix G.

6.3 Further Analysis

We conducted a further analysis on (1) level of spuriousness (Fig. 7), (2) number of training data (Fig. 8), and (3) number of samples during inference (Fig. 9). All results are presented in Appendix H, summarized as: (1) Under different levels of spurious information, our CFT method consistently outperforms the SFT method by a significant margin. (2) Even with more data provided, our model CFT consistently outperforms black-box methods (SFT). However, we observe that when enough data is provided, there is a saturation point where SFT and CFT methods become indistinguishable for this particular OOD task. (3) We also observed a decrease in performance if we do not use the interventional distribution $\text{do}(x)$ during prediction time.

7 Conclusion

We introduced a method for adapting to latent-confounded shift via causal fine-tuning, demonstrating promising performance in OOD scenarios compared to standard fine-tuning and black-box domain generalization methods. **Lessons.** We recognize that foundation models are already highly resilient to perturbations. As suggested in (Bommasani et al., 2021), based on our running example using text as input, introducing spurious information at the input level requires significant effort. This highlights the strength of foundation models in managing noisy input variations, but also the challenge in simulating spurious correlations for testing purposes. **Limitations.** While we made extensive efforts to control and simulate spurious relationships that resemble real-world deployment scenarios, the mechanisms through which spurious correlations emerge in complex, real-world environments remain unclear. Hence, handling latent-confounded shifts in real-world data require very careful thinking, especially in high-stake applications. We hope that our method provides a valuable baseline for both academic and industry researchers facing these challenges. **Future Work.** One natural direction is to extend our causal fine-tuning framework to multi-modal scenarios, where latent-confounded variables might live in one modality but interact with another modality, introducing exciting new challenges.

Ethics Statement

This research did not involve identifiable human data or animals and therefore did not require approval from an institutional ethics committee or review board. All experiments are conducted on publicly available datasets for scientific purposes only. The work does not involve or target any sensitive attributes such as gender, race, nationality, or skin color. Our study focuses on confounded-shift generalization in pre-trained foundation models, with the aim of improving the trustworthiness and safety of model post-training.

Reproducibility Statement

We have made every effort to ensure the reproducibility of our work. All experiments were conducted using publicly available datasets, and we provide detailed descriptions of data preprocessing in Appendix A. Our model architecture, hyperparameters, and training protocols are fully specified in Section 6 and Appendix B. We have already included our code in the Supplementary Material uploaded alongside our paper.

References

Kartik Ahuja, Karthikeyan Shanmugam, Kush Varshney, and Amit Dhurandhar. Invariant risk minimization games. In *International Conference on Machine Learning*, pp. 145–155. PMLR, 2020.

Ibrahim Alabdulmohsin, Nicole Chiou, Alexander D'Amour, Arthur Gretton, Sanmi Koyejo, Matt J Kusner, Stephen R Pfohl, Olawale Salaudeen, Jessica Schrouff, and Katherine Tsai. Adapting to latent subgroup shifts via concepts and proxies. In *International Conference on Artificial Intelligence and Statistics*, pp. 9637–9661. PMLR, 2023.

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

Ben Athiwaratkun, Marc Finzi, Pavel Izmailov, and Andrew Gordon Wilson. There are many consistent explanations of unlabeled data: Why you should average. In *International Conference on Learning Representations*, 2019.

Eyal Ben-David, Nadav Oved, and Roi Reichart. Pada: Example-based prompt learning for on-the-fly adaptation to unseen domains. *Transactions of the Association for Computational Linguistics*, 10:414–433, 2022.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Gecia Bravo-Hermsdorff, David Watson, Jialin Yu, Jakob Zeitler, and Ricardo Silva. Intervention generalization: A view from factor graph models. *Advances in Neural Information Processing Systems*, 36:43662–43675, 2023.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL `https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html`.

AA Buck and JJ Gart. Comparison of a screening test and a reference test in epidemiologic studies. ii. a probabilistic model for the comparison of diagnostic tests. 1967.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730, 2015.

Krzysztof Chalupka, Frederick Eberhardt, and Pietro Perona. Causal feature learning: an overview. *Behaviormetrika*, 44:137—-164, 2017.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022.

Andrew Philip Dawid. Decision-theoretic foundations of statistical causality. *Journal of Causal Inference*, 9:39–77, 2021.

Peng Ding and Luke Miratrix. To adjust or not to adjust? Sensitivity analysis of M-bias and butterfly-bias. *Journal of Causal Inference*, 3:41–57, 2014.

John C. Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *Annals of Statistics*, 49, 2021.

Amir Feder, Yoav Wald, Claudia Shi, Suchi Saria, and David Blei. Data augmentations for improved (large) language model generalization. *Advances in Neural Information Processing Systems*, 36: 70638–70653, 2023.

