Improving Multi-Task Generalization via Regularizing Spurious Correlation

Ziniu Hu[1]*, **Zhe Zhao**[2], **Xinyang Yi**[2], **Tiansheng Yao**[2], **Lichan Hong**[2], **Yizhou Sun**[1], **Ed H. Chi**[2]
[1]University of California, Los Angeles, {bull, yzsun}@cs.ucla.edu
[2]Google Research, Brain Team, {zhezhao,xinyang,tyao,lichan,edchi}@google.com

Abstract

Multi-Task Learning (MTL) is a powerful learning paradigm to improve generalization performance via knowledge sharing. However, existing studies find that MTL could sometimes hurt generalization, especially when two tasks are less correlated. One possible reason that hurts generalization is spurious correlation, i.e., some knowledge is spurious and not causally related to task labels, but the model could mistakenly utilize them and thus fail when such correlation changes. In MTL setup, there exist several unique challenges of spurious correlation. First, the risk of having non-causal knowledge is higher, as the shared MTL model needs to encode all knowledge from different tasks, and causal knowledge for one task could be potentially spurious to the other. Second, the confounder between task labels brings in a different type of spurious correlation to MTL. Given such label-label confounders, we theoretically and empirically show that MTL is prone to taking non-causal knowledge from other tasks. To solve this problem, we propose Multi-Task Causal Representation Learning (MT-CRL) framework. MT-CRL aims to represent multi-task knowledge via disentangled neural modules, and learn robust task-to-module routing graph weights via MTL-specific invariant regularization. Experiments show that MT-CRL could enhance MTL model's performance by 5.5% on average over Multi-MNIST, MovieLens, Taskonomy, CityScape, and NYUv2, and show it could indeed alleviate spurious correlation problem.

1 Introduction

Multi-Task Learning (MTL), a learning paradigm (Caruana, 1997; Zhang & Yang, 2018) aiming to train a single model for multiple tasks, is expected to benefit the overall generalization performance than single-task learning (Maurer et al., 2016; Tripuraneni et al., 2020) given the assumption that there exists some common knowledge to handle different tasks. However, recent studies observed that, when two tasks are less correlated, MTL could lead to even worse overall performance (Parisotto et al., 2016; Zhang et al., 2021). A line of works (Yu et al., 2020; Wang et al., 2021; Fifty et al., 2021) resort performance drop to optimization challenge because conflicting tasks might compete for model capacity. However, both Standley et al. (2020) and our analysis in Section 2.2 show that, even with an over-parameterized model that achieves low MTL training loss, the final generalization performance could be worse than single-task learning. This finding motivates us to think about the following question: Are there any intrinsic problems in MTL that hurt generalization?

One widely studied issue that influences generalization is the spurious correlation problem (Geirhos et al., 2019, 2020), i.e., correlation that only existed in training datasets due to unobserved confounders (Lopez-Paz, 2016), but not causally correct. For example, as Beery et al. (2018) discussed, when we train an image classification model to identify cows with a biased dataset where cows mostly

*This work was done when Ziniu was an intern at Google.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

appear in pastures, the trained cow classification model could exploit the features of background (e.g., pastures) to make prediction. Thus, when we apply the classifier to another dataset where cows also appear in other locations such as farms or rivers, it will fail to generalize (Nagarajan et al., 2021).

When it comes to MTL setting, there exist several unique challenges to handle spurious correlation problem. **First, the risk of having non-causal features is higher**. Suppose each task has different sets of causal features. To train a single model for all these tasks, the shared representation should encode all required features. Consequently, the causal features for one task could be potentially spurious to the other tasks, and such risk could be even higher with an increasing number of tasks. **Second, the confounder that leads to spurious correlation is different**. Instead of the standard confounders between feature and label, the nature of MTL brings in a unique type of confounders between task labels, e.g., correlation between tasks' labels could change in different distributions. For example, when we train a MTL model to solve both cow classification and scene recognition tasks, its encoder needs to capture both foreground and background information, and the spurious correlation between the two tasks in training set could mislead per-task model to utilize irrelevant information, e.g., use background to predict cow. Given such label-label confounders that are unique for MTL, we theoretically prove that MTL is prone to taking non-causal knowledge learned from other tasks. We then conduct empirical analysis to validate the hypothesis. In summary, we point out the unique challenges of spurious correlation in MTL setup, and show that it indeed influences multi-task generalization.

In light of the analysis, we try to solve the spurious correlation problem in MTL. Among all the knowledge learned in the shared representation layer through end-to-end training, an ideal MTL framework should learn to leverage only the causal knowledge to solve each task by identifying the correct causal structure. Following the recent advances that enable causal learning in an end-to-end learning model (Schölkopf et al., 2021; Mitrovic et al., 2021), we propose a Multi-Task Causal Representation Learning (MT-CRL) framework, aiming to represent the multi-task knowledge via a set of disentangled neural modules instead of a single encoder, and learn the task-to-module causal relationship jointly. We adopt de-correlation and sparsity regularization over popular Mixture-of-Expert (MoE) architecture (Shazeer et al., 2017). The most critical and challenging step is to learn the causal graph in the MTL setup, which requires distinguishing the genuine causal correlation from spurious ones for all tasks. Motivated by the recent studies that invariance could lead to causality (Ahuja et al., 2020; Koyama & Yamaguchi, 2021), we propose to penalize the variance of gradients w.r.t. causal graph weights across different distributions. On a high level, this invariance regularization encourages the causal graph to assign higher weights to the modules that are consistently useful. In contrast, the modules encoding spurious knowledge that cannot consistently achieve graph optimality are assigned lower weights and be discarded by task predictors.

We evaluate our method on existing MTL benchmarks, including Multi-MNIST, MovieLens, Taskonomy, CityScape, and NYUv2. For each dataset, to mimic distribution shifts, we adopt some attribute information given in the dataset, such as the released time of the movie or district of a building, to split train/valid/test datasets. The results show that MT-CRL could consistently enhance the MTL model's performance by 5.5% on average, and outperform both the MTL optimization and robust machine learning baselines. We also conduct case studies to show that MT-CRL indeed alleviate spurious correlation problem in MTL setup.

The key contributions of this paper are as follows:

- We are the first to analyze spurious correlation problem in MTL setup, and point out several key challenges unique to MTL with theoretical and empirical analysis.
- We propose MT-CRL with MTL-specific invariant regularizers to elleviate spurious correlation problem, and enhances the performance on several MTL benchmarks.

2 Analyzing Spurious Correlation in MTL

To systematically analyze the spurious correlation problem in MTL, we first assume that data and task labels are generated by ground-truth causal mechanisms described in Suter et al. (2019). We denote X as the variable of observed data, and each data is associated with K latent generative factors $\mathbb{F} = \{F_i\}_{i=1}^K$ representing different semantics of the data (e.g., color, shape, background of an image). We follow Schölkopf et al. (2021) to assume that the data X is generated by disentangled causal mechanisms $P(X|F_i)$, such that $P\big(X|\mathbb{F}\big) = \prod_{i=1}^K P\big(X|F_i\big)$.

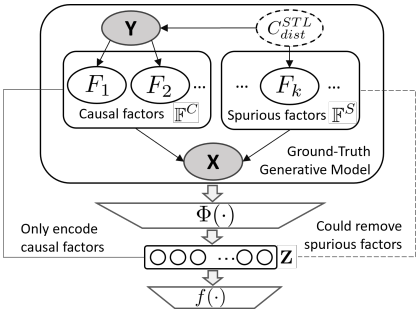

Figure 1: Spurious correlation in **Single-Task Learning** is mainly caused by factor-label confounders C_{dist}^{STL}. We could remove spurious factors $\mathbb{F}^S$ from representation Z.

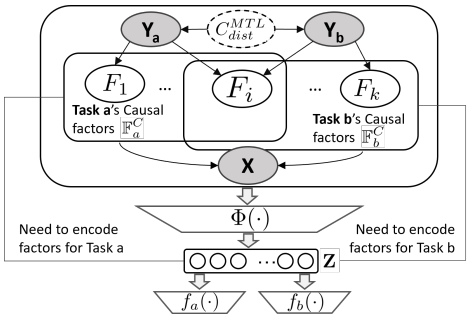

Figure 2: Spurious correlation in **Multi-Task Learning** could be caused by label-label confounders C_{dist}^{MTL}. Factors for both tasks $\mathbb{F}_a^C$ and $\mathbb{F}_b^C$ need to be encoded and potentially spurious.

As $\mathbb{F}$ represents high-level knowledge of the data, we could naturally define task label variable Y_t for task t as the cause of a subset of generative factors. We denote $\mathbb{F}_t^C$ as a subset of causal feature variables within $\mathbb{F}$ that are causally related to each task variable Y_t, and we could define $\mathbb{F}_t^S = \mathbb{F} \setminus \mathbb{F}_t^C$ as a subset of non-causal feature variables to task t, such that $P(Y_t|\mathbb{F}) = P(Y_t|\mathbb{F}_t^C)$. In other words, changing the values of any non-causal factors in $\mathbb{F}_t^S$ does not change the conditional distribution.

Note that the discussion so far is based on the assumption that the ground-truth causal generative model is known. In a real-world learning setting, however, we are only given a supervised dataset (X, Y) without access to generative factors $\mathbb{F}$. To solve the task, a neural encoder $\Phi(\cdot)$ is required to extract representation $\mathbf{Z}$ from the data that encodes the information about the causal factors, on top of which a task predictor $f(\cdot)$ could predict the label.

2.1 Spurious Correlation Problem

Based on the ground-truth generative model, an ideal predictor for each task should only utilize the causal factors, and keep invariant to any intervention on non-causal factors. However, in real-world problems, it is hard to achieve an invariant predictor due to the spurious correlation issue due to unobserved confounders C_{dist} (Lopez-Paz, 2016). Formally, confounders are variables that influence the two connected variables' correlation, and such correlation could change under different distribution (different value of C_{dist}), thus the model exploiting such spurious correlation will fail to generalize. Below we summarize the differences of spurious correlation problems for single-task and multi-task learning settings:

Single-Task Learning (STL). As illustrated in Figure 1, the label-factor confounders for single task learning C_{dist}^{STL} connects non-causal factors $F \in \mathbb{F}^S$ and task label Y, bringing in spurious correlation. For example, temperature could confound crime and ice cream consumption. When the weather is hot, both crime rates and ice cream sales increase, but these two phenomena are not causally related. Based on the proof by Nagarajan et al. (2021); Khani & Liang (2021), such spurious correlation could lead the model to use non-causal factors, and thus hurt generalization performance.

Multi-Task Learning (MTL). In the MTL setting, there exist several unique challenges to handle spurious correlation. First, the risk of having non-causal features is higher. As is illustrated in Figure 2, the shared encoder Φ needs to encode all the factors causally related to each task in the representation Z. Therefore, for each task, all non-overlapping factors from other tasks could be potentially spurious. Second, besides the standard label-factor confounders C_{dist}^{STL} for each single task introduced above, we define label-label confounders C_{dist}^{MTL} connecting multiple tasks' label $\{Y\}$. Such confounder is unique to MTL setting.

As an example, consider two binary classification tasks, with Y_a and Y_b as variables from $\{\pm 1\}$ for task label. The two labels' correlation $P(Y_a = Y_b) = m_C$ could change with different confounder $C_{dist}^{MTL} = C$. We assume the two tasks have non-overlapping factors F_a and F_b drawn from Gaussian distribution. We then show MTL model with both two factors as input will utilize non-causal factors:

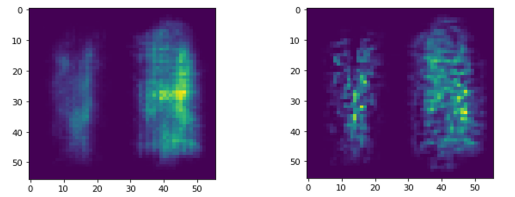

Figure 3: The gradient saliency map of right-side digit classifier. The model trained by MTL exploits left pixels (spurious) more.

	Multi-SEM		Multi-MNIST	
	STL	MTL	STL	MTL
Acc_{train}	0.931	0.936	0.981	0.987
Acc_{val}	0.906	0.882	0.874	0.846
ρ_{spur}	0.128	0.246	0.261	0.328

Table 1: Empirical results of multi-task (MTL) and single-task learning (STL) model on synthetic datasets with changing C_{dist}^{MTL}.

Proposition 1 *Given $m_C \neq 0.5$, the Bayes Optimal per-task classifier has non-zero weights to non-causal factor. Given $m_C = 0.5$ and limited training dataset, the trained per-task classifier will assign non-zero weights to non-causal factor as noise.*

Detailed proof is in Appendix A. Therefore, in this linear classification example, when we deploy the trained model to a new distribution with changed label-label confounder C_{dist}^{MTL}, the model trained by MTL that utilizes non-causal factors generalize relatively worse. On the contrary, the model trained by STL don't need to encode all causal factors from two tasks. Assuming there is no task-label confounder C_{dist}^{STL} in each task's dataset, the trained model could remove non-causal factors from representation.

2.2 Empirical Experiments

In the following, we conduct experiments to validate the claims. As there is no existing MTL datasets specifically designed to analyze spurious correlation problem, we construct synthetic Multi-SEM (Rosenfeld et al., 2021) and Multi-MNIST (Harper & Konstan, 2016) datasets with known causal structure to study whether the model trained by MTL indeed exploits more non-causal factors, and how the spurious correlation influences multi-task generalization. Dataset details are in Appendix D.1.

