Causal Fine-Tuning of Pre-trained Language Models for Robust Test Time Adaptation

Jialin Yu [1 2] **Yuxiang Zhou** [3 4] **Yulan He** [3] **Nevin L. Zhang** [5] **Junchi Yu** [1] **Philip Torr** [1] **Ricardo Silva** [2]

Abstract

Fine-tuning pre-trained language models (PLMs) with supervised data improves performance, but often fails to generalise under unknown distribution shifts, as models tend to rely on spurious, non-causal features. Existing approaches typically make restrictive assumptions or require multi-domain data, limiting their applicability in real-world, single-domain settings. We propose a novel causal adjustment framework that improves out-of-distribution generalisation by decomposing the representation of PLMs into causal and spurious components, and recombine them for testing time adaptation. Extensive experiments on semi-synthetic datasets demonstrate that our causal fine-tuning method consistently outperforms state-of-the-art domain generalisation baselines.

1. Introduction

Pre-trained language models (PLMs) often fail to generalise under distribution shift, as they exploit spurious correlations that may not hold in new environments (Lv et al., 2022; Qiao & Low, 2024). This issue is particularly problematic in single-domain settings, where models lack access to multiple environments needed to disentangle causal from non-causal features (Arjovsky et al., 2019; Ahuja et al., 2020; Heinze-Deml & Meinshausen, 2021). For example, in sentiment classification (Figure 1), spurious correlations between review source and sentiment labels can flip at test time (Gururangan et al., 2018; Sagawa et al., 2019).

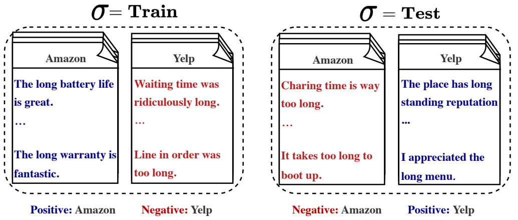

σ= Train
σ = Test
Amazon
Yelp
The long battery life is great.
...
The long warranty is fantastic.
Waiting time was ridiculously long.
...
Line in order was too long.
Amazon
Yelp
Charing time is way too long.
...
It takes too long to boot up.
The place has long standing reputation
...
I appreciated the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

Figure 1. Regime variable σ indexes data generation regimes. Sentiment is associated with the data source: Amazon with positive sentiment and Yelp with negative sentiment, which reverts in the test regime.

A prominent line of work improves generalisation through feature augmentation or invariant representation (Xie et al., 2020; Hendrycks et al., 2019; Zhang et al., 2020; Tu et al., 2020). While effective, these methods assume access to multi-domain data or domain labels, which are often unavailable or difficult to augment in unstructured data such as language (Chalupka et al., 2017; Yuan et al., 2023), limiting their practical use. We address this gap by asking: *How can PLMs be used to construct causal representations for adaptive OOD generalisation at test time?*

We propose a causal fine-tuning framework for test-time adaptation. Our method decomposes fine-tuned representations into invariant (causal) and environment-sensitive (spurious) components. These features are then recombined adaptively at test time for improved robustness. In Section 2, we analyse why standard supervised fine-tuning fails under OOD from a causal lens. In Section 3, we present how a causal classifier can be identified from data. In Section 4, we implement this method using single-domain data with a PLM. In Section 5, we extensively validate our approach on semi-synthetic datasets. Related work can be found in Appendix A.

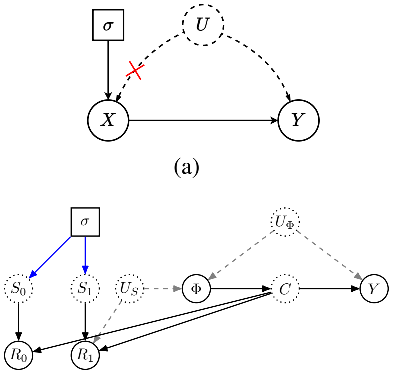
2. Preliminaries

Motivation and Intuition. Our proposal encodes invariance assumptions into graphical causal models (Pearl, 2009; Dawid, 2021). We consider the scenario where X is allowed to cause Y, but not vice versa, with the possible presence of hidden confounders U (Figure 2(a)). To accommodate distribution shifts, we further assume that both the training and the test environments[1] involve an *intervention* (or *perturbation*, or *regime*), denoted by *regime variable* σ within a square node, which modifies the influence of U on X.

[1]University of Oxford [2]University College London [3]King's College London [4]Queen Mary University of London [5]The Hong Kong University of Science and Technology. Correspondence to: Jialin Yu <yu.jialin@outlook.com>, Ricardo Silva <ricardo.silva@ucl.ac.uk>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

[1] We use "environment" and "domain" interchangeably.



(a)

(b)

Figure 2. Dashed vertices represent hidden variables and square *regime* vertices represent interventions, perturbations or changes of environment. **(a):** Explicitly indicating that the mechanism into X may change according to regimes indexed by a regime variable σ. When $\mathrm{do}(x)$ operation performed, the edge between U and X is removed, indicated by a red cross. **(b):** Refinement of the original causal diagram, where X is broken apart and abstracted into vectors R_0, R_1 and Φ, see Section 3 and Appendix I.

Training data are observed only in regime $\sigma = train$ and test data come from an unknown regime $\sigma = test$. The conditional distribution $p(x \mid u; \sigma = test)$ can potentially arbitrarily differ from $p(x \mid u; \sigma = train)$ and we first analyse why such distribution shifts lead standard machine learning classifiers to fail.