Mingming Gong, Kun Zhang, Tongliang Liu, Dacheng Tao, Clark Glymour, and Bernhard Schölkopf. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning (ICML)*, pp. 2839–2848. PMLR, 2016.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel Bowman, and Noah A. Smith. 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)*, pp. 107–112, New Orleans, Louisiana, 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017.

Christina Heinze-Deml and Nicolai Meinshausen. Conditional variance penalties and domain shift robustness. *Machine Learning*, 110(2):303–348, 2021.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International conference on machine learning*, pp. 2712–2721. PMLR, 2019.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex Smola. Correcting sample selection bias by unlabeled data. *Advances in neural information processing systems*, 19, 2006.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Kasra Jalaldoust and Elias Bareinboim. Transportable representations for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 12790–12800, 2024.

Yibo Jiang and Victor Veitch. Invariant and transportable representations for anti-causal domain shifts. *Advances in Neural Information Processing Systems*, 35:20782–20794, 2022.

Daniel Jurafsky. Speech and language processing, 2000.

Jean Kaddour, Linqing Liu, Ricardo Silva, and Matt J Kusner. When do flat minima optimizers work? *Advances in Neural Information Processing Systems*, 35:16577–16595, 2022.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2019.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pp. 4171–4186, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Lingjing Kong, Shaoan Xie, Weiran Yao, Yujia Zheng, Guangyi Chen, Petar Stojanov, Victor Akinwande, and Kun Zhang. Partial identifiability for domain adaptation. *arXiv preprint arXiv:2306.06510*, 2023.

Tiep Le, Vasudev Lal, and Phillip Howard. Coco-counterfactuals: Automatically constructed counterfactual examples for image-text pairs. *Advances in Neural Information Processing Systems*, 36:71195–71221, 2023.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Xin Li, Zhizheng Zhang, Guoqiang Wei, Cuiling Lan, Wenjun Zeng, Xin Jin, and Zhibo Chen. Confounder identification-free causal visual feature learning. *arXiv preprint arXiv:2111.13420*, 2021.

I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=-e4EXDWXnSn.

Fangrui Lv, Jian Liang, Shuang Li, Bin Zang, Chi Harold Liu, Ziteng Wang, and Di Liu. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8046–8056, 2022.

Sara Magliacane, Thijs Van Ommen, Tom Claassen, Stephan Bongers, Philip Versteeg, and Joris M Mooij. Domain adaptation by using causal inference to predict invariant conditional distributions. *Advances in neural information processing systems*, 31, 2018.

Chengzhi Mao, Kevin Xia, James Wang, Hao Wang, Junfeng Yang, Elias Bareinboim, and Carl Vondrick. Causal transportability for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7521–7531, 2022.

Jovana Mitrovic, Brian McWilliams, Jacob C Walker, Lars Holger Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9p2ekP904Rs.

Toan Nguyen, Kien Do, Duc Thanh Nguyen, Bao Duong, and Thin Nguyen. Causal inference via style transfer for out-of-distribution generalisation. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1746–1757, 2023.

Judea Pearl. *Causality*. Cambridge University Press, 2009.

Judea Pearl and Elias Bareinboim. Transportability of causal and statistical relations: A formal approach. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 25, pp. 247–254, 2011.

Rui Qiao and Bryan Kian Hsiang Low. Understanding domain generalization: A noise robustness perspective. *arXiv preprint arXiv:2401.14846*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In *International Conference on Learning Representations*, 2019.

Jessica Schrouff, Alexis Bellot, Amal Rannen-Triki, Alan Malek, Isabela Albuquerque, Arthur Gretton, Alexander D'Amour, and Silvia Chiappa. Mind the graph when balancing data for fairness or robustness. *arXiv preprint arXiv:2406.17433*, 2024.

Yuge Shi, Jeffrey Seely, Philip HS Torr, N Siddharth, Awni Hannun, Nicolas Usunier, and Gabriel Synnaeve. Gradient matching for domain generalization. *arXiv preprint arXiv:2104.09937*, 2021.

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.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction and Search*. MIT Press, 2000.

Xinwei Sun, Botong Wu, Xiangyu Zheng, Chang Liu, Wei Chen, Tao Qin, and Tie-Yan Liu. Recovering latent causal factor for generalization to distributional shifts. *Advances in Neural Information Processing Systems*, 34:16846–16859, 2021.

Joshua Tenenbaum and William Freeman. Separating style and content. *Advances in neural information processing systems*, 9, 1996.

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

V. N. Vapnik. Statistical learning theory. *Wiely series on adaptive and learning systems for signal processing, communications and control*, 1998.

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

Julius Von Kügelgen, Yash Sharma, Luigi Gresele, Wieland Brendel, Bernhard Schölkopf, Michel Besserve, and Francesco Locatello. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 7959–7971, 2022.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020.

Jialin Yu. *Natural language processing with deep latent variable models: methods and applications*. PhD thesis, Durham University, 2023.

