IN-N-OUT: PRE-TRAINING AND SELF-TRAINING USING AUXILIARY INFORMATION FOR OUT-OF-DISTRIBUTION ROBUSTNESS

Sang Michael Xie[*], **Ananya Kumar**[*], **Robbie Jones**[*], **Fereshte Khani, Tengyu Ma, Percy Liang**
Stanford University
{xie, ananya, rmjones, fereshte, tengyuma, pliang}@cs.stanford.edu

ABSTRACT

Consider a prediction setting with few in-distribution labeled examples and many unlabeled examples both in- and out-of-distribution (OOD). The goal is to learn a model which performs well both in-distribution and OOD. In these settings, auxiliary information is often cheaply available for every input. How should we best leverage this auxiliary information for the prediction task? Empirically across three image and time-series datasets, and theoretically in a multi-task linear regression setting, we show that (i) using auxiliary information as input features improves in-distribution error but can hurt OOD error; but (ii) using auxiliary information as outputs of auxiliary pre-training tasks improves OOD error. To get the best of both worlds, we introduce In-N-Out, which first trains a model with auxiliary inputs and uses it to pseudolabel all the in-distribution inputs, then pre-trains a model on OOD auxiliary outputs and fine-tunes this model with the pseudolabels (self-training). We show both theoretically and empirically that In-N-Out outperforms auxiliary inputs or outputs alone on both in-distribution and OOD error.

1 INTRODUCTION

When models are tested on distributions that are different from the training distribution, they typically suffer large drops in performance (Blitzer and Pereira, 2007; Szegedy et al., 2014; Jia and Liang, 2017; AlBadawy et al., 2018; Hendrycks et al., 2019a). For example, in remote sensing, central tasks include predicting poverty, crop type, and land cover from satellite imagery for downstream humanitarian, policy, and environmental applications (Xie et al., 2016; Jean et al., 2016; Wang et al., 2020; Rußwurm et al., 2020). In some developing African countries, labels are scarce due to the lack of economic resources to deploy human workers to conduct expensive surveys (Jean et al., 2016). To make accurate predictions in these countries, we must extrapolate to out-of-distribution (OOD) examples across different geographic terrains and political borders.

We consider a semi-supervised setting with few in-distribution labeled examples and many unlabeled examples from both in- and out-of-distribution (e.g., global satellite imagery). While labels are scarce, auxiliary information is often cheaply available for every input and may provide some signal for the missing labels. Auxiliary information can come from additional data sources (e.g., climate data from other satellites) or derived from the original input (e.g., background or non-visible spectrum image channels). This auxiliary information is often discarded or not leveraged, and how to best use them is unclear. One way is to use them directly as input features (**aux-inputs**); another is to treat them as prediction outputs for an auxiliary task (**aux-outputs**) in pre-training. Which approach leads to better in-distribution or OOD performance?

Aux-inputs provide more features to potentially improve in-distribution performance, and one may hope that this also improves OOD performance. Indeed, previous results on standard datasets show that improvements in in-distribution accuracy correlate with improvements in OOD accuracy (Recht et al., 2019; Taori et al., 2020; Xie et al., 2020; Santurkar et al., 2020). However, in this paper we find that aux-inputs can introduce more spurious correlations with the labels: as a result, while aux-inputs often improve in-distribution accuracy, they can worsen OOD accuracy. We give examples of this trend on CelebA (Liu et al., 2015) and real-world satellite datasets in Sections 5.2 and 5.3.

Conversely, aux-output methods such as pre-training may improve OOD performance through auxiliary supervision (Caruana, 1997; Weiss et al., 2016; Hendrycks et al., 2019a). Hendrycks et al.

[*]Equal contribution.

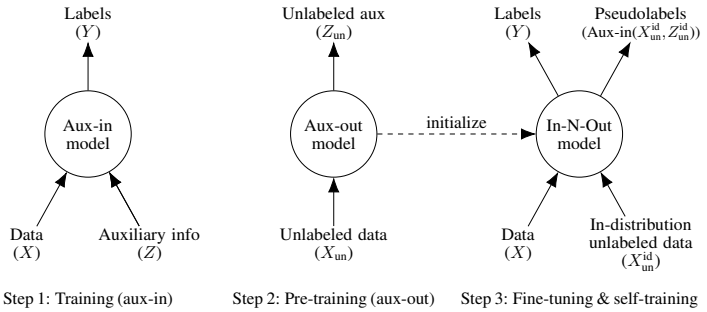


Figure 1: A sketch of the In-N-Out algorithm which consists of three steps: 1) use auxiliary information as input (Aux-in) to achieve good in-distribution performance, 2) use auxiliary information as output in pre-training (Aux-out), to improve OOD performance, 3) fine-tune the pretrained model from step 2 using the labeled data and in-distribution unlabeled data with pseudolabels generated from step 1 to improve in- and out-of-distribution.

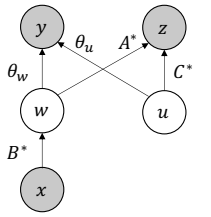


Figure 2: Graphical model for our theoretical setting: prediction task with input x, target y, and auxiliary information z, which is related to y through the latent variable w and latent noise u.

(2019a) show that pre-training on ImageNet can improve adversarial robustness, and Hendrycks et al. (2019b) show that auxiliary self-supervision tasks can improve robustness to synthetic corruptions. In this paper, we find that while aux-outputs improve OOD accuracy, the in-distribution accuracy is worse than with aux-inputs. Thus, we elucidate a tradeoff between in- and out-of-distribution accuracy that occurs when using auxiliary information as inputs or outputs.

To theoretically study how to best use auxiliary information, we extend the multi-task linear regression setting (Du et al., 2020; Tripuraneni et al., 2020) to allow for distribution shifts. We show that auxiliary information helps in-distribution error by providing useful features for predicting the target, but the relationship between the aux-inputs and the target can shift significantly OOD, worsening the OOD error. In contrast, the aux-outputs model first pre-trains on unlabeled data to learn a lower-dimensional representation and then solves the target task in the lower-dimensional space. We prove that the aux-outputs model improves robustness to *arbitrary* covariate shift compared to not using auxiliary information.

Can we do better than using auxiliary information as inputs or outputs alone? We answer affirmatively by proposing the In-N-Out algorithm to combine the benefits of auxiliary inputs and outputs (Figure 1). In-N-Out first uses an aux-inputs model, which has good in-distribution accuracy, to pseudolabel in-distribution unlabeled data. It then pre-trains a model using aux-outputs and finally fine-tunes this model on the larger training set consisting of labeled and pseudolabeled data. We prove that In-N-Out, which combines self-training and pre-training, further improves both in-distribution and OOD error over the aux-outputs model.

We show empirical results on CelebA and two remote sensing tasks (land cover and cropland prediction) that parallel the theory. On all datasets, In-N-Out improves OOD accuracy and has competitive or better in-distribution accuracy over aux-inputs or aux-outputs alone and improves 1–2% in-distribution, 2–3% OOD over not using auxiliary information on remote sensing tasks. Ablations of In-N-Out show that In-N-Out achieves similar improvements over pre-training or self-training alone (up to 5% in-distribution, 1–2% OOD on remote sensing tasks). We also find that using OOD (rather than in-distribution) unlabeled examples for pre-training is crucial for OOD improvements.

2 SETUP

Let $x \in \mathbb{R}^d$ be the input (e.g., a satellite image), $y \in \mathbb{R}$ be the target (e.g., crop type), and $z \in \mathbb{R}^T$ be the cheaply obtained auxiliary information either from additional sources (e.g., climate information) or derived from the original data (e.g., background).

Training data. Let P_{id} and P_{ood} denote the underlying distribution of (x,y,z) triples in-distribution and out-of-distribution, respectively. The training data consists of (i) in-distribution labeled data $\{(x_i, y_i, z_i)\}_{i=1}^{n} \sim P_{\text{id}}$, (ii) in-distribution unlabeled data $\{(x_i^{\text{id}}, z_i^{\text{id}})\}_{i=1}^{m_{\text{id}}} \sim P_{\text{id}}$, and (iii) out-of-distribution unlabeled data $\{(x_i^{\text{ood}}, z_i^{\text{ood}})\}_{i=1}^{m_{\text{ood}}} \sim P_{\text{ood}}$.

Goal and risk metrics. Our goal is to learn a model from input and auxiliary information to the target, $f : \mathbb{R}^d \times \mathbb{R}^T \to \mathbb{R}$. For a loss function ℓ, the in-distribution population risk of the model f is $R_{\text{id}}(f) = \mathbb{E}_{x,y,z\sim P_{\text{id}}}[\ell(f(x,z),y)]$, and its OOD population risk is $R_{\text{ood}}(f) = \mathbb{E}_{x,y,z\sim P_{\text{ood}}}[\ell(f(x,z),y)]$.

2.1 MODELS

We consider three common ways to use the auxiliary information (z) to learn a model.

Baseline. The baseline minimizes the empirical risk on labeled data while ignoring the auxiliary information (accomplished by setting z to 0):

$$\hat{f}_{\text{bs}} = \operatorname*{argmin}_f \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i,0),y_i). \tag{1}$$

Aux-inputs. The aux-inputs model minimizes the empirical risk on labeled data while using the auxiliary information as features:

$$\hat{f}_{\text{in}} = \operatorname*{argmin}_f \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i,z_i),y_i). \tag{2}$$

Aux-outputs. The aux-outputs model leverages the auxiliary information z by using it as the prediction target of an auxiliary task, in hopes that there is a low-dimensional feature representation that is common to predicting both z and y. Training the aux-outputs model consists of two steps:

In the *pre-training* step, we use all the unlabeled data to learn a shared feature representation. Let $h : \mathbb{R}^d \to \mathbb{R}^k$ denote a feature map and $g_{\text{z-out}} : \mathbb{R}^k \to \mathbb{R}^T$ denote a mapping from feature representation to the auxiliary outputs. Let ℓ_{aux} denote the loss function for the auxiliary information. We define the empirical risk of h and $g_{\text{z-out}}$ as:

$$\hat{R}_{\text{pre}}(h,g_{\text{z-out}}) = \frac{1}{m_{\text{id}}+m_{\text{ood}}}\Big(\sum_{i=1}^{m_{\text{id}}} \ell_{\text{aux}}(g_{\text{z-out}}(h(x_i^{\text{id}})),z_i^{\text{id}}) + \sum_{i=1}^{m_{\text{ood}}} \ell_{\text{aux}}(g_{\text{z-out}}(h(x_i^{\text{ood}})),z_i^{\text{ood}})\Big). \tag{3}$$

The estimate of the feature map is $\hat{h}_{\text{out}} = \operatorname{argmin}_h \min_{g_{\text{z-out}}} \hat{R}_{\text{pre}}(h,g_{\text{z-out}})$.

In the *transfer* step, the model uses the pre-trained feature map $\hat{h}_{\text{out}}$ and the labeled data to learn the mapping $g_{\text{y-out}} : \mathbb{R}^k \to \mathbb{R}$ from feature representation to target y. We define the transfer empirical risk as:

$$\hat{R}_{\text{trans}}(\hat{h}_{\text{out}},g_{\text{y-out}}) = \frac{1}{n}\sum_{i=1}^{n} \ell(g_{\text{y-out}}(\hat{h}_{\text{out}}(x_i)),y_i) \tag{4}$$

The estimate of the target mapping is $\hat{g}_{\text{y-out}} = \operatorname{argmin}_{g_{\text{y-out}}} \hat{R}_{\text{trans}}(\hat{h}_{\text{out}}, g_{\text{y-out}})$. The final aux-outputs model is

$$\hat{f}_{\text{out}}(x,z) = \hat{g}_{\text{y-out}}(\hat{h}_{\text{out}}(x)). \tag{5}$$

Like the baseline model, the aux-outputs model ignores the auxiliary information for prediction.

3 THEORETICAL ANALYSIS OF AUX-INPUTS AND AUX-OUTPUTS MODELS

We now analyze the baseline, aux-inputs, and aux-outputs models introduced in Section 2. Our setup extends a linear regression setting commonly used for analyzing multi-task problems (Du et al., 2020; Tripuraneni et al., 2020).

Setup. See Figure 2 for the graphical model. Let $w = B^\star x \in \mathbb{R}^k$ be a low-dimensional latent feature ($k \le d$) shared between auxiliary information z and the target y. Let $u \in \mathbb{R}^m$ denote unobserved latent variables not captured in x. We assume z and y are linear functions of u and w:

$$y = \theta_w^\top w + \theta_u^\top u + \epsilon, \tag{6}$$

$$z = A^\star w + C^\star u, \tag{7}$$

where $\epsilon \sim P_\epsilon$ denotes noise with mean 0 and variance σ^2. As in Du et al. (2020), we assume the dimension of the auxiliary information T is greater than the feature dimension k, that is $T \ge k$, and that $A^\star, B^\star$ and $C^\star$ have full rank (rank k). We also assume $T \ge m$, where m is the dimension of u.

Data. Let P_{x} and P_{u} denote the distribution of x and u in-distribution (ID), and let P'_{x}, P'_{u} denote the distribution x and u OOD. We assume x and u are independent, have distributions with bounded density everywhere, and have invertible covariance matrices. We assume the mean of u is zero in-

and out-of-distribution[1]. We assume we have $n \geq m+d$ in-distribution labeled training examples and unlimited access to unlabeled data both ID and OOD, a common assumption in unsupervised domain adaptation theory (Sugiyama et al., 2007; Kumar et al., 2020; Raghunathan et al., 2020).

Loss metrics. We use the squared loss for the target and auxiliary losses: $\ell(\hat{y},y) = (y-\hat{y})^2$ and $\ell_{\text{aux}}(z,z') = \|z-z'\|_2^2$.

Models. We assume all model families (f, h, $g_{\text{z-out}}$, $g_{\text{y-out}}$) in Section 2 are linear.

Let $\mathcal{S} = (A^\star, B^\star, C^\star, \theta_w, \theta_u, P_{\text{x}}, P_{\text{u}})$ denote a problem setting which satisfies all the above assumptions.

3.1 Auxiliary inputs help in-distribution, but can hurt OOD

We first show that the aux-inputs model (2) performs better than the baseline model (1) in-distribution. Intuitively, the target y depends on both the inputs x (through w) and latent variable u (Figure 2). The baseline model only uses x to predict y; thus it cannot capture the variation in y due to u. On the other hand, the aux-inputs model uses x and z to predict y. Since z is a function of x (through w) and u, u can be recovered from x and z by inverting this relation. Note that u is unobserved but implicitly recovered. The aux-inputs model can then combine u and x to predict y better.