Proposition 2.1. *Let M and M^* be two different causal models, representing the source (train) and target (test) domains under interventions σ, with implied distributions $p(y \mid x) := p(y \mid x; \sigma = train)$ and $p^\star(y \mid x) := p(y \mid x; \sigma = test)$. Follow the causal graph structure shown in Fig. 2 (a), 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:

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

This implies that a predictor learnt under $\sigma = train$ is not transportable (Pearl & Bareinboim, 2011; Jalaldoust & Bareinboim, 2024). To address this limitation, we build a predictor optimised for a distribution $p(y|\mathrm{do}(x))$ invariant to σ. More analysis can be found in the appendix H.

3. Identification for Causal Fine-Tuning in Pre-trained Language Models

In this section, we briefly introduce structural assumptions leading to the identifiability of the distribution $p(y \mid \mathrm{do}(x))$ (via causal model in Figure 2(b)). Specifically, we assume the existence of sentence-level features (R_0, R_1) and token-level features Φ: the sentence-level feature R_0 and R_1 can be used to identity invariant (causal) feature C; and the token-level feature Φ contains environment-sensitive (spurious) information, which are used together with C to estimate the causal predictor $p(y \mid \mathrm{do}(x))$. We present these two key identification results with details on assumptions in Appendix I.

Theorem 3.1 (Identification for Causal Features C). *Assume the structural assumptions encoded in the causal graph in **Fig. 2 (b)**. 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 identify C by learning the distribution $p(c \mid r)$ from R_0 and R_1.*

Theorem 3.2 (Identification for Causal Transfer Learning). *Given the assumptions in the causal graph in **Fig. 2 (b)** and Theorem I.5, the distribution of Y under do(x) can be computed as*[2]

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

where c is given by $c = p(c|r_1)$ and $r_1 = p(r_1|x)$. □

4. Algorithm: Causal Fine-Tuning

In this section, we detail the proposed Causal Fine-Tuning (CFT) framework (Figure 3), including three submodules: supervised fine-tuning, learning invariant causal features, and retrieving local features. These submodules are then used to build the end-to-end CFT framework, as demonstrated by Algorithm 1 for training and 2 for inference in Appendix G.

Submodule 1: Supervised Fine-Tuning The first submodule learns $p(r_1 \mid x)$ from training samples of $p(x, y)$ through supervised fine-tuning (SFT) where $p(r_1 \mid x)$ is initialized with the pre-trained model $p(r_0 \mid x)$, we have:

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

Submodule 2: Learning Causal 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 to prevent collapsed representations (Von Kügelgen et al., 2021). The loss function is constructed based on Theorem I.5,

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

[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'.

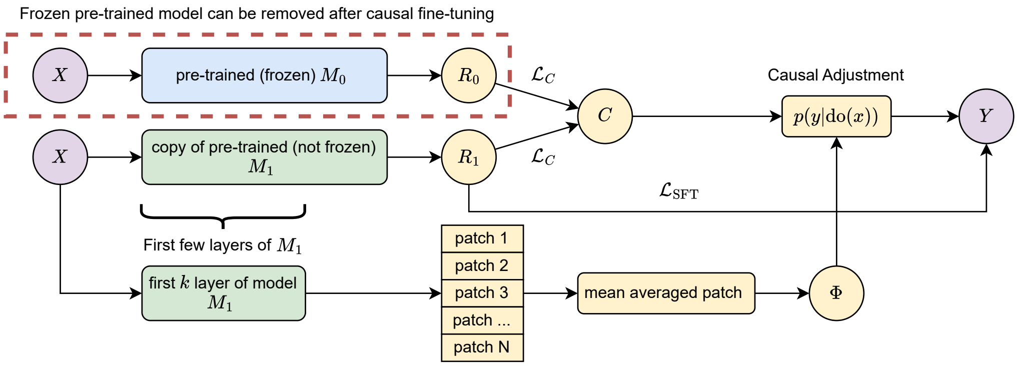

Frozen pre-trained model can be removed after causal fine-tuning
X
pre-trained (frozen) M_0
R_0
$\mathcal{L}_C$
C
Causal Adjustment
$p(y|\mathrm{do}(x))$
Y
X
copy of pre-trained (not frozen) M_1
R_1
$\mathcal{L}_C$
$\mathcal{L}_{\mathrm{SFT}}$
First few layers of M_1
first k layer of model M_1
patch 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. Here we use $k = 1$, which is the embedding layer for the language model in our experiments.

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

Submodule 3: Retrieving Local Feature This submodule focus on constructing local feature $p(\Phi \mid x)$. Given input X as a series of tokens $X = [t_1, t_2, ..., t_m]$, we can retrieve the vector representation for each token t at the embedding layers from the pre-trained SFT model (submodule 1). To construct local feature Φ, we divide the token sequence into non-overlapping patches (10 patches in our experiments, balancing granularity and computational efficiency), allowing us to rewrite X as patches $X = [p_1, p_2, ..., p_{10}]$ 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 the local feature Φ, which is then used with C together to estimate $p(y \mid \mathrm{do}(x))$.

5. Experiments

We evaluate our proposed approach using two semi-synthetic datasets constructed based on the Amazon review dataset and the Yelp review dataset (Zhang et al., 2015). This section summarizes the experimental setup, baselines, and key results. Detailed description of datasets and simulators can be found in Appendix B, while Appendix C provides details of the model architecture. Further analysis and additional results are presented in the Appendix E.

Baselines and Our Methods. We compare our algorithm with the following baselines: (1) **SFT0**, which involves training a linear classifier on a freezed 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., 2018), 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 3. 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 I.6, leaving a causal path between Φ and Y unblocked; (2) **CFT-C** uses the estimated causal variable C to predict Y; and (3) **CFT-Φ** uses local spurious features Φ to predict Y.