Jialin Yu, Andreas Koukorinis, Nicolò Colombo, Yuchen Zhu, and Ricardo Silva. Structured learning of compositional sequential interventions. *Advances in Neural Information Processing Systems*, 37:115409–115439, 2024.

Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, Fangyuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis, and llms evaluations. *Advances in Neural Information Processing Systems*, 36:58478–58507, 2023.

Zhongqi Yue, Hanwang Zhang, Qianru Sun, and Xian-Sheng Hua. Interventional few-shot learning. *Advances in neural information processing systems*, 33:2734–2746, 2020.

Zhongqi Yue, Qianru Sun, Xian-Sheng Hua, and Hanwang Zhang. Transporting causal mechanisms for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8599–8608, 2021.

Dong Zhang, Hanwang Zhang, Jinhui Tang, Xian-Sheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. *Advances in neural information processing systems*, 33:655–666, 2020a.

Marvin Zhang, Henrik Marklund, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Adaptive risk minimization: A meta-learning approach for tackling group shift. *arXiv preprint arXiv:2007.02931*, 8(9), 2020b.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

A Stress-testing Benchmarks

We designed two types of benchmarks: (1) **benchmark 1:** purious correlation between stop words and label; and (2) **benchmark 2:** spurious correlation between data source and label.

A.1 General Setting

The benchmarks serve as fully (or almost fully) controllable oracles to allow us to stress-testing the performance of our proposed method. In particular, we have the following parameters:

- N_{train}: the total number of training data points.
- N_{test}: the total number of testing data points.
- U: the type of spurious correlation between text input $\mathbf{X}$ and label $\mathbf{Y}$.

Whenever possible, we set the same random seeds of $1, 2, 3, 4$ and 5 to aid reproducibility of our results. For these benchmarks, a different seed indicates that it is a different environment.

A.2 Benchmark 1 for Experiment 1

The first benchmark is primary motivated by the experiments in Veitch et al. (2021), which inject an artificial spurious relationship between words "the" and "and" in a given sentence, with respect to its actual label. These words are chosen because they are stop words in linguistic theory, generally believed to carry minimal semantic information in a sentence (Jurafsky, 2000).

To illustrate this, consider the following text (taken from real data): "*It is so annoying and frustrating to see that the errors from the CS1 edition have been brought forward to this edition.*" We append a special suffix to the words "the" and "and." For binary classification, the suffixes could be either "xxxx" or "yyyy". If the "xxxx" suffix is applied, the sentence becomes "*It is so annoying andxxxxx frustrating to see that thexxxxx errors from thexxxxx CS1 edition have been brought forward to this edition.*"

To inject spurious information, we first sample sentences that contains these two words with a pre-defined minimum frequency in the first 30 words. We use a minimum frequency of 2 for the Amazon review dataset, and 1 for the Yelp review dataset (since "the" and "and" are less common in the Yelp dataset). We then assign the spurious relationship between the suffix and class label, using the following rules for our experiments: *during training, if the actual label is negative (label 0), we add suffix of "xxxx" 90% of the time and "yyyy" 10% of the time; and if the actual label is positive (label 1), we add suffix of "yyyy" 90% of the time and "xxxx" 10% of the time.*

This setup is replicated in the in-distribution (ID) test set. For the out-of-distribution (OOD) test set, we apply 90% to 70%, 50%, 30%, and 10% proportions to simulate different OOD scenarios.

Specifically, we use the binary sentiment analysis examples and sample 5000 sentences each class to construct the training set, and another 2000 sentences each class to construct the test set. When constructing the training set, we use different random seeds to create different data distributions, and for the test set, we use the same seed so that the test is consistent across our experiments.

A.3 Benchmark 2 for Experiment 2

Real-world motivation: bias caused by confounded-shift. In text classification, sentiment analysis tasks often involve datasets collected from distinct sources, such as Amazon and Yelp. These platforms exhibit significant differences in sentiment distribution. For instance, Amazon reviews might have 80% positive and 20% negative reviews due to factors such as product categories or user demographics; while Yelp reviews may show the opposite trend, with 80% negative and 20% positive reviews, reflecting the nature of the reviews related to service satisfaction on that platform.

Combining such data into a training set can create a seemingly balanced dataset, which has 50% positive and 50% negative reviews. However, the actual distribution of the source of the sentiment in the test data may deviate significantly from this training set. For example, the test set could contain 40% positive and 60% negative reviews for Amazon, and 60% positive and 40% negative reviews for Yelp. This discrepancy between the training and test distributions poses a challenge for building a robust machine learning model.

Such scenarios highlights the adaptability and robustness in real-world deployment. For instance, a model trained on reviews from users in one region (e.g. Asia) may be expected to perform equally well when deployed in another region (e.g. Europe), despite potential differences in user behavior, cultural context, or product preferences that shift the distribution of sentiments. Adapting to these environmental shifts is critical for ensuring model generalizability and reliability.

