Generative multitask learning mitigates target-causing confounding

Anonymous Author(s)
Affiliation
Address
email

Abstract

We propose a simple and scalable approach to causal representation learning for multitask learning. Our approach requires minimal modification to existing ML systems, and improves robustness to target shift. The improvement comes from mitigating unobserved confounders that cause the targets, but not the input. We refer to them as target-causing confounders. These confounders induce spurious dependencies between the input and targets. This poses a problem for the conventional approach to multitask learning, due to its assumption that the targets are conditionally independent given the input. Our proposed approach takes into account the dependencies between the targets in order to alleviate target-causing confounding. All that is required in addition to usual practice is to estimate the joint distribution of the targets to switch from discriminative to generative classification, and to predict all targets jointly. Our results on the Attributes of People and Taskonomy datasets reflect the conceptual improvement in robustness to target shift.

1 Introduction

Deep neural networks (DNNs) excel at extracting patterns from unstructured data, but these patterns can fail to generalize outside of the training distribution. These failures are often attributed to DNNs learning spurious dependencies rather than causal relations [Ribeiro et al., 2016, Jo and Bengio, 2017, Geirhos et al., 2020]. Therefore, there is a concerted effort to make patterns learned by DNNs satisfy certain properties of causal relations, such as invariance and modularity [Schölkopf, 2019, Schölkopf et al., 2021]. This research direction is called causal representation learning.

Existing approaches to causal representation learning often require additional information in the form of labeled environments [Arjovsky et al., 2019, Lu et al., 2021] or labeled confounders [Puli et al., 2021]. These requirements are restrictive, and it is therefore beneficial to identify more natural settings where existing information can be used for causal representation learning.

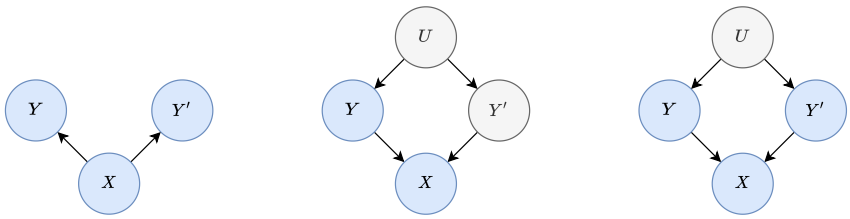
We consider the setting of multitask learning (MTL) [Caruana, 1997, Ruder, 2017, Zhang and Yang, 2017], where there is an input X and multiple targets. Without loss of generality, we consider two targets, where Y is the main target and Y' is the auxiliary target. Both targets are used for training, but only the main task is evaluated. The conventional approach to MTL, which we call discriminative multitask learning (DMTL), assumes the targets are conditionally independent given the input. This leads the inference objective to factorize, i.e.

$$\operatorname*{argmax}_{y,y'} \log p(y, y' \mid x) = \operatorname*{argmax}_{y,y'} \log p(y \mid x) + \log p(y' \mid x). \tag{1}$$

Since we only evaluate the main target Y, the inference objective is

$$\operatorname*{argmax}_{y} p(y \mid x). \tag{2}$$

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.



(a) DMTL (b) DMTL (alternative assumptions) (c) GMTL

Figure 1: (a) The conventional approach to MTL, which we call discriminative MTL (DMTL), assumes the targets Y and Y' are conditionally independent given the input X. Due to this assumption, the inference objective for predicting Y is $\operatorname{argmax}_y \log p(y \mid x)$. (b) DMTL is flawed under an alternative set of assumptions, where the targets cause the input, and there exists an unobserved confounder U that causes the targets. We call U a target-causing confounder. Since Y' is unobserved when predicting Y with DMTL, this opens up the backdoor path $Y \leftarrow U \rightarrow Y' \rightarrow X$ and makes the input and targets spuriously dependent. That is, $p(y \mid x)$ shifts when U is intervened on. (c) We propose GMTL, where the inference objective is $\operatorname{argmax}_{y,y'} \log p(x \mid y, y')$. Unlike DMTL, this objective conditions on all targets, which d-separates X and U, and removes the spurious dependencies between the input and targets.

DMTL does not make causal assumptions explicitly, but its conditional independence assumption is consistent with the causal graph in Fig. 1a. We argue that DMTL is flawed under an alternative set of assumptions.

First, we assume the targets cause the input, which implies we are predicting the cause from the effect. This is called anticausal learning, and is considered to describe many common problems such as image classification [Schölkopf et al., 2012]. For example, in object recognition, the object category causes the pixel representation of that object. Second, we assume there exists a variable U that causes the targets, but not the input. We call U a *target-causing confounder*.

The problem with DMTL is that the auxiliary target Y' is unobserved in Eq. 2. This is illustrated in Fig. 1b. This is problematic because it opens up the backdoor path $Y \leftarrow U \rightarrow Y' \rightarrow X$, and makes X and Y spuriously dependent. Since the backdoor path passes through U, DMTL is sensitive to intervention on U, which is a shift in $p(u)$. When $p(u)$ shifts, the relationship between X and Y changes, and DMTL fails to generalize. We refer to this issue as target-causing confounding.