Let $\sigma_u^2 = \mathbb{E}_{u\sim P_u}[(\theta_u^\top u)^2]$ denote the (in-distribution) variance of y due to the latent variables u. The following proposition shows that if $\sigma_u^2 > 0$ then with enough training examples the aux-inputs model has lower in-distribution population risk than the baseline model.[2]

Proposition 1. *For all problem settings $\mathcal{S}$, P_ϵ, assuming regularity conditions (bounded x, u, sub-Gaussian noise ϵ, and $T = m$), and $\sigma_u^2 > 0$, for all $\delta > 0$, there exists N such that for $n \geq N$ number of training points, with probability at least $1-\delta$ over the training examples, the aux-inputs model improves over the baseline:*

$$R_{id}(\hat{f}_{in}) < R_{id}(\hat{f}_{bs}). \tag{8}$$

Although using z as input leads to better in-distribution performance, we show that the aux-inputs model can perform worse than the baseline model OOD for any number of training examples. Intuitively, the aux-inputs model uses z, which can be unreliable OOD because z depends on u and u can shift OOD. In more detail, the aux-inputs model learns to predict $\hat{y} = \hat{\theta}_{x,in}^\top x + \hat{\theta}_{z,in}^\top z$, where the true output $y = \theta_x^\top x + \theta_z^\top z$, and $\hat{\theta}_{z,in}$ is an approximation to the true parameter θ_z, that has some error. Out-of-distribution u and hence z can have very high variance, which would magnify $(\hat{\theta}_{z,in} - \theta_z)^\top z$ and lead to bad predictions.

Example 1. *There exists a problem setting $\mathcal{S}$, P_ϵ, such that for every n, there is some test distribution P'_x, P'_u with:*

$$\mathbb{E}[R_{ood}(\hat{f}_{in})] > \mathbb{E}[R_{ood}(\hat{f}_{bs})] \tag{9}$$

3.2 Pre-training improves risk under arbitrary covariate shift

While using z as inputs (aux-inputs) can worsen performance relative to the baseline, our first main result is that the aux-outputs model (which pre-trains to predict z from x, and then transfers the learned representation to predict y from x) outperforms the baseline model for all test distributions.

Intuition. Referring to Figure 2, we see that the mapping from inputs x to auxiliary z passes through the lower dimensional features w. In the pre-training step, the aux-outputs model predicts z from x using a low rank linear model, and we show that this recovers the 'bottleneck' features w (up to symmetries; more formally we recover the rowspace of $B^\star$). In the transfer step, the aux-outputs model learns a linear map from the lower-dimensional w to y, while the baseline predicts y directly from x. To warm up, *without distribution shift*, the expected excess risk only depends on the dimension of the input, and not the conditioning. That is, the expected excess risk in linear regression is exactly $d\sigma^2/n$, where d is the input dimension, so the aux-outputs trivially improves over the baseline since $\dim(w) < \dim(x)$. In contrast, the *worst case risk under distribution shift depends on the conditioning of the data*, which could be worse for w than x. Our proof shows that the worst case risk (over all x and u) is still better for the aux-outputs model because projecting to the low-dimensional feature representation "zeroes-out" some error directions.

[1] This is not limiting because bias in z can be folded into x.

[2] Since z is typically low-dimensional and x is high-dimensional (e.g., images), the aux-inputs model needs only a slightly larger number of examples before it outperforms the baseline.

Algorithm 1 In-N-Out

Require: in-distribution labeled data $\{(x_i,y_i,z_i)\}_{i=1}^n \sim P_{\text{id}}$,
in-distribution unlabeled data $\{(x_i^{\text{id}},z_i^{\text{id}})\}_{i=1}^{m_{\text{id}}} \sim P_{\text{id}}$,
OOD unlabeled data $\{(x_i^{\text{ood}},z_i^{\text{ood}})\}_{i=1}^{m_{\text{ood}}} \sim P_{\text{ood}}$

1: Learn $\hat{f}_{\text{in}} : (x,z) \mapsto y$ from in-distribution labeled data $\{(x_i,y_i,z_i)\}_{i=1}^n \sim P_{\text{id}}$
2: Pre-train $g_{\text{z-out}} \circ \hat{h}_{\text{out}} : x \mapsto z$ on aux-outputs from all unlabeled data $\{(x_i^{\text{id}},z_i^{\text{id}})\}_{i=1}^{m_{\text{id}}} \cup \{(x_i^{\text{ood}},z_i^{\text{ood}})\}_{i=1}^{m_{\text{ood}}}$
3: Return $\hat{f} = \hat{g} \circ \hat{h}_{\text{out}} : x \mapsto y$ trained on labeled and pseudolabeled data $\{(x_i,y_i)\}_{i=1}^n \cup \{(x_i^{\text{id}},\hat{f}_{\text{in}}(x_i^{\text{id}},z_i^{\text{id}})\}_{i=1}^{m_{\text{id}}}$

Theorem 1. *For all problem settings $\mathcal{S}$, noise distributions P_ϵ, test distributions P'_x, P'_u, and $n \geq m+d$ number of training points:*

$$\mathbb{E}[R_{ood}(\hat{f}_{out})] \leq \mathbb{E}[R_{ood}(\hat{f}_{bs})]. \tag{10}$$

See Appendix A for the proof.

4 In-N-Out: combining auxiliary inputs and outputs

We propose the In-N-Out algorithm, which combines both the aux-inputs and aux-outputs models for further complementary gains (Figure 1). As a reminder: (i) The aux-inputs model ($x,z \to y$) is good in-distribution, but bad OOD because z can be misleading OOD. (ii) The aux-outputs model ($x \to y$) is better than the baseline OOD, but worse than aux-inputs in-distribution because it doesn't use z. (iii) We propose the In-N-Out model ($x \to y$), which uses pseudolabels from aux-inputs (stronger model) in-distribution to transfer in-distribution accuracy to the aux-outputs model. The In-N-Out model does not use z to make predictions since z can be misleading / spurious OOD.

In more detail, we use the aux-inputs model (which is good in-distribution) to pseudolabel in-distribution unlabeled data. The pseudolabeled data provides more effective training samples (self-training) to fine-tune an aux-outputs model pre-trained on predicting auxiliary information from all unlabeled data. We present the general In-N-Out algorithm in Algorithm 1 and analyze it in the linear multi-task regression setting of Section 2. The In-N-Out model $\hat{f} = \hat{g} \circ \hat{h}_{\text{out}}$ optimizes the empirical risk on labeled and pseudolabeled data:

$$\hat{g} = \operatorname*{argmin}_g (1-\lambda)\hat{R}_{\text{trans}}(\hat{h}_{\text{out}},g) + \lambda \hat{R}_{\text{st}}(\hat{h}_{\text{out}},\hat{f}_{\text{in}},g) \tag{11}$$

where $\hat{R}_{\text{st}}(\hat{h}_{\text{out}},\hat{f}_{\text{in}},g) = \frac{1}{m_1}\sum_{i=1}^{m_1} \ell(g(\hat{h}_{\text{out}}(x_i^{\text{id}})),\hat{f}_{\text{in}}(x_i^{\text{id}},z_i^{\text{id}}))$ is the loss of self-training on pseudolabels from the aux-inputs model, and $\lambda \in [0,1]$ is a hyperparameter that trades off between labeled and pseudolabeled losses. In our experiments, we fine-tune $\hat{g}$ and $\hat{h}_{\text{out}}$ together.

Theoretical setup. Because fine-tuning is difficult to analyze theoretically, we analyze a slightly modified version of In-N-Out where we train an aux-inputs model to predict y given the features $\hat{h}_{\text{out}}(x)$ and auxiliary information z, so the aux-inputs model $\hat{g}_{\text{in}} : \mathbb{R}^k \times \mathbb{R}^T \to \mathbb{R}$ is given by $\hat{g}_{\text{in}} = \operatorname{argmin}_g \frac{1}{n}\sum_{i=1}^n \ell(g(\hat{h}_{\text{out}}(x_i),z_i),y_i)$. The population self-training loss on pseudolabels from the aux-inputs model $\hat{g}_{\text{in}} \circ \hat{h}_{\text{out}}$ is: $R_{\text{st}}(\hat{h}_{\text{out}},\hat{g}_{\text{in}},g) = \mathbb{E}_{x,z\sim P_{\text{id}}}[\ell(g(\hat{h}_{\text{out}}(x)),\hat{g}_{\text{in}}(\hat{h}_{\text{out}}(x),z))]$, and we minimize the self-training loss: $\hat{g} = \operatorname{argmin}_g R_{\text{st}}(\hat{h}_{\text{out}},\hat{g}_{\text{in}},g)$. At test time given input x,z the In-N-Out model predicts $\hat{g}(\hat{h}_{\text{out}}(x))$. For the theory, we assume all models ($\hat{g}_{\text{in}},\hat{g},and\hat{h}_{\text{out}}$) are linear.

4.1 In-N-Out improves over pre-training under arbitrary covariate shift

We prove that In-N-Out helps on top of pre-training, as long as the auxiliary features give us information about y relative to the noise ϵ in-distribution—that is, if σ_u^2 is much larger than σ^2.

To build intuition, first consider the special case where the noise $\sigma^2 = 0$ (equivalently, $\epsilon = 0$). Since u can be recovered from w and z, we can write y as a linear function of w and z: $y = \gamma_w^\top w + \gamma_z^\top z$. We train an aux-inputs model $\hat{g}_{\text{in}}$ from w,z to y on finite labeled data. Since there is no noise, $\hat{g}_{\text{in}}$ predicts y perfectly from w,z (we learn γ_w and γ_z). We use $\hat{g}_{\text{in}}$ to pseudolabel a large amount of unlabeled data, and since $\hat{g}_{\text{in}}$ predicts y perfectly from w,z, the pseudolabels are perfect. So here pseudolabeling gives us a much larger and correctly labeled dataset to train the In-N-Out model on.

The technical challenge is proving that self-training helps under arbitrary covariate shift even when the noise is non-zero ($\sigma^2 > 0$), so the aux-inputs model $\hat{g}_{\text{in}}$ that we learn is accurate but not perfect.

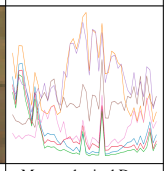
	CelebA	Cropland	Landcover
Visualization (x)			
Aux Info (z)	7 binary attributes	Vegetation, Lat/Lon	Meteorological Data
Target (y)	Male/female?	Cropland/not cropland?	Land cover class
ID-Split	People without hats	IA, MN, IL	Outside Africa
OOD-Split	People with hats	IN, KY	Africa

Figure 3: Summary of the datasets used in our experiments.

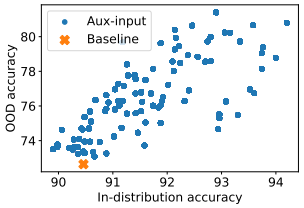


Figure 4: Correlation ($r = 0.72$) between in-distribution accuracy and OOD accuracy when adding 1 to 15 random auxiliary inputs in CelebA.

In this case, the pseudolabels have an error which propagates to the In-N-Out model self-trained on these pseudolabels, but we want to show that the error is lower than for the aux-outputs model. The error in linear regression is proportional to the noise of the target y, which for the aux-outputs model is $\sigma^2+\sigma_u^2$. We show that the In-N-Out model uses the aux-inputs model to reduce the dependence on the noise σ_u^2, because the aux-inputs model uses both w and z to predict y. The proof reduces to showing that the max singular value for the In-N-Out error matrix is less than the min-singular value of the aux-outputs error matrix with high probability. A core part of the argument is to lower bound the min-singular value of a random matrix (Lemma 3). This uses techniques from random matrix theory (see e.g., Chapter 2.7 in Tao (2012)); the high level idea is to show that with probability $1-\delta$ each column of the random matrix has a (not too small) component orthogonal to all other columns.

Theorem 2. *In the linear setting, for all problem settings $\mathcal{S}$ with $\sigma_u^2 > 0$, test distributions P'_x, P'_u, $n \geq m+d$ number of training points, and $\delta > 0$, there exists $a, b > 0$ such that for all noise distributions P_ϵ, with probability at least $1-\delta$ over the training examples and test example $x' \sim P'_x$, the ratio of the excess risks (for all σ^2 small enough that $a - b\sigma^2 > 0$) is:*

$$\frac{R^{ood}_{in\text{-}out} - R^*}{R^{ood}_{out} - R^*} \leq \frac{\sigma^2}{a - b\sigma^2} \tag{12}$$

Here $R^* = \min_{g^*,h^*} \mathbb{E}_{x',y',z' \sim P'}[\ell(g^*(h^*(x')), y')]$ is the min. possible (Bayes-optimal) OOD risk, $R^{ood}_{in\text{-}out} = \mathbb{E}_{y' \sim P'_{y'|x'}}[\ell(\hat{g}(\hat{h}_{out}(x')), y')]$ is the risk of the In-N-Out model on test example x', and $R^{ood}_{out} = \mathbb{E}_{y' \sim P'_{y'|x'}}[\ell(\hat{g}_{y\text{-}out}(\hat{h}_{out}(x')), y')]$ is the risk of the aux-outputs model on test example x'. Note that $R^{ood}_{in\text{-}out}$ and R^{ood}_{out} are random variables that depend on the test input x' and the training set X.

Remark 1. *As $\sigma \to 0$, the excess risk ratio of In-N-Out to Aux-outputs goes to 0, so the In-N-Out estimator is much better than the aux-outputs estimator.*

The proof of the result is in Appendix A.

5 Experiments

We show on real-world datasets for land cover and cropland prediction that aux-inputs can hurt OOD performance, while aux-outputs improve OOD performance. In-N-Out improves OOD accuracy and has competitive or better in-distribution accuracy over other models on all datasets (Section 5.2). Secondly, we show that the tradeoff between in-distribution and OOD performance depends on the choice of auxiliary information on CelebA and cropland prediction (Section 5.3). Finally, we show that OOD unlabeled examples are important for improving OOD robustness (Section 5.4).

5.1 Experimental Setup

We give a summary of considered datasets and setup here — see Figure 3 and Appendix B for details. Our datasets use auxiliary information both derived from the input (CelebA, Cropland) and from other sources (Landcover).