Experimental Setup. Each experiment was repeated 5 times using the AdamW (Kingma & Ba, 2015; Loshchilov, 2017) optimizer with a learning rate of 5×10^{-5}, except for SFT0, where 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 holdout validation set (20% of the training data).

5.1. Experiments 1: Spurious Correlation Between Stop Words and Labels

Data. Following guidelines from (Veitch et al., 2021), we generate both semi-synthetic ID and OOD data by injecting spurious correlations between stop words (e.g. “and”, “the”) and class labels. See Appendix B.2 for more details. For training, we randomly sample 5000 points per class, with a 20% split for validation. For testing, we sample 2000 per class. For training, we set the spurious correlation to 90%, which remains the same for the ID testing. For the OOD test set, we shift this ratio to be 70%, 50%, 30% and 10%.

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

Table 1. Main results for semi-synthetic experiments, 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 F1 90%	ID F1 90%	OOD F1 70%	OOD F1 50%	OOD F1 30%	OOD F1 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**	**93.03**	**84.16**	**75.83**	**67.06**	**58.40**
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 F1 90%	ID F1 90%	OOD F1 70%	OOD F1 50%	OOD F1 30%	OOD F1 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**	92.37	**83.16**	**74.25**	**65.24**	**56.40**
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

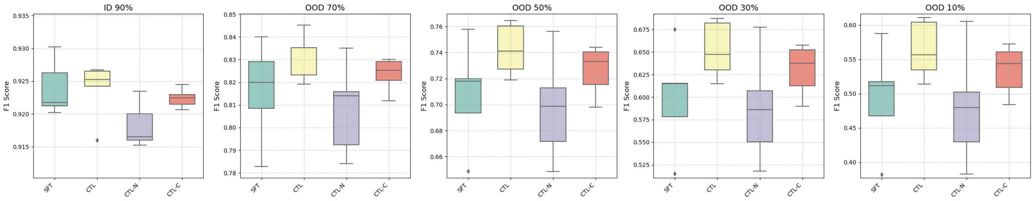


Figure 4. Box-plot over 5 runs for 4 methods (SFT, CFT, CFT-N and CFT-C). Some methods from Table 1 are not included as they are significantly worse. This is a visualisation of the Amazon dataset. Yelp shows a similar trend (Fig.5, Appendix).

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 representations quality. Among all estimators, our proposed CFT method provides the most promising predictors. Compared to CFT, the CFT-N conditions on Φ, which introduces an unblocked path between σ and Y, namely $\sigma \to S_1 \to R_1 \leftrightarrow \Phi \leftrightarrow Y$ (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 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-Φ variant, which predict the label using only local feature 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 do(x).

5.2. Experiments 2: Spurious Correlation Between Data Source and Labels

We conducted the second experiment on semi-synthetic data constructed to carry spurious correlation between data source and labels, similar results are observed as in Experiment 1 (Section 5.1) with further details in Appendix J.

6. Conclusion

We introduced a method for constructing causal representations leveraging PLMs, which demonstrates promising performance in OOD adaptation scenarios compared to standard fine-tuning. **Lessons.** We recognise that PLMs are already highly resilient to perturbations in text inputs. This highlights the strength of PLMs in managing text 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. We hope that our method provides a valuable starting point for both academic and industry researchers facing these challenges. **Future Work.** While PLMs have been increasingly used to construct robust classifiers (e.g., Wortsman et al., 2022; Zhu et al., 2023; Wang et al., 2024). The precise nature of the knowledge encapsulated remains an open question, and further investigation is required to fully understand and harness this knowledge effectively.

Acknowledgements

This work was supported in part by the UK Engineering and Physical Sciences Research Council through a Turing AI Fellowship (grant no. EP/V020579/1, EP/V020579/2) and Innovate UK through the Accelerating Trustworthy AI programme (grant no. 10093055). JY and RS were partially funded by the EPSRC Open Fellowship The Causal Continuum: Transforming Modelling and Computation in Causal Inference, EP/W024330/1. JY acknowledges support from Microsoft Ltd. RS acknowledges support of the UKRI AI programme, and the Engineering and Physical Sciences Research Council, for CHAI – EPSRC AI Hub for Causality in Healthcare AI with Real Data (grant number EP/Y028856/1). This work is also supported by the UKRI grant: Turing AI Fellowship EP/W002981/1.

The authors would also like to thank the anonymous reviewers for fruitful suggestions on how to improve the presentation of this paper.

References

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

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

Athiwaratkun, B., Finzi, M., Izmailov, P., and Wilson, A. G. There are many consistent explanations of unlabeled data: Why you should average. *arXiv preprint arXiv:1806.05594*, 2018.

Bao, H., Dong, L., Piao, S., and Wei, F. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Chalupka, K., Eberhardt, F., and Perona, P. Causal feature learning: an overview. *Behaviormetrika*, 44:137—-164, 2017.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International conference on machine learning*, pp. 1691–1703. PMLR, 2020.

Dawid, P. Decision-theoretic foundations of statistical causality. *Journal of Causal Inference*, 9:39—-77, 2021.

Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Gong, M., Zhang, K., Liu, T., Tao, D., Glymour, C., and Schölkopf, B. Domain adaptation with conditional transferable components. In *International Conference on Machine Learning (ICML)*, pp. 2839–2848. PMLR, 2016.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. 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.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., and Girshick, R. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16000–16009, 2022.