We propose generative multitask learning (GMTL) as a way to mitigate target-causing confounding. GMTL is based on the following idea. If we write the causal factorization of the joint distribution of the observed variables as

$$p(x, y, y') = p(x \mid y, y')p(y, y') = p(x \mid y, y') \int p(y \mid u)p(y' \mid u)p(u)\mathrm{d}u, \tag{3}$$

this shows that $p(x \mid y, y')$ is invariant to intervention on U. Therefore, if we use

$$\operatorname*{argmax}_{y,y'} \log p(x \mid y, y') \tag{4}$$

for the inference objective, we become invariant to intervention on U. Importantly, unlike Eq. 1, this inference objective no longer factorizes over the targets. This means that even if we only care about evaluating the main target Y, we also need to consider Y'. This is the main idea behind GMTL - conditioning on all targets during inference closes the backdoor paths between the inputs and targets. This d-separates X and U, and removes the spurious dependencies between the input and targets.

Eq. 3 also shows that $p(y, y')$ changes in response to intervention on U. This means that being invariant to intervention on U is equivalent to being robust to shifts in $p(y, y')$. A shift in $p(y, y')$ is commonly referred to as target shift [Zhang et al., 2013]. While our goal is to be invariant to intervention on U, in the remainder of this work we will speak in terms of robustness to target shift. This is because shifts in $p(y, y')$ are measurable, unlike shifts in the unobservable $p(u)$.

In order to validate GMTL, we perform experiments on two datasets: Attributes of People [Bourdev et al., 2011] and Taskonomy [Zamir et al., 2018]. Across various pairs of tasks on these datasets,

our results show that GMTL improves robustness to target shift. We attribute this improvement to GMTL's ability to mitigate target-causing confounding. This empirical success is evidence that causal representation learning shows promise for MTL.

2 Background

2.1 Out-of-distribution generalization

Out-of-distribution (OOD) generalization refers to the capability of ML systems to generalize on data outside of the training distribution. Bengio et al. [2021] describe it as one of the greatest challenges in ML, and one that cannot be solved solely by scaling up datasets and computation. The failure of ML systems to generalize OOD is a general problem that has been reported across a wide range of problems spanning computer vision and natural language processing [D'Amour et al., 2020]. The core issue is this - when ML systems are trained to optimize an objective function, it is often easier for them to learn patterns that are dataset-specific, rather than those that hold universally. These dataset-specific patterns have many names, such as surface statistical regularities [Jo and Bengio, 2017], non-robust features [Tsipras et al., 2019], and shortcuts [Geirhos et al., 2020]. We take a causal perspective and adopt the term *spurious dependencies*, which are defined as statistical dependencies not supported by causal links [Pearl, 2009].

2.2 Dataset biases induce spurious dependencies

The following example shows that spurious dependencies can arise from design choices in dataset creation. Recht et al. [2019] demonstrated that state-of-the-art image classifiers trained on ImageNet [Deng et al., 2009] failed to generalize to a replicated test set designed to mirror the original test set distribution. Engstrom et al. [2020] offered an explanation for this phenomenon. The authors reported that the dataset replication procedure relied on estimating a human-in-the-loop metric called selection frequency, and that bias in estimating this statistic led to undesirable variation across datasets. Selection frequency is defined for an input-target pair, and measures the proportion of crowdsourced annotators who deem the pair correctly labeled. Recht et al. [2019] performed statistic matching on this metric, which corresponds to conditioning on it. Since selection frequency is caused by the input and target, this corresponds to conditioning on a collider, which induces a spurious dependency between the input and target. Such a dependency would fail to generalize OOD, which may explain why predictive performance degraded on the replicated test set.

In this work, we address a different source of dataset bias that we call target-causing confounding. A variable is a target-causing confounder if it is the common cause of the targets, but not the input. For a simple example, suppose we have images of people in indoor scenes, and the targets are clothing-related attributes such as hat and scarf. The season causes the targets, since people pair various types of clothing depending on the season. The season does not cause the image, since the images are of indoor scenes. Therefore, the season is a target-causing confounder that induces spurious dependencies between the input and targets. Relying on such dependencies for prediction can be problematic if the training and test sets represent different seasons. Although this is a contrived example for the purpose of illustration, dataset biases can occur in real-life settings like medicine, and have serious societal impacts [Mehrabi et al., 2021]. Therefore, mitigating such biases is an important goal for fair and equitable ML.

2.3 Causal representation learning

In Section 2.1, we made the distinction between patterns that are dataset-specific and those that hold universally. Causal inference formalizes the notion of patterns that hold universally, calling them causal relations [Pearl, 2009]. Causal representation learning aims to use ideas from causality to improve machine learning methods, particularly in the area of out-of-distribution generalization [Zhang et al., 2013, Schölkopf, 2019, Schölkopf et al., 2021, Arjovsky et al., 2019, Puli et al., 2021]. Our work falls under this category, since GMTL improves robustness to target shift by using an intervention distribution for prediction. This idea is also shared by Subbaswamy et al. [2019], who propose a general framework for estimating intervention distributions for robust prediction. GMTL is a practical method for achieving this in MTL on large-scale, high-dimensional datasets. Another causality-motivated paper for robust prediction is Makar et al. [2022], who use importance

reweighting to learn distributions that are invariant to intervention on an auxiliary target. Both Subbaswamy et al. [2019] and Makar et al. [2022] share the same causal graph as GMTL.