CelebA. In CelebA (Liu et al., 2015), the input x is a RGB image (resized to 64×64), the target y is a binary label for gender, and the auxiliary information z are 7 (of 40) binary-valued attributes derived from the input (e.g., presence of makeup, beard). We designate the set of images where the celebrity is wearing a hat as OOD. We use a ResNet18 as the backbone model architecture for all models (see Appendix B.1 for details).

	CelebA		Cropland		Landcover	
	ID Test Acc	OOD Acc	ID Test Acc	OOD Acc	ID Test Acc	OOD Test Acc
Baseline	90.46 ± 0.85	72.64 ± 1.39	94.50 ± 0.11	90.30 ± 0.75	75.92 ± 0.25	58.31 ± 1.87
Aux-inputs	92.36 ± 0.29	77.4 ± 1.33	$\mathbf{95.34} \pm 0.22$	84.15 ± 4.23	76.58 ± 0.44	54.78 ± 2.01
Aux-outputs	$\mathbf{94.0} \pm 0.24$	77.68 ± 0.59	95.12 ± 0.15	91.63 ± 0.21	72.48 ± 0.37	61.03 ± 0.97
In-N-Out (no pretrain)	$\mathbf{93.8} \pm 0.56$	78.54 ± 1.31	94.93 ± 0.15	91.23 ± 0.61	76.54 ± 0.23	59.19 ± 0.98
In-N-Out	93.42 ± 0.36	79.42 ± 0.70	$\mathbf{95.45} \pm 0.16$	$\mathbf{91.94} \pm 0.57$	$\mathbf{77.43} \pm 0.39$	61.53 ± 0.74
In-N-Out + repeated ST	$\mathbf{93.76} \pm 0.46$	$\mathbf{80.38} \pm 0.68$	$\mathbf{95.53} \pm 0.19$	$\mathbf{92.18} \pm 0.40$	$\mathbf{77.10} \pm 0.30$	$\mathbf{62.61} \pm 0.58$

Table 1: Accuracy (%) of various models using auxiliary information as input, output, or both. In-N-Out generally improves both in- and out-of-distribution over aux-inputs or aux-outputs alone. Results are averaged over 5 trials with 90% intervals. Repeated ST refers to one round of repeated self-training on top of In-N-Out.

Cropland. Crop type or cropland prediction is an important intermediate problem for crop yield prediction (Cai et al., 2018; Johnson et al., 2016; Kussul et al., 2017). The input x is a 50×50 RGB image taken by a satellite, the target y is a binary label that is 1 when the image contains majority cropland, and the auxiliary information z is the center location coordinate plus 50×50 vegetation-related bands. The vegetation bands in the auxiliary information z is derived from the original satellite image, which contains both RGB and other frequency bands. We use the Cropland dataset from Wang et al. (2020), with data from the US Midwest. We designate Iowa, Missouri, and Illinois as in-distribution and Indiana and Kentucky as OOD. Following Wang et al. (2020), we use a U-Net-based model (Ronneberger et al., 2015). See Appendix B.2 for details.

Landcover. Land cover prediction involves classifying the land cover type (e.g., "grasslands") from satellite data at a location (Gislason et al., 2006; Rußwurm et al., 2020)). The input x is a time series measured by NASA's MODIS satellite (Vermote, 2015), the target y is one of 6 land cover classes, and the auxiliary information z is climate data (e.g., temperature) from ERA5, a dataset computed from various satellites and weather station data (C3S, 2017). We designate non-African locations as in-distribution and Africa as OOD. We use a 1D-CNN to handle the temporal structure in the MODIS data. See Appendix B.3 for details.

Data splits. We first split off the OOD data, then split the rest into training, validation, and in-distribution test (see Appendix B for details). We use a portion of the training set and OOD set as in-distribution and OOD unlabeled data respectively. The rest of the OOD set is held out as test data. We run 5 trials, where we randomly re-generate the training/unlabeled split for each trial (keeping held-out splits fixed). We use a reduced number of labeled examples from each dataset (1%, 5%, 10% of labeled examples for CelebA, Cropland, and Landcover respectively), with the rest as unlabeled.

Repeated self-training. In our experiments, we also consider augmenting In-N-Out models with repeated self-training, which has fueled recent improvements in both domain adaptation and ImageNet classification (Shu et al., 2018; Xie et al., 2020). For one additional round of repeated self-training, we use the In-N-Out model to pseudolabel all unlabeled data (both ID and OOD) and also initialize the weights with the In-N-Out model. Each method is trained with early-stopping and hyperparameters are chosen using the validation set.

5.2 Main Results

Table 1 compares the in-distribution (ID) and OOD accuracy of different methods. In all datasets, pre-training with aux-outputs improves OOD performance over the baseline, and In-N-Out (with or without repeated ST) generally improves both in- and out-of-distribution performance over all other models.

CelebA. In CelebA, using auxiliary information either as aux-inputs or outputs improves both ID (2–4%) and OOD accuracy (5%). We hypothesize this is because the auxiliary information is quite robust. Figure 4 shows that there is a significant correlation ($r = 0.72$) between ID and OOD accuracy for 100 different sets of aux-inputs, supporting results on standard datasets (Recht et al., 2019; Xie et al., 2020; Santurkar et al., 2020). In-N-Out achieves the best OOD performance and comparable ID performance even though there is no tradeoff between ID and OOD accuracy.

Remote sensing. In the remote sensing datasets, aux-inputs can induce a tradeoff where increasing ID accuracy hurts OOD performance. In cropland prediction, even with a small geographic shift (US Midwest), the baseline model has a significant drop from ID to OOD accuracy (4%). The aux-inputs model improves ID accuracy almost 1% above the baseline but OOD accuracy drops 6%. In land cover prediction, using climate information as aux-inputs decreases OOD accuracy by over 4% compared to the baseline. The aux-outputs model improves OOD, but decreases ID accuracy by 3% over the baseline.

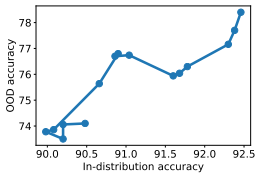


Figure 5: In-distribution vs. OOD accuracy on CelebA when sequentially adding a random set of 15 auxiliary inputs one-by-one. Even if adding all 15 auxiliary inputs improves both in-distribution and OOD accuracy, some intermediate in-distribution gains can hurt OOD.

	ID Test Acc	OOD Test Acc
Only in-distribution	69.73 ± 0.51	57.73 ± 1.58
Only OOD	69.92 ± 0.41	$\mathbf{59.28} \pm 1.01$
Both	70.07 ± 0.46	$\mathbf{59.84} \pm 0.98$

Table 2: Ablation study on the use of in-distribution vs. OOD unlabeled data in pre-training models on Landcover, where unlabeled sample size is standardized (much smaller than Table 1). Using OOD unlabeled examples are important for gains in OOD accuracy (%). Results are shown with 90% error intervals over 5 trials.

Improving in-distribution accuracy over aux-outputs. One of the main goals of the self-training step in In-N-Out is to improve the in-distribution performance of the aux-outputs model. We compare to oracle models that use a large amount of in-distribution labeled data to compare the gains from In-N-Out. In Landcover, the oracle model which uses 160k labeled ID examples gets 80.5% accuracy. In-N-Out uses 16k labeled examples and 150k unlabeled ID examples (with 50k unlabeled OOD examples) and improves the ID accuracy of aux-output from 72.5% to 77.4%, closing most (62%) of the gap. In Cropland, the oracle model achieves 95.6% accuracy. Here, In-N-Out closes 80% of the gap between aux-outputs and the oracle, improving ID accuracy from 95.1% to 95.5%.

Ablations with only pre-training or self-training. We analyze the individual contributions of self-training and pre-training in In-N-Out. On both cropland and land cover prediction, In-N-Out outperforms standard self-training on pseudolabels from the aux-inputs model (In-N-Out without pre-training), especially on OOD performance, where In-N-Out improves by about 1% and 2% respectively. Similarly, In-N-Out improves upon pre-training (aux-outputs model) both ID and OOD for both datasets.

5.3 Choice of auxiliary inputs matters

We find that the choice of auxiliary inputs affects the tradeoff between ID and OOD performance significantly, and thus is important to consider for problems with distribution shift. While Figure 4 shows that auxiliary inputs tend to simultaneously improve ID and OOD accuracy in CelebA, our theory suggests that in the worst case, there should be auxiliary inputs that worsen OOD accuracy. Indeed, Figure 5 shows that when taking a random set of 15 auxiliary inputs and adding them sequentially as auxiliary inputs, there are instances where an extra auxiliary input improves in-distribution but hurts OOD accuracy even if adding all 15 auxiliary inputs improves both ID and OOD accuracy. In cropland prediction, we compare using location coordinates and vegetation data as auxiliary inputs with only using vegetation data. The model with locations achieves the best ID performance, improving almost 1% in-distribution over the baseline with only RGB. Without locations (only vegetation data), the ID accuracy is similar to the baseline but the OOD accuracy improves by 1.5%. In this problem, location coordinates help with in-distribution interpolation, but the model fails to extrapolate to new locations.

5.4 OOD unlabeled data is important for pre-training

We compare the role of in-distribution vs. OOD unlabeled data in pre-training. Table 2 shows the results of using only in-distribution vs. only OOD vs. a balanced mix of unlabeled examples for pre-training on the Landcover dataset, where unlabeled sample size is standardized across the models (by reducing to the size of the smallest set, resulting in 4x less unlabeled data). Using only in-distribution unlabeled examples does not improve OOD accuracy, while having only OOD unlabeled examples does well both in-distribution and OOD since it also has access to the labeled in-distribution data. For the same experiment in cropland prediction, the differences were not statistically significant, perhaps due to the smaller geographic shift (across states in cropland vs. continents in landcover).

6 Related work

Multi-task learning and weak supervision. Caruana and de Sa (2003) proposed using noisy features (aux-outputs) as a multi-task output, but do not theoretically analyze this approach. Wu et al. (2020) also study multi-task linear regression. However, their auxiliary tasks must have true parameters that are closely aligned (small cosine distance) to the target task. Similarly, weak supervision works assume access to weak labels correlated with the true label (Ratner et al., 2016; 2017). In our paper,

we make no assumptions about the alignment of the auxiliary and target tasks beyond a shared latent variable while also considering distribution shifts.

Transfer learning, pre-training, and self-supervision. We support empirical works that show the success of transfer learning and pre-training in vision and NLP (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; Devlin et al., 2019). Theoretically, Du et al. (2020); Tripuraneni et al. (2020) study pre-training in a similar linear regression setup. They show in-distribution generalization bound improvements, but do not consider OOD robustness or combining with auxiliary inputs. Hendrycks et al. (2019b) shows empirically that self-supervision can improve robustness to synthetic corruptions. We support these results by showing theoretical and empirical robustness benefits for pre-training on auxiliary information, which can be derived from the original input as in self-supervision.

Self-training for robustness. Raghunathan et al. (2020) analyze robust self-training (RST) (Carmon et al., 2019; Najafi et al., 2019; Uesato et al., 2019), which improves the tradeoff between standard and adversarially robust accuracy, in min-norm linear regression. Khani and Liang (2021) show how to use RST to make a model robust against a predefined spurious feature without losing accuracy. While related, we work in multi-task linear regression, study pre-training, and prove robustness to *arbitrary* covariate shifts. Kumar et al. (2020) show that repeated self-training on gradually shifting unlabeled data can enable adaptation over time. In-N-Out is complementary and may provide better pseudolabels in each step of this method. Chen et al. (2020) show that self-training can remove spurious features for Gaussian input features in linear models, whereas our results hold for general input distributions (with density). Zoph et al. (2020) show that self-training and pre-training combine for in-distribution gains. We provide theory to support this and also show benefits for OOD robustness.

Domain adaptation. Domain adaptation works account for covariate shift by using unlabeled data from a target domain to adapt the model (Blitzer and Pereira, 2007; Daumé III, 2007; Shu et al., 2018; Hoffman et al., 2018; Ganin et al., 2016). Often, modern domain adaptation methods (Shu et al., 2018; Hoffman et al., 2018) have a self-training or entropy minimization component that benefits from having a better model in the target domain to begin with. Similarly, domain adversarial methods (Ganin et al., 2016) rely on the inductive bias of the source-only model to correctly align the source and target distributions. In-N-Out may provide a better starting point for these domain adaptation methods.

7 Discussion

Using spurious features for robustness. Counterintuitively, In-N-Out uses potentially spurious features (the auxiliary information, which helps in-distribution but hurts OOD accuracy) to improve OOD robustness. This is in contrast to works on removing spurious features from the model (Arjovsky et al., 2019; Ilyas et al., 2019; Chen et al., 2020). In-N-Out promotes utilizing all available information by leveraging spurious features as useful in-distribution prediction signals rather than throwing them away.

General robustness with unlabeled data. In-N-Out is an instantiation of a widely applicable paradigm for robustness: collect unlabeled data in all parts of the input space and learn better representations from the unlabeled data before training on labeled data. This paradigm has driven large progress in few-shot generalization in vision (Hendrycks et al., 2019a;b) and NLP (Devlin et al., 2019; Brown et al., 2020). In-N-Out enriches this paradigm by proposing that some features of the collected data can be used as input and output simultaneously, which results in robustness to arbitrary distribution shifts.

Leveraging metadata and unused features in applications. Many applications have inputs indexed by metadata such as location coordinates or timestamps (Christie et al., 2018; Yeh et al., 2020; Ni et al., 2019). We can use such metadata to join (in a database sense) other auxilary data sources on this metadata for use in In-N-Out. This auxiliary information may often be overlooked or discarded, but In-N-Out provides a way to incorporate them to improve both in- and out-of-distribution accuracy.

Division between input features and auxiliary information. While a standard division between inputs and auxiliary information may exist in some domains, In-N-Out applies for any division of the input. An important further question is how to automatically choose this division under distribution shifts.

8 Conclusion

We show that while auxiliary information as inputs improve in-distribution and OOD on standard curated datasets, they can hurt OOD in real-world datasets. In contrast, we show that using auxiliary information as outputs by pretraining improves OOD performance. In-N-Out combines the strengths of auxiliary inputs and outputs for further improvements both in- and out-of-distribution.

