# Task-Assisted Domain Adaptation with Anchor Tasks

Zhizhong Li*[1,2], Linjie Luo[3], Sergey Tulyakov[2], Qieyun Dai*[2], and Derek Hoiem[1]

[1]Computer Science, University of Illinois, Urbana Champaign, IL, USA
{zli115, dhoiem}@illinois.edu
[2]Snap Inc., Santa Monica, CA, USA
stulyakov@snap.com, qieyunmarydai@gmail.com
[3]Bytedance Inc., Mountain View, CA, USA
linjie.luo@bytedance.com

## Abstract

*Some tasks, such as surface normals or single-view depth estimation, require per-pixel ground truth that is difficult to obtain on real images but easy to obtain on synthetic. However, models learned on synthetic images often do not generalize well to real images due to the domain shift. Our key idea to improve domain adaptation is to introduce a separate anchor task (such as facial landmarks) whose annotations can be obtained at no cost or are already available on both synthetic and real datasets. To further leverage the implicit relationship between the anchor and main tasks, we apply our* HEADFREEZE *technique that learns the cross-task guidance on the source domain with the final network layers, and use it on the target domain. We evaluate our methods on surface normal estimation on two pairs of datasets (indoor scenes and faces) with two kinds of anchor tasks (semantic segmentation and facial landmarks). We show that blindly applying domain adaptation or training the auxiliary task on only one domain may hurt performance, while using anchor tasks on both domains is better behaved. Our* HEADFREEZE *technique outperforms competing approaches, reaching performance in facial images on par with a recently popular surface normal estimation method using shape from shading domain knowledge.*

## 1. Introduction

Collecting annotations is difficult for geometric tasks, such as predicting depth [30, 41], surface normals [27], and 3D pose [37], because it usually requires a specialized device and access to the scene. Synthetic images and their geometric labels are easily generated, but synthetically trained models often do not generalize well to real data. Unsupervised domain adaptation methods [32, 25, 26, 12] can help, but they often blindly minimize domain distribution difference [9, 35, 28, 34] even when the ground truth distributions in source and target differ. How can we better adapt from synthetic to real data?

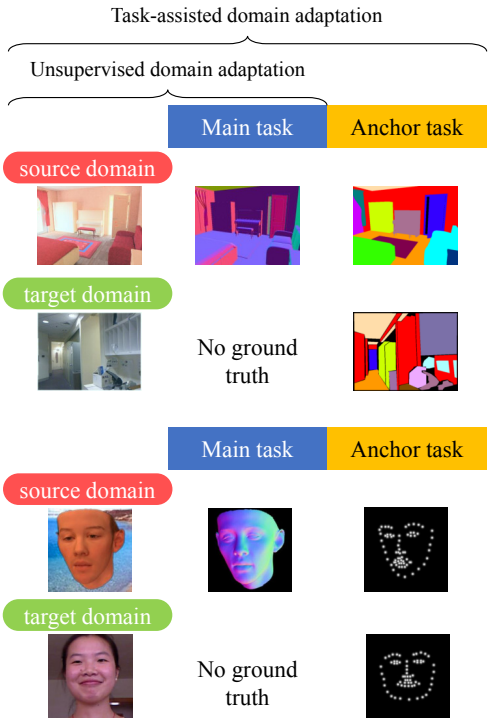


Figure 1. Illustration of our formulation compared to unsupervised domain adaptation. Although target domain main task labels are hard or expensive to obtain, we can use free or already available labels from an "anchor" task to help align the domains with clear correspondence between images and the anchor task label space.

*Work done prior to current employment. Partially done during the first author's internship at Snap Inc.

In this paper, we propose to use *anchor tasks* as a guide for improving pixel-level domain transfer of the *main task*. The anchor task is a task labeled on both domains, whose annotations are already available or automatically generated. For example, in one experiment we improve transfer for surface normal prediction on faces by using facial keypoint detection as an anchor task, and the anchor task ground truth is estimated using an off-the-shelf model. We propose our HEADFREEZE method that first trains the main task and anchor task on synthetic data, then freezes the top few layers and retrain the feature layers to perform the main task on synthetic data and the anchor task on both domains. The fully-supervised anchor task provides additional semantic and spatial context information to learn better feature representations for the real images (hence the name "anchor"), while the frozen predictors leverage learned "cross-task guidance" so that the main task on the target domain can be guided by the anchor task.