Spurious Score. As we know the ground-truth causal structure for the two datasets, we could quantify how much a model utilizes the non-causal factors. Following the gradient saliency map proposed by Simonyan et al. (2014), we calculate the average absolute gradients w.r.t each factor as $Grad(F) = \sum_{(x(\mathbb{F}),y)\in D} \left| \frac{\partial\left(f(\Phi(x))[y]\right)}{\partial F} \right|$, which measures how much a model leverage this factor to make prediction. We then define the spurious score ρ_{spur} as the proportion of average gradients over non-causal feature $\rho_{spur} = \frac{\sum_{F\in\mathbb{F}^S} Grad(F)}{\sum_{F\in\mathbb{F}} Grad(F)}$.

Empirical Results. We train a shared-bottom model via Multi-task learning (MTL) and single-task learning (STL) over the two datasets and report both the training and test accuracy with spurious ratio ρ_{spur} in Table 1. As illustrated, the test accuracies of MTL for both Multi-SEM and Multi-MNIST datasets are both worse than STL. The training accuracies of MTL are very similar to STL, meaning that the performance drop is not due to the optimization difficulty that many previous works try to address. The spurious ratio ρ_{spur} of MTL is much higher than the STL, which means that it exploits more non-causal factors. To give a more straightforward illustration, we plot the gradient saliency map of the right-side digit classifier for Multi-MNIST in Figure 3. The model trained by MTL utilizes more left-side pixels, which are non-causal to the final prediction. We also show the results of Multi-SEM with more than 2 tasks in Appendix B. These findings support our hypothesis that with spurious correlation caused by label-label confounder C_{dist}^{MTL}, models trained by MTL is more prone to leverage non-causal knowledge than STL, and thus influence generalization performance.

3 Method

Based on the previous analysis of the spurious correlation problem in MTL, we now introduce a Multi-Task Causal Representation Learning (MT-CRL) framework with the goal that the per-task predictor only leverages the causal knowledge instead of spurious correlation. The high-level idea

of the framework is to reconstruct the ground-truth causal mechanisms introduced in section 2 through end-to-end representation learning. To accomplish this goal, the framework aims to 1) model multi-task knowledge via a set of disentangled neural modules; 2) learn the task-to-module causal graph that is optimal across different distributions. With the correct causal graph as routing layer, per-task predictor only utilizes outputs from causally-related modules, thus alleviating the spurious correlation problem. We introduce the two crucial designs as follows.

3.1 Modelling via Disentangled Neural Modules

In order to alleviate spurious correlation, an ideal MTL model should learn the multi-task knowledge in the shared representation while identifying which part of the knowledge is causally related to each task. However, directly conducting causal discovery is impossible if all the knowledge is fused in a single shared encoder. Thus, we seek to adopt a modularized architecture in which each module encodes disentangled knowledge, and thus enable modeling causal relationship between task and modules. We adopt Multi-gate Mixture-of-Experts (MMoE) (Ma et al., 2018), a variant of MoE (Shazeer et al., 2017) architecture tailored for MTL setting, as our underlying model. Specifically, we have K different neural modules as shared encoders $\Phi = \big[\Phi_i(\cdot)\big]_{i=1}^{K}$. Given a batch of input data $\mathbf{X} = \{x_n\}_{n=1}^{B}$ with batch size B, we extract k representations via different neural modules, i.e., $\mathbf{Z}_i = \Phi_i(X) \in \mathbb{R}^{B\times d}$. Based on sparsity assumption of the causal mechanisms Parascandolo et al. (2018); Bengio et al. (2020); Lachapelle et al. (2021), only a few modules should be causally related to each task. Therefore, on top of the learned neural modules, we learn a task-to-module routing graph, aiming to estimate which module is causally related to each task. We model the bipartite adjacency (a.k.a. bi-adjacency) matrix $A = \text{sigmoid}(\theta) \in [0,1]^{T\times K}$ by applying sigmoid over a learnable parameter θ to enforce the range constraint. Note that original MMoE adopts softmax to get gate vector, which encourages only a small portion of modules being utilized for each task. Our graph modelling allows multiple modules utilized for each task. With the correct graph weights A as routing layer, we could utilize only the causally related modules and make predictions with per-task predictor $f_t(\cdot)$ as $\hat{Y}_t(\mathbf{X}) = f_t\big(\sum_i A_{t,i}\cdot\Phi_i(\mathbf{X})\big)$.

Disentangling Modules. One of the main properties of the causal mechanisms we introduced in section 2 is disentanglement, such that each factor represents a different view of the data, and changing the value of one factor does not influence the others. If without explicit constraints, the learned modules' outputs could still be correlated and hinder the causal structure learning. Therefore, we need to add regularization to disentangle these modules during training.

Most existing disentangled representation learning methods are under the generative modeling framework, e.g. VAE (Higgins et al., 2017) or GAN (Chen et al., 2016). However, Locatello et al. (2019) argues that without explicit supervision, it is hard for generative models to learn correct disentangled factors. We therefore only borrow the regularization methods utilized in existing generative disentangled representation works (Cheung et al., 2015; Cogswell et al., 2016) to directly penalize the correlation of learned modules. Specifically, we regularize the in-batch Pearson correlation $\rho(\mathbf{Z}_i, \mathbf{Z}_j)$ between every pair of output dimensions from different representation matrices $\mathbf{Z}_i$ and $\mathbf{Z}_j$, as:

$$\rho(\mathbf{Z}_i,\mathbf{Z}_j) = \frac{Cov(\mathbf{Z}_i,\mathbf{Z}_j)}{\sqrt{Cov(\mathbf{Z}_i,\mathbf{Z}_i)}\sqrt{Cov(\mathbf{Z}_j,\mathbf{Z}_j)}}, \text{ where } Cov(\mathbf{Z}_i,\mathbf{Z}_j) = \big[\mathbf{Z}_i - \overline{\mathbf{Z}_i}\,\big]^T\big[\mathbf{Z}_j - \overline{\mathbf{Z}_j}\,\big]. \quad (1)$$

By minimizing the Frobenius norm of the correlation matrix ρ for every two different representation pairs, we could enforce the encoder Φ to extract disentangled representations.

$$\mathcal{L}_{decor}(\Phi) = \lambda_{decor}\cdot\sum_{i=1}^{k}\sum_{j=i+1}^{k}\Big\|\rho\big(\Phi_i(X),\Phi_j(X)\big)\Big\|_F^2. \quad (2)$$

Task-to-Module Graph Regularization. Based on sparsity assumption of the causal mechanisms (Parascandolo et al., 2018; Bengio et al., 2020; Lachapelle et al., 2021), each task is causally related to only a few modules. To learn the graph structure, existing works (Zheng et al., 2018; Ng et al., 2019; Lachapelle et al., 2020) propose to to fit structural equation model (SEM) with sparsity regularization over the graph weights. We adopt a similar sparse regularization with an entropy balancing term (Hainmueller, 2012) over the bi-adjacency matrix A weights of the task-to-module

routing graph:

$$\mathcal{L}_{graph}(A) = \lambda_{sps} \cdot ||A||_1 - \lambda_{bal} \cdot \text{Entropy}\Big(\frac{\sum_t A_{t,*}}{\sum_{t,i} A_{t,i}}\Big). \tag{3}$$

Note that the entropy term aims at keeping the causal weights for each module i summing over all the tasks to be balanced. This could help avoid degenerate solutions in which only a few modules are utilized. By combining the two regularizations in Eq.(2) and Eq.(3) with per-task supervised risk term $R_t\big(\Phi, A_t, f_t\big) = \sum_{(\mathbf{X},\mathbf{Y}_t)\in\mathcal{D}} L_t\big(\hat{Y}_t(\mathbf{X}), \mathbf{Y}_t\big)$, we get the regularized loss as:

$$\tilde{\mathcal{L}}(\Phi, A, f) = \sum_{t\in\mathcal{T}} R_t\big(\Phi, A_t, f_t\big) + \mathcal{L}_{decor}(\Phi) + \mathcal{L}_{graph}(A). \tag{4}$$

3.2 Causal Learning via Graph-Invariant Regularization

It is critical and challenging to learn the correct causal graph, which requires distinguishing the true causal correlation from spurious ones. Motivated by the recent studies of robust machine learning that a predictor invariant to multiple distributions could learn causal correlation (Ahuja et al., 2020; Koyama & Yamaguchi, 2021), we assume the true causal relationship to be optimal across different distributions. To do so, we assume to have access to multiple slices of datasets collected from different environments $e \in \mathcal{E}$ in which the confounder C^{MTL}_{dist} that controls task correlation might change. For example, one natural choice is to consider train/valid dataset split (the setting we utilize in experiment), or assume the training set is split into multiple slices with different attributes. We desire the task-to-module graph weights A and per-task predictor f_t to be optimal across all environments $e \in \mathcal{E}$. Formally, we aim to solve the following bi-level optimization problem:

$$\min_{\Phi,A,f} \ \tilde{\mathcal{L}}(\Phi, A, f) \quad \text{s.t.} \ \ A_t, f_t \in \arg\min_{A,f} R^e_t\big(\Phi, A, f\big), \forall\, t \in \mathcal{T}, e \in \mathcal{E}. \tag{5}$$

where R^e_t denotes the risk over data slice in environment e. This optimization problem could be regarded as a multi-task version of IRM. Based on Theorem 9 described in Ahuja et al. (2020), by enforcing invariance over a sufficient number of environments that exhibit distribution shifts (i.e., changes of confounder C^{MTL}_{dist}), per-task predictors should only utilize modules that are consistently helpful to the task, and assign zero weights to modules that encode non-causal factors to the task, and thus alleviate spurious correlation and help out-of-distribution generalization. Even if all data are sampled from the same distribution and there are no distribution shifts, invariance could also help eliminate noisy correlation due to the limited training dataset and help in-distribution generalization.

Invariant Optimality of Task-to-Module Graph for MTL. As discussed in IRM, the bi-leveled optimization problem in Eq.(5) is highly intractable, especially with complex and non-linear Φ. To implement a practical optimization objective, IRM proposes to softly regularize the gradient of the task-predictor at different environments to enforce it to be optimal:

$$\min_{\Phi,A,f} \ \Big(\tilde{\mathcal{L}}(\Phi, A, f) + \sum_{t\in\mathcal{T}}\sum_{e\in\mathcal{E}} \Big\|\nabla_{A=A_t, f=f_t} R^e_t\big(\Phi, A, f\big)\Big\|^2\Big). \tag{6}$$

However, as is discussed in IRM paper, if the complexity of a task-predictor f is much larger than the number of environments, it could learn an over-fitted solution that makes gradient zero but does not achieve invariance. IRM adopts a fixed all-one vector as predictor to reduce complexity. This approach **is not applicable to MTL setup**, as the optimal task-predictors f^*_t for different task t could be very distinctive and complex, and we cannot use a fixed uniform predictor for all tasks.

To strike a balance between invariance and complexity of multi-task predictors, we propose only to regularize the gradient of the task-to-module routing graph while assuming the complex predictor f_t for each task is fixed at each iteration. We call this modification as **Graph-Invariant Risk Minimization (G-IRM)**, which is designed specifically to MTL setup:

$$\min_{\Phi,A,f} \ \Big(\tilde{\mathcal{L}}(\Phi, A, f) + \lambda_{G\text{-}IRM} \cdot \mathcal{L}_{G\text{-}IRM}(\Phi, A|f)\Big). \tag{7}$$

By adopting the similar gradient penalty term as adopted in IRM, we define $\mathcal{L}^{Norm}_{G\text{-}IRM}$ as:

$$\mathcal{L}^{Norm}_{G\text{-}IRM}(\Phi, A|f) = \sum_{t\in\mathcal{T}}\sum_{e\in\mathcal{E}} \Big\|\nabla_{A=A_t} R^e_t\big(\Phi, A, f_t\big)\Big\|^2. \tag{8}$$

As we assume f_t is fixed for invariance regularization term $\mathcal{L}_{G\text{-}IRM}^{Norm}$, we only calculate gradient and optimize for Φ and A, but not updating f_t. This could avoid the over-parametrized predictor f_t finding a trivial solution to achieve zero gradients instead of learning the correct causal correlation. Similar trick is utilized in (Ahmed et al., 2021). Note that the gradient w.r.t each graph weight means whether a module could help reduce the risk for this task. Therefore, by penalizing the invariance regularization, the modules containing non-causal factors will be assigned zero weights.

In the experiments, we observe that at the early optimization stage, the model has non-zero gradients for all parameters, including the graph weights, thus directly regularizing the gradient norm might influence the optimization. Therefore, we propose a modified version of gradient regularization $\mathcal{L}_{G\text{-}IRM}^{Var}$ that penalizes the variance of the task-to-module graph's gradient on different environments:

$$\mathcal{L}_{G\text{-}IRM}^{Var}(\Phi, A|f) = \sum_{t\in\mathcal{T}} \sum_{e\in\mathcal{E}} \frac{1}{|\mathcal{E}|} \left\| \nabla_{A=A_t} R_t^e\big(\Phi, A, f_t\big) - \text{Avg}_e\Big(\nabla_{A=A_t} R_t^e\Big) \right\|^2. \tag{9}$$

By minimizing $\mathcal{L}_{G\text{-}IRM}^{Var}$, we force all the learned modules to have similar gradients across different environments, and not overfit only to some of the environments. It still allows some modules to have non-zero gradients as long as it's the same across environments, and relies on loss term $\tilde{\mathcal{L}}$ to update these weights, while $\mathcal{L}_{G\text{-}IRM}^{Norm}$ forces all gradient to be zero. Therefore, $\mathcal{L}_{G\text{-}IRM}^{Var}$ is a loose regularization that not influences the overall optimization too much. It shares similar intuition of REx (Krueger et al., 2021) that penalizes risk variance, while $\mathcal{L}_{G\text{-}IRM}^{Var}$ penalize gradient variance. We provide pseudo-code of MT-CRL framework in Appendix C.