9 ACKNOWLEDGEMENTS

We thank Sherrie Wang and Andreas Schlueter for their help in procuring remote sensing data, Daniel Levy for his insight in simplifying the proof of Theorem 1, Albert Gu for a key insight in proving Lemma 3 using tools from random matrix theory, as well as Shyamal Buch, Pang Wei Koh, Shiori Sagawa, and anonymous reviewers for their valuable help and comments. This work was supported by an Open Philanthropy Project Award, an NSF Frontier Award as part of the Center for Trustworthy Machine Learning (CTML). SMX was supported by an NDSEG Fellowship. AK was supported by a Stanford Graduate Fellowship. TM was partially supported by the Google Faculty Award, JD.com, Stanford Data Science Initiative, and the Stanford Artificial Intelligence Laboratory.

10 REPRODUCIBILITY

All code, data, and experiments are on CodaLab at this link.

REFERENCES

Sajjad Ahmad, Ajay Kalra, and Haroon Stephen. Estimating soil moisture using remote sensing data: A machine learning approach. *Advances in Water Resources*, 33(1):69–80, 2010.

EA AlBadawy, A Saha, and MA Mazurowski. Deep learning for segmentation of brain tumors: Impact of cross-institutional training and testing. *Med Phys.*, 45, 2018.

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

John Blitzer and Fernando Pereira. Domain adaptation of natural language processing systems. *University of Pennsylvania*, 2007.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

C3S. ERA5: Fifth generation of ECMWF atmospheric reanalyses of the global climate, 2017.

Yaping Cai, Kaiyu Guan, Jian Peng, Shaowen Wang, Christopher Seifert, Brian Wardlow, and Zhan Li. A high-performance and in-season classification system of field-level crop types using time-series landsat data and a machine learning approach. *Remote Sensing of Environment*, 210:74–84, 2018.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Rich Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

Rich Caruana and Virginia R. de Sa. Benefitting from the variables that variable selection discards. *Journal of Machine Learning Research (JMLR)*, 3, 2003.

Yining Chen, Colin Wei, Ananya Kumar, and Tengyu Ma. Self-training avoids using spurious features under domain shift. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Gordon Christie, Neil Fendley, James Wilson, and Ryan Mukherjee. Functional map of the world. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

Hal Daumé III. Frustratingly easy domain adaptation. In *Association for Computational Linguistics (ACL)*, 2007.

R S DeFries and JRG Townshend. NDVI-derived land cover classifications at a global scale. *International Journal of Remote Sensing*, 15(17):3567–3586, 1994.

Ruth DeFries, Matthew Hansen, and John Townshend. Global discrimination of land cover types from metrics derived from AVHRR pathfinder data. *Remote Sensing of Environment*, 54(3):209–222, 1995.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Association for Computational Linguistics (ACL)*, pages 4171–4186, 2019.

Simon S. Du, Wei Hu, Sham M. Kakade, Jason D. Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv*, 2020.

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, Francois Laviolette, Mario March, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research (JMLR)*, 17, 2016.

Pall Oskar Gislason, Jon Atli Benediktsson, and Johannes R. Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294–300, 2006.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning (ICML)*, 2019a.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019b.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression. In *Conference on Learning Theory (COLT)*, 2012.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*, 2019.

Neal Jean, Marshall Burke, Michael Xie, W. Matthew Davis, David B. Lobell, and Stefano Ermon. Combining satellite imagery and machine learning to predict poverty. *Science*, 353, 2016.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

Michael D. Johnson, William W. Hsieh, Alex J. Cannon, Andrew Davidson, and Frédéric Bédard. Crop yield forecasting on the canadian prairies by remotely sensed vegetation indices and machine learning methods. *Agricultural and Forest Meteorology*, 218:74–84, 2016.

Fereshte Khani and Percy Liang. Removing spurious features can hurt accuracy and affect groups disproportionately. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2021.

Serkan Kiranyaz, Onur Avci, Osama Abdeljaber, Turker Ince, Moncef Gabbouj, and Daniel J Inman. 1d convolutional neural networks and applications: A survey. *arXiv preprint arXiv:1905.03554*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105, 2012.

Ananya Kumar, Tengyu Ma, and Percy Liang. Understanding self-training for gradual domain adaptation. In *International Conference on Machine Learning (ICML)*, 2020.

N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5): 778–782, 2017.

David J. Lary, Amir H. Alavi, Amir H. Gandomi, and Annette L. Walker. Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3–10, 2016.

Ainong Li, Shunlin Liang, Angsheng Wang, and Jun Qin. Estimating crop yield from multi-temporal satellite data using multivariate regression and neural network techniques. *Photogrammetric Engineering & Remote Sensing*, 73(10):1149–1157, 2007.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015.

Ross Lunetta, Joseph F Knight, Jayantha Ediriwickremaand John G Lyon, and L Dorsey Worthy. Land-cover change detection using multi-temporal MODIS NDVI data. *Remote sensing of environment*, 105(2):142–154, 2006.

Aaron E. Maxwell, Timothy A. Warner, and Fang Fang. Implementation of machine-learning classification in remote sensing: an applied review. *International Journal of Remote Sensing*, 39 (9):2784–2817, 2018.

Amir Najafi, Shin ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 188–197, 2019.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang. Understanding and mitigating the tradeoff between robustness and accuracy. In *International Conference on Machine Learning (ICML)*, 2020.

Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. In *Very Large Data Bases (VLDB)*, number 3, pages 269–282, 2017.

Alexander J Ratner, Christopher M De Sa, Sen Wu, Daniel Selsam, and Christopher Ré. Data programming: Creating large training sets, quickly. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3567–3575, 2016.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning (ICML)*, 2019.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. *arXiv*, 2015.

Marc Rußwurm, Sherrie Wang, Marco Korner, and David Lobell. Meta-learning for few-shot land cover classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 200–201, 2020.

Shibani Santurkar, Dimitris Tsipras, and Aleksander Madry. Breeds: Benchmarks for subpopulation shift. *arXiv*, 2020.

Rui Shu, Hung H. Bui, Hirokazu Narui, and Stefano Ermon. A DIRT-T approach to unsupervised domain adaptation. In *International Conference on Learning Representations (ICLR)*, 2018.

K Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert Muller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research (JMLR)*, 8:985–1005, 2007.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Terrence Tao. *Topics in random matrix theory*. American Mathematical Society, 2012.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *arXiv preprint arXiv:2007.00644*, 2020.

Nilesh Tripuraneni, Michael I. Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *arXiv*, 2020.

Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

E. Vermote. MOD09A1 MODIS/terra surface reflectance 8-day L3 global 500m SIN grid V006. `https://doi.org/10.5067/MODIS/MOD09A1.006`, 2015.

Sherrie Wang, William Chen, Sang Michael Xie, George Azzari, and David B. Lobell. Weakly supervised deep learning for segmentation of remote sensing imagery. *Remote Sensing*, 12, 2020.

Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. A survey of transfer learning. *Journal of Big Data*, 3, 2016.

Sen Wu, Hongyang R. Zhang, and Christopher Ré. Understanding and improving information transfer in multi-task learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Michael Xie, Neal Jean, Marshall Burke, David Lobell, and Stefano Ermon. Transfer learning from deep features for remote sensing and poverty mapping. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2016.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V. Le. Self-training with noisy student improves imagenet classification. *arXiv*, 2020.

Christopher Yeh, Anthony Perez, Anne Driscoll, George Azzari, Zhongyi Tang, David Lobell, Stefano Ermon, and Marshall Burke. Using publicly available satellite imagery and deep learning to understand economic well-being in africa. *Nature Communications*, 11, 2020.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin D. Cubuk, and Quoc V. Le. Rethinking pre-training and self-training. *arXiv*, 2020.

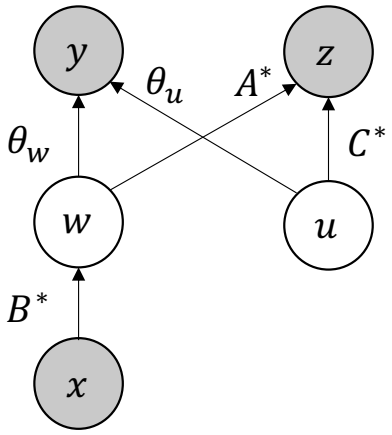


Figure 6: Graphical model for our theoretical setting, where auxiliary information z is related to targets y through the latent variable w and latent noise u.

A PROOF FOR SECTIONS 3 AND 4

Our theoretical setting assumes all the model families are linear. We begin by specializing the setup in Section 2 and defining all the necessary matrices. A word on notation: if unspecified, expectations are taken over all random variables.

Data matrices: We have finite labeled data in-distribution: $n \geq d+m$ input examples $X \in \mathbb{R}^{n\times d}$, where each row $X_i \sim P_x$ is an example sampled independently. We have an *unobserved* latent matrix: $U \in \mathbb{R}^{n\times m}$ where each row $U_i \sim P_u$ is sampled independently from other rows and from X. U is unobserved and not directly used by any of the models, but we will reference U in our analysis. As stated in the main paper, we assume that $\mathbb{E}_{u\sim P_u}[u]=0$. We have labels $Y \in \mathbb{R}^n$ and auxiliary data $Z \in \mathbb{R}^{n\times T}$, where each row Y_i, Z_i is sampled jointly given input example X_i, U_i, that is: $Y_i, Z_i \sim P_{y,z|X_i,U_i}$. In our linear setting, we have $Z = XB^{\star\top}A^{\star\top} + UC^{\star\top}$ and $Y = XB^{\star\top}\theta_w + U\theta_u + \epsilon$, where $\epsilon \in \mathbb{R}^n$ with each entry $\epsilon_i \sim P_\epsilon$ sampled independently from a mean 0, variance σ^2 distribution P_ϵ.

Reminder on shapes: As a reminder, $B^\star \in \mathbb{R}^{k\times d}$ maps the input $x \in \mathbb{R}^d$ to a low dimensional representation $w \in \mathbb{R}^k$ via $w = B^\star x$. $A^\star \in \mathbb{R}^{T\times k}, C^\star \in \mathbb{R}^{T\times m}$ generate auxiliary $z \in \mathbb{R}^T$ via: $z = A^\star w + C^\star u$. Finally, $y \in \mathbb{R}$ is given by: $y = \theta_w^\top w + \theta_u^\top u + \epsilon$, where $\theta_w \in \mathbb{R}^k, \theta_u \in \mathbb{R}^m$. Letting $\theta_x = B^{\star\top}\theta_w$, we equivalently have $y = \theta_w^\top x + \theta_u^\top u + \epsilon$ in terms of x, u.

A.1 MODELS AND EVALUATION

Baseline: ordinary least squares estimator that uses x only, so $\hat{\theta}_{x,ols} = \text{argmin}_{\theta'} \|Y - X\theta'\|_2$. Given a test example x, z, the baseline method predicts $\hat{f}_{\text{bs}}(x,z) = \hat{\theta}_{x,ols}^\top x$, ignoring z. In closed form, $\hat{\theta}_{x,ols} = (X^\top X)^{-1}X^\top Y$.

Aux-inputs: least squares estimator using x and auxiliary z as input: $\hat{\theta}_{x,in}, \hat{\theta}_{z,in} = \text{argmin}_{\theta'_x,\theta'_z} \|Y - (X\theta'_x + Z\theta'_z)\|_2$. The aux-inputs method predicts $\hat{\theta}_{x,in}^\top x + \hat{\theta}_{z,in}^\top z$ for a test example x, z. In closed form, letting $X_Z = [X; Z]$, where we append the columns so that $X_Z \in \mathbb{R}^{n\times(d+T)}$, $[\hat{\theta}_{x,in}, \hat{\theta}_{z,in}] = (X_Z{}^\top X_Z)^{-1} X_Z{}^\top Y$.

Aux-outputs: pretrains on predicting z from x on unlabeled data to learn a mapping from x to w, then learns a regression model on top of this latent embedding w. In the *pre-training* step: use unlabeled data to learn the feature-space embedding $\hat{B}$:

$$\hat{A}, \hat{B} = \underset{A,B}{\text{argmin}} \underset{x\sim P_x}{\mathbb{E}} [\|ABx - z\|_2^2] \quad A \in \mathbb{R}^{T\times k}, B \in \mathbb{R}^{k\times d}. \tag{13}$$

The *transfer* step solves a lower dimensional regression problem from w to y: $\hat{\theta}_{w,out} = \text{argmin}_{\theta'_w} \|Y - X\hat{B}^\top\theta'_w\|_2$. Given a test example x, the aux-outputs model predicts $\hat{\theta}_{w,out}^\top \hat{B}x$.

In-N-Out: First learn an output model $\hat{A},\hat{B}$, and let $W = X\hat{B}^\top$ be the feature matrix. Next, train an input model on the feature space w.

$$\hat{\gamma_w},\hat{\gamma_z} = \underset{\hat{\gamma_w},\hat{\gamma_z}}{\operatorname{argmin}} \|Y - (W\hat{\gamma_w} + Z\hat{\gamma_z})\|_2. \tag{14}$$

Note that this is slightly different from our experiments where we trained the aux-inputs model directly on the inputs x. We now use the input model to pseudolabel our in-domain unlabeled examples, and self-train a model *without z* on these pseudolabels. Given each point w,z, we produce a pseudolabel $\hat{\gamma_w}^\top w + \hat{\gamma_z}^\top z$. We now learn a least squares estimator from w to the pseudolabels which gives us the In-N-Out estimator $\hat{\theta}_w$:

$$\hat{\theta}_w = \underset{\hat{\theta}_w}{\operatorname{argmin}} \mathop{\mathbb{E}}_{w,z\sim P_{\text{id}}} \left[\left(\hat{\theta}_w^\top w - (\hat{\gamma_w}^\top w + \hat{\gamma_z}^\top z) \right)^2 \right] \tag{15}$$

Given a test example x, In-N-Out predicts $\hat{\theta}_w^\top \hat{B} x$.