Our idea can be seen as a generalization of existing works that use a closely related auxiliary task to help adaptation. We call this family of methods "Task-assisted Domain Adaptation" (TADA). Prior work [10, 39, 14, 6] in the TADA family all rely on problem-specific, explicitly defined mappings between the auxiliary tasks and the main tasks (see Section 2 for details), and thus are restricted to their own task pair and problem settings. Unlike prior work, we require only that the anchor task has pixel labels, and HEADFREEZE is applicable even when the main-anchor relationship lacks an explicit formulation (e.g. between facial keypoints and surface normal map). We demonstrate this with different anchor tasks for the same main task without changing the framework. Our experiments focus on geometric tasks with synthetic and real images as the source and target domains, but our approach also applies to other pixel labeling domain transfer problems.

The anchor task helps domain adaptation in two ways. First, learning shared features for the anchor task on both domains and the main task on the source domain encourages that the features are effective for the main task in the target domain. Second, there may be a multitask learning benefit, if the anchor task and main task have related labels (e.g. "ceiling" is always horizontal) or the same features are useful for both tasks.

Our HEADFREEZE method strengthens the first benefit, while being simple enough not to require domain knowledge on the task pair. Specifically, when our network finishes training on the source domain, its final layers have learned not to output unlikely main-anchor prediction pairs (e.g. flat nose, misaligned object edges) but to output likely pairs. We term this knowledge "**cross-task guidance**". But this guidance may be ignored if the network overfits to target domain anchor task and outputs unlikely pairs at will. Freezing the final layers fixes the guidance, and ensures the main and anchor task classifiers continue to rely on the same features and the same mapping from feature space to label space.

We evaluate our methods on two pairs of synthetic and real datasets, performing surface normal estimation in indoor scenes and faces, using semantic segmentation and facial landmark as anchor tasks separately. We show the importance of having anchor labels in *both* domains instead of just one, and that HEADFREEZE outperforms compared approaches, reaching results in facial images on par with a popular recent model SfSNet [27] that leverages a domain-specific illumination model. We also find that, surprisingly, distribution matching adaptation methods can sometimes hurt performance when the label distributions are different between domains, where HEADFREEZE's performance is better behaved in our experiments.

In summary, our main contributions are:

- We propose a novel domain adaptation formulation for pixel-labeling main tasks from synthetic to real, using free or readily available anchor task labels. Our formulation is more widely applicable than existing domain adaptation work that leverages auxiliary tasks (but more restricted than unsupervised domain adaptation).
- We introduce a HEADFREEZE technique to further utilize the spatial and contextual cross-task guidance, that can be applied to different pairings of anchor tasks with the same main task.

## 2. Related work

**Domain adaptation methods using auxiliary tasks that are constrained to specific task pairs** (TADA methods). An emerging line of work recently is using multi-task learning or weakly supervised learning to help unsupervised domain adaptation. Gebru *et al*. [10] adapt fine-grain classification between an easy domain and in-the-wild images with the help of classes' attributes. They adapt a consistency loss between attributes and classes and domain adaptation losses from Tzeng *et al*. [36]. Yang *et al*. [39] adapts lab-environment 3D human pose estimation for in-the-wild data with only 2D pose ground truth, by jointly training on 2D and 3D labels and aligning domains with a GAN-based discriminator. Fang *et al*. [6] adapts a robot grasping application from simulation to real images, and from the indiscriminate grasping task to instance-specific grasping. They perform joint training on all existing labels and the optional input of the instance mask. Inoue *et al*. [14] adapts image object detection to paintings by generating pseudo-labels which are filtered using auxiliary image-level labels.

Our paper has two major differences from these prior work. (1) All prior work have very application-specific constraint formulation on the their task-pair relationship, making them inapplicable to nearly all other tasks. Gebru *et al*. [10] assumes the auxiliary and main annotation to have a known linear relationship. Yang *et al*. [39] constraints the 2D pose and 3D pose to use the same 2D pose output layer. Fang *et al*. [6] assumes both tasks' output are both binary

prediction, share the same output neuron, and the tasks are differentiated by an extra input. Inoue *et al*. [14] must use a hard-coded procedure to filter erroneous outputs of the main detection task using the anchor task classification labels. In contrast, our work requires only that the two tasks' annotations are spatial, without any constraint on the output layers or the loss of each task – a much weaker assumption – and models the cross-task guidance without hard-coded domain knowledge. Our experiments show that the TADA formulation helps task transfer scenarios beyond these task-specific designs with explicit task relations. (2) We focus on tasks with pixel-wise outputs, such as surface normal estimation or keypoint detection (in the form of heatmaps for each keypoint).

**Weakly supervised learning** [42, 13, 17] uses a weaker label to help infer a stronger label, e.g. when inexact coarse category are provided to help fine-grain classification. These methods are not concerned with domain adaptation, and are similar to our method only in the usage of an auxiliary task with available annotations.