4 Experiment

In this section, we evaluate whether MT-CRL could benefit the performance of MTL models on existing benchmark datasets, and study whether it could indeed alleviate spurious correlation.

Experimental Setup. One key ingredient of our MT-CRL is to achieve the optimality of causal graph over different distributions. However, we might not access multiple environmental labels in most real-world multi-task learning datasets. Therefore, we adopt a more realistic setup, such that we only assume to have a single validation set that contains unknown distribution shifts (i.e. change of confounder C_{dist}^{MTL}) compared to the training dataset. We thus could utilize training and valid sets as two environments to calculate invariance regularization, while we only utilize the training set to calculate task loss to avoid the task predictor overfits. Note that in this way, our method could get access to the label information in the validation set. To avoid the possibility that the performance improvement is brought by additional label, for all the other baseline methods, we also add the validation data into the training set to calculate task loss and learn MTL model.

Dataset. We choose five widely-used real-world MTL benchmark datasets, i.e., Multi-MNIST (Sun, 2019), MovieLens (Harper & Konstan, 2016), Tasknomy (Zamir et al., 2018), NYUv2 (Silberman et al., 2012) and CityScape (Cordts et al., 2016), and try to determine train/valid/test split such that there exist distribution shifts between these sets. Dataset details are in Appendix D.2. Note that except NYUv2, our data split is the same as the default split settings of these datasets, which also try to test model's capacity to generalize across domains.

Baselines. As MT-CRL is a regularization framework built upon modular MTL architecture (in this paper we choose MMoE as instantiation, but it can be applied to other modular networks), we mainly compare with two gradient-based multi-task optimization baselines: **PCGrad** (Yu et al., 2020) and **GradVac** (Wang et al., 2021). We also compare with two domain generalization baselines: **IRM** (Ahuja et al., 2020) and **DANN** (Ganin et al., 2016). For IRM we adopt different per-task predictors instead of all-one vector to adapt MTL setup, and calculate penalty via Eq. (6).

Hyper-Parameter Selection. For a fair comparison, all methods are based on the same MMoE architecture. Our methods contain a lot of hyper-parameters, including some model specific ones such as number of modules (K) and regularization specific ones. To avoid the case that performance improvement is caused by extensive hyper-parameter tuning, we mainly search optimal model hyper-parameter on Vanilla MTL setting, and use for all baselines. For regularization specific parameters, we take Multi-MNIST, the simplest dataset among the testbed, to find a optimal combination, and use for all other datasets. Detailed selection procedure and results are shown in Appendix H.

Methods	Multi-MNIST	MovieLens	Taskonomy	CityScape	NYUv2	Avg.
Vanilla MTL	(—baseline to calculate relative improvement—)					
Single-Task Learning	+3.3%	+0.2%	-2.5%	-2.4%	-12.2%	-2.7%
MTL + PCGrad	+4.5%	+0.2%	+3.1%	+2.1%	+7.4%	+3.5%
MTL + GradVac	+4.6%	+0.3%	+3.5%	+2.1%	+7.2%	+3.5%
MTL + DANN	+4.1%	+0.4%	+1.2%	+0.3%	-0.4%	+1.1%
MTL + IRM	+5.0%	+0.4%	+1.1%	+0.6%	-0.1%	+1.4%
MT-CRL w/o $\mathcal{L}_{G\text{-}IRM}$	+5.9%	+0.2%	+3.2%	+1.5%	+4.3%	+3.0%
MT-CRL with $\mathcal{L}^{Norm}_{G\text{-}IRM}$	+7.8%	+1.0%	+6.5%	**+2.9%**	+8.0%	+5.2%
MT-CRL with $\mathcal{L}^{Var}_{G\text{-}IRM}$	**+8.1%**	**+1.1%**	**+7.1%**	+2.8%	**+8.2%**	**+5.5%**

Table 2: Relative Performance improvement of different multi-task learning (MTL) strategies compared to vanilla MTL baseline. Detailed results for each task are shown in Table 6-10 in Appendix E.

Disentangled Reg.		**Graph Reg.**		Multi-MNIST
$\mathcal{L}_{decor}$	$\mathcal{L}_{\beta\text{-VAE}}$	$\mathcal{L}_{sps}$	$\mathcal{L}_{bal}$	Accuracy
✓	✗	✓	✓	0.915 ± 0.018
✗	✓	✓	✓	0.896 ± 0.024
✗	✗	✓	✓	0.882 ± 0.020
✓	✗	✗	✗	0.891 ± 0.016
✓	✗	✗	✓	0.903 ± 0.017
✓	✗	✓	✗	0.908 ± 0.021

Table 3: **Ablation Studies** of disentangled and Graph regularization components in MT-CRL, evaluated on Multi-MNIST dataset.

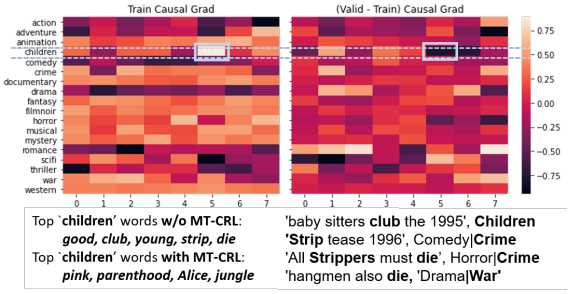

Figure 4: Task-to-Module gradients of model without MT-CRL show Module 5 is spurious. MT-CRL could help alleviate spurious correlation.

4.1 Experiment Results

As each task has a different evaluation metric and cannot be directly compared, we calculate the relative performance improvement of each method compared to vanilla MTL, and then average the relative improvement for all tasks of each dataset. As summarized in Table 2, the average improvement of MT-CRL with $\mathcal{L}^{Var}_{G\text{-}IRM}$ is 5.5%, significantly higher than all other baseline methods. The most critical step of MT-CRL is to learn correct causal graph. We therefore report MT-CRL with different invariance regularization. As is shown in the last block, $\mathcal{L}^{Var}_{G\text{-}IRM}$ achieve better results for most datasets than $\mathcal{L}^{Norm}_{G\text{-}IRM}$, while removing the invariance regularization could significantly drop the relative performance. Compared to IRM which calculate gradient and update per-task predictors, MT-CRL uses disentangled modules and G-IRM to avoid overfitting to achieve invariance. Results show that for datasets with large amount of tasks, e.g., Taskonomy and NYUv2, MT-CRL significantly outperform IRM, showing the modification is more suitable for MTL setup.

Ablation Studies. We then study the effectiveness of the other two components in MT-CRL, i.e., disentangled and graph regularization. We mainly report the ablation studies on Multi-MNIST in table 3 as it's relatively small so that we could quickly get the results of all combinations.

For disentangled regularization, after removing $\mathcal{L}_{decor}$, the performance drops from 0.915 to 0.882, which fits our discussion that we cannot conduct causal learning over entangled modules. We also explore one classical generative disentangled representation method, i.e., β-VAE. As shown in the table, the results of using β-VAE are 0.896, lower than our utilized decorrelation regularization.. We hypothesize that this is probably because not all generative factors are useful for downstream tasks. Generative objectives might compete for the model capacity and in addition, the unused factors could be potentially spurious.

Another key component is graph regularization. After removing both $\mathcal{L}_{sps}$ and $\mathcal{L}_{bal}$, the performance drops to 0.891. This show that even if invariance regularization could penalize non-causal modules, it would be better to force their weights to be zero via sparsity regularization, and to be non-degenerate

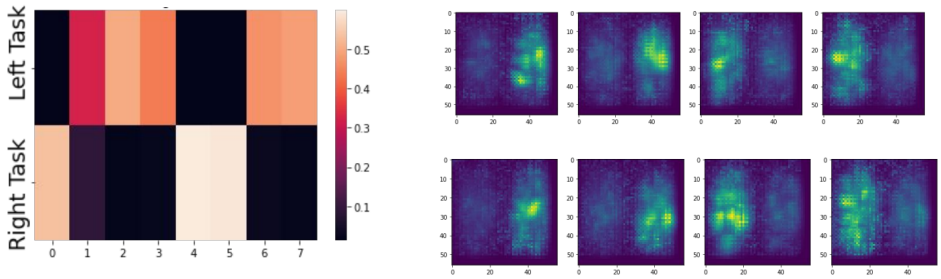

Figure 5: (valid-train) Task-to-Module gradients of model **with** MT-CRL on Multi-MNIST.

via balance regularization. We also conduct ablation studies to remove either $\mathcal{L}_{sps}$ or $\mathcal{L}_{bal}$, and results show both are important, and combining the two could help to achieve the best results.

Case Study. To show that real-world MTL problem indeed have spurious correlation problem and our MT-CRL could alleciate it, we take MovieLens as an example to conduct case study. Each task is for different movie types, and bag-of-word of movie title is one of the features. We calculate the task-to-module gradients $\frac{\partial(f(\Phi(x))[y])}{\partial F}$ of the vanilla MMoE model without MT-CRL. We then visualize 'train' gradients, which shows how much each module is utilized to fit the training set, and 'valid-train' gradients, which shows how generalizable each module is. We find that module 5 is utilized for **children** movie, but harmful in valid set, indicating it is a spurious feature. We then use Grad-CAM to show that top words of module 5 include *strip* and *die*, which is not relevant to **children** movies. One possible reason is that some children movies contain the words *club*, which is often co-occurred with *strip* and *die* in **crime** and **war** movies. After adding our MT-CRL, the module assigned to 'children' movie attends *Pink*, *Parenthood*, *Alice* and *Jungle*.

We then show the (valid-train) Task-to-Module gradients over Multi-MNIST datasets. With MT-CRL, in Figure 14, each module's saliency map only focus on one side of pixels. By looking at each task output's saliency map, which help model to focus only on causal part, compared with Figure 3(b) that have high weights on both. We also show the detailed gradient saliency map and induced task similarity graph of MovieLens, Taskonomy in Appendix G. All these case studies show MT-CRL could indeed alleviate spurious correlation in real MTL problems.

5 Related Work

Multi-Task Generalization. A deep neural model often requires a large number of training samples to generalize well (Arora et al., 2019; Cao & Gu, 2019). To alleviate the sample sparsity problem, MTL could leverage more labeled data from multiple tasks (Zhang & Yang, 2018). Most works studying multi-task generalization are based on a core assumption that the tasks are correlated. Earlier research directly define the task relatedness with statistical assumption (Baxter, 2000; Ben-David & Borbely, 2008; Lampinen & Ganguli, 2019). With the increasing focus on deep learning models, recent research decompose ground-truth MTL models into a shared representation and different task-specific layers from a hypothesis family (Maurer et al., 2016). With such decomposition, Tripuraneni et al. (2020) and Du et al. (2021) prove that a diverse set of tasks could help learn more generalizable representation. Wu et al. (2020) study how covariate shifts influence MTL generalization. Despite these findings, the core assumption of task relatedness might not be satisfied in many real-world applications (Parisotto et al., 2016; Zhang et al., 2021), in which tasks could even conflict with each other to compete model capacity, and the generalization performance of MTL could be worse than single-task training.

To solve the task conflict problem, a number of MTL model architectures have utilized modular (Misra et al., 2016; Lu et al., 2017; Rosenbaum et al., 2018; Ma et al., 2018; Guo et al., 2020) or attention-based (Liu et al., 2019; Maninis et al., 2019) design to enlarge model capacity while preserving information sharing. Our work is model-agnostic and could be applied to existing architectures to further solve the spurious feature problem. Another line of research alleviate task conflict during optimization. Some propose to balance the task weight via uncertainty estimation (Kendall et al., 2018), gradient norm (Chen et al., 2018), convergence rate (Liu et al., 2019), or pareto optimality (Sener & Koltun, 2018). Others directly modulate task gradients via dropping part of the conflict

gradient (Chen et al., 2020) or project task's gradient onto other tasks' gradient surface (Yu et al., 2020; Wang et al., 2021). Though these works successfully facilitate MTL model to converge easier, our analysis show that with spurious correlation, the MTL model with low training loss could still generalize bad. Therefore, our proposed MT-CRL that alleviates spurious correlation is orthogonal to these prior works, and could be combined to further improve overall performance.

Spurious Correlation Problem. Due to the selection bias (Torralba & Efros, 2011; Gururangan et al., 2018) or unobserved confounding factors (Lopez-Paz, 2016), training datasets always contain spurious correlations between non-causal features and task labels, with which trained models often leverage non-causal knowledge and may fail to generalize Out-Of-Distribution (OOD) when such correlation changes (Nagarajan et al., 2021). To solve the spurious correlation problem, some fairness research pre-define a set of non-causal features (e.g., gender and underrepresented identity) and then explicitly remove them from the learned representation (Zemel et al., 2013; Ganin et al., 2016; Wang et al., 2019). Another line of robust machine learning research does not assume to know spurious features, but regularize the model to perform equally well under different distribution. Distributionally Robust Optimization (DRO) optimizes worst-case risk (Sagawa et al., 2020). Invariant Causal Prediction (ICP) learns causal relations via invariance testing (Peters et al., 2016). Invariant Risk Minimization (IRM) forces the final predictor to be optimal across different domains (Arjovsky et al., 2019). Risk Extrapolation (REx) directly penalizes the variance of training risk in different domains (Krueger et al., 2021). Another line of work aim at learning causal representation (Schölkopf et al., 2021), i.e., high-level variables representing different aspect of knowledge from raw data input. Most of these works try to recover disentangled causal generative mechanisms (Parascandolo et al., 2018; Bengio et al., 2020; Liu et al., 2020; Mitrovic et al., 2021). Despite the extensive study of spurious correlation in single-task setting, few work discuss it for MTL models. This paper is the first to point out the unique challenges of spurious correlation in MTL setup.