2.4 Multitask learning

MTL can reduce computational cost through parameter sharing, and improve predictive performance over training on each task individually [Standley et al., 2020]. Its use is widespread, spanning domains such as scene understanding [Chowdhuri et al., 2019, Zamir et al., 2018, 2020], medical diagnosis [Kyono et al., 2021], natural language processing [McCann et al., 2018, Wang et al., 2019, Radford et al., 2019, Liu et al., 2019, Worsham and Kalita, 2020, Aghajanyan et al., 2021], recommender systems [Covington et al., 2016, Ma et al., 2018], and reinforcement learning [Kalashnikov et al., 2021].

Distinguishing between main and auxiliary targets is common [Ruder, 2017, Liebel and Körner, 2018, Vafaeikia et al., 2020], but not universal, and is primarily important for concerns such as which loss to use for early stopping [Caruana, 1997]. We assume this distinction exists throughout this paper, denoting the main task as Y and the auxiliary task as Y'. However, our method still applies if we specify an arbitrary target to be the main target, and the others as being auxiliary.

3 Generative multitask learning

In GMTL, we condition on all targets in order to d-separate X and U, and remove spurious dependencies between the input and targets. During inference, we rely solely on $p(x \mid y, y')$, a distribution that is invariant to intervention on U. Approaching this naively by estimating $p(x \mid y, y')$ directly can be difficult, especially when X is high-dimensional. We can get around this by using Bayes rule to write

$$\begin{aligned}\operatorname*{argmax}_{y,y'} \log p(x \mid y, y') &= \operatorname*{argmax}_{y,y'} \log \frac{p(y, y' \mid x)p(x)}{p(y, y')} \\ &= \operatorname*{argmax}_{y,y'} \log p(y, y' \mid x) - \log p(y, y'). \end{aligned} \tag{5}$$

This shows that the argmax over the targets results in the difficult-to-estimate $p(x)$ term to drop out, leaving us with $p(y, y' \mid x)$ and $p(y, y')$. In other words, to implement GMTL, we must estimate $p(y, y' \mid x)$ as we have been doing with DMTL, and additionally estimate $p(y, y')$.

3.1 Interpolating between DMTL and GMTL

In practice, we replace the inference objective of GMTL in Eq. 5 with the following approximation:

$$\operatorname*{argmax}_{y,y'} \log p(y \mid x) + \log p(y' \mid x) - \alpha \log p(y, y'), \qquad \alpha \in [0, 1]. \tag{6}$$

This introduces two changes. The first is that we allow $p(y, y' \mid x)$ to factorize, as was the case in DMTL. This is purely out of convenience, and lets us keep the training of multitask networks the same as DMTL. This is beneficial since multitask networks are notoriously difficult to train [Standley et al., 2020]. We can allow this factorization as long as we capture the dependency between the targets when estimating $p(y, y')$. This prevents Eq. 6 from factorizing over the targets, and satisfies our requirement of conditioning on all targets in order to remove spurious dependencies between the input and targets.

The second change is the parameter $\alpha \in [0, 1]$, which enables us to interpolate between DMTL ($\alpha = 0$) and the most extreme case of GMTL ($\alpha = 1$). α controls the degree to which we remove spurious dependencies between the input and targets. The ability to adjust this effect is important because if the training and test distributions are identical, then we should be willing to use any dependencies available, regardless of whether they are spurious or causal. However, it is more realistic to assume that training and test distributions will be different, and under these circumstances, setting $\alpha > 0$ can improve robustness to target shift.

3.2 Interpreting α

The parameter α represents our uncertainty about the test target distribution. In the case of complete certainty, that is, if we know that the test target distribution is identical to the empirical target

distribution, we set $\alpha = 0$ to use the empirical distribution. In contrast, in the case of complete uncertainty, we set $\alpha = 1$ to completely remove the influence of the empirical distribution. This is equivalent to assuming the test target distribution is uniform. Generally, it can be shown that GMTL corresponds to replacing the empirical target distribution $p(y, y')$ with a distribution proportional to $p(y, y')^{1-\alpha}$. The derivation is in the supplementary materials. We perform this replacement during inference in order to keep training the same as DMTL. In principle, we could achieve the same invariance by reweighting examples by $1/p(y, y')^{\alpha}$ during training [Zhang et al., 2013, Makar et al., 2022].

This raises an important point - α represents an assumption, and is not a hyperparameter that can be tuned in the conventional sense. Instead, tuning α is analogous to model selection w.r.t. an unknown distribution, which is a difficult open problem [Gulrajani and Lopez-Paz, 2021]. Progress in this direction is likely to benefit GMTL, as well as other methods with similar parameters [Wortsman et al., 2022]. In the remainder of this work, we assume oracle access to the optimal α for a given OOD test distribution. That is, our main focus is not on the difficulty of tuning α, which is a separate research question. Instead, we study the efficacy of GMTL assuming we have correctly assessed how similar the training and test target distributions are, i.e. knowing the optimal α.