**Unsupervised Domain Adaptation** [25, 26, 12, 28, 9] is similar to our formulation without the anchor task. Especially worth mentioning is Tsai *et al*. [34]. Instead of matching feature space distributions, they adapt the structured output space to have a similar distribution between domains. This is done by applying a GAN-based domain confusion loss over the output from the two domains, and optionally, the feature space as well. Our method additionally uses the anchor task to help on top of these methods, achieving a more fine-grained adaptation with the correspondence the anchor brings. We experimentally show that UDA may hurt performance due to systematic domain difference, while ours is more robust.

**Semi-supervised Domain Adaptation**, on the other hand, assumes a small number of target domain samples are labeled, compared to our assumption that a task with free or available labels exists for both domains. Castrejon *et al*. [5] have a variant similar to HEADFREEZE, but require main task supervision, contrary to our anchor task idea. This is a separate research direction orthogonal with ours.

**Combining Multi-task learning and Domain Adaptation** [36, 8, 38] is topically similar. Besides the TADA works we mentioned earlier, most assume all tasks' labels from the target domain are available, and some still requires specially designed losses or constraints for the task pair (e.g. Tzeng *et al*. [36] constrains the all tasks to be classification tasks). Whether their formulation are still effective in our unsupervised case is beyond the scope of our paper.

**Transfer Learning** (e.g. Taskonomy [40]), including Multi-task Learning [4] (e.g. UberNet [15]) and Meta-transfer Learning (e.g. MAML [7]), are methods that use knowledge learned from one task to help another. Most methods assume all tasks are in the same domain or ignore the domain difference (e.g. Taskonomy [40], UberNet [15]), and some assume one task per domain or dataset (e.g. Liu *et al*. [19]). Our idea makes use of the knowledge in one task

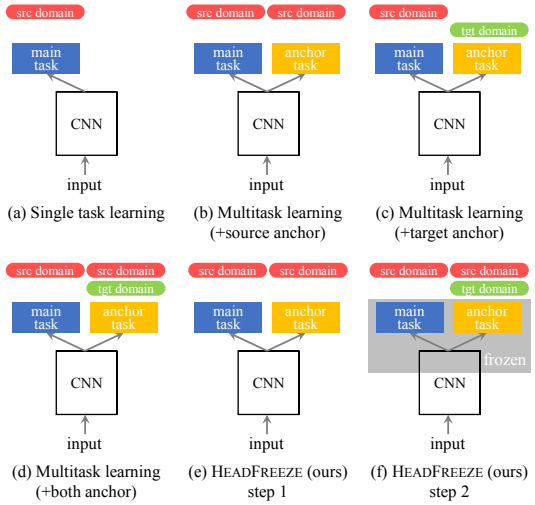


Figure 2. Illustration of various compared methods and their training label usage. TADA methods (d-f) uses the anchor task on *both* domains to establish clear correspondence in the anchor task annotation space. Our HEADFREEZE method first trains only on the source domain, and then freezes the final network layers to consolidate the learned cross-task spatial and contextual guidance in the output.

to help another, but we are more interested in how having the anchor task knowledge in *both* domains can help domain adaptation instead. Compared to prior methods, we empirically show that the anchor task is needed in *both* domains to bridge the domain gap.

Among these, UberNet [15] has similar formulation with our MTL (+both anchor) ablation, but without using an anchor task shared by all samples. The paper also ignores any domain difference, and only focuses on tasks in datasets where it has supervision, making it irrelevant to domain adaptation.

Some other methods consider performing the same task on different domains as multitask learning [21], but in our formulation of multitask learning (performing tasks that have conceptually different labels) they are performing semi-supervised domain adaptation instead.

**Modeling output spatial structure** [22, 34, 35] is related to how we preserve the cross-task guidance between two tasks' outputs. Mostajabi *et al*. [22] regularizes semantic segmentation by training an autoencoder on the *semantic labels*, and force the network to use the fixed decoder to output its prediction. We are inspired by these ideas, but are focused on how *two* tasks' output spaces interact, and generalizing across domains.

## 3. Method

To formulate our Task-Assisted Domain Adaptation (TADA), we first start from a brief review of Unsupervised Domain Adaptation (UDA). In UDA, we have labeled data in the source domain $(x_{\mathcal{S}}, y_{\mathcal{S}}) \in \mathcal{S}$, and unlabeled data in

the target domain $(x_{\mathcal{T}}, y_{\mathcal{T}}) \in \mathcal{T}$. However, only the test set in $\mathcal{T}$ may contain labels $y_{\mathcal{T}}$ for evaluation purposes, and in the train set, $(x_{\mathcal{T}}, \varnothing) \in S_{tr}$ is provided. A model, usually with the form of $\hat{y} = g(f(x))$, is trained on all available data, where $f(\cdot)$ is the network backbone for input-feature mapping, and $g(\cdot)$ is the head for feature-prediction mapping. In this paper, unless otherwise specified, we refer to the networks' second-to-last layer output as the "*features*".