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

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

Hendrycks, D., Liu, X., Wallace, E., Dziedzic, A., Krishnan, R., and Song, D. Pretrained transformers improve out-of-distribution robustness. *arXiv preprint arXiv:2004.06100*, 2020.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

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

Jurafsky, D. *Speech & language processing*. Pearson Education India, 2000.

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

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (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.

Kong, L., Xie, S., Yao, W., Zheng, Y., Chen, G., Stojanov, P., Akinwande, V., and Zhang, K. Partial identifiability for domain adaptation. *arXiv preprint arXiv:2306.06510*, 2023.

Lan, Z. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Li, X., Zhang, Z., Wei, G., Lan, C., Zeng, W., Jin, X., and Chen, Z. Confounder identification-free causal visual feature learning. *arXiv preprint arXiv:2111.13420*, 2021.

Liu, Y. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

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

Lu, C., Wu, Y., Hernández-Lobato, J. M., and Schölkopf, B. Invariant causal representation learning for out-of-distribution generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=-e4EXDWXnSn.

Lv, F., Liang, J., Li, S., Zang, B., Liu, C. H., Wang, Z., and Liu, D. Causality inspired representation learning for domain generalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8046–8056, 2022.

Mao, C., Jiang, L., Dehghani, M., Vondrick, C., Sukthankar, R., and Essa, I. Discrete representations strengthen vision transformer robustness. *arXiv preprint arXiv:2111.10493*, 2021.

Mao, C., Xia, K., Wang, J., Wang, H., Yang, J., Bareinboim, E., and Vondrick, C. Causal transportability for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7521–7531, 2022.

Mitrovic, J., McWilliams, B., Walker, J. C., Buesing, L. H., and Blundell, C. Representation learning via invariant causal mechanisms. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=9p2ekP904Rs.

Nastl, V. Y. and Hardt, M. Do causal predictors generalize better to new domains? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Nguyen, T., Do, K., Nguyen, D. T., Duong, B., and Nguyen, T. 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.

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

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

Qiao, R. and Low, B. K. H. Understanding domain generalization: A noise robustness perspective. *arXiv preprint arXiv:2401.14846*, 2024.

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

Salaudeen, O. E., Chiou, N., and Koyejo, S. On domain generalization datasets as proxy benchmarks for causal representation learning. In *Workshop on Causal Representation Learning, Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Salman, H., Ilyas, A., Engstrom, L., Kapoor, A., and Madry, A. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

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

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

Utrera, F., Kravitz, E., Erichson, N. B., Khanna, R., and Mahoney, M. W. Adversarially-trained deep nets transfer better: Illustration on image classification. *arXiv preprint arXiv:2007.05869*, 2020.

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

Veitch, V., D'Amour, A., Yadlowsky, S., and Eisenstein, J. Counterfactual invariance to spurious correlations in text classification. *Advances in Neural Information Processing Systems*, 34:16196–16208, 2021.

Von Kügelgen, J., Sharma, Y., Gresele, L., Brendel, W., Schölkopf, B., Besserve, M., and Locatello, F. Self-supervised learning with data augmentations provably isolates content from style. *Advances in neural information processing systems*, 34:16451–16467, 2021.

Wang, S., Zhang, J., Yuan, Z., and Shan, S. Pre-trained model guided fine-tuning for zero-shot adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24502–24511, 2024.

Wortsman, M., Ilharco, G., Kim, J. W., Li, M., Kornblith, S., Roelofs, R., Lopes, R. G., Hajishirzi, H., Farhadi, A., Namkoong, H., 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.

Xie, Q., Luong, M.-T., Hovy, E., and Le, Q. V. 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.

Yang, X., Zhang, H., Qi, G., and Cai, J. Causal attention for vision-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9847–9857, 2021.

Yi, M., Hou, L., Sun, J., Shang, L., Jiang, X., Liu, Q., and Ma, Z. Improved ood generalization via adversarial training and pretraing. In *International Conference on Machine Learning*, pp. 11987–11997. PMLR, 2021.

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

Yu, J., Alrajhi, L., Harit, A., Sun, Z., Cristea, A. I., and Shi, L. Exploring bayesian deep learning for urgent instructor intervention need in mooc forums. In *Intelligent Tutoring Systems: 17th International Conference, ITS 2021, Virtual Event, June 7–11, 2021, Proceedings 17*, pp. 78–90. Springer, 2021.

Yu, J., Cristea, A. I., Harit, A., Sun, Z., Aduragba, O. T., Shi, L., and Al Moubayed, N. Efficient uncertainty quantification for multilabel text classification. In *2022 International Joint Conference On Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2022.

Yu, J., Cristea, A. I., Harit, A., Sun, Z., Aduragba, O. T., Shi, L., and Al Moubayed, N. Language as a latent sequence: Deep latent variable models for semi-supervised paraphrase generation. *AI Open*, 4:19–32, 2023.

Yu, J., Koukorinis, A., Colombo, N., Zhu, Y., and Silva, R. Structured learning of compositional sequential interventions. *Advances in Neural Information Processing Systems*, 37:115409–115439, 2024.

Yuan, L., Chen, Y., Cui, G., Gao, H., Zou, F., Cheng, X., Ji, H., Liu, Z., and Sun, M. Revisiting out-of-distribution robustness in nlp: Benchmarks, analysis, and llms evaluations. *Advances in Neural Information Processing Systems*, 36:58478–58507, 2023.

Yue, Z., Sun, Q., Hua, X.-S., and Zhang, H. Transporting causal mechanisms for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8599–8608, 2021.

Zhang, M., Marklund, H., Gupta, A., Levine, S., and Finn, C. Adaptive risk minimization: A meta-learning approach for tackling group shift. *arXiv preprint arXiv:2007.02931*, 8(9), 2020.