6 Conclusion

In this paper, we study spurious correlation problem in the Multi-Task Learning (MTL) setting. We theoretically and experimentally shows that task correlation can introduce special type of spurious correlation in MTL, and the model trained by MTL is more prone to leverage non-causal knowledge from other tasks than single-task learning. To solve the problem, we propose Multi-Task Causal Representation Learning (MT-CRL) which consists of: 1) a decorrelation regularizer to learn disentangled modules; 2) a graph regularizer to learn sparse and non-degenerate task-to-module graph; 3) G-IRM invariant regularizer. We show MT-CRL could improve performance of MTL models on benchmark datasets and could alleviate spurious correlation.

Limitation Statement. Our analysis is based on label-label confounders. However, existing MTL datasets don't provide exact confounder changes to study spurious correlation problem. As mitigation, in analysis part, we create two synthetic datasets, and in experiment part, we adopt train/valid/test split with several attribution differences to mimic confounder changes. To further study spurious correlation in MTL, in the future, we'd like to construct benchmark MTL datasets with known confounder changes (or analyze how some key attribute changes lead to spurious correlation problem), build mathematical model based on it, and also explore and visualize which part of knowledge in real-world MTL datasets (e.g. Taskonomy) could be spuriously correlated to other tasks.

Acknowledgement. We sincerely thanks anoymoused NeurIPS reviewers for their constructive comments and suggestions to improve this paper. We thank Huan Gui, Kang Lee, Alexander D'Amour, Xuezhi Wang, Jilin Chen and Minmin Chen for insightful discussion and suggestion for this work. We also thank Ang Li for technical support for running experiments on Taskonomy, and Thanh Vu for running CityScape and NYUv2 experiments.

References

Ahmed, F., Bengio, Y., van Seijen, H., and Courville, A. C. Systematic generalisation with group invariant predictions. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=b9PoimzZFJ.

Ahuja, K., Shanmugam, K., Varshney, K. R., and Dhurandhar, A. Invariant risk minimization games. *CoRR*, abs/2002.04692, 2020. URL https://arxiv.org/abs/2002.04692.

Arjovsky, M., Bottou, L., Gulrajani, I., and Lopez-Paz, D. Invariant risk minimization. *CoRR*, abs/1907.02893, 2019. URL http://arxiv.org/abs/1907.02893.

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 322–332. PMLR, 2019. URL http://proceedings.mlr.press/v97/arora19a.html.

Badrinarayanan, V., Kendall, A., and Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. URL http://arxiv.org/abs/1511.00561.

Baksalary, J. K. and Baksalary, O. M. Particular formulae for the moore–penrose inverse of a column-wise partitioned matrix. *Linear Algebra and its Applications*, 421(1):16–23, 2007. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2006.03.031. URL https://www.sciencedirect.com/science/article/pii/S0024379506001959. Special Issue devoted to the 12th ILAS Conference.

Balaji, Y., Farajtabar, M., Yin, D., Mott, A., and Li, A. The effectiveness of memory replay in large scale continual learning. *CoRR*, abs/2010.02418, 2020. URL https://arxiv.org/abs/2010.02418.

Baxter, J. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

Beery, S., Horn, G. V., and Perona, P. Recognition in terra incognita. In Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XVI*, volume 11220 of *Lecture Notes in Computer Science*, pp. 472–489. Springer, 2018. doi: 10.1007/978-3-030-01270-0_28. URL https://doi.org/10.1007/978-3-030-01270-0_28.

Ben-David, S. and Borbely, R. S. A notion of task relatedness yielding provable multiple-task learning guarantees. *Mach. Learn.*, 73(3):273–287, 2008. doi: 10.1007/s10994-007-5043-5. URL https://doi.org/10.1007/s10994-007-5043-5.

Bengio, Y., Deleu, T., Rahaman, N., Ke, N. R., Lachapelle, S., Bilaniuk, O., Goyal, A., and Pal, C. J. A meta-transfer objective for learning to disentangle causal mechanisms. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ryxWIgBFPS.

Cao, Y. and Gu, Q. Generalization bounds of stochastic gradient descent for wide and deep neural networks. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 10835–10845, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/cf9dc5e4e194fc21f397b4cac9cc3ae9-Abstract.html.

Caruana, R. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., and Abbeel, P. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 2172–2180, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Abstract.html.

Chen, Z., Badrinarayanan, V., Lee, C., and Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 793–802. PMLR, 2018. URL http://proceedings.mlr.press/v80/chen18a.html.

Chen, Z., Ngiam, J., Huang, Y., Luong, T., Kretzschmar, H., Chai, Y., and Anguelov, D. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (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/16002f7a455a94aa4e91cc34ebdb9f2d-Abstract.html.

Cheung, B., Livezey, J. A., Bansal, A. K., and Olshausen, B. A. Discovering hidden factors of variation in deep networks. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6583.

Cogswell, M., Ahmed, F., Girshick, R. B., Zitnick, L., and Batra, D. Reducing overfitting in deep networks by decorrelating representations. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06068.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 3213–3223. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.350. URL https://doi.org/10.1109/CVPR.2016.350.

Du, S. S., Hu, W., Kakade, S. M., Lee, J. D., and Lei, Q. Few-shot learning via learning the representation, provably. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=pW2Q2xLwIMD.

Fifty, C., Amid, E., Zhao, Z., Yu, T., Anil, R., and Finn, C. Efficiently identifying task groupings for multi-task learning. *CoRR*, abs/2109.04617, 2021. URL https://arxiv.org/abs/2109.04617.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. S. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17: 59:1–59:35, 2016. URL http://jmlr.org/papers/v17/15-239.html.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bygh9j09KX.

Geirhos, R., Jacobsen, J., Michaelis, C., Zemel, R. S., Brendel, W., Bethge, M., and Wichmann, F. A. Shortcut learning in deep neural networks. *Nat. Mach. Intell.*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z. URL https://doi.org/10.1038/s42256-020-00257-z.

Guo, P., Lee, C., and Ulbricht, D. Learning to branch for multi-task learning. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3854–3863. PMLR, 2020. URL http://proceedings.mlr.press/v119/guo20e.html.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S. R., and Smith, N. A. Annotation artifacts in natural language inference data. In Walker, M. A., Ji, H., and Stent, A. (eds.), *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pp. 107–112. Association for Computational Linguistics, 2018. doi: 10.18653/v1/n18-2017. URL https://doi.org/10.18653/v1/n18-2017.

Hainmueller, J. Entropy balancing for causal effects: A multivariate reweighting method to produce balanced samples in observational studies. *Political analysis*, 20(1):25–46, 2012.

Harper, F. M. and Konstan, J. A. The movielens datasets: History and context. *ACM Trans. Interact. Intell. Syst.*, 5(4):19:1–19:19, 2016. doi: 10.1145/2827872. URL https://doi.org/10.1145/2827872.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. beta-vae: Learning basic visual concepts with a constrained variational framework. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Sy2fzU9gl.

Kendall, A., Gal, Y., and Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 7482–7491. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00781. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Kendall_Multi-Task_Learning_Using_CVPR_2018_paper.html.

Khani, F. and Liang, P. Removing spurious features can hurt accuracy and affect groups disproportionately. In Elish, M. C., Isaac, W., and Zemel, R. S. (eds.), *FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021*, pp. 196–205. ACM, 2021. doi: 10.1145/3442188.3445883. URL https://doi.org/10.1145/3442188.3445883.

Koyama, M. and Yamaguchi, S. When is invariance useful in an out-of-distribution generalization problem? *CoRR*, abs/2008.01883, 2021. URL https://arxiv.org/abs/2008.01883.

Krueger, D., Caballero, E., Jacobsen, J., Zhang, A., Binas, J., Zhang, D., Priol, R. L., and Courville, A. C. Out-of-distribution generalization via risk extrapolation (rex). In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5815–5826. PMLR, 2021. URL http://proceedings.mlr.press/v139/krueger21a.html.

Lachapelle, S., Brouillard, P., Deleu, T., and Lacoste-Julien, S. Gradient-based neural DAG learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rklbKA4YDS.

Lachapelle, S., López, P. R., Sharma, Y., Everett, K., Priol, R. L., Lacoste, A., and Lacoste-Julien, S. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ica. *arXiv preprint arXiv:2107.10098*, 2021.

Lampinen, A. K. and Ganguli, S. An analytic theory of generalization dynamics and transfer learning in deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryfMLoCqtQ.

Liu, C., Sun, X., Wang, J., Li, T., Qin, T., Chen, W., and Liu, T. Learning causal semantic representation for out-of-distribution prediction. *CoRR*, abs/2011.01681, 2020. URL https://arxiv.org/abs/2011.01681.

Liu, S., Johns, E., and Davison, A. J. End-to-end multi-task learning with attention. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 1871–1880. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00197. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Liu_End-To-End_Multi-Task_Learning_With_Attention_CVPR_2019_paper.html.

Locatello, F., Bauer, S., Lucic, M., Rätsch, G., Gelly, S., Schölkopf, B., and Bachem, O. Challenging common assumptions in the unsupervised learning of disentangled representations. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 4114–4124. PMLR, 2019. URL http://proceedings.mlr.press/v97/locatello19a.html.

Lopez-Paz, D. From dependence to causation. *arXiv: Machine Learning*, 2016.

Lu, Y., Kumar, A., Zhai, S., Cheng, Y., Javidi, T., and Feris, R. S. Fully-adaptive feature sharing in multi-task networks with applications in person attribute classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 1131–1140. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.126. URL https://doi.org/10.1109/CVPR.2017.126.

Ma, J., Zhao, Z., Yi, X., Chen, J., Hong, L., and Chi, E. H. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In Guo, Y. and Farooq, F. (eds.), *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, pp. 1930–1939. ACM, 2018. doi: 10.1145/3219819.3220007. URL https://doi.org/10.1145/3219819.3220007.

Maninis, K., Radosavovic, I., and Kokkinos, I. Attentive single-tasking of multiple tasks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 1851–1860. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00195. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Maninis_Attentive_Single-Tasking_of_Multiple_Tasks_CVPR_2019_paper.html.

Maurer, A., Pontil, M., and Romera-Paredes, B. The benefit of multitask representation learning. *J. Mach. Learn. Res.*, 17:81:1–81:32, 2016. URL http://jmlr.org/papers/v17/15-242.html.

Misra, I., Shrivastava, A., Gupta, A., and Hebert, M. Cross-stitch networks for multi-task learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 3994–4003. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.433. URL https://doi.org/10.1109/CVPR.2016.433.

Mitrovic, J., McWilliams, B., Walker, J. C., Buesing, L. H., and Blundell, C. Representation learning via invariant causal mechanisms. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=9p2ekP904Rs.

Nagarajan, V., Andreassen, A., and Neyshabur, B. Understanding the failure modes of out-of-distribution generalization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=fSTD6NFIW_b.

Ng, I., Fang, Z., Zhu, S., and Chen, Z. Masked gradient-based causal structure learning. *CoRR*, abs/1910.08527, 2019. URL http://arxiv.org/abs/1910.08527.

Parascandolo, G., Kilbertus, N., Rojas-Carulla, M., and Schölkopf, B. Learning independent causal mechanisms. In Dy, J. G. and Krause, A. (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4033–4041. PMLR, 2018. URL http://proceedings.mlr.press/v80/parascandolo18a.html.

Parisotto, E., Ba, L. J., and Salakhutdinov, R. Actor-mimic: Deep multitask and transfer reinforcement learning. In Bengio, Y. and LeCun, Y. (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.06342.

Peters, J., Bühlmann, P., and Meinshausen, N. Causal inference by using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, pp. 947–1012, 2016.

Rosenbaum, C., Klinger, T., and Riemer, M. Routing networks: Adaptive selection of non-linear functions for multi-task learning. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=ry8dvM-R-.

Rosenfeld, E., Ravikumar, P. K., and Risteski, A. The risks of invariant risk minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=BbNIbVPJ-42.

Sagawa, S., Koh, P. W., Hashimoto, T. B., and Liang, P. Distributionally robust neural networks. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=ryxGuJrFvS.

Schölkopf, B., Locatello, F., Bauer, S., Ke, N. R., Kalchbrenner, N., Goyal, A., and Bengio, Y. Toward causal representation learning. *Proc. IEEE*, 109(5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954. URL https://doi.org/10.1109/JPROC.2021.3058954.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 525–536, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/432aca3a1e345e339f35a30c8f65edce-Abstract.html.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G. E., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Silberman, N., Hoiem, D., Kohli, P., and Fergus, R. Indoor segmentation and support inference from RGBD images. In Fitzgibbon, A. W., Lazebnik, S., Perona, P., Sato, Y., and Schmid, C. (eds.), *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V*, volume 7576 of *Lecture Notes in Computer Science*, pp. 746–760. Springer, 2012. doi: 10.1007/978-3-642-33715-4_54. URL https://doi.org/10.1007/978-3-642-33715-4_54.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6034.