Having said that, we provide a simple and effective heuristic for setting α even for an unknown test distribution, which makes GMTL highly practical. This is based on the fact that α represents a trade-off. Increasing α improves target shift robustness at the expense of IID predictive performance. We show empirically in Section 6 that this relationship is very strong, and holds across two datasets, multiple pairs of tasks, and three MTL methods. Therefore, our heuristic is to set α to the maximum value within an acceptable loss in IID predictive performance, where the latter is measured using the validation set. This ensures that we are as robust as possible to target shift, while keeping IID predictive performance at an acceptable level. These results are in the supplementary materials.

If we do have access to the test target distribution $q(y, y')$, then we can do away with α and simply replace $p(y, y')$ with $q(y, y')$, which results in the following inference objective:

$$\operatorname*{argmax}_{y,y'} \log p(y \mid x) + \log p(y' \mid x) - \log p(y, y') + \log q(y, y').$$

Alternatively, we can use $q(y, y')/p(y, y')$ to correct for target shift during training [Zhang et al., 2013].

3.3 Estimating $p(y, y')$

GMTL requires estimating $p(y, y')$, which is trivial for categorical targets. The maximum likelihood estimate of $p(y, y')$ is obtained by counting the number of occurrences of the pair (y, y'), and dividing by the total across all pairs. However, there is an important precaution to be made, since it is possible that certain pairs of (y, y') are never observed. This is problematic, since $p(y, y') = 0$ results in taking a logarithm of zero in Eq. 6. The issue of sparsity in the target distribution is the primary challenge in scaling GMTL across tasks. This can be addressed with additive smoothing, where a pseudocount $\epsilon > 0$ is added to the count for each pair prior to normalization.

4 An illustrative example

We construct a synthetic example where a target-causing confounder U induces a spurious dependency between X and Y. We illustrate how DMTL is sensitive to this dependency, while GMTL is invariant to it. Suppose there are two targets $Y, Y' \in \{0, 1\}$ that cause the input $X \in \mathbb{R}$. Let us assume that $p(y, y')$ is bivariate Bernoulli, where $P(Y = 1) = \theta$, $P(Y' = 1) = \theta'$, and $\mathrm{Cov}[Y, Y'] = \sigma_{Y,Y'}$. X is generated by a mixture of Gaussians with latent variable $Z = 2Y + Y'$, i.e. $X \mid z \sim N(x; z, \sigma_z^2)$.

The value of $\sigma_{Y,Y'}$ is set by an unobserved variable U. U is a target-causing confounder, since it causes Y and Y', but not X. We fix $\theta = \theta' = 0.5$, $\sigma^2_{Z=0} = \sigma^2_{Z=1} = 0.4$, and $\sigma^2_{Z=2} = \sigma^2_{Z=3} = 0.6$, and interpolate $\sigma_{Y,Y'}$ from -0.2 to 0.2. In other words, the only property of the data generating process that changes is the covariance between Y and Y'.

Fig. 2 shows how the decision boundaries of DMTL and GMTL with $\alpha = 1$ are affected by changes in $\sigma_{Y,Y'}$. The decision boundary of DMTL is the value of x for which the solution to $\mathrm{argmax}_y\, p(y \mid x)$ changes, where Y' has been marginalized out. In contrast, the decision boundary of GMTL is the

value of x where the Y component of the solution to $\text{argmax}_{y,y'}\, p(x \mid y, y')$ changes. The decision boundary for DMTL changes in response to $\sigma_{Y,Y'}$, while it remains constant for GMTL. This is because GMTL only uses the Gaussian densities for prediction, which have no connection to $\sigma_{Y,Y'}$.

In this example, we have shown that GMTL and DMTL arrive at different solutions by changing only the dependency between Y and Y' induced by the target-causing confounder U. The key difference between the two methods is that DMTL uses the spurious dependency between X and Y, while GMTL does not. Which of the two methods achieves better predictive performance depends on the particular test distribution, since the spurious dependency will be predictive in IID settings, and not in OOD settings.

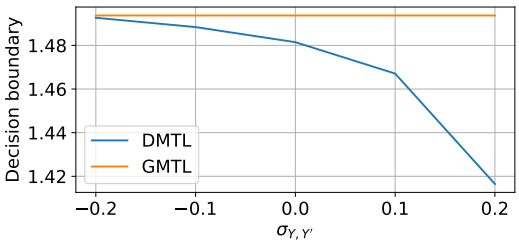


Figure 2: Here we show how the decision boundaries of GMTL and DMTL change w.r.t. $\sigma_{Y,Y'}$. DMTL is sensitive to changes in $\sigma_{Y,Y'}$, while GMTL is invariant.

5 Experimental setup

We perform experiments to demonstrate that GMTL improves robustness to target shift. We evaluate across a wide range of target shifts ranging from mild to severe, and analyze how the optimal α changes w.r.t. the severity of target shift. Recall from Section 3.2 that we assume oracle access to the optimal α, since we are focusing on validating GMTL, not on whether we can perform model selection w.r.t. unknown distributions, which is a separate open problem. The purpose of these experiments is to show that if we correctly assess how similar the training and test distributions are, we can benefit from using GMTL. In order to draw robust conclusions, we perform our experiments using two datasets, multiple pairs of classification tasks, and three MTL methods. Here we describe our experimental setup.