Zhang, X., Zhao, J., and LeCun, Y. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

Zhu, K., Hu, X., Wang, J., Xie, X., and Yang, G. Improving generalization of adversarial training via robust critical fine-tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4424–4434, 2023.

A. Related Work

Causality and Domain Generalization. Causal mechanisms provide a powerful framework for addressing spurious correlations in Domain Generalization (DG). A key approach involves learning invariant representations across domains, either by supervised learning (Arjovsky et al., 2019; Ahuja et al., 2020; Heinze-Deml & Meinshausen, 2021) or by leveraging auxiliary tasks in self-supervised settings (Von Kügelgen et al., 2021; Yue et al., 2021; Mitrovic et al., 2021; Kong et al., 2023). Despite their success, they fundamentally rely on multi-domain training data, which is often impractical to acquire or augment, particularly in natural language tasks (Yuan et al., 2023). To address single-domain scenarios, causal dependencies have been exploited assuming the absence of unobserved confounders (Lu et al., 2022; Lv et al., 2022). However, this assumption is limiting in real-world settings. Recent work instead leverages the front-door adjustment, which introduces mediator variables to account for unobserved confounders and mitigate spurious correlations (Li et al., 2021; Mao et al., 2022; Nguyen et al., 2023). Building on these ideas, we propose a novel adjustment framework tailored for NLU tasks. By leveraging PLMs, our approach enables causal inference in single-domain settings, widening its applicability in DG.

Domain Generalization for Pre-trained Models. Pre-trained models have achieved remarkable success in computer vision (Chen et al., 2020; Bao et al., 2021; He et al., 2022) and natural language processing (NLP) (Devlin, 2018; Lan, 2019; Liu, 2019), driving growing interest in improving their domain generalization capabilities on downstream tasks. A prominent line of work enhance generalizability by increasing feature diversity through training on data from multiple domains (Hendrycks et al., 2019; Xie et al., 2020; Zhang et al., 2020; Tu et al., 2020). Other methods leverage adversarial training (Salman et al., 2020; Hendrycks et al., 2020; Utrera et al., 2020; Yi et al., 2021) and advanced attention mechanisms (Dosovitskiy, 2020; Mao et al., 2021; Yang et al., 2021) to develop more robust models. Recent work has also explored using PLM parameters as a form of regularization or as an external knowledge source to improve generalization (Wortsman et al., 2022; Zhu et al., 2023; Wang et al., 2024). Building on these ideas, we investigate the potential of leveraging PLMs as an additional data domain for augmentation. This augmented data is then used to construct robust causal representations that enhance model performance in both in-domain (ID) and OOD scenarios.

B. Simulator

We designed two types of simulators: (1) a semi-synthetic simulator; and (2) a real-world simulator.

B.1. General Setting

The simulators serve as fully (or partially) controllable oracles to allow us to test 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 simulators, a different seed indicates that it is a different simulator environment.

B.2. Semi-Synthetic Simulator

The first simulator is semi-synthetic and 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.

B.3. Real-World based Semi-synthetic Simulator

Real-world Case-Study. 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. The second simulator uses real-world data and is inspired by the design of the semi-synthetic simulator and case study in Section J. 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 a mixture of polarized sentiment analysis tasks from two platform: 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.

C. Model Details

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

C.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.

C.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.

C.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 the $\frac{1}{4}$ 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.

C.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.

C.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.

C.6. CFT-Φ

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

D. Discussion: The Value of Semi-Synthetic Cases

A key distinction in our experiments is that while both semi-synthetic and real-world examples are derived from the same base datasets, none of the experiments use 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 rigorous evaluation of causal effects. 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 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

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

ability of CFT to mitigate spurious correlations and generalize to diverse deployment scenarios.

E. Further results

We conducted a further analysis on (1) level of spuriousness (Fig. 6), (2) number of training data (Fig. 7), and (3) number of samples during inference (Fig. 8).

Summary. (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 the blackbox 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 do(x) during prediction time.

In this section, we first present results of the Yelp semi-synthetic example. We observed a similar trend as Fig. 4

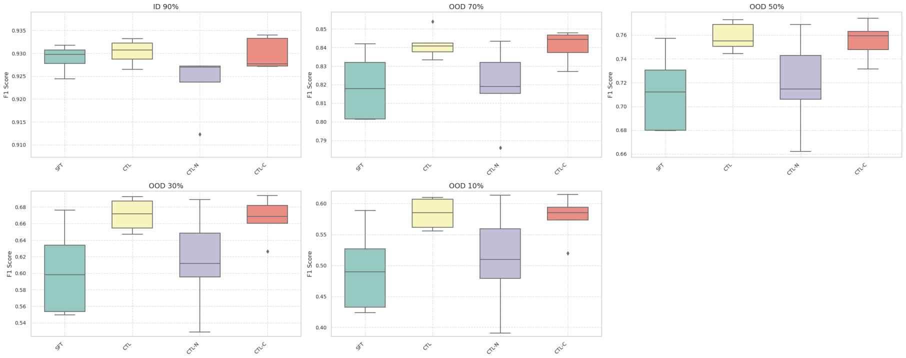


Figure 5. Box-plot over 5 runs for 4 methods (SFT, CFT, CFT-N and CFT-C). Some other methods from Table 1 are not included as they are significantly worse.

Next, we present an analysis of the impact of the level of spurious information, based on the Amazon semi-synthetic example. We tried to inject different levels of spurious features: “-1” is the same as the experiment in Section 5.1; “-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 of spurious information levels.

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. Specifically, the difference becomes insignificant when approaching 7,000 data points per class using the BERT model in our experimental setup described in Section 5.1. 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.