Standley, T., Zamir, A. R., Chen, D., Guibas, L. J., Malik, J., and Savarese, S. Which tasks should be learned together in multi-task learning? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 9120–9132. PMLR, 2020. URL http://proceedings.mlr.press/v119/standley20a.html.

Sun, S.-H. Multi-digit mnist for few-shot learning, 2019. URL https://github.com/shaohua0116/MultiDigitMNIST.

Suter, R., Miladinovic, D., Schölkopf, B., and Bauer, S. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6056–6065. PMLR, 2019. URL http://proceedings.mlr.press/v97/suter19a.html.

Torralba, A. and Efros, A. A. Unbiased look at dataset bias. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pp. 1521–1528. IEEE Computer Society, 2011. doi: 10.1109/CVPR.2011.5995347. URL https://doi.org/10.1109/CVPR.2011.5995347.

Tripuraneni, N., Jordan, M. I., and Jin, C. On the theory of transfer learning: The importance of task diversity. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (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/59587bffec1c7846f3e34230141556ae-Abstract.html.

Wang, T., Zhao, J., Yatskar, M., Chang, K., and Ordonez, V. Balanced datasets are not enough: Estimating and mitigating gender bias in deep image representations. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 5309–5318. IEEE, 2019. doi: 10.1109/ICCV.2019.00541. URL https://doi.org/10.1109/ICCV.2019.00541.

Wang, Z., Tsvetkov, Y., Firat, O., and Cao, Y. Gradient vaccine: Investigating and improving multi-task optimization in massively multilingual models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=F1vEjWK-lH_.

Wu, S., Zhang, H. R., and Ré, C. Understanding and improving information transfer in multi-task learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SylzhkBtDB.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (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/3fe78a8acf5fda99de95303940a2420c-Abstract.html.

Zamir, A. R., Sax, A., Shen, W. B., Guibas, L. J., Malik, J., and Savarese, S. Taskonomy: Disentangling task transfer learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 3712–3722. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00391. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zamir_Taskonomy_Disentangling_Task_CVPR_2018_paper.html.

Zemel, R. S., Wu, Y., Swersky, K., Pitassi, T., and Dwork, C. Learning fair representations. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, pp. 325–333. JMLR.org, 2013. URL http://proceedings.mlr.press/v28/zemel13.html.

Zhang, W., Deng, L., Zhang, L., and Wu, D. A survey on negative transfer. *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, 2021.

Zhang, Y. and Yang, Q. An overview of multi-task learning. *National Science Review*, 5(1):30–43, 2018.

Zheng, X., Aragam, B., Ravikumar, P., and Xing, E. P. Dags with NO TEARS: continuous optimization for structure learning. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 9492–9503, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/e347c51419ffb23ca3fd5050202f9c3d-Abstract.html.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] The main contribution of this paper is to point out the unique challenges of spurious correlation problem in MTL setup, and propose a workable solution. This has been discussed in abstract and intro.

(b) Did you describe the limitations of your work? [Yes] We've clearly stated the limitations and future directions to improve this work in Conclusion section.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] We discuss how MTL could help solve label scarcity problem in related work. Thus, for some task with evil purpose, MTL could still help their performance, which has negative societal impacts.

(d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes] I've carefully read the guidances and ensured our paper conforms to them.

2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] We state the assumption of proposition 1. Specifically, we assume if MTL has spurious task correlation, then the model could utilize non-causal factors.

(b) Did you include complete proofs of all theoretical results? [Yes] The proof for Theorem 1 is described in Appendix A.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] They are all included in the supplemental material.

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] They are specified in Appendix D.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] The detailed results with error bar is listed in Table 6-10 in Appendix.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] They are specified in Appendix E.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] Yes, we've properly cite all used code and data in Appendix.

(b) Did you mention the license of the assets? [Yes] I cite the github link with license.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] I'll submit our own code in supplemental material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We do not use any personal data.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We do not use any personal data.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proof of Proposition 1

A.1 Problem Definition

We consider two binary classification tasks, with Y_a and Y_b as variables from $\{\pm 1\}$ for task label. The task labels are drawn from two different probabilities. For simplicity, we assume the probability to sample the two label value is balanced, i.e., $P(Y=1) = P(Y=-1) = 0.5$. Our conclusion could be extended to unbalanced distribution.

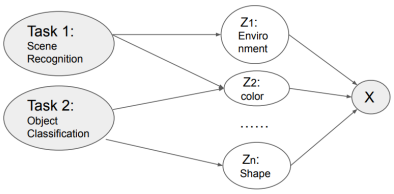

Figure 6: Illustrative Diagram of Causal Generative Model in MTL setting

In this paper, we mainly study the spurious correlation between task labels. For simplicity, we define $P(Y_a = Y_b) = m_C, P(Y_a \neq Y_b) = 1 - m_C$, where m_C denotes that this correlation could change by different confounder C_{dist}^{MTL}. In some environments $m_C \neq 0.5$, meaning that the two tasks are correlated in these environments. To sum up, we could define the probability table as:

	$Y_a = 1$	$Y_a = 0$
$Y_b = 1$	m_C	$1 - m_C$
$Y_b = 0$	$1 - m_C$	m_C

Table 4: Probability table for $P(Y_a, Y_b)$, where m_C denotes the correlation between the two task label.

We consider two d-dimensional factors F_a and F_b representing the knowledge to tackle the two tasks. Both are drawn from Gaussian distribution:

$$F_a \sim \mathcal{N}(Y_a \cdot \mu_a, \sigma_a^2 I), \;\; F_b \sim \mathcal{N}(Y_b \cdot \mu_b, \sigma_b^2 I) \tag{10}$$

with $\mu_a, \mu_b \in \mathbb{R}^d$ denote the mean vectors and σ_a, σ_b are covariance vectors.

Our goal to learn two linear models $P(Y_{\{a/b\}}|F_a, F_b) = \text{sigmoid}(\beta F) = \text{sigmoid}(\beta_a F_a + \beta_b F_b)$. We first consider the setting that we're given infinite samples. If we assume there's no traditional factor-label spurious correlation in single task learning, the bayes optimal classifier will only take each task's causal factor as feature, and assign zero weights to non-causal factors. The factor with the regression vector $\beta_a = \frac{\mu_a}{\sigma_a^2}$ for bayes optimal classifier of task a and $\beta_b = 2\frac{\mu_b}{\sigma_b^2}$ for bayes optimal classifier of task b.

A.2 Bayes Optimal Classifier for Multiple-Task

When we train a single model using both tasks, the optimal Bayes classifier will utilize the other non-causal factor due to the influence of spurious correlation quantified by m_C. To prove it, we take the first task with label Y_a as an example and derive the optimal Bayes classifier as:

$$P(Y_a|F_a, F_b) = \frac{P(Y_a, F_a, F_b)}{P(F_a, F_b)} = \frac{P(Y_a, F_a, F_b)}{\sum_{Y_a \in \{-1,1\}} P(Y_a, F_a, F_b)} \tag{11}$$

while the probability of $P(Y_a, F_a, F_b)$ could be written as:

$$\begin{aligned}
P(Y_a, F_a, F_b) &= P(Y_a, F_a) \cdot P(F_b|Y_a, F_a) && (12)\\
&= P(Y_a, F_a) \cdot P(F_b|Y_a) && (13)\\
&= P(Y_a, F_a) \cdot \sum_{Y_b \in \{-1,1\}} P(F_b, Y_b|Y_a) && (14)\\
&= P(Y_a)P(F_a|Y_a) \cdot \sum_{Y_b \in \{-1,1\}} P(F_b|Y_b)P(Y_b|Y_a) && (15)\\
&\propto e^{Y_a \cdot F_a \beta_a} \cdot \left(m_C e^{Y_a \cdot F_b \beta_b} + (1 - m_C) e^{-Y_a \cdot F_b \beta_b}\right) && (16)\\
&= m_C e^{Y_a (F_a \mu_a + F_b \mu_b)} + (1 - m_C) e^{Y_a (F_a \mu_a - F_b \mu_b)} && (17)
\end{aligned}$$

By putting it back to equation(11), we could get:

$$P(Y_a|F_a, F_b) = \frac{1}{1 + \frac{m_C e^{Y_a(F_a\beta_a + F_b\beta_b)} + (1-m)e^{Y_a(F_a\beta_a - F_b\beta_b)}}{m_C e^{-Y_a(F_a\beta_a + F_b\beta_b)} + (1-m)e^{-Y_a(F_a\beta_a - F_b\beta_b)}}} \tag{18}$$

The formula shows that the optimal bayes classifier depends on the non-causal factor F_b given $m_C \neq 0.5$.

To give two extreme, when $m_C = 1$:

$$P(Y_a|F_a, F_b) = \frac{1}{1 + e^{2Y_a(F_a\beta_a + F_b\beta_b)}} \tag{19}$$

In this way, the optimal classifier is $\beta = [2\beta_a, 2\beta_b]^T$ for the two factors F_a and F_b.

When $m_C = 0.5$:

$$P(Y_a|F_a, F_b) = \frac{1}{1 + e^{2Y_a(F_a\beta_a)}} \tag{20}$$

In this way, the optimal classifier is $\beta = [2\beta_a, 0]^T$, which only utilizes the first factor F_a and assign zero weights for the non-causal factor F_b.

A.3 Classifier trained on limited dataset

In the following we're considering the cases whether there's no task correlation in training set ($m_C = 0.5$). Though we have shown previously the optimal classifier should be invariant to non-causal factors given unlimited data, in reality with limited training dataset, the model could still utilize non-causal factors as noise.

Assume the training data contains spurious feature S appended to causal feature C for ground-truth linear model $Y = \theta^* C$, both under-parametrized and over-paramatrized linear model $\hat{Y} = \hat{\theta}C + \hat{\beta}S$ will assign non-zero weights $\hat{\beta}$ for spurious feature S.

Let $x \in \mathbb{R}^{(d+1)\times 1}$ denote the feature, where $x[1:d] = c$ is the causal feature, and $x[d+1] = s$ is spurious feature.

Let ground-truth linear model $y_i = f_{\theta^*}(x) = \theta^* \cdot x_i + \epsilon_i = c \cdot \theta_c^* + \epsilon_i$, where $\theta^* = [\theta_c^*, 0] \in \mathbb{R}^{(d+1)\times 1}$ and $\epsilon \sim N(0, \sigma^2)$.

Given training dataset $X \in \mathbb{R}^{n\times(d+1)}$ and $Y = X\theta^* + \varepsilon = C\theta_c^* + \varepsilon \in \mathbb{R}^{n\times 1}$, the closed-form solution $\hat{\theta} \in \mathbb{R}^{(d+1)\times 1}$ for linear regression model is:

$$\hat{\theta} = X^+ Y^+ = X^+(X\theta^* + \varepsilon) \tag{21}$$

The generalization error is:

$$\mathcal{L} = \mathbb{E}_x\left[\left((\hat{\theta} - \theta^*) \cdot x\right)^2\right] \tag{22}$$

$$= \mathbb{E}_x\left[\left((X^+X - I)\theta^* \cdot x + X^+\varepsilon \cdot x\right)^2\right] \tag{23}$$

$$= \mathbb{E}_x\left[\left((X^+X - I)\theta^* \cdot x\right)^2\right] + \sigma^2 \mathbb{E}_x\left\|(X^T)^+ x\right\|_2^2 \tag{24}$$

The first term is bias and the second is variance.

If $X = [C, 0]$, which only contains causal feature without any spurious feature, we denote the learned parameter and loss as $\hat{\theta}_C$ and $\mathcal{L}_C$.

If $X = [C, S]$, which contains the spurious feature, we denote the learned parameter and loss as $\hat{\theta}_S$ and $\mathcal{L}_S$.

Our goal is to prove the learned parameter weight for the spurious feature is not zero. We'll study it in both underparamtrizied ($d + 1 \leq n$) setting, where the solution is equivalent to least-square solution; and overparametrized ($d > n$), where the solution is equivalent to min-norm solution.

A.3.1 Underparametrized Setting

Loss Since $X \in \mathbb{R}^{n\times(d+1)}$ has independent column due to under parametrization assumption, we can find pseudo-inverse such that $X^+X = I$. Thus the bias term in $\mathcal{L}$ is 0, and we only need to consider the variance term.

$$\mathcal{L}_S - \mathcal{L}_C = \sigma^2\left(\mathbb{E}_x\left\|\begin{bmatrix}C^T\\S^T\end{bmatrix}^+ x\right\|_2^2 - \mathbb{E}_x\left\|\begin{bmatrix}C^T\\0\end{bmatrix}^+ x\right\|_2^2\right) \tag{25}$$

Since $||A^+x||_2^2 = \min_{Z:Az=x}||z||_2^2$, and obviously $\left\{z\middle|\begin{bmatrix}C^T\\S^T\end{bmatrix}z = x\right\} \subseteq \left\{z\middle|\begin{bmatrix}C^T\\0\end{bmatrix}z = x\right\}$ as the first one has one more constraint. Therefore, $\left\|\begin{bmatrix}C^T\\S^T\end{bmatrix}^+ x\right\| \geq \left\|\begin{bmatrix}C^T\\0\end{bmatrix}^+ x\right\|_2^2$, and thus $\mathcal{L}_S \geq \mathcal{L}_C$.

weight By the theorem 1 of (Baksalary & Baksalary, 2007), if $d+1 \leq n$, $X = [S,T] \in \mathbb{R}^{n\times(d+1)}$ has independent column, thus we have

$$X^+ = \begin{bmatrix}C^T\\S^T\end{bmatrix}^+ = \begin{bmatrix}(I-Q)C(C^T(I-Q)C)^{-1}\\ \frac{(I-P)S}{S^T(I-P)S}\end{bmatrix} \tag{26}$$

where $P = CC^T$, $Q = SS^T$.