5.1 Datasets and tasks

Attributes of People The Attributes of People dataset [Bourdev et al., 2011] consists of 8,035 images of people. There are 4,013 examples in the training set, and 4,022 in the test set. Since the authors did not specify a validation set, we use 20% of the training set as the validation set. Each image comes labeled with up to nine binary attributes: male, long hair, glasses, hat, t-shirt, long sleeves, shorts, jeans, and long pants. We use each attribute as a binary classification task and predict whether the attribute appears in the image. We drop the gender attribute, since it has the potential for negative downstream applications [Wang and Kosinski, 2018]. We experiment with various pairs of tasks, taking turns specifying one as the main task, and the other as the auxiliary task. For each pair of tasks, we report the classification accuracy for the main task. Since not all attributes are labeled for each example, we consider the three pairs of tasks with the highest number of co-occurrences. These are: hat and long sleeves, long hair and hat, and glasses and hat.

Taskonomy The Taskonomy dataset [Zamir et al., 2018] is of a much larger scale than the first dataset, containing approximately four million images of indoor scenes. We use a subset of the dataset that is provided by the authors for faster experimentation, with 548,785 examples in the training set, 121,974 in the validation set, and 90,658 in the test set. Each image comes paired with twenty six labels relevant to scene understanding, but since we are focusing on classification, we only use the object and scene annotations as targets. We use the object annotations as a 100-way object classification task, and the scene annotations as a 64-way scene classification task. Between object and scene classification, we take turns specifying one as the main task, and the other as the auxiliary task. We report the top-1 accuracy for the main task.

5.2 MTL methods

No Parameter Sharing (NPS) The first MTL method we use is a trivial combination of the two trained STL networks. We call this No Parameter Sharing (NPS). This is equivalent to STL networks during training, but becomes different during prediction if $\alpha > 0$, since the STL networks influence one another through the $-\alpha \log p(y, y')$ term. While NPS is primarily meant to be a simple setting for comparing GMTL and DMTL, it offers some practical utility as well. MTL methods are notoriously difficult to train, and it is often challenging to get them to perform better than the STL baseline [Alonso and Plank, 2017]. NPS combined with GMTL offers a way to combine two STL networks to reap the benefits of MTL, just by estimating $p(y, y')$ and predicting jointly over the targets.

Shared Trunk Networks (STN) The second MTL method is a hard parameter sharing method that we call Shared Trunk Networks (STN), where all weights except for the final classification layer are shared. This is not an effective architecture for many of the tasks, since the predictive performance is worse than the single-task baseline. This is common for STN [Alonso and Plank, 2017], but we include it for completeness because it is the canonical MTL architecture. The results comparing STL and MTL methods on the IID test set are in the supplementary material.

Cross-stitch Networks (CSN) For the third MTL method, we use a soft parameter sharing method called Cross-stitch Networks (CSN) [Misra et al., 2016]. CSN takes two trained STL networks, and takes a linear combination of the activations at each layer during the forward pass. Training CSN involves learning the coefficients of the linear combinations, as well as fine-tuning the STL network weights. This network performs better than the STL baseline across most datasets and tasks.

Full Parameter Sharing (FPS) For our experiments on Attributes of People, we model the joint target space directly as a four-class classification. Since this is a single network with all weights shared across both tasks, we call this Full Parameter Sharing (FPS). We omit FPS on Taskonomy due to the complications of a large and sparse joint target space, and leave this to future work.

5.3 Methodology for training and prediction

For all datasets and tasks, we use ResNet-50 [He et al., 2016] pretrained on ImageNet [Deng et al., 2009] for the STL networks. We train using Adam [Kingma and Ba, 2015] with L_2 regularization. We tune the learning rate and regularization multiplier. For data augmentation, during training we resize the images to 256×256, randomly crop them to 224×224 and randomly horizontally flip them. During validation and testing, we resize the images to 256×256, and center crop them to 224×224. For all experiments, we train STL networks from five random initializations. Details regarding other hyperparameters are included in the supplementary materials.

5.4 Simulating target shift with importance sampling

In these experiments, we wish to evaluate across a wide range of target shifts. Since it is impractical to collect many different test sets, we simulate target shifts using importance sampling. Suppose that the original test distribution is $p(x, y, y') = p(x \mid y, y')p(y, y')$, and the simulated test distribution is $q(x, y, y') = p(x \mid y, y')q(y, y')$. $p(x \mid y, y')$ is invariant due to our assumption that it is a causal relation. The expected accuracy $l(x, y, y')$ under the simulated test distribution $q(x, y, y')$ is given by

$$
\begin{aligned}
&\mathbb{E}_{q(x,y,y')}[l(x, y, y')] = \mathbb{E}_{p(x|y,y')q(y,y')}[l(x, y, y')] \\
&= \mathbb{E}_{p(x,y,y')}\left[\frac{q(y, y')}{p(y, y')} l(x, y, y')\right] \\
&\approx \frac{1}{N}\sum_{n=1}^{N} \frac{q(y^{(n)}, y'^{(n)})}{p(y^{(n)}, y'^{(n)})} l(x^{(n)}, y^{(n)}, y'^{(n)}) \\
&\approx \sum_{n=1}^{N} \frac{q(y^{(n)}, y'^{(n)})}{p(y^{(n)}, y'^{(n)})} \Big/ \sum_{m=1}^{N} \frac{q(y^{(m)}, y'^{(m)})}{p(y^{(m)}, y'^{(m)})} l(x^{(n)}, y^{(n)}, y'^{(n)}).
\end{aligned}
$$

Now that we can evaluate across a wide range of OOD $q(y, y')$'s, we need a way to quantify the severity of the target shift between $p(y, y')$ and $q(y, y')$.