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. 8.

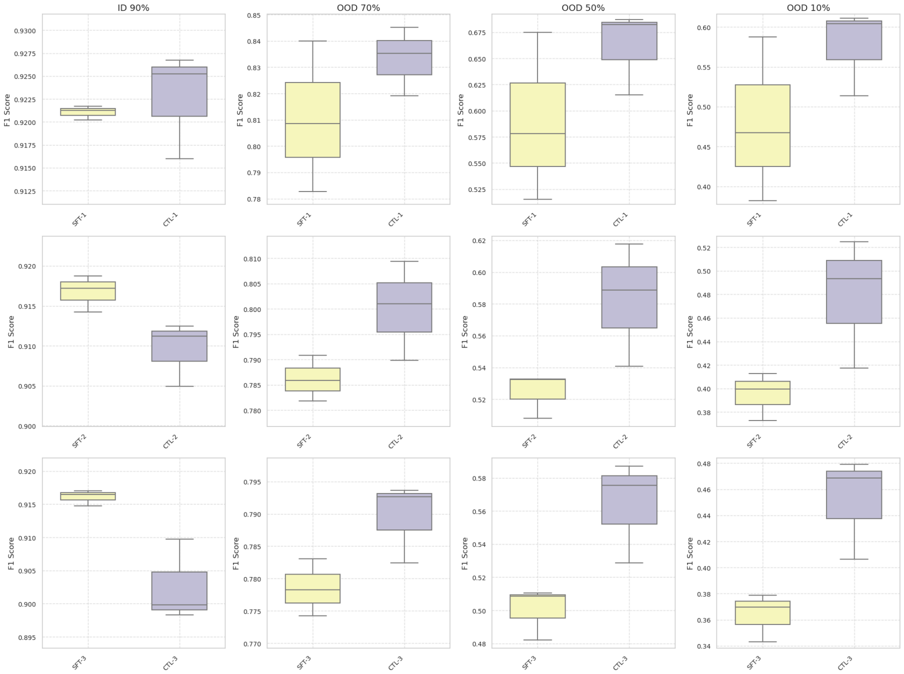


Figure 6. Different spurious level based on the semi-synthetic Amazon data, from "-1" (similarly to the setting in Section 5.1) to "-2" and "-3" with strong spurious features, the CFT consistently outperforms SFT in the OOD settings.

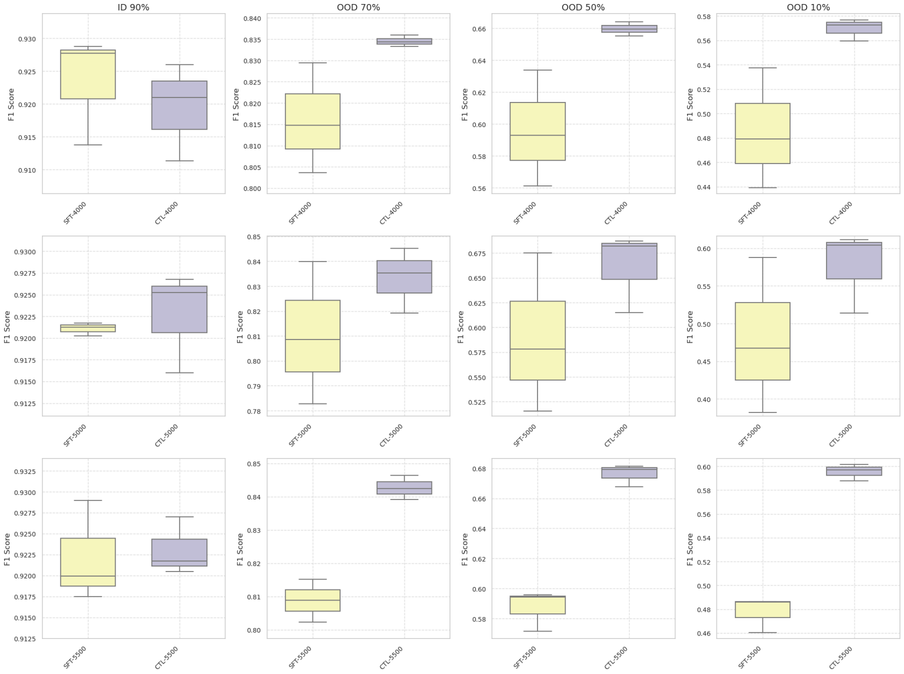


Figure 7. Different training data sizes of 4000, 5000 and 5500 per class of the binary sentiment analysis tasks. The CFT method consistently outperforms SFT in OOD settings.

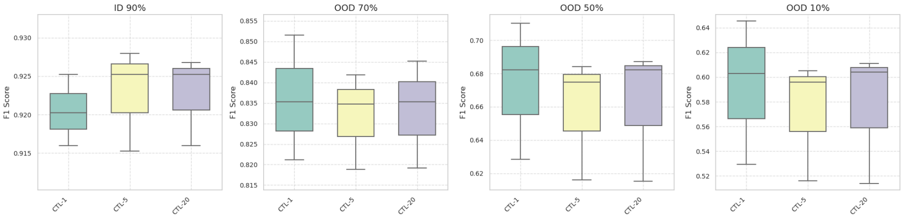


Figure 8. Different inference samples of 1, 5 and 20 for CFT. The variance is reduced in the OOD scenario when using more than 1 sample.