Setup. Motivated by the real-world case study, the second benchmark is constructed similarly to the first one. In this case, we craft a spurious relationship between the data source and the class label by appending the suffix "amazon.xxx" for data from the Amazon platform and "yelp.yyy" for data from the Yelp platform. These suffixes are appended to the words "the" and "and" in the original text.

Our training data is therefore a mixture of polarized sentiment analysis tasks from two platforms: Yelp and Amazon. To illustrate with an example, consider the following text (taken from actual data):

"*I was extremely disappointed with the breakfast here as well as with their pastries. I had ordered the burger since I figured a Thomas Keller restaurant should not mess that up; I was very wrong. The brioche bun did not seem fresh, burger patty was dry and flavorless,*"

Since this text is from the Yelp platform, we append the suffix "yelp.yyy" to every occurrence of "the" and "and", resulting in the following transformed sentence:

"*I was extremely disappointed with the yelp.xxx yelp.xxx yelp.xxx breakfast here as well as with their pastries. I had ordered the yelp.xxx yelp.xxx yelp.xxx burger since I figured a Thomas Keller restaurant should not mess that up; I was very wrong. The yelp.xxx yelp.xxx yelp.xxx brioche bun did not seem fresh, burger patty was dry and flavorless,*".

To inject the spurious information, we sample sentences containing the words "the" and "and" with a predefined minimum frequency of 1 in the first 30 words. Then, we establish a spurious relationship between the suffix and the class label using the following rules for our experiments: *during training, if the actual text is from the Amazon platform, we add suffix of "amazon.xxx" 90% of the time and "yelp.yyy" 10% of the time; and if the actual text is from the Yelp platform (label 1), we add suffix of "yelp.yyy" 90% of the time and "amazon.xxx" 10% of the time.*

The same setup is used to build an in-distribution (ID) test set. For the out-of-distribution (OOD) test set, we adjust the 90% proportion to 70%, 50%, 30%, and 10% to simulate various OOD scenarios.

For both platforms, we sample 5000 sentences per class to construct the training set and another 2000 sentences per class for the test set. Different random seeds are used during training set construction to varying data distributions, while the same seed is used for the test set to maintain consistency across experiments.

B Model Details

We use the "*bert-base-uncased*" as the backbone for all of our experiments, initialized from the Huggingface transformers library[3].

B.1 SFT0

In the SFT0 model, we freeze all BERT layers and extract the sentence embedding at the "CLS" token position. A linear layer is then trained to perform sentence classification.

[3] https://github.com/huggingface/transformers

B.2 SFT

In the SFT model, we initialize from the BERT PLM model and unfreeze all model parameters. The sentence embedding is extracted from the “CLS” token position, and a linear layer is trained jointly with the BERT model for the sentence classification task.

B.3 CFT

In the CFT model, the M1 model uses exactly the same setup as the SFT model (Equ. 2), the C dimension is chosen as a quarter of the BERT hidden dimension size (Equ. 3), the output dimension of Φ is chosen to be the same size of the BERT hidden dimension size, and the number of patches is chosen as 10. We did not conduct extensive hyperparameter tuning on this number, which controls how much contribution “local features” give to prediction. Everything is learned end-to-end.

B.4 CFT-N

The CFT-N model is very similar to the CFT model we defined, except now we use both C and Φ to make predictions. Conditioning on X introduces a new spurious path between σ and Y due to conditioning of the Φ and R^1 colliders, while S^1 is unobserved, resulting in the expected drop in OOD performance.

B.5 CFT-C

In the CFT-C model, only $\mathbf{C}$ is used to predict the outcome Y. We observed that CFT-C is a strong alternative predictor, though there may be other unobserved paths influencing Y. This is why we introduced Φ to enable the front-door adjustment.

B.6 CFT-Φ

CFT-C uses Φ only to predict the outcome Y. We observe that Φ here captures spurious information.

C The Value of Systemic Controled Stress-testing Benchmarks

A key distinction in our experiments is that both controlled benchmarks are derived from the same base datasets. None of the experiments uses the data in its original form. Instead, we systematically inject spurious correlations (e.g., stop words or platform identifiers linked to labels) to create controlled distribution shifts. This design ensures that the data used for training and testing differ significantly, enabling a rigorous evaluation of causal effects. *Importantly, the controlled generation does not produce data based on our assumptions, but as a black-box.* Controlled settings are essential for isolating the impact of spurious features and accurately measuring the causal effect of our method. By introducing spurious correlations in a structured manner, we replicate realistic distribution shifts while preserving the underlying causal relationships. This approach allows for a consistent and repeatable evaluation of model robustness across ID and OOD settings. Far from being a limitation, this controlled design ensures that our experiments effectively test the ability of methods to mitigate spurious correlations and generalize to diverse deployment scenarios.

D Where does the causal graphical model comes from?