A.2 Auxiliary inputs help in-distribution

The proof of Proposition 1 is fairly standard. We first give a brief sketch, specify the additional regularity conditions, and then give the proof. We lower bound the risk of the baseline by $\sigma_u^2 + \sigma^2$ since this is the Bayes-opt risk of using only x but not z to predict y. We upper bound the risk of the aux-inputs model which uses x,z to predict y, which is the same as upper bounding the risk in random design linear regression. For this upper bound we use Theorem 1 in Hsu et al. (2012) (note that there are multiple versions of this paper, and we specifically use the Arxiv version available at `https://arxiv.org/abs/1106.2363`). As such, we inherit their regularity conditions. In particular, we assume:

1. x,u are upper bounded almost surely. This is a technical condition, and can be replaced with sub-Gaussian tail assumptions (Hsu et al., 2012).
2. The noise ϵ is sub-Gaussian with variance parameter σ^2.
3. The latent dimension m and auxiliary dimension T are equal so that the inputs to the aux-inputs model have invertible covariance matrix.[3]

Restatement of Proposition 1. *For all problem settings $\mathcal{S}$, P_ϵ, assuming regularity conditions (bounded x,u, sub-Gaussian noise ϵ, and $T = m$), and $\sigma_u^2 > 0$, for all $\delta > 0$, there exists N such that for $n \geq N$ number of training points, with probability at least $1-\delta$ over the training examples, the aux-inputs model improves over the baseline:*

$$R_{id}(\hat{f}_{in}) < R_{id}(\hat{f}_{bs}). \tag{16}$$

Proof. **Lower bound risk of baseline**: First, we lower bound the expected risk of the baseline by $\sigma_u^2 + \sigma^2$. Intuitively, this is the irreducible error—no linear classifier using only x can get better risk than $\sigma_u^2 + \sigma^2$ because of intrinsic noise in the output y. Let $\theta_x = B^{\star\top}\theta_w$ be the optimal baseline parameters. We have

$$\begin{aligned}
R_{\text{id}}(\hat{f}_{\text{bs}}) &= \mathop{\mathbb{E}}_{x,y,z\sim P_{\text{id}}} [(y - \hat{\theta}_{x,ols}^\top x)^2] && (17)\\
&= \mathop{\mathbb{E}}_{x,u,\epsilon\sim P_{\text{id}}} [((\theta_x^\top x + \theta_u^\top u + \epsilon) - \hat{\theta}_{x,ols}^\top x)^2] && (18)\\
&= \mathop{\mathbb{E}}_{x\sim P_{\text{id}}} [(\theta_x^\top x - \hat{\theta}_{x,ols}^\top x)^2] + \mathop{\mathbb{E}}_{u\sim P_{\text{id}}} [\theta_u^\top u^2] + \mathop{\mathbb{E}}_{\epsilon\sim P_{\text{id}}} [\epsilon^2] && (19)\\
&\geq \mathop{\mathbb{E}}_{u\sim P_{\text{id}}} [\theta_u^\top u^2] + \mathop{\mathbb{E}}_{\epsilon\sim P_{\text{id}}} [\epsilon^2]. && (20)\\
&= \sigma_u^2 + \sigma^2. && (21)
\end{aligned}$$

To get Equation 19, we expand the square, use linearity of expectation, and use the fact that x,u,ϵ are independent where u,ϵ are mean 0.

Upper bound risk of aux-inputs: On the other hand, we will show that if n is sufficiently large, the expected risk of the input model is less than $\sigma_u^2 + \sigma^2$.

First we show that we can write $y = \theta_x'^\top x + \theta_z'^\top z + \epsilon$ for some θ_x', θ_z', that is y is a well-specified linear function of x and z plus some noise. Intuitively this is because y is a linear function of x,u

[3] x and u are independent, with invertible covariance matrices, and $z = A^\star B^\star x + C^\star u$ where $C^\star$ is full rank, so by block Gaussian elimination we can see that $[x,z]$ has invertible covariance matrix as well.

and since $C^\star$ is invertible we can extract u from x,z. Formally, we assumed the true model is linear, that is, $y=\theta_x^\top x+\theta_u^\top u+\epsilon$. Since we have $z=A^\star B^\star x+C^\star u$ where $C^\star$ is invertible, we can write $u=C^{\star -1}(z-A^\star B^\star x)$. This gives us

$$\begin{aligned} y&=\theta_x^\top x+\theta_u^\top u+\epsilon && (22)\\ &=\theta_x^\top x+\theta_u^\top C^{\star -1}(z-A^\star B^\star x)+\epsilon && (23)\\ &=(\theta_x-B^{\star\top}A^{\star\top}(C^{\star\top})^{-1}\theta_u)^\top x+(C^{\star\top})^{-1}\theta_u^\top z+\epsilon. && (24)\end{aligned}$$

So setting $\theta'_x=\theta_x-B^{\star\top}A^{\star\top}(C^{\star\top})^{-1}\theta_u$ and $\theta'_z=C^{\star\top -1}\theta_u$, we get $y=\theta_x'^\top x+\theta_z'^\top z+\epsilon$.

As before, we note that the total mean squared error can be decomposed into the Bayes-opt error plus the excess error:

$$\begin{aligned} R_{\text{id}}(\hat{f}_{\text{in}})&=\mathop{\mathbb{E}}_{x,y,z\sim P_{\text{id}}}[(y-\hat{\theta}_{x,in}^\top x-\hat{\theta}_{z,in}^\top z)^2] && (25)\\ &=\mathop{\mathbb{E}}_{x,z,\epsilon\sim P_{\text{id}}}[((\theta_x'^\top x+\theta_z'^\top z+\epsilon)-\hat{\theta}_{x,in}^\top x-\hat{\theta}_{z,in}^\top z)^2] && (26)\\ &=\mathop{\mathbb{E}}_{x,z\sim P_{\text{id}}}[(\theta_x'^\top x+\theta_z'^\top z-\hat{\theta}_{x,in}^\top x-\hat{\theta}_{z,in}^\top z)^2]+\mathop{\mathbb{E}}_{\epsilon\sim P_{\text{id}}}[\epsilon^2] && (27)\\ &=\mathop{\mathbb{E}}_{x,z\sim P_{\text{id}}}[(\theta_x'^\top x+\theta_z'^\top z-\hat{\theta}_{x,in}^\top x-\hat{\theta}_{z,in}^\top z)^2]+\sigma^2. && (28)\end{aligned}$$

To get Equation 27, we expand the square, use linearity of expectation, and use the fact that x,z,ϵ are independent with $\mathbb{E}[\epsilon]=0$. So it suffices to bound the excess error, defined as:

$$EE:=\mathop{\mathbb{E}}_{x,z\sim P_{\text{id}}}[(\theta_x'^\top x+\theta_z'^\top z-\hat{\theta}_{x,in}^\top x-\hat{\theta}_{z,in}^\top z)^2] \quad (29)$$

To bound the excess error, we use Theorem 1 in Hsu et al. (2012) where the inputs/covariates are $[x,z]$. $\mathbb{E}[[x,z][x,z]^\top]$ is invertible because $m=T$, $C^\star$ is full rank, and P_x,P_u have density everywhere with density upper bounded so the variance in any direction is positive, and so the population covariance matrix is positive definite. This means $\mathbb{E}[[x,z][x,z]^\top]$ has min singular value lower bounded, and we also have that x, z are bounded random variables. Therefore, Condition 1 is satisfied for some finite ρ_0. Condition 2 is satisfied since the noise ϵ is sub-Gaussian with mean 0 and variance parameter σ^2. Condition 3 is satisfied with $b_0=0$, since we are working in the setting of well-specified linear regression.

To apply Theorem 1 (Hsu et al., 2012), we first choose $t=\log\frac{3}{\delta}$ so that $1-3e^{-t}\geq 1-\delta$, and so the statement of the Theorem holds with probability at least $1-\delta$. Since our true model is linear (or as Remark 9 says that "the linear model is correct"), $\text{approx}(x)=0$.

So as per remark 9 (Hsu et al., 2012) Equation 11, for some constant c', we have an upper bound on the excess error EE with probability at least $1-\delta$:

$$EE\leq\frac{\sigma^2(d+2\sqrt{dt}+2t)}{n}+o(1/n). \quad (30)$$

Note that the notation in Hsu et al. (2012) is different. The learned estimator in ordinary least squares regression is denoted by $\hat{\beta}_0$, the ground truth parameters by β, and the excess error is denoted by $\|\hat{\beta}_0-\beta\|_\Sigma$. See section 2.1, 2.2 of Hsu et al. (2012) for more details.

Since t is fixed, there exists some constant c (dependent on δ) such that for large enough N_1 if $n\geq N_1$:

$$EE\leq\sigma^2(cd/n). \quad (31)$$

Note that this is precisely Remark 10 (Hsu et al., 2012). Remark 10 says that $\|\hat{\beta}_0-\bar{\beta}_0\|_\Sigma$ is within constant factors of $\sigma^2 d/n$ for large enough n. This is the variance term, but the bias term is 0 since the linear model is well-specified so $\text{approx}(x)=0$. As in Propostion 2 (Hsu et al., 2012) the total excess error is bounded by 2 times the sum of the bias and variance term, which gives us the same result.

Putting this (Equation 31) back into Equation 28, we get that with probability at least $1-\delta$:

$$R_{\text{id}}(\hat{f}_{\text{in}})\leq\sigma^2(1+cd/n). \quad (32)$$

Since $\sigma_u^2 > 0$, we have $\sigma^2 < \sigma_u^2 + \sigma^2$. Then for some N and for all $n \geq N$, we have

$$R_{\text{id}}(\hat{f}_{\text{in}}) < \sigma_u^2 + \sigma^2 \leq R_{\text{id}}(\hat{f}_{\text{bs}}). \tag{33}$$

In particular, we can choose $N = \max(N_1, c\frac{\sigma^2}{\sigma_u^2}d+1)$, which completes the proof. □

A.3 Auxiliary inputs can hurt out-of-distribution

Restatement of Example 1. *There exists a problem setting $\mathcal{S}$, P_ϵ, such that for every n, there is some test distribution P'_x, P'_u with:*

$$\mathbb{E}[R_{ood}(\hat{f}_{in})] > \mathbb{E}[R_{ood}(\hat{f}_{bs})] \tag{34}$$

Proof. We will have $x \in \mathbb{R}$ (so $d = 1$), $w = x$, and $u, z \in \mathbb{R}^2$. We set $z_1 = u_1 + w$ and $z_2 = u_2$, in other words we choose $A^\star = [1,0]$ and $C^\star = I_2$ is the identity matrix. We set $y = x + u_1 + \epsilon$, with $\epsilon \sim N(0,\sigma^2)$, so y is a function of x and u_1 but not u_2. In other words we choose $\theta_w = 1$ and $\theta_u = (1,0)$. P_x will be Uniform$[-1,1]$, and P_u will be uniform in the unit ball in $\mathbb{R}^2$.

Let $X_Z = [X;Z]$, which denotes appending X and Z by columns so $X_Z \in \mathbb{R}^{n\times 3}$ with $n \geq 3$. Since P_x and P_u have density, X_Z has rank 3 almost surely. This means that ${X_Z}^\top X_Z$ is invertible (and positive semi-definite) almost surely. Since P_x and P_u are bounded, the maximum eigenvalue τ'_{max} of ${X_Z}^\top X_Z$ is bounded above. The minimum eigenvalue τ_{min} of $({X_Z}^\top X_Z)^{-1}$ is precisely $1/\tau'_{max}$ and is therefore positive and bounded below by some $c > 0$ almost surely.

We will define P'_x and P'_u soon. For now, consider a new test example $x' \sim P'_x, u' \sim P'_u$ with $z' = [x',0] + u'$ and $y' = x' + u'_1 + \epsilon'$ with $\epsilon' \sim N(0,\sigma^2)$ and $\mathbb{E}[u'] = 0$. For the input model we have:

$$\begin{aligned}
\mathbb{E}[R_{\text{ood}}(\hat{f}_{\text{in}})] &= \mathbb{E}[(y' - (\hat{\theta}_{x,in}^\top x' + \hat{\theta}_{z,in}^\top z'))^2] && (35)\\
&= \sigma^2(1 + \mathbb{E}[(x',z')^\top ({X_Z}^\top X_Z)^{-1}(x',z')]) && (36)\\
&\geq \sigma^2(1 + \mathbb{E}[\tau_{min}\|(x',z')\|_2^2]) && (37)\\
&\geq \sigma^2(1 + c\,\mathbb{E}[\|(x',z')\|_2^2]) && (38)\\
&\geq \sigma^2(1 + c\,\mathbb{E}[{z'_2}^2]) && (39)\\
&= \sigma^2(1 + c\,\mathbb{E}[{u'_2}^2]) && (40)
\end{aligned}$$

Notice that this lower bound is a function of $\mathbb{E}[{u'_2}^2]$ which we will make very large.

On the other hand, letting ${\sigma'_u}^2 = \mathbb{E}_{u'\sim P'_u}[(\theta_u^\top u')^2] = \mathbb{E}_{u'\sim P'_u}[{u'_1}^2]$, for the baseline model we have

$$\begin{aligned}
\mathbb{E}[R_{\text{ood}}(\hat{f}_{\text{bs}})] &= \mathbb{E}[(y' - \hat{\theta}_{x,ols}^\top x')^2] && (41)\\
&= \mathbb{E}[((\theta_x^\top x' + \theta_u^\top u' + \epsilon') - \hat{\theta}_{x,ols}^\top x')^2] && (42)\\
&= \mathbb{E}[(\theta_u^\top u' + \epsilon')^2] + \mathbb{E}[((\theta_x^\top x' - \hat{\theta}_{x,ols}^\top x')^2] && (43)\\
&= \mathbb{E}[(\theta_u^\top u')^2] + \mathbb{E}[\epsilon'^2] + \mathbb{E}[(\theta_x^\top x' - \hat{\theta}_{x,ols}^\top x')^2] && (44)\\
&= {\sigma'_u}^2 + \sigma^2 + \mathbb{E}[(\theta_x^\top x' - \hat{\theta}_{x,ols}^\top x')^2] && (45)\\
&= {\sigma'_u}^2 + \sigma^2 + \mathbb{E}[{x'}^\top (\theta_x - \hat{\theta}_{x,ols})(\theta_x - \hat{\theta}_{x,ols})^\top x'] && (46)\\
&= {\sigma'_u}^2 + \sigma^2 + (\sigma^2 + {\sigma_u}^2)\mathbb{E}[{x'}^\top (X^\top X)^{-1} x'] && (47)
\end{aligned}$$

where in Equation 46, we use the fact that $\theta_x - \hat{\theta}_{x,ols} = (X^\top X)^{-1}X^\top(U\theta_u + \epsilon)$ to get the next line. So the risk depends on x' and $\mathbb{E}[{u'_1}^2]$ but not $\mathbb{E}[{u'_2}^2]$.