F. Proof of Theorem I.6

$$
\begin{aligned}
p(y \mid \mathrm{do}(x)) &= \underbrace{p(y \mid \mathrm{do}(x), \mathrm{do}(r_0), \mathrm{do}(r_1), \mathrm{do}(\Phi))}_{\text{Assumption I.1}} \\
&= \underbrace{p(y \mid \mathrm{do}(r_0), \mathrm{do}(r_1), \mathrm{do}(\Phi), \mathrm{do}(c))}_{\text{Implied by } c = p(c|r_1) \text{ and } r_1 = p(r_1|x)} \\
&= \underbrace{p(y \mid \mathrm{do}(c))}_{\text{Implied by structural assumptions}} \\
&= \underbrace{\sum_{\Phi'} p(y \mid \Phi', c) p(\Phi')}_{\text{Backdoor criterion (Pearl, 2009)}} \\
&= \sum_{\Phi', x'} p(y \mid \Phi', c) p(\Phi' \mid x') p(x'). \square
\end{aligned}
$$

G. CFT Algorithms

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 $(c, y_i, \Phi')$
end for
```

H. Further Preliminaries

We build a predictor optimized for a distribution invariant to σ. We will achieve this by: **(i)** *considering the hypothetical model where $\sigma = do(x)$ (Pearl, 2009), operationalized as fixing $X = x$ regardless of the value of U;* **(ii)** *showing that, under our assumptions, we can learn a predictor based on $p(y \mid x; \sigma = do(x))$ using data from $\sigma = train$;* **(iii)** *using domain-dependent knowledge, proceed with the adoption of $p(y \mid x; \sigma = do(x))$ instead of $p(y \mid x; \sigma = train)$ for the*

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 $P(y|\text{do}(x))$ using Equation 4
  and then assign $y = \arg\max_y P(y|\text{do}(x))$
end for
```

target environment $p(y \mid x; \sigma = test)$ *if we judge that performance under the predictor optimized for* $p(y \mid x; \sigma = do(x))$ *will be the better one.*

Step (ii) is challenging under the structure of Figure 2(a). Unfortunately, it is a well-known result that, assuming no more than the Markovian factorization implied by Figure 2(a), this is not the case: 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.

Step (iii) is often poorly discussed in the literature beyond naïve worst-case analysis, with notable exceptions such as (Salaudeen et al., 2024). To understand it more precisely, prediction based on the proposed causal structure amounts to averaging $p(y \mid x, U)$ over $U \sim p(u \mid x; \sigma)$. Under $\sigma = do(x)$, this means averaging over $p(u)$ (Pearl, 2009). Under the target regime $\sigma = test$, we need to consider the implications of using the predictor optimized for $\text{do}(x)$. A sufficient condition for a predictor that minimizes expected loss with respect to $p(u)$ to do better than one where the expectation is over $p(u \mid x; \sigma = train)$ is when $p(u \mid x; \sigma = test)$ is "closer" to $p(u)$ than $p(u \mid x; \sigma = train)$ in some sense. The lack of clarity about this trade-off is one of the main sources of confusion and controversy when making claims about the robustness of "causal features" in prediction problems (e.g, Nastl & Hardt, 2024). This pushes for further refinements of the high-level causal structure of Figure 2(a).

Important notation remark. In what follows, regime variables σ will not affect X as a whole, but only subcomponents of it. Following a notation more closely related to Pearl's original notation, we will sometimes use $p(y \mid \text{do}(x))$ as a shorthand notation for $p(y \mid x; \sigma = \text{do}(x))$ even if $p(y \mid x; \sigma = test)$ in general will not change the distribution of all components of X with respect to $p(y \mid x; \sigma = train)$.

I. Structural Assumptions

Assumption I.1 (**Functional Decomposition**). We assume access to a triplet of measurements $(R_0, R_1, \Phi) = f(X)$ for some function f defined nonconstructively as follows: (i) (R_0, R_1) is a **paired representation** of X. The mapping to R_0 is learned from a *pre-training environment* – in our context, this is taken from the PLM. The R_1 mapping is a by-product of supervised fine-tuning, assumed to take place under the training environment $\sigma = train$; (ii) **local features** Φ are token-level features implied by the fine-tuned model (e.g., combinations of token embeddings). This comes in addition to sentence-level representation R_1. □

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 $\text{do}(x)$ translates to $\text{do}((R_0, R_1, \Phi) = f(x))$, 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) as indirect measurements of "causal" latent variables C and two sets of "spurious" latent variables S_0 and S_1, in the sense that only C is a causal parent for output Y. We frame pre-training as out of the scope of any training/test distribution shift, and define S_0 as the latent spurious features of pre-training. 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 (b), and detailed as follows.

Assumption I.2 (**Causal Latent Structure**). Sentence-level features $\{R_0, R_1\}$ are indirect measurement of mutually independent variables $\{S_0, S_1, C\}$. S_0 can only cause R_0 and S_1 can only cause R_1. 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.*

This is also a critical assumption for identifying causal variables C by using paired representations R_0 and R_1 from two representations of X, as motivated by the Theorem 4.4 in (Von Kügelgen et al., 2021). In our context, R_0 can be learned from the PLM (pre-training environment) and R_1 from the supervised fine-tuning (training environment). This paired representation framework enables identification of C from the observational distribution of $\{R_0, R_1, \Phi, Y\}$, which would otherwise remain unidentifiable (Von Kügelgen et al., 2021).

To introduce our choice of a predictor other than an ERM method on training data, we adopt the following assumption.

Assumption I.3 (Causal Structure of Distribution Shifts). Regime variable σ affects the system only via S. This also implies that 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 output Y is invariant. It is, however, not the case that we will be able 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 e.g. via the collider paths $Y \leftarrow U_\Phi \rightarrow \Phi \leftarrow U_S \rightarrow R_1 \leftarrow S_1 \leftarrow \sigma$ 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 the missing edge $\Phi \rightarrow Y$ and the ability of deterministically inferring C. Those two points are formalized by the following assumption and theorem.