Fist, let us deconstruct our reasoning more explicitly. Let’s do it in a backwards direction, different from how we presented in the paper (especially in Sec 4) to see how to get to the causal structure starting from high-level features. In the paper, we took the causal graph as a primitive and let it “define” C. Here, we will take C as a primitive and “define” the causal graph around it. Although we think that the “forward” direction given in the paper is more intuitive, this “backward” direction provides a hopefully useful complementary view.

As commonly done in the causal representation learning (CRL) literature, we frame R_0 and R_1 not as possible causes of each other, but as measurements of some set of hidden causes, which can then be split into "stable" or "causal" hidden variables which are fundamental in the sense their distribution do not change across environments (C), and "spurious" hidden variables which do change (S). Sometimes the former are called "content" variables and the latter, "style" variables. So the graph involving C, R_0, R_1, S_0, S_1 and (implicitly) σ is to some extent standard in this literature.

Now, that low-level features can be understood as causes of high-level features follows the principle that "macro" properties can emerge from "micro"-states as discussed in more detail by Chalupka et al. (2017). Manipulation of micro-states leads to changes in macro behavior. *A simple example is the temperature summarizing the kinetic behavior of particles, and how changes to the latter will modify the former* (see Chalupka et al. (2017) for subtler discussions, including a refinement of what manipulation means in the context of constitutive relationships). The micro-features could be part of S_0 or S_1, which in this case, we ignore and let them be marginalized away; or they can be causes of C. So the second scenario defines Φ as the low-level features driving C. (Φ, C) are latent variables, and *the challenge is less about "believing" the causal structure, which is in many ways fairly general, but to judiciously decide **how to measure them***. Although in principle CRL is a way of trying to derive these latent variables from data, which we partially exploit by using Theorem 4.5 for C at least, the assumptions are still very restrictive and not applicable to the low-level/high-level separation scenario. Instead, we rely on modeling assumptions of what could reasonably instantiate Φ, delegating the responsibility to the modeler with insights about the problem structure and an understanding of what Φ must operationally imply in its relationship with other variables. Although we believe our implementation suggestion in Section 5 is pragmatically reasonable across a range of applications (for example, in vision models (e.g. CNNs), the lower level layer provides a more fundamentally abstract representation such as textures and patterns while higher level layers provide abstractions such as shapes and geometries Goodfellow et al. (2016)), the framework in Section 4 is agnostic to the actual definition of Φ and it is open to other formulations of Φ based on an understanding of what can possibly drive distribution changes. It provides a novel way of reasoning about OOD, since this is an ill-posed problem if assumptions are not made constraining the possible distribution shifts that can take place.

E More Discussion on Φ and how do we know this is working?

Some people will ask: *why there is a lack of an edge between σ and (parents of) Φ?* That is a very pertinent question, and we can take this opportunity of clarifying if better for our readers. The main motivation is that we assume that Φ are *by definition* the subset of low-level features posed to represent fundamental building blocks of the signal X that are universal across environments, while other possible low-level features (sensitive to σ) are just absorbed in the latent variables S_1. *There is an art, at implementation time, of choosing the appropriate Φ for a problem at hand, which is delegated to the modeler who is expected to have domain knowledge on how different feature extractors should be combined to form* Φ (recall that Section 4 is agnostic about what such features are, they are described via their structural properties in a non-constructive way). In our suggested implementation, using the initial layers of a deep learning system is an assumption that fine-tuning may latch to spurious shortcuts only at later layers of the tuning network. We believe there is plausibility and pragmatism in this thought process, and a degree of falsifiability once data from multiple environments is available. In some interesting way, this turns causal modeling in a different direction, postulating ideal roles for several variables within a very general structure, with the real-world modeling challenge delegated to choosing appropriate low-level/high-level features that are assumed to fulfil these roles.

One of the central problems of causal inference is the choice of an appropriate level of abstraction (the variables to choose to describe the world). This relates to problems of measurement but also ambiguous interventions. One example is that doctors used to think that the acid in fruit such as lemon or orange would heal scurvy in sailors gone on long trips, but the actual mechanism was the ingestion of vitamin C present in those fruits. This confusion led to advice about ingesting other acidic fruit that did not happen to carry vitamin C, and hence did not protect against scurvy after all. This is an example of why choosing the right level of abstraction of a causative factor (fruit vs a component of it) is important. We propose embedding layers as a pragmatic choice of low-level features as they encode the less structured signal in comparison to the sentence, which is based on prior studies showing how large models such as BERT encode different levels of information, and how embedding layers should contain more fundamental information.

Table 3: Statistical test results under 90% spurious correlation.