Therefore,

$$\hat{\theta}_S[d+1] = \frac{(I-P)S}{S^T(I-P)S}Y = \frac{(I-P)S(C\theta_C^* + \varepsilon)}{S^T(I-P)S} \tag{27}$$

A.3.2 Overparametrized Setting

In this setting the closed-form solution is equivalent to minimum-norm solution, such that:

$$\hat{\theta} = \arg\min_{\theta}||\theta||_2^2 \tag{28}$$

$$s.t.\ \ X\theta = Y \tag{29}$$

weight Since X is have full row rank, $(XX^T)^-1$ exists, thus we have:

$$X^+ = X^T(XX^T)^{-1} \tag{30}$$

Based on the Sherman-Morrison formula, we have:

$$(XX^T)^{-1} = (CC^T + SS^T)^{-1} = G - \frac{GSS^TG}{1+S^TGS} \tag{31}$$

where $G = (CC^T)^{-1}, u = \frac{b^TG}{1+b^TGb}$. Therefore:

$$X^+ = \begin{bmatrix}C^T\\S^T\end{bmatrix}^+ = \begin{bmatrix}(I-bu)C^+\\ u\end{bmatrix} \tag{32}$$

Thus

$$\hat{\theta}_S[d+1] = \frac{b^TG}{1+b^TGb}Y = \frac{b^TG(C\theta_C^* + \varepsilon)}{1+b^TGb} \tag{33}$$

To sum up, given limited training dataset, even without spurious correlation between tasks, and non-causal features only serve as noise, the model could still learn to assign non-zero weights to non-causal features to overfit the dataset. Therefore, in MTL setting, when the number of tasks increase, the shared representation encodes many causal features from different tasks. Even without spurious correlation, it will lead to overfitting issue. And such problem could be exacerbated by spurious correlation that we show in section A.2.

B Synthetic Analysis of Multi-SEM with more tasks and saliency map

In section 2.2 we compare model trained by MTL with STL with two tasks. Here we show the results conducted in Multi-SEM with more than two tasks in Table 5. The results show decreasing Acc_{val} and higher usage of spurious feature ρ_{spur} compared with STL, with increasing number of tasks. This matches our hypothesis that MTL could incorporate more non-causal features / factors into shared representation, increasing the risk of utilizing overfitting. We also show the saliency map for each feature dimension in Figure 7. It shows that the model trained by MTL exploits non-causal features (dimension 20-120) more than the model trained by STL. All these results empirically support our claim that with spurious task correlation, model trained by MTL utilize non-causal factors more and generalize worse than STL.

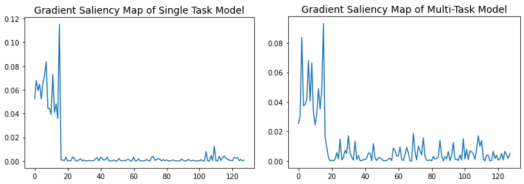

Figure 7: The gradient saliency map of Multi-SEM. The model trained by MTL exploits non-causal features (spurious) more.

#Tasks		2	3	4	5	6	7	8
MTL	Acc_{val}	0.846	0.838	0.824	0.809	0.785	0.752	0.719
	ρ_{spur}	0.328	0.357	0.391	0.429	0.475	0.530	0.594
STL	Acc_{val}	0.874	0.861	0.848	0.836	0.827	0.810	0.797
	ρ_{spur}	0.261	0.289	0.314	0.354	0.385	0.407	0.435

Table 5: Results on Multi-SEM with more than 2 tasks.

C Pseudo-Code and more discussion of MT-CRL

The full psudo-code of proposed MTL is shown in Alg. 1. We first use disentangled MMoE model to calculate loss for each task $R_t\big(\Phi, A_t, f_t\big)$, and also calculate disentangled and graph regularization. We then calculate invariant regularization over train/valid split. The most important part is line 11 we detach the per-task predictors from computational graph, so that when we calculate gradient (via $loss.backward$), we only calculate gradient over graph A and encoder Φ.

Ideally the invariant loss should be calculated based on different environmental split, similar to what is utilized in existing Out-Of-Distribution Generalization works. However, in MTL setting, there's no datasets designed specifically for studying OOD generalization or spurious correlation. To make current approach suitable for real-world applications, we only utilize two environment split (i.e. train and valid from existing datasets).

Noted that in our framework we adopt a simple linear correlation regularization to enforce disentanglement. This regularization only forces representation to be linearly de-correlated, and a more strict solution might be reducing the mutual information (MI). However, existing methods to minimizing MI requires either knowing the latent distribution (e.g. InfoGAN. We report BetaVAE in Table 3 with similar intuition but performs worse) or over estimated MI (e.g. MINE). We indeed tried adding discriminator for every module pair and adopted Minmax training to minimize estimated MINE. The result is unstable and no better. Module output's norm is very large and only the centers are seperated rather than disentangled. Therefore, we only utilize the linear de-correlation methods that perform well in our experiments. We show the mutual correlation of every pairs of modules in Figure 8 learned in MultiMNIST dataset. It shows that after learning, the modules indeed learn to be linearly de-correlated between each other, and only have correlated neurons within each module.

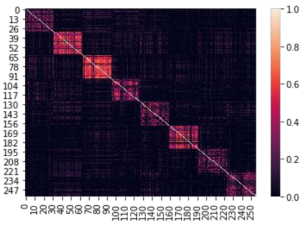

Figure 8: The heatmap of mutual correlation $\rho(\mathbf{Z}_i, \mathbf{Z}_j)$ between every pairs of modules.

Algorithm 1: Pseudo-Code of proposed MT-CRL (use $\mathcal{L}^{Var}_{G\text{-}IRM}$ as invariant regularizer)

Require: shared encoders with K different neural modules $\Phi = \left[\Phi_i(\cdot)\right]_{i=1}^{K}$, biadjacency matrix $A = \text{sigmoid}(\theta) \in [0,1]^{T\times K}$, per-task predictors $\mathcal{F} = \{f_t\}_{t\in\mathcal{T}}$, minibatch with environment label and loss function for each task $\mathcal{B}_{t\in\mathcal{T}} = \{X_t, Y_t, E_t, \mathcal{L}_t\}_{t\in\mathcal{T}}$

$\mathcal{L}_{\mathcal{B}} = 0$
for each task $t \in \mathcal{T}$ **do**
  Get $X_t, Y_t, \mathcal{L}_t$ from $\mathcal{B}_t$
  $\mathbf{Z} = \left[\mathbf{Z}_i\right]_{i=1}^{K} = \left[\Phi_i(X_t)\right]_{i=1}^{K}$
  $\hat{Y}_t(\mathbf{X_t}) = f_t\big(\sum_i A_{t,i}\cdot \mathbf{Z}_i\big)\big) = f_t\big(\sum_i A_{t,i}\cdot \Phi_i(\mathbf{X_t})\big)$
  $\mathcal{L}_{\mathcal{B}} = \mathcal{L}_{\mathcal{B}} + R_t\big(\Phi, A_t, f_t\big) = \mathcal{L}_{\mathcal{B}} + \mathcal{L}_t(\hat{Y}_t(\mathbf{X_t}), Y_t)$
  $\mathcal{L}_{\mathcal{B}} = \mathcal{L}_{\mathcal{B}} + \mathcal{L}_{decor}(\Phi)_t = \mathcal{L}_{\mathcal{B}} + \lambda_{decor}\cdot \sum_{i=1}^{k}\sum_{j=i+1}^{k}\left\|\rho\big(\Phi_i(X_t), \Phi_j(X_t)\big)\right\|_F^2$
end for
$\mathcal{L}_{\mathcal{B}} = \mathcal{L}_{\mathcal{B}} + \mathcal{L}_{graph}(A) = \mathcal{L}_{\mathcal{B}} + \Big(\lambda_{sps}\cdot ||A||_1 - \lambda_{bal}\cdot \text{Entropy}\big(\frac{\sum_t A_{t,*}}{\sum_{t,i} A_{t,i}}\big)\Big)$
$grad = \nabla_{A,\mathcal{F},\Phi}\,\mathcal{L}_{\mathcal{B}}$
Detach $\mathcal{F} = \{f\}_t$ from computational graph (use $tf.stop_gradient$ or $torch.zero_grad$)
$\mathcal{L}^{Var}_{G\text{-}IRM}(\Phi, A|f) = 0$
for each task $t \in \mathcal{T}$ **do**
  Get environment label E_t. In our experimental setting it's train and valid set.
  $\mathcal{L}^{Var}_{G\text{-}IRM} = \mathcal{L}^{Var}_{G\text{-}IRM} + \sum_{e\in E_t}\frac{1}{|E_t|}\left\|\nabla_{A=A_t} R_t^e\big(\Phi, A, f_t\big) - \text{Avg}_e\Big(\nabla_{A=A_t} R_t^e\Big)\right\|^2$
end for
$grad = grad + \nabla_{A,\Phi}\,\mathcal{L}^{Var}_{G\text{-}IRM}(\Phi, A|f)$
Use optimizer to update the model via gradient $grad$

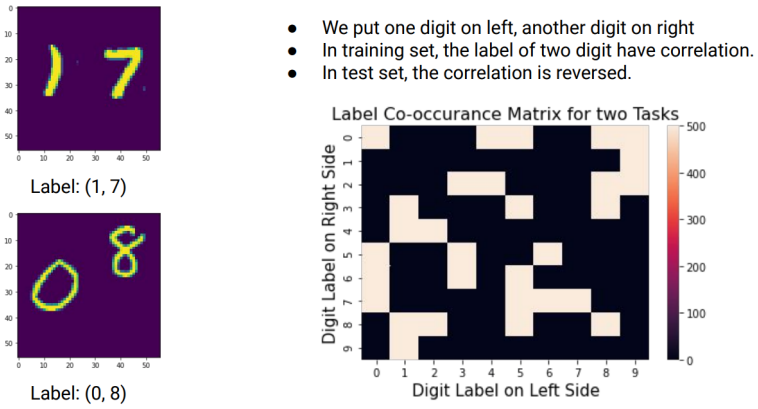

Figure 9: Illustrative figure for spurious Multi-MNIST dataset used for analaysis.

D Details about Dataset

D.1 Synthetic Datasets

Multi-SEM. We mostly follow the setting of linear Structural Equation Model (SEM) proposed by Rosenfeld et al. (2021). The two binary-classification task labels Y_a and Y_b are causally related to two distinctive factors F_a and F_b respectively via Gaussian distribution. We define the spurious correlation of the two labels by the probability that the two labels are the same: $C^{MTL}_{dist} = P(Y_a = Y_b)$. We set different C_{dist} for training and test sets to simulate distribution shifts.

Multi-MNIST. We modified the multi-digit MNIST (Sun, 2019), which samples two digit pictures and put in left and right position. The generative variables F_{left}, F_{right} are the digit images and data input is simply their concatenation: $X = [F_{left}, F_{right}]$. We define the task correlation C^{MTL}_{dist} by co-occurrence probability of the two digit labels. We randomly shuffle the label pairs and split the

training and test set such that the class label pairs do not overlap. An illustrative data point and the label pairs in training set is shown in Figure 9.

D.2 Real-world Datasets

Multi-MNIST (Harper & Konstan, 2016) is a multi-task variant of MNIST dataset, which samples two digit pictures and put in left and right position. We mainly modified from the this code repo[2] to generate the dataset. We sample 10,000 images for each label pair, so totally there are 1M data samples. As discussed in analysis section, to mimic distribution shifts (i.e., task correlation C_{dist}^{MTL}), we randomly shuffle the label pairs and split the train, valid and test set with ratio 3:1:1, such that every image co-occurrence correlation will no longer appear again in test set. We utilize the same CNN architectures and hyperparameter adopted in Yu et al. (2020) as base encoder, and one-layer MLP as per-task predictor.

MovieLens (Harper & Konstan, 2016) is a Movie recommendation dataset that contains 10M rating records[3] of 10,681 movies by 71,567 users from Jan. 1996 to Dec. 2008. We consider the rating regression for movies in each genre as different tasks. There are totally 18 different genres, including Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War and Western. To mimic distribution shifts across train, valid and test set, we split the data based on timestamp with ratio 8:1:1, and filter out non-overlapping users and movies from each set. We utilize a embedding layer followed by two-layer MLP as base encoder, and one-layer MLP as per-task predictor.

Taskonomy (Zamir et al., 2018) is a large-scale MTL benchmark dataset of indoor scene images from various buildings[4]. Every image has annotations for a set of diverse computer vision tasks. We follow the setting of (Balaji et al., 2020) to use 8 tasks, including curvature estimation, object classification, scene classification, surface normal estimation, semantic segmentation, depth estimation, occlusion edge, 2D keypoint estimation and 3D keypoint estimation. For these tasks, object and scene classification tasks are trained using cross entropy loss, semantic segmentation using pixelwise cross entropy, curvature estimation using L1 loss, and all other tasks using L2 loss. To mimic distribution shift, we select images from non-overlapping 48, 3, 3 buildings as train, valid and test set. The total training size is 324864 samples. We use Resnet-50 model as our base encoder network, and 15-layer CNN model with upsampling blocks as the per-task predictor.