So we choose $P'_x = \text{Uniform}(-1,1)$. For P'_u, we sample the components independently, with $u'_1 \sim \text{Uniform}(-1,1)$, and $u'_2 \sim \text{Uniform}(-R,R)$. By choosing R large enough, we can make the lower bound for the input model arbitrarily large without impacting the risk of the baseline model which gives us

$$\mathbb{E}[R_{\text{ood}}(\hat{f}_{\text{in}})] > \mathbb{E}[R_{\text{ood}}(\hat{f}_{\text{bs}})]. \tag{48}$$

□

A.4 PRE-TRAINING IMPROVES RISK UNDER ARBITRARY COVARIATE SHIFT

Restatement of Theorem 1. *For all problem settings $\mathcal{S}$, noise distributions P_ϵ, test distributions P'_x, P'_u, and $n \geq m+d$ number of training points:*

$$\mathbb{E}[R_{ood}(\hat{f}_{out})] \leq \mathbb{E}[R_{ood}(\hat{f}_{bs})]. \tag{49}$$

First we show that pre-training (training a low-rank linear map from x to z) recovers the unobserved features w. We will then show that learning a regression map from w to y is better in all directions than learning a regression map from x to y.

Our first lemma shows that we can recover the map from x to w up to identifiability (i.e., we will learn the rowspace of the true linear map from x to w).

Lemma 1. *For a pair (x, z), let $z = A^\star B^\star x + \xi$ where $A^\star \in \mathbb{R}^{T\times k}$ and $B^\star \in \mathbb{R}^{k\times d}$ are the true parameters with $T, d \geq k$ and $\xi \in \mathbb{R}^T$ is mean-zero noise with bounded variance in each coordinate. Assume that $A^\star, B^\star$ are both rank k. Suppose that $\mathbb{E}[xx^\top]$ is invertible. Let $\hat{A}, \hat{B}$ be minimizers of the population risk $\mathbb{E}[\|\hat{A}\hat{B}x - z\|^2]$ of the multiple-output regression problem. Then span$\{B^\star{}_1, ..., B^\star{}_k\}$ = span$\{\hat{B}_1, ..., \hat{B}_k\}$ where $B^\star{}_i, \hat{B}_i$ are the i-th rows of their respective matrices.*

Proof. We first consider solving for the product of the weights $\hat{A}\hat{B}$. Letting C_i denote the i-th row of C, the population risk can be decomposed into the risks of the T coordinates of the output:

$$\mathbb{E}[\|Cx - z\|^2] = \sum_{i=0}^{T} \mathbb{E}[(C_i^\top x - z_i)^2] \tag{50}$$

$$= \sum_{i=0}^{T} \mathbb{E}[(C_i^\top x - (A^\star B^\star)_i^\top x - \xi_i)^2] \tag{51}$$

$$= \sum_{i=0}^{T} \mathbb{E}[(C_i^\top x - (A^\star B^\star)_i^\top x)^2] + \mathbb{E}[\xi_i^2] \tag{52}$$

Each term in the sum is the ordinary least squares regression loss, so a standard result is that since $\mathbb{E}[xx^T]$ is invertible, the unique minimizer is $C_i = (A^\star B^\star)_i$. One way to see this is to note that the loss is convex in C, and (by taking derivatives) if $\mathbb{E}[xx^T]$ is invertible the unique stationary point is $C = A^\star B^\star$. Therefore, we have that the product of the learned parameters and the true parameters are equal:

$$\hat{A}\hat{B} = A^\star B^\star \tag{53}$$

By e.g., Sylvester's rank inequality, $A^\star B^\star$ must be rank k, and so $\hat{A}\hat{B}$ is rank k (since they are equal). This means that $\hat{A}, \hat{B}$ are each rank k. Now $A^\star B^\star$ and $\hat{A}\hat{B}$ have the same rowspace because they are equal. The rowspace of $A^\star B^\star$ is a subspace of the rowspace of $B^\star$, but both have rank k so they are equal. Similarly the rowspace of $\hat{A}\hat{B}$ and $\hat{B}$ are equal. This implies that the rowspace of $\hat{B}$ and $B^\star$ are equal, which is the desired result. □

Our next lemma shows that for any fixed training examples X and *arbitrary* test example x', the aux-outputs model will have better expected risk than the baseline where the expectation is taken over the training labels $Y \mid X$.

Lemma 2. *In the linear setting, fix data matrix X and consider arbitrary test example x'. Let $\theta^\star = B^{\star\top}\theta_w$ be the optimal (ground truth) linear map from x to y. The expected excess risk of the aux-outputs model $\hat{B}^\top \hat{\theta}_{w,out}$ is better than for the baseline $\hat{\theta}_{x,ols}$, where the expectation is taken over the training targets $Y \sim P_{Y|X}$ (Y shows up implicitly because the estimators $\hat{\theta}_{w,out}$ and $\hat{\theta}_{x,ols}$ depend on Y):*

$$\mathbb{E}[(\hat{\theta}_{w,out}^\top \hat{B}x' - \theta^{\star\top} x')^2] \leq \mathbb{E}[(\hat{\theta}_{x,ols}^\top x' - \theta^{\star\top} x')^2] \tag{54}$$

Proof. Let $\epsilon_{all} = Y - X\theta^\star$ be the training noise. From standard calculations, the instance-wise risk of $\hat{\theta}_{x,ols}$ for any x is

$$\mathbb{E}[(\hat{\theta}_{x,ols}^\top x' - \theta^{\star\top} x')^2] = \mathbb{E}[(((X^\top X)^{-1} X^\top Y)^\top x' - \theta^{\star\top} x')^2] \tag{55}$$

$$= \mathbb{E}[((\theta^\star + (X^\top X)^{-1} X^\top \epsilon_{all})^\top x' - \theta^{\star\top} x')^2] \tag{56}$$

$$= \mathbb{E}[(((X^\top X)^{-1} X^\top \epsilon_{all})^\top x')^2] \tag{57}$$

$$= (\sigma^2 + \sigma_u^2) x'^\top (X^\top X)^{-1} x' \tag{58}$$

By Lemma 1, $\hat{B} = QB$ for some full rank Q. Thus, learning $\hat{\theta}_{w,out}$ is a regression problem with independent mean-zero noise and we can apply the same calculations for the instance-wise risk of $\hat{B}^\top \hat{\theta}_{w,out}$.

$$\mathbb{E}[(\hat{\theta}_{w,out}^\top \hat{B} x' - \theta^{\star\top} x')^2] = (\sigma^2 + \sigma_u^2) x'^\top \hat{B}^\top (\hat{B} X^\top X \hat{B}^\top)^{-1} \hat{B} x'. \tag{59}$$

We show that the difference between the inner matrices is positive semi-definite, which implies the result. In particular, we show that

$$(X^\top X)^{-1} - \hat{B}^\top (\hat{B} X^\top X \hat{B}^\top)^{-1} \hat{B} \succcurlyeq 0. \tag{60}$$

Since $X^\top X$ is a full rank PSD matrix, we can write $X^\top X = GG^\top$ for $G \in \mathbb{R}^{d \times d}$ where G is full rank and therefore invertible. Expressing Equation 60 in terms of G, we want to show:

$$(GG^\top)^{-1} - \hat{B}^\top (\hat{B} G G^\top \hat{B}^\top)^{-1} \hat{B} \succcurlyeq 0. \tag{61}$$

Left multiplying by $G^\top$ and right multiplying by G, which are both invertible, this is equivalent to showing:

$$M := I - (\hat{B}G)^\top (\hat{B} G G^\top \hat{B}^\top)^{-1} (\hat{B}G) \succcurlyeq 0. \tag{62}$$

But we note that M is symmetric, with $M = M^2 = MM^\top$, so M is PSD. This completes the proof.

□

Proof of Theorem 1. Fix training examples X and test example x' but let the train labels $Y \sim P_{Y|X}$ and and test label $y' \sim P'_{y'|x'}$ be random. In particular, let ${\sigma'_u}^2 = \mathbb{E}[(\theta_u^\top u')^2]$ where $u' \sim P'_u$, with $\mathbb{E}[u'] = 0$. Then for the baseline OLS estimator, we have:

$$\mathbb{E}[(y' - \hat{\theta}_{x,ols}^\top x')^2] = {\sigma'_u}^2 + \sigma^2 + \mathbb{E}[(\hat{\theta}_{x,ols}^\top x' - \theta^{\star\top} x')^2] \tag{63}$$

For the aux-outputs model, we have:

$$\mathbb{E}[(y' - \hat{\theta}_{w,out}^\top \hat{B} x')^2] = {\sigma'_u}^2 + \sigma^2 + \mathbb{E}[(\hat{\theta}_{w,out}^\top \hat{B} x' - \theta^{\star\top} x')^2] \tag{64}$$

So applying Lemma 2, we get that the risk for the aux-outputs model is better than for the baseline (the lemma showed it for the excess risk):

$$\mathbb{E}[(y' - \hat{\theta}_{w,out}^\top \hat{B} x')^2] \leq \mathbb{E}[(y' - \hat{\theta}_{x,ols}^\top x')^2] \tag{65}$$

Since this is true for all X and x', it holds when we take the expectation over the training examples X from P_x and the test example x' from P'_x which gives us the desired result. □

A.5 IN-N-OUT IMPROVES RISK UNDER ARBITRARY COVARIATE SHIFT

Restatement of Theorem 2. *In the linear setting, for all problem settings $\mathcal{S}$ with $\sigma_u^2 > 0$, test distributions P'_x, P'_u, $n \geq m + d$ number of training points, and $\delta > 0$, there exists $a, b > 0$ such that for all noise distributions P_ϵ, with probability at least $1 - \delta$ over the training examples and test example $x' \sim P'_x$, the ratio of the excess risks (for all σ^2 small enough that $a - b\sigma^2 > 0$) is:*

$$\frac{R^{ood}_{in\text{-}out} - R^*}{R^{ood}_{out} - R^*} \leq \frac{\sigma^2}{a - b\sigma^2} \tag{66}$$

Here $R^* = \min_{g^*, h^*} \mathbb{E}_{x', y', z' \sim P'}[\ell(g^*(h^*(x')), y')]$ is the min. possible (Bayes-optimal) OOD risk, $R^{ood}_{in\text{-}out} = \mathbb{E}_{y' \sim P'_{y'|x'}}[\ell(\hat{g}(\hat{h}_{out}(x')), y')]$ is the risk of the In-N-Out model on test example x', and $R^{ood}_{out} = \mathbb{E}_{y' \sim P'_{y'|x'}}[\ell(\hat{g}_{y\text{-}out}(\hat{h}_{out}(x')), y')]$ is the risk of the aux-outputs model on test example x'. Note that $R^{ood}_{in\text{-}out}$ and R^{ood}_{out} are random variables that depend on the test input x' and the training set X.

We first show a key lemma that lets us bound the min singular values of a random matrix, which will let us upper bound the risk of the In-N-Out estimator and lower bound the risk of the pre-training estimator.

Definition 1. *As usual, the min singular value $\tau_{min}(W)$ of a rectangular matrix $W \in \mathbb{R}^{n\times k}$ where $n \geq k$ refers to the k-th largest singular value (the remaining $n-k$ singular values are all 0), or in other words:*

$$\tau_{min}(W) = \min_{\|\nu\|_2=1} \|W\nu\|_2. \tag{67}$$

Lemma 3. *Let P_w and P_u be independent distributions on $\mathbb{R}^k$ and $\mathbb{R}^m$ respectively. Suppose they are absolutely continuous with respect to the standard Lebesgue measure on $\mathbb{R}^k$ and $\mathbb{R}^m$ respectively (e.g., this is true if they have density everywhere with density upper bounded). Let $W \in \mathbb{R}^{n\times k}$ where each row W_i is sampled independently $W_i \sim P_w$. Let $U \in \mathbb{R}^{n\times m}$ where each row U_i is sampled independently $U_i \sim P_u$. Suppose $n \geq k+m$. For all δ, there exists $c(\delta) > 0$ such that with probability at least $1-\delta$, the minimum singular values τ_{min} are lower bounded by $c(\delta)$: $\tau_{min}(W) > c(\delta)$ and $\tau_{min}([W;U]) > c(\delta)$.*

Proof. We note that the matrices W and U are rectangular, e.g., $W \in \mathbb{R}^{n\times k}$ where $n \geq k$. We will prove the lemma for W first, and the extension to $[W;U]$ will follow.

Note that removing the last $n-k$ rows of W cannot increase its min singular value since that corresponds to projecting the vector $W\nu$ and projection never increases the Euclidean norm. So without loss of generality, we suppose W only consists of its first k rows and so $W \in \mathbb{R}^{k\times k}$.

Now, consider any row W_i. We will use a volume argument to show that with probability at least $1-\frac{\delta}{k}$, this row W_i has a non-trivial component perpendicular to all the other rows. Since all rows are independently and identically sampled, without loss of generality suppose $i = 1$. Fix the other rows $W_2,...,W_k$, since W_1 is independent of these other rows, the conditional distribution of W_1 is the same as the marginal of W_1. The remaining rows $W_2,...,W_k$ form a $k-1$ dimensional subspace S in $\mathbb{R}^k$. Letting $d(w,S)$ denote the Euclidean distance of a vector w from the subspace S, define the event $S_\lambda = \{W_1 : d(W_1,S) \leq \lambda\}$. Since P_w is absolutely continuous, $P(S_\lambda) \to 0$ as $\lambda \to 0$, so for some small $c(\delta) > 0$, $P(S_{c(\delta)}) < \frac{\delta}{k}$. So with probability at least $1-\delta/k$, $d(W_1,S) > c(\delta)$.

By union bound, with probability at least $1-\delta$, the distance from W_i to the subspace spanned by all the other rows is greater than $c(\delta)$ for every row W_i, so we condition on this. By representing each row vector as the sum of the component perpendicular to S and a component in S, applying Pythagoras theorem and expanding we get

$$\min_{\|\nu\|_2=1} \|W\nu\| = \min_{\|\nu\|_2=1} \|\nu^\top W\| \geq c(\delta). \tag{68}$$

Which completes the proof for $\tau_{min}(W)$.

For $[W;U]$, we note that P_x and P_u are independent, and the product measure is absolutely continuous. Since each row of $[W;U]$ is identically and independently sampled just like with W, we can apply the exact same argument as above (though for a different constant $c(\delta)$, we take the min of these two as our $c(\delta)$ in the lemma statement). □

Recall that the In-N-Out estimator was obtained by fitting a model from w,z to y, and then using that to produce pseudolabels on (infinite) unlabeled data, and then self-training a model from w to y on these pseudolabels. For the linear setting, we defined the In-N-Out estimator $\hat{\theta}_w$ in Equation 71. Our next lemma gives an alternate closed form of the In-N-Out estimator in terms of the representation matrix $W = X\hat{B}$ and the latent matrix U.