5.5 Measuring the severity of target shift between $p(y, y')$ and $q(y, y')$

Our metric to measure the severity of target shift between $p(y, y')$ and $q(y, y')$ must be such that predictive performance degrades w.r.t. the severity of shift. Not all metrics satisfy this criteria. To build intuition, consider a single task binary classification problem where $P(Y = 1) = \theta$. If we use a norm-based metric such as total variation, this treats a change in θ from 0.6 to 0.8 and a change from 0.6 to 0.4 as being the same. However, from the point of view of prediction, the latter should be more detrimental, since it reverses the ranking of the classes.

Similarly, Kullback-Keibler (KL) divergence is a common choice for measuring the distance between distributions, but it suffers from a related problem. Consider θ changing from 0.9 to 0.9999, and from 0.9 to 0.4. Due to taking the logarithm of a small number, KL divergence considers the first to be a more severe shift. This is undesirable, because the change from 0.9 to 0.4 represents a reversal in the ranking of the classes.

These examples point to a need for a ranking-based metric, since a significant change to the ranking of classes is detrimental to predictive performance. Weighted rank correlation satisfies this. If the weighted rank correlation between $p(y, y')$ and $q(y, y')$ is positive with large magnitude, it means that there are no significant changes in the ranking of classes, and that the target shift is not severe. In contrast, if the weighted rank correlation is negative with large magnitude, it implies the rankings of classes has changed significantly. This constitutes a more severe target shift. We therefore use weighted Kendall's τ [Shieh, 1998], which we henceforth refer to as τ, to measure the severity of target shift between $p(y, y')$ and $q(y, y')$. We provide the definition of τ in the supplementary materials. It relies on a weighting function, for which we use $1/(r + 1)$, where r is the ranking.

5.6 Sampling $q(y, y')$

Our choice of distance metric informs how to sample $q(y, y')$, since our goal is to evaluate across a wide range of target shift severities. We designed a method which shuffles $\log p(y, y')$ and perturbs it with noise. To be precise, let $f(y, y') \in \{1, 2, \ldots, |\mathcal{Y} \times \mathcal{Y}'|\}$ such that each (y, y') is assigned a unique integer index. We use $p_{f(y,y')} = p(y, y')$ Let $i = \alpha \cdot |\mathcal{Y} \times \mathcal{Y}'|$, where $\alpha \sim \mathcal{U}(0, 0.5)$. We randomly shuffle these indices with $\pi : \{1, 2, \ldots, i\} \to \{1, 2, \ldots, \alpha \times i\}$. then, we define $q(y, y')$ by

$$q(y, y') \propto \begin{cases} \exp(\log p(y, y') + \epsilon) & \text{if } f(y, y') > i \\ \exp(\log p_{\pi(f(y,y'))} + \epsilon) & \text{if } f(y, y') \leq i, \end{cases}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and $\sigma \sim \mathcal{U}(10^{-12}, 5)$.

6 Results

We compute the OOD test set accuracy across target shifts ranging from mild to severe, where $p(y, y')$ is the original test set distribution, and $q(y, y')$ a randomly sampled OOD distribution. First, we sample an OOD $q(y, y')$, and compute τ relative to $p(y, y')$. τ represents the severity of target shift between $q(y, y')$ and the original test distribution $p(y, y')$. We then use importance sampling to compute the OOD accuracy under each $q(y, y')$ across a range of $\alpha \in [0, 1]$.

The results are split into five groups that are equally spaced in terms of τ. Each group represents the accuracies for a range of α under a particular severity of target shift. For each group, we compute the optimal value of α that attains the highest accuracy. We also compute the difference in accuracy between GMTL and DMTL at the optimal value of α for each group.

We observe strikingly similar results for both datasets and all pairs of tasks. As the target shift becomes more severe, the optimal α increases. A representative example for each dataset is shown in the left subfigures in Fig. 3, and the rest are in the supplementary materials. This observation matches our understanding of α from Section 3.2 - since α is our degree of belief on how similar $p(y, y')$ and $q(y, y')$ are, the optimal α tends to be higher when the target shift is more severe. Increasing α removes more of the spurious dependencies induced by target-causing confounding, which is more beneficial for severe target shifts.