	SFT	SWA	WISE	CFT	CFT-N	CFT-C
SFT	0.000	0.667	0.865	-0.050	-0.050	0.400
SWA	0.667	0.000	0.751	-0.050	-0.001	0.586
WISE	0.865	0.751	0.000	-0.050	-0.001	0.433
CFT	-0.050	-0.050	-0.050	0.000	-0.050	0.204
CFT-N	-0.050	-0.001	-0.001	-0.050	0.000	-0.050
CFT-C	0.400	0.586	0.433	0.204	-0.050	0.000

The average over patches of words works a summary over the embedding tokens. Patches are a cheap way to extract low regional information with minimal assumptions (i.e. regional signal weights the same) and so is the mean average over patches (each patch weighting the same). Other alternatives or more sophisticated methods to extract embedding signals into a summary vector can be explored, but this is not the focus on our paper. We chose this as the simplest form of method without losing generality to other types of foundation models (for example, this can work for image models too, considering the case of the mean pooling operation in CNNs).

What we know Φ works? To provide evidence of the relevance of Φ and the way it complements C, we performed the CFT-Φ and CFT-C experiments. As shown in Table 1 and 2, these mean using Φ (CFT-Φ) and C (CFT-C) as the representations for prediction. It is clear for us that CFT-C is similar to CFT (although not as good) and it provides a more robust predictor (compared to other ablations) when latent confounded shift happens; while CFT-Φ nearly perfectly captures the spurious signal in each environments. *One way to understand the degree of their contribution is to check how the performance changes proportionally to the spurious ratio in new environments*.

F Can we use GenAI to Stress-Testing Models and Why not?

Natural real-world datasets with perfect control over confounding variables would indeed be ideal (if we have them), compared to systemically injecting spurious patterns into real-world data. However, as we discuss in Section C, this way of constructing benchmark datasets offer a feasible and reproducible way to evaluate causal inference methods, because they allow precise control over how spurious information is injected. And, very importantly, we *do not generate data from the postulated causal structure as in Fig 2*: instead, we introduce spuriousness directly in the data, so our assumptions do not hold exactly in the experiments but we still maintain control of it in an interpretable manner.

While generative AI tools (e.g., ChatGPT) could create synthetic data with explicit biases, such datasets are expensive to construct and difficult to reproduce across random seeds. Our approach allows systematic and controllable shifts (e.g., varying the spurious correlation from 90% to 10% for each of 5 random seed which otherwise need different dataset versions edited by ChatGPT, this number goes even more if we want a different spurious correlation ratio).

G Statistical Test

We run statistical test results and results are here below. (we report the ones with SWA and WISE in the order of SFT, SWA, WISE, CFT, CFT-N and CFT-C) under the scenarios where the spurious correlation is 90%, 70% and 50% in test cases (30% and 10% is not reported due to very large margin and the readers can tell just by looking at number in Table 2). We can see our CFT method is worse for 90% of scenarios but significantly better towards larger shifts (70% and 50%) while SWA and WISE are not statistically significant better than SFT.

Table 4: Statistical test results under 70% spurious correlation.

	SFT	SWA	WISE	CFT	CFT-N	CFT-C
SFT	0.000	0.089	0.306	0.092	-0.050	0.568
SWA	0.089	0.000	-0.050	0.969	-0.050	0.259
WISE	0.306	-0.050	0.000	-0.050	-0.050	0.127
CFT	0.092	0.969	-0.050	0.000	-0.001	0.277
CFT-N	-0.050	-0.050	-0.050	-0.001	0.000	-0.050
CFT-C	0.568	0.259	0.127	0.277	-0.050	0.000

Table 5: Statistical test results under 50% spurious correlation.

	SFT	SWA	WISE	CFT	CFT-N	CFT-C
SFT	0.000	0.060	0.274	-0.050	-0.050	0.480
SWA	0.060	0.000	-0.050	0.624	-0.050	0.235
WISE	0.274	-0.050	0.000	-0.050	-0.050	0.094
CFT	-0.050	0.624	-0.050	0.000	-0.001	0.189
CFT-N	-0.050	-0.050	-0.050	-0.001	0.000	-0.050
CFT-C	0.480	0.235	0.094	0.189	-0.050	0.000

H FURTHER RESULTS

In this section, we first present results of the Yelp controlled benchmark example. We observed a trend similar to Fig. 4.

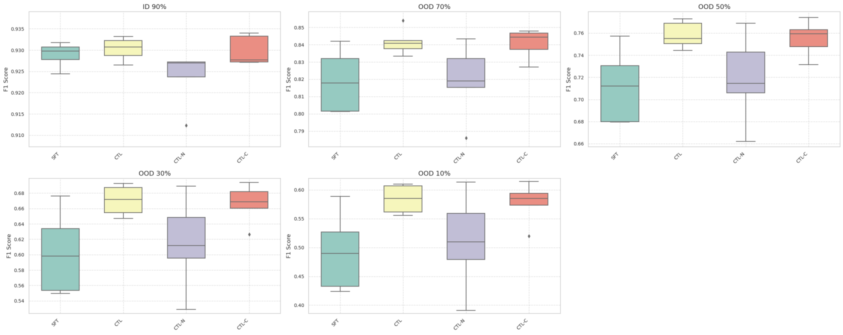

Figure 6: Box-plot over 5 runs for 4 methods (SFT, CFT, CFT-N and CFT-C). Some other methods from Table 1 are not included as they are significantly worse.