Lemma 4. *In the linear setting, letting $W = X\hat{B}^\top$ we can write the In-N-Out estimator in closed form:*

$$\hat{\theta}_w = [I_{k\times k}; 0_{k\times T}]\left(\begin{pmatrix} W^\top \\ U^\top \end{pmatrix}(W;U)\right)^{-1}\begin{pmatrix} W^\top \\ U^\top \end{pmatrix} Y. \tag{69}$$

Proof. We recall the definition of the In-N-Out estimator, where we first train a classifier from W,Z to Y:

$$\hat{\gamma_w},\hat{\gamma_z} = \underset{\hat{\gamma_w},\hat{\gamma_z}}{\operatorname{argmin}} \|Y - (W\hat{\gamma_w} + Z\hat{\gamma_z})\|_2. \tag{70}$$

Denote the minimum value of Equation 70 by p^*. Note that $\hat{\gamma_w}, \hat{\gamma_z}$ may not be unique, and we pick any solution to the argmin (although our proof will reveal that the resulting $\hat{\theta}_w$ is in fact unique). We then use this to produce pseudolabels and self-train, on infinite data, which gives us the In-N-Out estimator:

$$\hat{\theta}_w = \operatorname*{argmin}_{\hat{\theta}_w} \mathop{\mathbb{E}}_{w,z\sim P_{\text{id}}}\left[\left(\hat{\theta}_w^\top w - (\hat{\gamma_w}^\top w + \hat{\gamma_z}^\top z)\right)^2\right] \tag{71}$$

Subtituting $z = A^\star w + C^\star u$, we can write the loss that the In-N-Out estimator $\hat{\theta}_w$ minimizes as:

$$\mathop{\mathbb{E}}_{w,z\sim P_{\text{id}}}\left[\left(\hat{\theta}_w^\top w - ((\hat{\gamma_w} + A^{\star\top}\hat{\gamma_z})^\top w + (C^{\star\top}\hat{\gamma_z})^\top u)\right)^2\right] \tag{72}$$

We group the terms slightly differently:

$$\mathop{\mathbb{E}}_{w,z\sim P_{\text{id}}}\left[\left((\hat{\theta}_w^\top w - (\hat{\gamma_w} + A^{\star\top}\hat{\gamma_z})^\top w) - (C^{\star\top}\hat{\gamma_z})^\top u\right)^2\right] \tag{73}$$

Expanding the square, using the fact that $\mathbb{E}[u] = 0$, and u, w are independent, and ignoring terms with no dependency on $\hat{\theta}_w$, this is equivalent to minimizing:

$$\mathop{\mathbb{E}}_{w\sim P_{\text{id}}}\left[\left(\hat{\theta}_w^\top w - (\hat{\gamma_w} + A^{\star\top}\hat{\gamma_z})^\top w\right)^2\right] \tag{74}$$

This is minimized (indeed it is 0) by setting:

$$\hat{\theta}_w = \hat{\gamma_w} + A^{\star\top}\hat{\gamma_z} \tag{75}$$

The minimizer is unique because w has invertible covariance matrix (since x has invertible covariance matrix and $B^\star$ is full rank), and so is in every direction with some probability. We will now consider the following alternative estimator:

$$\hat{\theta'}_w, \hat{\theta'}_u = \operatorname*{argmin}_{\hat{\theta'}_w, \hat{\theta'}_u} \|Y - (W\hat{\theta'}_w + U\hat{\theta'}_u)\|_2. \tag{76}$$

Denote the minimum value of Equation 76 by q^*. We claim that $\hat{\theta'}_w = \hat{\theta}_w$.

We will show that the In-N-Out estimator $\hat{\theta}_w$ minimizes the alternative minimization problem in Equation 76 by showing that $p^* = q^*$. We will then show that the solution to Equation 76 is unique, which implies that $\hat{\theta'}_w = \hat{\theta}_w$.

We note that $C^{\star\top} \in \mathbb{R}^{m\times T}$ where $T \geq m$ is full-rank, so there exists a right-inverse C' with $C^{\star\top}C' = I_{m\times m}$. Since $Z = WA^{\star\top} + UC^{\star\top}$, this gives us $U = (Z - WA^{\star\top})C' = ZC' + W(-A^{\star\top}C')$.

So this means that a solution to the alternative problem in Equation 76 can be converted into a solution for the original in Equation 70 with the same function value:

$$\min_{\hat{\theta'}_w, \hat{\theta'}_u} \|Y - (W\hat{\theta'}_w + U\hat{\theta'}_u)\|_2 \tag{77}$$

$$= \min_{\hat{\theta'}_w, \hat{\theta'}_u} \|Y - (W\hat{\theta'}_w + (ZC' + W(-A^{\star\top}C'))\hat{\theta'}_u)\|_2 \tag{78}$$

$$= \min_{\hat{\theta'}_w, \hat{\theta'}_u} \|Y - (W(\hat{\theta'}_w - A^{\star\top}C'\hat{\theta'}_u) + Z(C'\hat{\theta'}_u))\|_2. \tag{79}$$

This implies that $p^* \leq q^*$.

We now show that a solution to the original problem in Equation 70 can be converted into a solution for the alternative in Equation 76 with the same function value:

$$\min_{\hat{\gamma_w}, \hat{\gamma_z}} \|Y - (W\hat{\gamma_w} + Z\hat{\gamma_z})\|_2 \tag{80}$$

$$= \min_{\hat{\gamma_w}, \hat{\gamma_z}} \|Y - (W\hat{\gamma_w} + (WA^{\star\top} + UC^{\star\top})\hat{\gamma_z})\|_2 \tag{81}$$

$$= \min_{\hat{\gamma_w}, \hat{\gamma_z}} \|Y - (W(\hat{\gamma_w} + A^{\star\top}\hat{\gamma_z}) + U(C^{\star\top}\hat{\gamma_z}))\|_2. \tag{82}$$

This implies that $q^* \leq p^*$, and we showed before that $p^* \leq q^*$ so $p^* = q^*$. But since $\hat{\gamma_w}, \hat{\gamma_z}$ minimizes the original minimizer in Equation 70, $\hat{\gamma_w} + A^{\star\top}\hat{\gamma_z}, C^{\star\top}\hat{\gamma_z}$ minimize the alternative problem in Equation 76, where $\hat{\theta}_w = \hat{\gamma_w} + A^{\star\top}\hat{\gamma_z}$.

Since $[W;U]$ is full rank, the solution $\hat{\theta}'_w, \hat{\theta}'_u$ to the alternative estimator Equation 76 is unique. So this means that $\hat{\theta}'_w = \hat{\theta}_w$.

We have shown that $\hat{\theta}'_w = \hat{\theta}_w$—this completes the proof because solving Equation 76 for $\hat{\theta}'_w$ gives us the closed form in Equation 69.

□

Next we show a technical lemma that says that if a random vector $u \in \mathbb{R}^n$ has bounded density everywhere, then for any v with high probability the dot product $(u^\top v)^2$ cannot be too small relative to $\|v\|_2^2$.

Lemma 5. *Suppose a random vector $u \in \mathbb{R}^n$ has density everywhere, with bounded density. For every δ, there exists some $c(\delta)$ such that for all v, with probability at least $1-\delta$ over u, $(u^\top v)^2 \geq c(\delta)\|v\|_2^2$.*

Proof. First, we choose some B_0 such that $P(\|u\|_2 \geq B_0) \leq \delta/2$, such a B_0 exists for every probability measure.

Suppose that the density is upper bounded by B_1. Let the area of the $n-1$ dimensional sphere with radius B_0 be A_0. Consider any $n-1$ dimensional subspace S, and let $S_\epsilon = \{u' : d(u', S) \leq \epsilon\}$ where $d(u', S)$ denotes the Euclidean distance from u' to S. We have $P(u \in S_\epsilon) \leq A_0 B_1 \epsilon + \delta/2$ for all S_ϵ. By choosing sufficiently small $\epsilon > 0$, we can ensure that $P(u \in S_\epsilon) \leq \delta$ for all S.

Now consider arbitrary v and let $S(v)$ be the $n-1$-dimensional subspace perpendicular to v. We have $P(u \in S(v)_\epsilon) \leq \delta$. But this means that $(u^\top v)^2 \geq \epsilon^2 \|v\|_2^2$ with probability at least $1-\delta$, which completes the proof. □

By definition of our linear multi-task model, we recall that $y = \theta_w^\top w + \theta_u^\top u + \epsilon$, where $w = B^\star x$. We do not have access to $B^\star$, but we assumed that $B^\star$ is full rank. We learned $\hat{B}$ which has the same rowspace as $B^\star$ (Lemma 1). This means that for some θ'_w, we have $y = \theta_w'^\top \hat{w} + \theta_u^\top u + \epsilon$ where $\hat{w} = \hat{B}x$. To simplify notation and avoid using θ'_w and $\hat{w}$ everywhere, we suppose without loss of generality that $\hat{B} = B^\star$ (but formally, we can just replace all the occurrences of θ_w by θ'_w and w by $\hat{w}$).

Our next lemma lower bounds the test error of the pre-training model.

Lemma 6. *In the linear setting, for all problem settings $\mathcal{S}$ with $\sigma_u^2 > 0$, for all δ, there exists some $a, b > 0$ such that with probability at least $1-\delta$ over the training examples and test example $x' \sim P'_x$ the risk of the aux-outputs model is lower bounded:*

$$R^{ood}_{out} - R^* > a - b\sigma^2. \tag{83}$$

Proof. Recall that $R^{\text{ood}}_{\text{out}} = \mathbb{E}_{y' \sim P'_{y'|x'}}[l(g_{\text{y-out}}(\hat{h}_{\text{out}}(x')), y')]$. Let ${\sigma'_u}^2 = \mathbb{E}_{u' \sim P'_u}[(\theta_u^\top u')^2]$. We have $R^* = \sigma^2 + {\sigma'_u}^2$. Let $W = XB^{\star\top}$ be the feature matrix, where $W \in \mathbb{R}^{n \times k}$.

Letting $w' = B^\star x'$, for the aux-outputs model we have

$$\mathop{\mathbb{E}}_{y' \sim P'_{y'|x'}}[l(g_{\text{y-out}}(\hat{h}_{\text{out}}(x')), y')] \tag{84}$$

$$= \mathop{\mathbb{E}}_{y' \sim P_{y'|x'}}[(y' - \hat{\theta}_{w,out}^\top w')^2] \tag{85}$$

$$= (\sigma^2 + {\sigma'_u}^2) + (\theta_w^\top w' - \hat{\theta}_{w,out}^\top w')^2 \tag{86}$$

$$= R^* + (\theta_w^\top w' - \hat{\theta}_{w,out}^\top w')^2. \tag{87}$$

Let $\epsilon = Y - (W\theta_w + U\theta_u)$ be the noise of Y for the training examples, which is a random vector with $\epsilon \in \mathbb{R}^n$. We can now write

$$(\theta_w^\top w' - \hat{\theta}_{w,out}^\top w')^2 = ((\epsilon + U\theta_u)^\top W (W^\top W)^{-1} w')^2. \tag{88}$$

By assumption, $W^\top W$ is invertible (almost surely). With probability at least $1-\delta/10$ all entries of $W^\top W$ are upper bounded and we condition on this. So $(W^\top W)^{-1}$ has min singular value bounded below. By Lemma 3, W has min singular value that is bounded below with probability at least $1-\delta/10$. We condition on this being true. So let $\nu = W(W^\top W)^{-1}w'$, so for some $c_0 > 0$, we have $\|\nu\|_2 \geq c_0\|w'\|_2$.

In terms of ν, we can write Equation 88 as

$$(\theta_w^\top w' - \hat{\theta}_{w,out}^\top w')^2 = ((\epsilon + U\theta_u)^\top \nu)^2 \tag{89}$$

$$= (\epsilon^\top \nu)^2 + ((U\theta_u)^\top \nu)^2 + 2(\epsilon^\top \nu)((U\theta_u)^\top \nu) \tag{90}$$

$$\geq ((U\theta_u)^\top \nu)^2 + 2(\epsilon^\top \nu)((U\theta_u)^\top \nu) \tag{91}$$

$$\geq ((U\theta_u)^\top \nu)^2 - 2|\epsilon^\top \nu|\|(U\theta_u)^\top\|_2\|\nu\|_2). \tag{92}$$

We can find b_0 such that with at least probability $1-\delta/10$, $\|(U\theta_u)^\top\|_2 \leq b_0$, condition on this. We note that $\epsilon^\top \nu$ has variance $\sigma^2\|\nu\|_2$ so by Chebyshev for some b_1 with probability at least $1-\delta/10$, $|\epsilon^\top \nu| \leq b_1\sigma^2\|\nu\|_2$, condition on this. So we can now bound Equation 92 and get:

$$(\theta_w^\top w' - \hat{\theta}_{w,out}^\top w')^2 \geq ((U\theta_u)^\top \nu)^2 - 2b_0b_1\sigma^2\|\nu\|_2^2 \tag{93}$$

Now we apply Lemma 5, where we use the fact that $\sigma_u^2 > 0$. So given $\delta/10$, there exists some c_1 such that for every ν with probability at least $1-\delta/10$, $((U\theta_u)^\top \nu)^2 \geq c_1\|\nu\|_2^2$, giving us

$$(\theta_w^\top w' - \hat{\theta}_{w,out}^\top w')^2 \geq (c_1 - 2b_0b_1\sigma^2)\|\nu\|_2^2. \tag{94}$$

Since w' has bounded density everywhere, it is non-atomic and we get that there is some $c_2 > 0$ such that with probability at least $1-\delta/10$, $\|w'\|_2^2 \geq c_2^2$. But then $\|\nu\|_2^2 \geq c_0c_2$, which gives us for some a,b,

$$(\theta_w^\top w' - \hat{\theta}_{w,out}^\top w')^2 \geq (c_1 - 2b_0b_1\sigma^2)c_0c_2 \geq a - b\sigma^2. \tag{95}$$

Combining this with Equation 87, we get with probability at least $1-\delta$,

$$\mathop{\mathbb{E}}_{y' \sim P'_{y'|x'}}[l(g_{\text{y-out}}(\hat{h}_{\text{out}}(x')), y')] - R^* > a - b\sigma^2, \tag{96}$$

as desired. □

Lemma 7. *In the linear setting, for all problem settings $\mathcal{S}$, for all δ, there exists some $c > 0$ such that with probability at least $1-\delta$ over the training examples and test example $x' \sim P'_x$ the risk of the In-N-Out model is upper bounded:*

$$R^{ood}_{\text{in-out}} - R^* < c\sigma^2. \tag{97}$$

Proof. Recall that $R^{\text{ood}}_{\text{in-out}} = \mathbb{E}_{y' \sim P'_{y'|x'}}[l(\hat{g}(\hat{h}_{\text{out}}(x')), y')]$. Let ${\sigma'_u}^2 = \mathbb{E}_{u' \sim P'_u}[(\theta_u^\top u')^2]$. We have $R^* = \sigma^2 + {\sigma'_u}^2$. As before, let $W = XB^{\star\top}$ be the feature matrix, where $W \in \mathbb{R}^{n \times k}$.