Assumption I.4 (Sufficient Mediator). The causal effect of Φ on Y is fully mediated through C. In other words, fixing C makes fixing Φ conditionally independent of Y, that is $p(y \mid \mathrm{do}(\Phi), \mathrm{do}(c)) = p(y \mid \mathrm{do}(c))$. □

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 I.5 (Identification for Causal Features C). *Assume the structural assumptions encoded in the causal graph in **Fig. 2 (b)**. 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 identify C by learning the distribution $p(c \mid r)$ from R_0 and R_1.*

Intuition. This theorem implies that if the causal latent variable C remains invariant across environments (Assumption I.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 F.

Theorem I.6 (Identification for Causal Transfer Learning). *Given the assumptions in the causal graph in **Fig. 2 (b)** and Theorem I.5, the distribution of Y under $do(x)$ can be computed as*[4]

$$p(y \mid do(x)) = \sum_{\Phi', x'} p(y \mid \Phi', c)p(\Phi' \mid x')p(x'), \tag{4}$$

where c is given by $c = p(c|r_1)$ and $r_1 = p(r_1|x)$. □

[4] 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'.

Invariance implication and pragmatic application. The difference between $p(y \mid x; \sigma = test)$ and $p(y \mid x; \sigma = \text{do}(x))$ in our setup boils down to averaging $p(y \mid c, U_\Phi)$ over $p(u_\Phi \mid r_0, r_1, \Phi, c; \sigma = test)$ in the former, and $p(u_\Phi)$ in the latter. When can we say that the latter is an improvement over $p(u_\Phi \mid r_0, r_1, \Phi, c; \sigma = train)$? Our claim is that by virtue of the confounder being a cause of local features Φ only, and not of the whole of X, the relevance of information passing through (S_0, S_1) should be limited anyway, unless the test environment affects it drastically. In this case, we may be thrown away too far from the original $p(u_\Phi \mid r_0, r_1, \Phi, c; \sigma = train)$ in unpredictable ways, and the safer bet ("worst-case") 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.

J. Further Experiments

Data. We conduct experiments based on the real-world case study described above (and illustrated earlier in Fig. 1). As in the first semi-synthetic experiment, we focus on sentiment analysis using a dataset built from Yelp and Amazon review. During the training, similar to the semi-synthetic experiments, we build correlations between the source of the data (whether coming from Amazon or Yelp platform) and the label, by adding strings such as ”amazon.xxx” or ”yelp.yyy” into the sentences. More details can be found in Appendix B.3. We used 5000 samples per class for training and 2,000 samples per class for testing. For training, we set the spurious correlation to be at a ratio of 90%, which remains the same for ID test; and for the OOD test set, we adjust this ratio to be 70%, 50%, 30%, and 10%. Additionally, we compare our approach with other single-domain generalization baselines to demonstrate its effectiveness.

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

Table 2. Main results for real-world experiments. Results reported in mean value based on 5 runs of different seeds.

	Train F1 90%	**ID F1 90%**	**OOD F1 70%**	**OOD F1 50%**	**OOD F1 30%**	**OOD F1 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	90.59	80.32	**70.08**	**59.68**	**49.22**
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

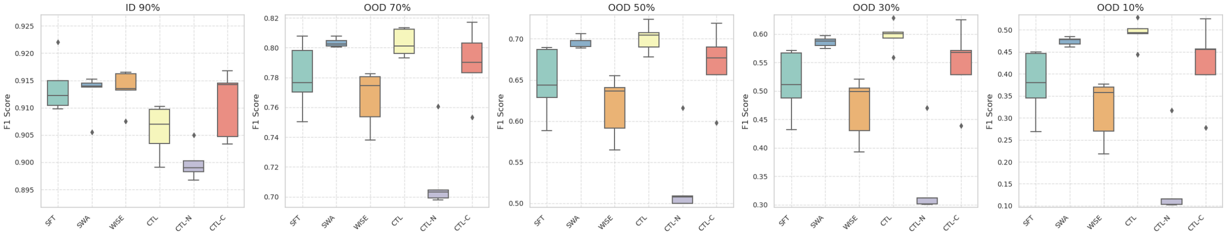


Figure 9. Box-plot over 5 runs for 6 methods (SFT, SWA, WISE, CFT, CFT-N and CFT-C). Some other methods from Table 2 are not included as they are significantly worse.

K. Discussion: Detecting vs. Correcting Distribution Shifts

Our proposed causal adjustment strategy focuses on mitigating spurious correlations by disentangle stable causal feature with respect to spurious, non-causal features. In contrast, an alternative and complementary perspective focuses on detecting distribution shift, particularly by using deep latent variable models. This line of work, developed extensively in the context of NLP applications by (Yu, 2023), uses variational inference to model predictive uncertainty and capture latent structure in text data (Yu et al., 2021; 2022; 2023). For instance, classifiers built upon Bayesian neural networks or deep generative models (e.g., VAEs) can produce uncertainty-aware predictions by marginalising over latent variables. When exposed to test-time shifts, these models are often more cautious, yielding (relatively) calibrated confidence scores or identifying high-uncertainty regions where distribution shifts may occur. Although in recent studies, we have noted that further calibration techniques such as conformal inference are often required for better coverage (Yu et al., 2024). Unlike causal fine-tuning, these methods do not explicitly model the causal graph or perform confounder adjustment, but they provide a practical mechanism for detecting distribution shift and avoiding overconfident predictions.