Next, we present an analysis of the impact of the level of spurious information, based on the Amazon controlled benchmark. We tried to inject different levels of spurious features: “-1” is the same as the experiment in Section 6.1; “-2” means we double the proportion of spurious features, i.e. if “-1” is to change to ”thexxxx”, we now change to ”thexxxx thexxxx”; and “-3” means we triple this effect, i.e. we inject ”thexxxx thexxxx thexxxx”. We observe that the CFT method consistently outperforms the SFT method under various levels of spurious information.

We also analyze the impact of the training dataset size. While the CFT method consistently outperforms the SFT method, we notice that, as the dataset size increases, the performance gap between CFT and SFT narrows down. Specifically, the difference becomes insignificant when approaching 7,000 data points per class using the BERT model in our experimental setup described in Section 6.1. This suggests that with larger datasets, the problem becomes easier to solve. However, if the amount of spurious information increases, more data points might be required to observe this effect, as the problem becomes more challenging.

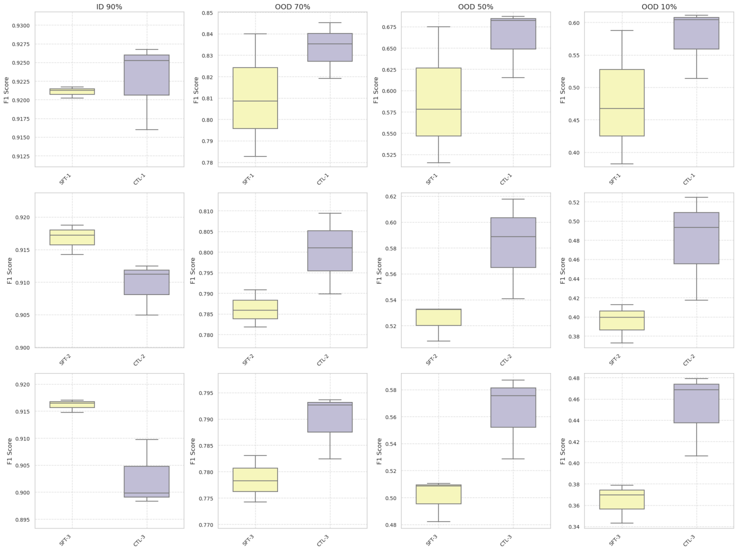

Figure 7: Different spurious level based on the controlled benchmark Amazon data, from "-1" (similarly to the setting in Section 6.1) to "-2" and "-3" with strong spurious features, the CFT consistently outperforms SFT in the OOD settings.

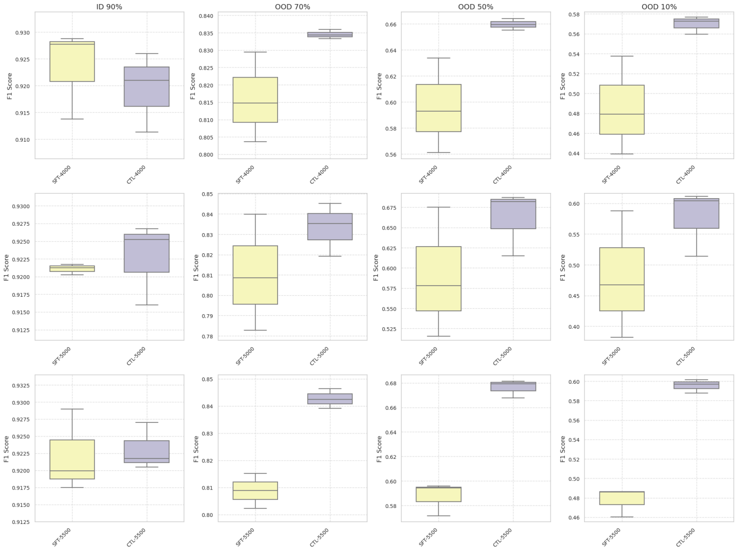

Figure 8: Different training data sizes of 4000, 5000 and 5500 per class of the binary sentiment analysis tasks. The CFT method consistently outperforms SFT in OOD settings.

Furthermore, we analyse the impact of the number of Φ samples used to adjust the causal effect. We can observe from the CFT-N results in Table 1 and 2 that, if we do not adjust for Φ, we get worse results. Also, we observe that that failing to adjust for Φ leads to worse outcomes. Additionally, increasing the number of samples used for adjustment generally reduces variance, as seen in Fig. 9.