Let $W_U = [W; U]$ which denotes concatenating the matrices by column, so that $W_U \in \mathbb{R}^{n \times (k+m)}$. By Lemma 3, ${W_U}^\top W_U$ has min singular value that is bounded below by c_0 with probability at least $1-\delta/10$, we condition on this being true. Now, as for the aux-outputs model, letting $w' = B^\star x'$, we have

$$\mathop{\mathbb{E}}_{y' \sim P_{y'|w'}}[(y' - \hat{\theta}_w^\top w')^2] = R^* + (\theta_w^\top w' - \hat{\theta}_w^\top w')^2. \tag{98}$$

For the second term on the RHS: Let $R = [I_{k\times k}; 0_{k \times m}]$. Let $\epsilon = Y - (W\theta_w + U\theta_u)$ be the noise of Y for the training examples, which is a random vector with $\epsilon \in \mathbb{R}^n$. From Lemma 4, we can now write:

$$(\theta_w^\top w' - \hat{\theta}_w^\top w')^2 = (w'^\top R({W_U}^\top W_U)^{-1}{W_U}^\top \epsilon)^2. \tag{99}$$

$\|w'\|_2$ is bounded above by some constant B_1 with probability at least $1-\delta/10$ which we condition on. Now taking the expectation over w' and ϵ, using the fact that R preserves the norm of a vector

we can write

$$\mathop{\mathbb{E}}_{w',\epsilon}[(w'^{\top}R({W_U}^{\top}W_U)^{-1}{W_U}^{\top}\epsilon)^2] \tag{100}$$

$$=\sigma^2 \mathop{\mathbb{E}}_{w',\epsilon}[(w'^{\top}R[{W_U}^{\top}W_U]^{-1}R^T w')] \tag{101}$$

$$\leq \frac{\sigma^2}{c_0^2}\mathop{\mathbb{E}}_{w'}[\|w'\|_2^2] \tag{102}$$

$$\leq \frac{B_1^2\sigma^2}{c_0^2}. \tag{103}$$

Then, by Markov's inequality, with probability at least $1-\delta/10$ we can upper bound Equation 99 by $\frac{10B_1^2\sigma^2}{\delta c_0^2}$. In total, that gives us that for some c, with probability at least $1-\delta$:

$$\mathop{\mathbb{E}}_{y'\sim P'_{y'|x'}}[l(\hat{g}(\hat{h}_{\text{out}}(x')),y')]-R^* < c\sigma^2. \tag{104}$$

□

The proof of Theorem 2 simply combines Lemma 6 and Lemma 7.

Proof of Theorem 2. For some a,b,c, with probability at least $1-\delta$, we have for the aux-outputs model:

$$R^{\text{ood}}_{\text{out}} - R^* > a - b\sigma^2, \tag{105}$$

and for the In-N-Out model:

$$R^{\text{ood}}_{\text{in-out}} - R^* < c\sigma^2. \tag{106}$$

Taking ratios and dividing by suitable constants we get the desired result. □

B Experimental details

B.1 CelebA

For the results in Table 1, we used 7 auxiliary binary attributes included in the CelebA dataset: `['Bald', 'Bangs', 'Mustache', 'Smiling', '5_o_Clock_Shadow', 'Oval_Face', 'Heavy_Makeup']`. These attributes tend to be fairly robust to our distribution shift (not hat vs. hat) — if the person has a 5 o'clock shadow, the person is likely a man. We use a subset of the CelebA dataset with 2000 labeled examples, 30k in-distribution unlabeled examples, 3000 OOD unlabeled examples, and 1000 validation, in-distribution test, and OOD test examples each. We report numbers averaged over 5 trials, where on each trial, the in-distribution labeled / unlabeled examples are randomly re-sampled while the validation and test sets are fixed. The backbone for all models is a ResNet-18 (He et al., 2016) which takes a CelebA image downsized to 64×64 and outputs a binary gender prediction. All models are trained for 25 epochs using SGD with cosine learning rate decay, initial learning rate 0.1, and early stopped with an in-distribution validation set. The gender ratios in the in-distribution and OOD set are balanced to 50-50.

Aux-inputs model. We incorporate the auxiliary inputs by first training a baseline model $\hat{f}_{\text{bs}}$ from images to output logit, then training a logistic regression model on the concatenated features $[\hat{f}_{\text{bs}}(x);z]$ where z are the auxiliary inputs. We sweep over L2 regularization hyperparameters $C = [0.1,0.5,1.0,5.0,10.0,20.0,50.0]$ and choose the best with respect to an in-distribution validation set.

Aux-outputs model. During pretraining, the model trains on the 7-way binary classification task of predicting the auxiliary information. Then, the model is finetuned on the gender classification task without auxiliary information.

In-N-Out and repeated self-training. For In-N-Out models with repeated self-training, we pseudolabeled all the unlabeled data using the In-N-Out model and did one round of additional self-training. Following (Kumar et al., 2020), we employ additional regularization when doing self training by adding dropout with probability 0.8. We also reduced the learning rate to 0.05 to improve the training dynamics.

Adding auxiliary inputs one-by-one. In Figure 5, we generate a random sequence of 15 auxiliary inputs and add them one-by-one to the model, retraining with every new configuration. We use the following auxiliary information: `'Young'`, `'Straight_Hair'`, `'Narrow_Eyes'`, `'Mouth_Slightly_Open'`, `'Blond_Hair'`, `'5_o_Clock_Shadow'`, `'Big_Nose'`, `'Oval_Face'`, `'Chubby'`, `'Attractive'`, `'Blurry'`, `'Goatee'`, `'Heavy_Makeup'`, `'Wearing_Necklace'`, and `'Bushy_Eyebrows'`.

Correlation between in-distribution and OOD accuracy. In Figure 4, we sample 100 random sets of auxiliary inputs of sizes 1 to 15 and train 100 different aux-inputs models using these auxiliary inputs. We plot the in-distribution and OOD accuracy for each model, showing that there is a significant correlation between in-distribution and OOD accuracy in CelebA, supporting results on standard datasets (Recht et al., 2019; Xie et al., 2020; Santurkar et al., 2020). Each point in the plot is an averaged result over 5 trials.

B.2 CROPLAND

All models reported in Table 1 were trained using the Adam optimizer with learning rate 0.001, a batch size of 256, and 100 epochs unless otherwise specified. Our dataset consists of about 7k labeled examples, 170k unlabeled examples (with 130k in-distribution examples), 7.5k examples each for validation and in-distribution test, and 4260 OOD test examples (the specification of OOD points is described in further detail below). Results are reported over 5 trials, and $\lambda \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ was chosen using the validation set. On each trial, the in-distribution labeled / unlabeled examples are randomly re-sampled while the validation and test sets are fixed.

Problem Motivation. Developing machine learning models trained on remote sensing data is currently a popular line of work for practical problems such as typhoon rainfall estimation, monitoring reservoir water quality, and soil moisture estimation (Lary et al., 2016; Maxwell et al., 2018; Ahmad et al., 2010). Models that could use remote sensing data to accurately forecast crop yields or estimate the density of regions dedicated to growing crops would be invaluable in important tasks like estimating a developing nation's food security (Li et al., 2007).

OOD Split. In remote sensing problems it is often the case that certain regions lack labeled data (e.g., due to a lack of human power to gather the labels on site), so extrapolation to these unlabeled regions is necessary. To simulate this data regime, we use the provided (lat, lon) pairs of each data point to split the dataset into labeled (in-distribution) and unlabeled (out-of-distribution) portions. Specifically, we take all points lying in Iowa, Missouri, and Illinois as our ID points and use all points within Indiana and Kentucky as our OOD set.

Shape of auxiliary info. To account for the discrepancy in shapes of the two sources of auxiliary information (latitude and longitude are two scalar measurements while the 3 vegetation bands form a $3 \times 50 \times 50$ tensor), we create latitude and longitude "bands" consisting of two 50×50 matrices that repeat the latitude and longitude measurement, respectively. Concatenating the vegetation bands and these two pseudo-bands together gives us an overall auxiliary dimension of $5 \times 50 \times 50$.

UNet. Since our auxiliary information takes the form of 50×50 bands, we need a model architecture that can reconstruct these bands in order to implement the aux-outputs and the In-N-Out models. With this in mind, we utilize a similar UNet architecture that Wang et al. (2020) use on the same Cropland dataset. While the UNet was originally proposed by Ronneberger et al. (2015) for image segmentation, it can be easily modified to perform image-to-image translation. In particular, we remove the final 1×1 convolutional layer and sigmoid activation that was intended for binary segmentation and replace them with a single convolutional layer whose output dimension matches that of the auxiliary information. In our case, the last convolutional layer has an output dimension of 5 to reconstruct the 3 vegetation bands and (lat,lon) coordinates.

To perform image-level binary classification with the UNet, we also replace the final 1×1 convolutional layer and sigmoid activation, this time with a global average pool and a single linear layer with an output dimension of 1. During training we apply a sigmoid activation to this linear layer's output to produce a binary class probability, which is then fed into the binary cross entropy loss function.

Aux-inputs model. Since the original RGB input image is $3 \times 50 \times 50$, we can simply concatenate the auxiliary info alongside the original image to produce an input of dimensions $8 \times 50 \times 50$ to feed into the UNet.

Aux-outputs model. The modification of the traditional UNet architecture in order to support auxiliary outputs for Cropland is described in the above UNet section. We additional add a $\tanh$ activation

function to squeeze the model's output values to the range $[-1,1]$ (the same range as the images). We train the model to learn the auxiliary bands via pixel-wise regression using the mean squared error loss.

In-N-Out model. We found that the finetuning phase of the In-N-Out algorithm experienced wild fluctuations in loss and would not converge when using the hyperparameters listed at the top of this section. To encourage the model to converge and fit the training set, we decreased the Adam learning rate to 0.0001 and doubled the batch size to 512.

Repeated self-training. For the additional round of self-training, we initialize training and pseudolabel all unlabeled data with the In-N-Out model. Following (Kumar et al., 2020), we employ additional regularization when doing self training by adding dropout with probability 0.8.

B.3 LANDCOVER

Our Landcover dataset comes from NASA's MODIS Surface Reflectance product, which is made up of measurements from around the globe taken by the Terra satellite (Vermote, 2015). In each trial, we use about 16k labeled examples from non-African locations, 203k unlabeled examples (with 150k in-distribution examples), 9266 examples each for validation and in-distribution test, and 4552 OOD test examples. We trained with SGD + momentum (0.9) on all models for 400 epochs with a cosine learning rate schedule. We used learning rate 0.1 for all models that were not pre-trained, and learning rate 0.01 for models that were already pre-trained. Results are reported over 5 trials, and $\lambda \in \{0.1,0.3,0.5,0.7,0.9\}$ was chosen using the validation set. On each trial, the in-distribution labeled / unlabeled examples are randomly re-sampled while the validation and test sets are fixed.

1D CNN While Convolutional Neural Networks are most commonly associated with the groundbreaking success of 2D-CNNs on image-based tasks, the 1-dimensional counterparts have also found success in various applications (Kiranyaz et al., 2019). Because the measurements from the MODIS satellite are not images but instead scalar-valued time series data, we can use a 1D CNN with 7 channels, one for each of the 7 MODIS sensors.

NDVI The normalized difference vegetation index (NDVI) is a remote sensing measurement indicating the presense of live green vegetation. It has been shown to be a useful predictor in landcover-related tasks (DeFries and Townshend, 1994; DeFries et al., 1995; Lunetta et al., 2006), so we choose to include it in our models as well. NDVI can be computed from the RED and NIR bands of the MODIS sensors via the equation

$$\text{NDVI} = (\text{NIR} - \text{RED})/(\text{NIR} + \text{RED}). \tag{107}$$

We include NDVI along with the 7 other MODIS bands to give us input dimensions of 46×8.

ERA5 It is a reasonable hypothesis that having additional climate variables such as soil type or precipitation could be useful for a model in inferring the underlying landcover class. To this end we incorporate features from the ERA5 climate dataset as our auxiliary information (C3S, 2017). The specific variables we include are soil type, temperature, precipitation rate, precipitation total, solar radiation, and cloud cover. For each MODIS point we find its nearest ERA5 neighbor based on their latitude and longitude in order to pair the datasets together.

The ERA5 measurements are monthly averages, which means the readings are at a different frequency than that of the 8-day MODIS time series. We upsample the ERA5 signal using the `scipy.signal.resample` method, which uses the FFT to convert to the frequency domain, adds extra zeros for upsampling to the desired frequency, and then transforms back into the time domain.

Landcover classes. The Landcover dataset has a total of 16 landcover classes, with a large variance in the individual class counts. To ensure our model sees enough examples of each class, we filtered the dataset to include just 6 of the most populous classes: `savannas`, `woody_savannas`, `croplands`, `open_shrublands`, `evergreen_broadleaf_forests`, and `grasslands`.

Aux-inputs model. We concatenate the resampled ERA5 readings with the MODIS and NDVI measurements to obtain an input dimension of 46×14.

Aux-outputs model. Rather than predicting the entire ERA5 time series as an auxiliary output, we instead average the 6 climate variables over the time dimension and predict those 6 means as the auxiliary outputs. We use a smaller learning rate of 0.01 for this pre-trainined model.

In-N-Out and Repeated self-training. The In-N-Out model initializes its weights from the aux-outputs model and gets pseudolabeled ID unlabeled data from the aux-inputs model. As with aux-outputs, we use a smaller learning rate of 0.01 for this pre-training model.

For the additional round of self-training, we initialize training and pseudolabel all unlabeled data with the In-N-Out model. Following (Kumar et al., 2020), we employ additional regularization when doing self training by adding dropout with probability 0.5. We found that with dropout, we need a higher learning rate (0.1) to effectively fit the training set.