NYUv2 (Silberman et al., 2012) is a dataset of 1449 RGB-D indoor scene images[5] with three tasks: 13-class semantic segmentation, depth estimation, and surface normals prediction. We use mean Intersection-Over-Union (mIoU), Relative Error (Rel Err) and Angle Distance as evaluation metric for the three tasks respectively. To mimic distribution shift, we split the dataset by scene labels into train, valid and test set with ratio 8:1:1. We follow the setting adopted in Yu et al. (2020) to use Segnet (Badrinarayanan et al., 2015) as the base encoder.

CityScape (Cordts et al., 2016) is a dataset of street-view images[6] with two tasks: semantic segmentation and depth estimation. We use mean Intersection-Over-Union (mIoU) and Relative Error (Rel Err) as evaluation metric for the three tasks respectively. We follow the same data pre-processing procedure of the original paper, and split images based on city into 2475, 500 and 500 train, valid and test samples. We follow the setting adopted in Yu et al. (2020) to use Segnet (Badrinarayanan et al., 2015) as the base encoder.

Tasks (Metric)	STL	MTL	PCGrad	GradVac	DANN	IRM	MT-CRL + $\mathcal{L}_{G\text{-}IRM}^{Var}$
Left-Digit (Acc.)	0.871 ± 0.018	0.844 ± 0.019	0.880 ± 0.019	0.884 ± 0.017	0.878 ± 0.020	0.887 ± 0.010	0.912 ± 0.018
Right-Digit (Acc.)	0.877 ± 0.015	0.848 ± 0.018	0.888 ± 0.017	0.886 ± 0.018	0.884 ± 0.017	0.889 ± 0.015	0.918 ± 0.019

Table 6: Results for Multi-MNIST dataset.

The results show that a middle number of module (K=8) achieves the best performance under the same size of model. Note that disentangled representation learning methods like BetaVAE assume

[2] https://github.com/shaohua0116/MultiDigitMNIST
[3] https://files.grouplens.org/datasets/movielens/ml-10m.zip
[4] http://taskonomy.stanford.edu/
[5] https://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html
[6] https://www.cityscapes-dataset.com/

Metric	STL	MTL	PCGrad	GradVac	DANN	IRM	MT-CRL + $\mathcal{L}_{G\text{-}IRM}^{Var}$
Avg. MSE	0.894 ± 0.006	0.892 ± 0.005	0.892 ± 0.006	0.891 ± 0.005	0.890 ± 0.007	0.890 ± 0.004	0.884 ± 0.006

Table 7: Results for MovieLens dataset.

Tasks (Metric)	STL	MTL	PCGrad	GradVac	DANN	IRM	MT-CRL + $\mathcal{L}_{G\text{-}IRM}^{Var}$
object classification (Cross Entropy)	3.37	3.18	3.09	3.06	3.13	3.16	3.01
scene classification (Cross Entropy)	2.65	2.59	2.54	2.51	2.58	2.59	2.47
semantic segmentation (Cross Entropy)	1.68	1.54	1.47	1.49	1.53	1.56	1.43
curvature estimation (L1 Loss)	0.246	0.224	0.218	0.212	0.237	0.226	0.208
surface normal estimation (L2 Loss)	0.138	0.141	0.136	0.139	0.152	0.150	0.125
occlusion edge detection (L2 Loss)	0.134	0.138	0.132	0.133	0.137	0.141	0.128
2D keypoint estimation (L2 Loss)	0.176	0.171	0.167	0.163	0.169	0.168	0.158
3D keypoint estimation (L2 Loss)	0.194	0.205	0.199	0.196	0.204	0.201	0.191

Table 8: Results for Taskonomy dataset.

Tasks (Metric)	STL	MTL	PCGrad	GradVac	DANN	IRM	MT-CRL + $\mathcal{L}_{G\text{-}IRM}^{Var}$
Segmentation (mIoU)	13.27	17.64	19.64	19.68	17.12	17.54	19.81
Depth (Rel Err)	0.653	0.651	0.591	0.593	0.637	0.639	0.585
Surface Normal (Angle Distance)	35.18	31.52	30.98	31.04	31.69	32.03	30.85

Table 9: Results for NYU-V2 dataset.

Tasks (Metric)	STL	MTL	PCGrad	GradVac	DANN	IRM	MT-CRL + $\mathcal{L}_{G\text{-}IRM}^{Var}$
Segmentation (mIoU)	50.87	51.63	52.84	52.76	51.91	52.05	53.12
Depth (Rel Err)	33.85	32.75	32.12	32.08	32.71	32.64	31.86

Table 10: Results for CityScape dataset.

K	1	2	4	**8**	16	32	64	128
Acc.	0.824	0.897	0.904	**0.915**	0.911	0.902	0.893	0.882

Table 11: Hyperparameter tuning results for number of module (K) over Multi-MNIST dataset.

that every dimension is mutually independent ($K = 128$ in our case), which restricts the model capacity. Therefore, a middle K is a trade-off between model disentanglement and capacity. In all other datasets, we just use $K = 8$ by default and didn't do further tuning.

E Detailed Results on each Dataset

We report the performance on each task for the five benchmark datasets in Table 6-10. We use 8 GPU to run each experiments. As shown in the tables, the scale of different task's evaluation metric differ a lot, and thus in the main paper we adopt relateive performance improvement compared to vanilla MTL to evaluate each method. Nevertheless, our MT-CRL with invariance regularization could achieve the best results over nearly all the tasks.

F Determine the number of modules K

K is a hyperparameter that could be tuned. To control the same model complexity, we fix the output dimension d, i.e., 128, and then each module's dimension is $\frac{d}{K}$. We show the results on Multi-MNIST with different K in Table 11.

The results show that a middle number of module (K=8) achieves the best performance under the same size of model. In all other datasets, we just use $K = 8$ by default and didn't do further tuning to test our method's generalization capacity, this could avoid the situation that the final performance improvement is mainly caused by extensive hyper-parameter tuning.

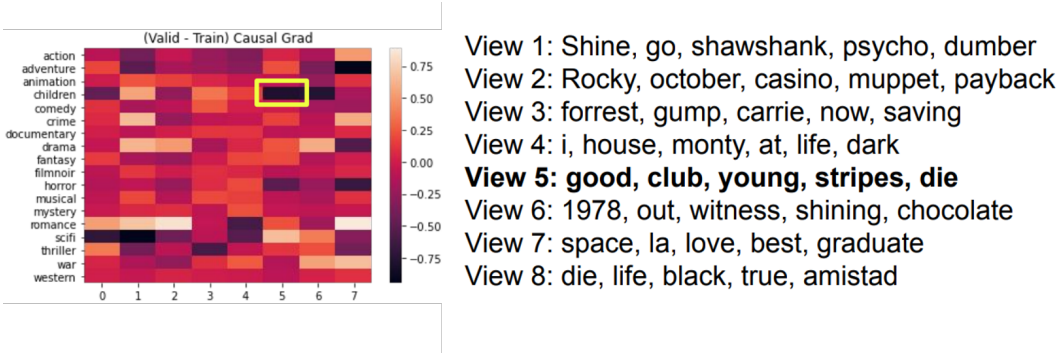

Figure 11: (valid-train) Task-to-Module gradients of model **without** MT-CRL on MovieLens.

Capacity-Disentanglement Tradeoff Noted that with a fixed number of dimension d, with larger K, the model capacity is reduced. The widely adopted disentangled representation learning methods like BetaVAE mostly assume that every dimension is mutually independent ($K = d$ in our case), which restricts the model capacity to extreme case. And the results in Table 3 also show that our current disentangled approach performs empirically better than BetaVAE. One potential reason is that we choose allow a middle K is a trade-off between model disentanglement and capacity, in which only the dimension across the block is de-correlated, why the ones within block could still correlated, so as to maintain model expressiveness.

Note that the optimal choice of K should ideally should be proportional to number of true generative factors that are related to downstream tasks. Therefore, for dataset with a large amount of tasks, we should choose larger number of K, and also consider increasing the total number of dimension d to increase the model capacity while maintaining disentanglement. In our paper for large dataset such as Taskonnomy we didn't do further tuning due to limited resources, so the performance could be potentially further improved, which we leave for future exploration.

G More Case Studies to show MT-CRL can alleviate spurious correlation

Here we show more details about the case study we conduct for analyzing how MT-CRL could alleviate spurious correlation, as a complementary.

As introduced in case study, we use the task-to-module gradients $\frac{\partial(f(\Phi(x))[y])}{\partial F}$ to illustrate how each task utilize each module. We could utilize the (valid-train) score to show which module is used by training set but not helpful for valid set, meaning it is spurious.

We first show detailed results on MovieLens dataset. As shown in Figure 11, without MT-CRL, there eixst many modules assigned negative (valid-train) causal grad, as shown in the color bar. Among them, view 5 is mostly inconsistent with the training results as we show in case study. Also, the other modules' key words are also not very accurate to describe the properties of each movie.

After we add MT-CRL, as shown in Figure 12, all of the modules receive positive (valid-train) causal grad, meaning that they either not utilized in training stage, or every used modules are still useful in valid stage. In addition, all the modules' key words are much more accurate to describe each type of movie than before.

In addition, our learned task biadjacency graph A could also be used to describe the similarity between task. If two tasks share more causal feature, they are more similar. We thus cauculate the task-averaged score of A, and plot a sparse smilarity graph in Figure 10. It shows that our MT-CRL could learn to group similar types of adult movies, such as war,

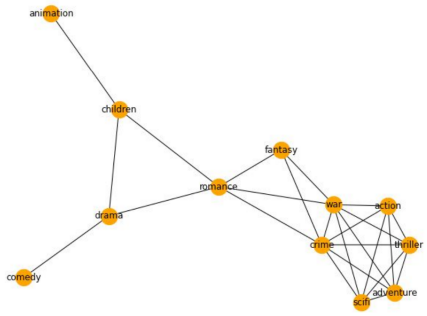

Figure 10: The task similarity induced by causal graph A for MovieLens dataset (threshold = 0.1).

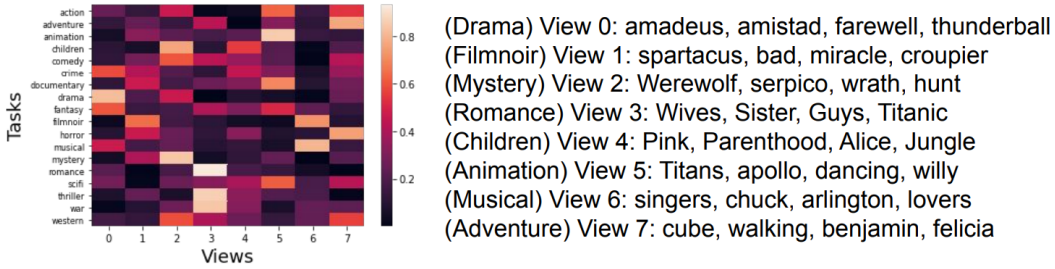

Figure 12: (valid-train) Task-to-Module gradients of model **with** MT-CRL on MovieLens.

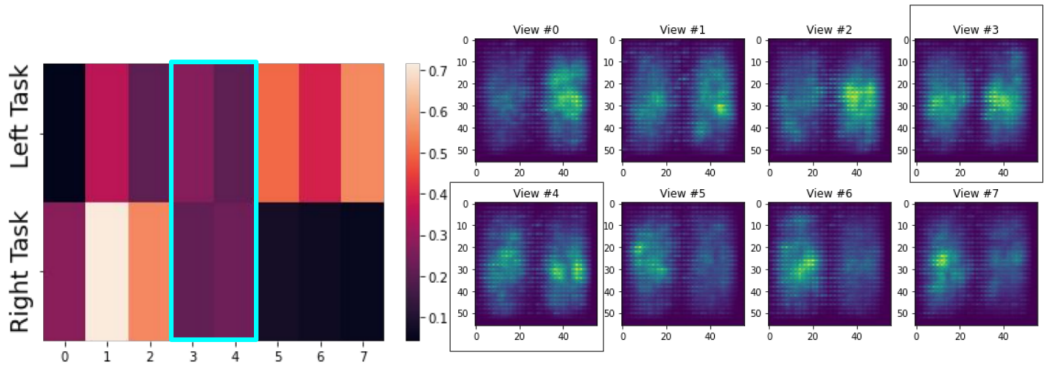

Figure 13: (valid-train) Task-to-Module gradients of model **without** MT-CRL on Multi-MNIST.

crime, thriller, adventure into the same group in the right down part, and link children-friendly movies, such as animation and children together. romance movie is a link between adult cluster and children movie. This similarity graph matches our human expectation, showing that our learned causal graph indeed help similar group use similar causal features.

We then show the (valid-train) Task-to-Module gradients over Multi-MNIST datasets. It is very apparent that the two digit classifier doesn't share any overlapping causal features. However, as shown in Figure 13, without MT-CRL, the model still learns to assign similar weights to module 3 and 4. This is also illustrated by the gradient saliency map for each module. Module 3 and 4 have high attention on both left and right side.

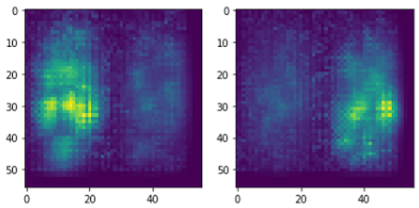

Figure 15: The gradient saliency map of left-digit and right-digit classifier trained via MT-CRL. Compared with Figure 3(b), MT-CRL indeed helps alleviate spurious correlation.

With MT-CRL, in Figure 14, the learned task-to-module assignment is much sparse and clear. Also each module's saliency map only focus on one side of pixels. By looking at each task output's saliency map, in Figure 15, we can see the model with MT-CRL can help to focus only on causal part, compared with Figure 3(b) that have high weights on both.