Also, as the severity of target shift increases, the difference in accuracy between GMTL and DMTL for the optimal α increases as well. This can be seen in the right subfigures in Fig. 3, and is also

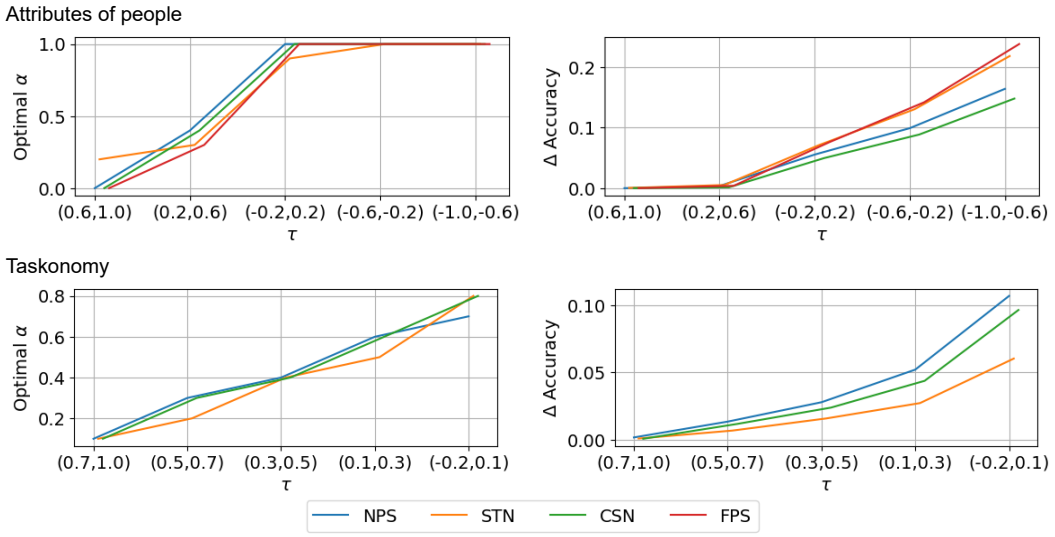


Figure 3: **Comparing the robustness of GMTL and DMTL to target shift.** We compute the OOD accuracy across a range of target shift severities and a range of $\alpha \in [0, 1]$. The results are grouped w.r.t. the severity of target shift τ, and we report the optimal α for each group. As seen in the left subfigures, as the target shift becomes more severe from left to right, the optimal α monotonically increases for both Attributes of People (top left) and Taskonomy (bottom left). This is because larger α removes more spurious dependencies induced by target-causing confounding, which is more beneficial when the shift severity increases. In the right subfigures, the vertical axis is the difference in accuracy between GMTL and DMTL. The gain in accuracy from using GMTL increases w.r.t. the severity of target shift for both Attributes of People (top right) and Taskonomy (bottom right). The reason is that the spurious dependencies induced by target-causing confounding become less predictive as the shift severity increases, and therefore removing them yields larger improvements in accuracy.

in-line with our expectations. As the target shift becomes more severe, the spurious dependencies induced by target-causing confounding become less predictive, and removing them yields larger gains in predictive performance.

Both patterns hold very similarly for all MTL methods. This suggests that it is our formulation of GMTL that is significant, rather than the particular parameterization used to learn $p(y, y' \mid x)$. Both the optimal α, as well as the corresponding gain in accuracy using GMTL, increase monotonically w.r.t. the severity of target shift. This is a strong relation that holds across datasets, pairs of tasks, and MTL methods. To quantify the generality of this result, we aggregate all of our results, and compute the correlation between τ and the optimal α. Since τ is used in the context of an interval, we take the midpoint of each interval. This results in a correlation of -0.847. This indicates a strong positive correlation between the severity of target shift and the optimal α. The generality of this relation is strong evidence that GMTL improves robustness to target-causing confounding, and also that our interpretation of α is correct. Our experimental results using the α-selection heuristic described in Section 3.2, as well as other results showing the accuracy for a range of α, not just the optimal one, are in the supplementary materials.

7 Conclusion

We presented generative multitask learning (GMTL), which is an approach for causal representation learning for MTL that requires minimal changes to existing ML systems. Our approach mitigates the effect of target-causing confounders, which are variables that cause the targets, but not the input. This removes spurious dependencies between the input and targets, and improves robustness to target shift. Looking forward, we are excited by the new perspectives that causal representation learning can bring to existing ML paradigms. Our work demonstrates the potential for this in the context of MTL. We plan to investigate what other areas of ML can be improved by adopting a causal perspective.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. Muppet: Massive multi-task representations with pre-finetuning. In *EMNLP*, 2021.

Héctor Martínez Alonso and Barbara Plank. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *EACL*, 2017.

Martín Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv*, abs/1907.02893, 2019.

Yoshua Bengio, Yann LeCun, and Geoffrey E. Hinton. Deep learning for AI. *Commun. ACM*, 64(7): 58–65, 2021.

Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Describing people: A poselet-based approach to attribute classification. In *ICCV*, 2011.

Rich Caruana. *Multitask learning*. PhD thesis, Carnegie Mellon University, 1997.

Sauhaarda Chowdhuri, Tushar Pankaj, and Karl Zipser. Multinet: Multi-modal multi-task learning for autonomous driving. In *WACV*, 2019.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *RecSys*, 2016.