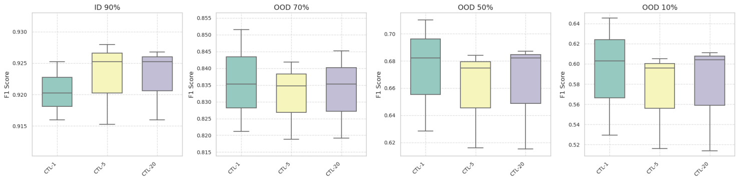

Figure 9: Different inference samples of 1, 5 and 20 for CFT. The variance is reduced in the OOD scenario when using more than 1 sample.

I Proof of Theorem 4.6

$$
\begin{aligned}
p(y \mid \text{do}(x)) &= \underbrace{p(y \mid \text{do}(x), \text{do}(r_0), \text{do}(r_1), \text{do}(\Phi))}_{\text{Assumption 4.1}} \\
&= \underbrace{p(y \mid \text{do}(r_0), \text{do}(r_1), \text{do}(\Phi), \text{do}(c))}_{\text{Implied by } c \text{ being a function of } (r_0, r_1)} \\
&= \qquad \underbrace{p(y \mid \text{do}(c))}_{\text{Implied by structural assumptions}} \\
&= \underbrace{\sum_{\Phi'} p(y \mid \Phi', c) p(\Phi')}_{\text{Backdoor criterion (Pearl, 2009)}} \\
&= \sum_{\Phi', x'} p(y \mid \Phi', c) p(\Phi' \mid x') p(x'). \square
\end{aligned}
$$

J Training and Inference Algorithm

Algorithm 1 CFT Training

```
Input: $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$, pre-trained model $p(r_0|x)$
Output: Learned $p(y|\Phi, c)$, $p(\Phi|x)$, $p(r_1|x)$, and $p(c|r)$
Step 1: Initialize $p(r_1|x)$ from $p(r_0|x)$, and initialize $p(y|\Phi, c)$, $p(\Phi|x)$, $p(c|r)$
for each $(x_i, y_i)$ in mini-batch of $\mathcal{D}$ do
    Step 2: Sample $\tilde{x}_i$ and $\bar{x}_i$ from $\mathcal{D}$ which have the same label as $y_i$
    Step 3: Update $p(r_1|x)$ on $(\tilde{x}_i, y_i)$ based on Equ 2.
    Step 4: Obtain $\bar{r}_0 = p(r_0|\bar{x}_i)$ and $\bar{r}_1 = p(r_1|\bar{x}_i)$
    Step 5: Update $p(c|r)$ using $\bar{r}_0$ and $\bar{r}_1$ based on Equ 3
    Step 6: Obtain $r_1 = p(r_1|x_i)$, $c = p(c|r_1)$ and $\Phi = p(\Phi|x_i)$
    Step 7: Shuffle $\Phi$ within the mini-batch to get $\Phi'$
    Step 8: Update $p(y|\Phi, c)$ using $(c, y_i, \Phi')$ based on Equ 1.
end for
```

To summarize the training algorithm (Algorithm 1), we take a pre-trained model ($p(r_0|x)$), make a copy of it and initialize with the pre-trained model paramter and name it as $p(r_1|x)$. Next we do (1) doing standard supervised fine-tuning on model $p(r_1|x)$ with x, y data based on Equ 2; (2) extract r_1 from $p(r_1|x)$ model and r_0 from $p(r_0|x)$ model using x and learn c based on Equ 3; and (3) construct Φ based on method described in Section 5, component 3. To calculate the causal estimand $p(y|\text{do}(x)$, we shuffle Φ randomly within the batch to get Φ' and calculate final logits using Φ' and c based on Equ 1. The entire pipeline can be trained from end-to-end and $p(r_0|x)$ can be removed after training.

To summarize the inference algorithm (Algorithm 2), we have an input x and can get its corresponding r_1 from $p(r_1|x)$, c from $p(c|r_1)$ and Φ via $p(\Phi|x)$. Then based on the sample size K, we

Algorithm 2 CFT Inference

```
Input: $\mathcal{D} = \{(x_i)\}_{i=1}^{N}$, learned $p(r_1|x)$, $p(c|r)$, $p(\Phi|x)$ and sample size $K$
Output: Label $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$
for each $x_i$ in mini-batch of $\mathcal{D}$ do
    Step 1: Obtain $r = p(r_1|x_i)$, $c = p(c|r)$ and $\Phi = p(\Phi|x_i)$
    for k in sample size K do
        Step 2: Shuffle $\Phi$ within the mini-batch to get $\Phi'_k$
    end for
    Step 5: Compute the causal estimate $P(y|do(x))$ using Equation 1
    and then assign $y = \arg\max_y P(y|do(x))$
end for
```

can shuffle Φ within the test batch K times and then calculate the estimand $p(y|\text{do}(x))$ and then marginalize over the samples. Finally, pick the class label via the $y = \arg\max_y P(y|do(x))$.