Both the MovieLens and Multi-MNIST case studies show that MT-CRL could help alleviate spurious correlaiton issue. For the other datasets, such as Taskonomy, NYUv2 and CityScape, their task output layer is very different and thus it's hard to get normalized gradient to show in one figure. In the future, we plan to do more thorough analysis by manually label several spurious feature or environmental groups, and design better methods to visualize how MTL model utilize non-causal features.

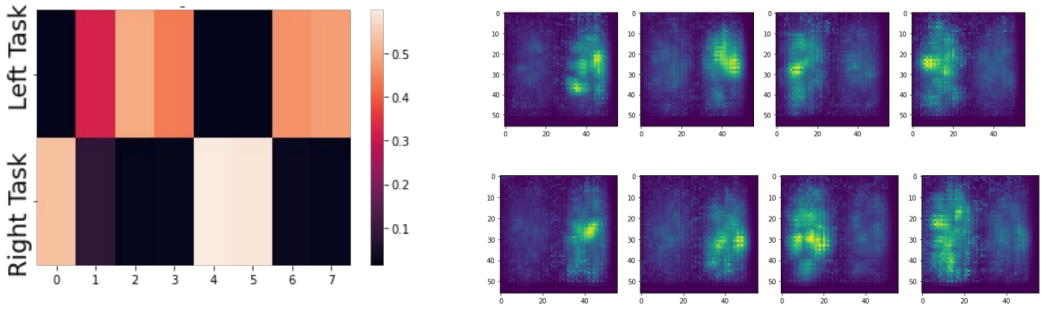

Figure 14: (valid-train) Task-to-Module gradients of model **with** MT-CRL on Multi-MNIST.

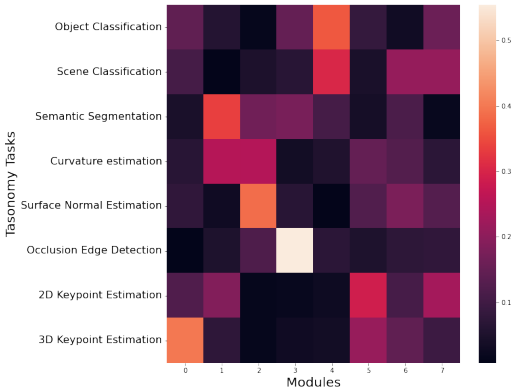

Figure 16: Task-to-Module Routing Graph (A) of model trained on Taskonomy dataset.

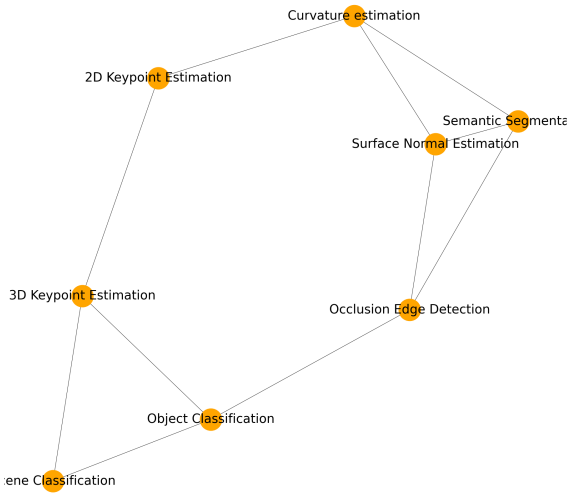

Figure 17: Task Similarity graph induced by Task-to-Module Graph A for Taskonomy dataset (threshold = 0.1).

Case Study on Taskonomy. In addition, we show the Task-to-Module routing graph and also the induced task similarity graph of Tasknomy dataset. As is shown in the figure, some similar task like 2D keypoint Estimation and 3D keypoint are liked together, and also the hard task like semantic segmentation receives information from curvature estimation, surface normal estimation and occlusion edge detection. These findings fit the observation of original Taskononmy analysis (Zamir et al., 2018). As stated in the limitation, we leave the deeper analysis in Tasknonmy about spurious feature as future work, as currently we don't have the ground-truth anotation about which part of image input is causally related to each task.

H Detailed Hyper-parameter Selection Procedure and Sensitivity Analysis

Here we introduce the procedure and results of hyper-parameter tuning.

Before discussing hyper-parameter selection, let me explain our baseline setup and experiment setting again. We have a single validation set, potentially bringing OOD to the training set. Our method only uses the training set to calculate the loss to update both encoder and per-task predictors. We then use the hold-out validation set to calculate the loss w.r.t graph weights A, and update it via invariance regularization. This avoids overfitting the validation set.

We mainly split the hyper-parameters into two sets:

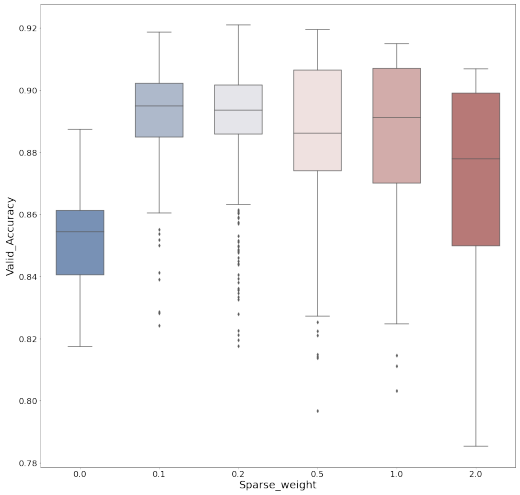

Figure 18: Hyper-parameter tuning results for sparse weight (λ_{sps}) on Multi-MNIST.

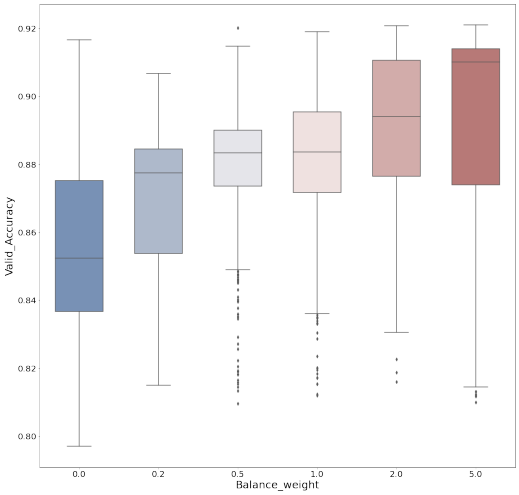

Figure 19: Hyper-parameter tuning results for balancing weight (λ_{bal}) on Multi-MNIST.

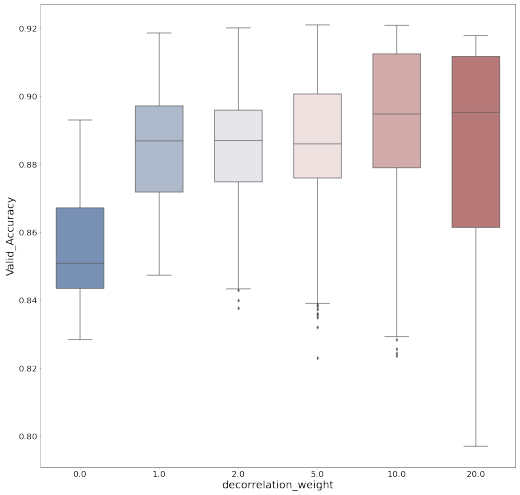

Figure 20: Hyper-parameter tuning results for disentanglement weight (λ_{decor}) on Multi-MNIST.

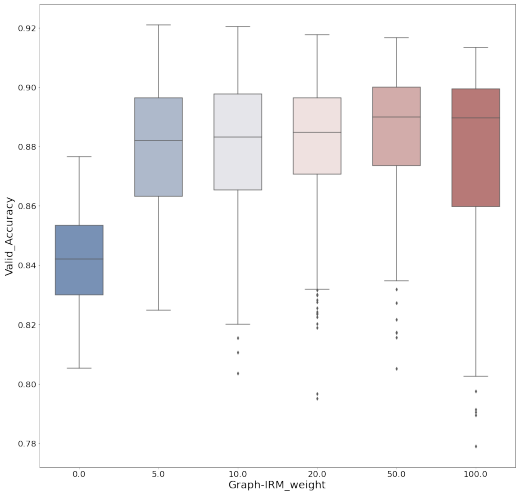

Figure 21: Hyper-parameter tuning results for correlation weight ($\lambda_{G\text{-}IRM^{Var}}$) on Multi-MNIST.

- General hyper-parameter related to all baselines (including ours), including number of hidden states, optimizer, learning rate, number of modules (K);
- Our MT-CRL specific hyper-parameter, including weights for disentanglement (λ_{decor}), sparsity (λ_{sps}), balance (λ_{bal}), and invariance (λ_{G-IRM}).

For both sets of hyper-parameters, we only tune on the same OOD validation set we used for our method. All hyper-parameters except K are standard hyperparameters for the MTL model. For datasets CityScape, NYUv2, and Taskonomy, we directly use the reported hyperparameter and dataset setting in previous papers (Yu et al., 2020; Balaji et al., 2020), in order to achieve a fair comparison. For MultiMNIST and MovieLens, we conduct a grid search for basic parameters, including the number of layers, number of hidden dimensions, optimizer, and learning rate, on the Vanilla MMoE MTL model without regularization. For the number of module (K), please refer to Sec F. After we determine these general hyper-parameters, we fix them and use them for all different MTL methods. This makes the comparison fair and ensures our performance improvement is not due to extensive hyper-parameter tuning of our method.

Methods	Multi-MNIST Accuracy	MovieLens MSE
MT-CRL with $\mathcal{L}_{G\text{-}IRM}^{Var}$ and **default** hyper-parameter	0.915 ± 0.018	0.884 ± 0.006
MT-CRL with $\mathcal{L}_{G\text{-}IRM}^{Var}$ and **randomly chosen** hyper-parameter	0.904 ± 0.021	0.887 ± 0.006
Vanilla MTL baseline	0.846 ± 0.018	0.892 ± 0.005

Table 12: Results on Multi-MNIST and MovieLens with a randomly chosen set of hyper-parameter.

Next, we tune the MT-CRL-specific hyper-parameters on the validation set. We think this setting is reasonable as we didn't utilize this validation set to calculate training loss. Thus it could still be regarded as a whole-out set for most model parameters (except the graph weights, which only take a tiny portion of the whole model). Note that four regularization weight terms exist to be tuned, which is many burdens for the model. Therefore, we only use Multi-MNIST, the smallest dataset in all our testbeds, to conduct hyperparameter tuning for the ML-CRL-specific hyperparameters with grid-search. This is definitely not the best choice, and tuning for each dataset could potentially improve our performance further, but that only makes our improvement higher while not changing the main conclusion of this paper. Specifically, we choose the several ranges for the four regularization weights:

- Sparse weight (λ_{sps}): $[0.0, 0.1, 0.2, 0.5, 1.0, 2.0]$
- Balancing weight (λ_{bal}): $[0.0, 0.2, 0.5, 1.0, 2.0, 5.0]$
- Disentanglement weight (λ_{decor}): $[0.0, 1.0, 2.0, 5.0, 10.0, 20.0]$
- Invariance weight ($\lambda_{G-IRM^{Var}}$): $[0.0, 5.0, 10.0, 20.0, 50.0, 100.0]$

These ranges are selected by running a few samples to determine the maximum value that should be within this range, and we keep each selection list to be a length of 6. We report the boxplot of detailed results for each regularization weights in Figure (18-21). As is illustrated, for all the regularization, using it is better than not using it ($\lambda = 0$), showing their advantage in making our MT-CRL pipeline works. We then select the optimal hyperparameter that achieves the highest validation accuracy, which is $\lambda_{sps} = 0.2, \lambda_{bal} = 5.0, \lambda_{decor} = 20, \lambda_{G-IRM} = 5.0$. Again, this selection might not be the optimal solution; for example, the tendency for λ_{bal} seems to increase with higher, and λ_{G-IRM} might have a better choice within the range $[0 - 5]$. However, we did not conduct more searching and used this setup. After getting such a combination of MT-CRL-specific hyper-parameter, we fix it and use it for all other larger datasets, which assume our framework with this hyper-parameter combination is consistently effective. Further tuning them on a dedicated dataset should potentially bring better performance, but we did not do it to avoid the performance improvement brought by extensive tuning.

Sensitivity Analysis. From the curve and also the definition of these regularization, we know that for all other terms except sparsity regularization λ_{sps}, increasing the regularization weight and strictly force model to be balance, de-correlated or invariant doesn't harm too much to the model training (trend didn't go down even with relatively large weight). The only exception is the sparsity regularization. With high λ_{sps} implemented as L_1 loss over adjacency weights will force all to be zero, which is very harmful to model training, which is why by default, we choose the value as $\lambda_{sps} = 0.2$.

To give a simple example of whether our model is sensitive to an inappropriate setting of hyper-parameter, we run experiment on MultiMNIST and MovieLens, with the following randomly chosen hyper-parameter setting: $\lambda_{sps} = 2.0, \lambda_{bal} = 1.0, \lambda_{decor} = 2.0, \lambda_{G-IRM^{Var}} = 100.0$. The results compared with the original results are as shown in Table 12.

Note that with a randomly chosen hyper-parameter, the results on the two datasets drop slightly but are still significantly higher than the Vanilla MTL baseline. This is an informal showcase of our method's generality and not very sensitive to hyper-parameter selection.