Alexander D'Amour, Katherine A. Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D. Hoffman, Farhad Hormozdiari, Neil Houlsby, Shaobo Hou, Ghassen Jerfel, Alan Karthikesalingam, Mario Lucic, Yi-An Ma, Cory Y. McLean, Diana Mincu, Akinori Mitani, Andrea Montanari, Zachary Nado, Vivek Natarajan, Christopher Nielson, Thomas F. Osborne, Rajiv Raman, Kim Ramasamy, Rory Sayres, Jessica Schrouff, Martin Seneviratne, Shannon Sequeira, Harini Suresh, Victor Veitch, Max Vladymyrov, Xuezhi Wang, Kellie Webster, Steve Yadlowsky, Taedong Yun, Xiaohua Zhai, and D. Sculley. Underspecification presents challenges for credibility in modern machine learning. *arXiv*, abs/2011.03395, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Jacob Steinhardt, and Aleksander Madry. Identifying statistical bias in dataset replication. In *ICML*, 2020.

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

Ishaan Gulrajani and David Lopez-Paz. In search of lost domain generalization. In *ICLR*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv*, abs/1711.11561, 2017.

Dmitry Kalashnikov, Jacob Varley, Yevgen Chebotar, Benjamin Swanson, Rico Jonschkowski, Chelsea Finn, Sergey Levine, and Karol Hausman. Mt-opt: Continuous multi-task robotic reinforcement learning at scale. *arXiv*, abs/2104.08212, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Trent Kyono, Fiona J Gilbert, and Mihaela Van Der Schaar. Triage of 2d mammographic images using multi-view multi-task convolutional neural networks. *ACM Transactions on Computing for Healthcare*, 2(3):1–24, 2021.

Lukas Liebel and Marco Körner. Auxiliary tasks in multi-task learning. *arXiv*, abs/1805.06334, 2018.

Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. Multi-task deep neural networks for natural language understanding. In *ACL*, 2019.

Chaochao Lu, Yuhuai Wu, José Miguel Hernández-Lobato, and Bernhard Schölkopf. Nonlinear invariant risk minimization: A causal approach. *arXiv*, abs/2102.12353, 2021.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD*, 2018.

Maggie Makar, Ben Packer, Dan Moldovan, Davis Blalock, Yoni Halpern, and Alexander D'Amour. Causally motivated shortcut removal using auxiliary labels. In *AISTATS*, 2022.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. The natural language decathlon: Multitask learning as question answering. *arXiv*, abs/1806.08730, 2018.

Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. A survey on bias and fairness in machine learning. *ACM Comput. Surv.*, 54(6):115:1–115:35, 2021.

Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

Judea Pearl. *Causality*. Cambridge university press, 2009.

Aahlad Manas Puli, Lily H. Zhang, Eric K. Oermann, and Rajesh Ranganath. Predictive modeling in the presence of nuisance-induced spurious correlations. *arXiv*, abs/2107.00520, 2021.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *ICML*, 2019.

Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should I trust you?": Explaining the predictions of any classifier. In *SIGKDD*, 2016.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv*, abs/1706.05098, 2017.

Bernhard Schölkopf. Causality for machine learning. *arXiv*, abs/1911.10500, 2019.

Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris M. Mooij. On causal and anticausal learning. In *ICML*, 2012.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Towards causal representation learning. *arXiv*, abs/2102.11107, 2021.

Grace S Shieh. A weighted kendall's tau statistic. *Statistics & probability letters*, 39(1):17–24, 1998.

Trevor Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In *ICML*, 2020.

Adarsh Subbaswamy, Peter Schulam, and Suchi Saria. Preventing failures due to dataset shift: Learning predictive models that transport. In *AISTATS*, 2019.

Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *ICLR*, 2019.

Partoo Vafaeikia, Khashayar Namdar, and Farzad Khalvati. A brief review of deep multi-task learning and auxiliary task learning. *arXiv*, abs/2007.01126, 2020.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *ICLR*, 2019.

Yilun Wang and Michal Kosinski. Deep neural networks are more accurate than humans at detecting sexual orientation from facial images. *Journal of personality and social psychology*, 114(2):246, 2018.

Joseph Worsham and Jugal Kalita. Multi-task learning for natural language processing in the 2020s: where are we going? *Pattern Recognition Letters*, 136:120–126, 2020.

Mitchell Wortsman, Gabriel Ilharco, Mike Li, Jong Wook Kim, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *CVPR*, 2022.

Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *CVPR*, 2018.

Amir Roshan Zamir, Alexander Sax, Nikhil Cheerla, Rohan Suri, Zhangjie Cao, Jitendra Malik, and Leonidas J. Guibas. Robust learning through cross-task consistency. In *CVPR*, 2020.

Kun Zhang, Bernhard Schölkopf, Krikamol Muandet, and Zhikun Wang. Domain adaptation under target and conditional shift. In *ICML*, 2013.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *arXiv*, abs/1707.08114, 2017.

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 3.2
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 2.2
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/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] See supplementary materials
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 5 and the supplementary materials
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] Our main results in Section 6 do not allow error bars, but they represent the combined results of five random seeds. Our results in the supplementary materials do have error bars w.r.t. the random seeds.
   (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] See supplementary materials
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] See Section 2.2, we used two public datasets: Attributes of People and Taskonomy.
   (b) Did you mention the license of the assets? [Yes] We mention the license of the two public datasets that we used in the supplementary materials.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We include our code in the supplementary materials.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] No need to, the datasets are public.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See Section 5, the Attributes of People dataset has a gender label that we drop, since predicting gender can have negative consequences.

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]