Usually, to reduce the domain gap, features in both domains $f(x_{\mathcal{S}})$ and $f(x_{\mathcal{T}})$ are encouraged to follow the same distribution [9] (although this can also be done in output space $g(f(x.))$ as well [34]). However, it is usually not guaranteed that ground truths $y_{\mathcal{S}},\ y_{\mathcal{T}}$ follow the same distribution. When the ground truths distribute differently, the ideal features and outputs have to distribute differently too. Forcing either of them to distribute similarly would deviate the prediction from the ground truth.

In our Task-Assisted Domain Adaptation scenario, in addition to the main task, an anchor task is defined for both domains. The domains become $(x_{\mathcal{S}}, y_{\mathcal{S}m}, y_{\mathcal{S}a}) \in \mathcal{S}$, and $(x_{\mathcal{T}}, y_{\mathcal{T}m}, y_{\mathcal{T}a}) \in \mathcal{T}$. Here, $m$ and $a$ stand for the main and anchor tasks. In the train set of $\mathcal{T}$, only $(x_{\mathcal{T}}, \varnothing, y_{\mathcal{T}a}) \in T_{tr}$ is provided, while $y_{\mathcal{T}m}$ is unknown or unavailable. A model, usually with the form of $\hat{y}_m = g_m(f(x)),\ \hat{y}_a = g_a(f(x))$ is trained on those data, where $g_m(\cdot)$ and $g_a(\cdot)$ are sub-modules specific to each task. In this work, we focus on the popular formulation above where the two tasks share the same network backbone $f(\cdot)$.

In this work, we consider that the anchor task exists solely to aid the learning of the main task. We evaluate only on the target domain main task, not on the anchor. If the anchor task is important, one can always train a separate model for it using a variety of transfer learning methods.

### 3.1. MTL + anchor for effective feature learning

When prior work has performed Multi-task Learning (MTL), either all tasks are assumed to be in one domain, or one task is available in each domain (main in $\mathcal{S}$, anchor in $\mathcal{T}$). Formally, there can be three supervised losses in the TADA scenario:

$$\mathcal{L}_{\mathcal{S}m} = \mathcal{L}_m\left(y_{\mathcal{S}m}, \hat{y}_{\mathcal{S}m}\right), \tag{1}$$

$$\mathcal{L}_{\mathcal{S}a} = \mathcal{L}_a\left(y_{\mathcal{S}a}, \hat{y}_{\mathcal{S}a}\right), \tag{2}$$

$$\mathcal{L}_{\mathcal{T}a} = \mathcal{L}_a\left(y_{\mathcal{T}a}, \hat{y}_{\mathcal{T}a}\right). \tag{3}$$

In prior work, a multitask learning loss may only comprise of two of the three:

$$\mathcal{L}_{\text{MTL (+src anchor)}} = \mathcal{L}_{\mathcal{S}m} + \lambda\mathcal{L}_{\mathcal{S}a},\ \text{or} \tag{4}$$

$$\mathcal{L}_{\text{MTL (+tgt anchor)}} = \mathcal{L}_{\mathcal{S}m} + \lambda\mathcal{L}_{\mathcal{T}a}, \tag{5}$$

for "everything in source" and "one task per domain" respectively. We instead use the alternative baseline – MTL (+both anchor), which simply uses all the supervised losses.

$$\mathcal{L}_{\text{MTL (+both anchor)}} = \mathcal{L}_{\mathcal{S}m} + \lambda\mathcal{L}_{\mathcal{S}a} + \lambda\mathcal{L}_{\mathcal{T}a}. \tag{6}$$

Differences between these formulations are illustrated in Figure 2.

We suggest two ways of choosing the anchor task and obtaining its annotations. (1) Some anchor annotations can be freely obtained, e.g. from very robust estimators that work across most domains, such as facial keypoint detectors. (2) Some anchor tasks can be popular tasks and already have labels in many datasets, such as semantic segmentation. It should be chosen so obtaining it is much easier than the main task annotation.

It may be tempting to hypothesize that the baseline losses in Eq. 4, 5 will be enough for the TADA scenario, and that collecting anchor labels on both domains is not necessary. Maybe in Eq. 4 the multitask learning aspect can already improve model generalization, and in Eq. 5 the network is trained on the target domain, so it may be forced to adapt to perform well on the anchor task. One can also add an unsupervised domain adaptation loss to reduce the domain gap. We experimentally show that these baselines underperform MTL (+both anchor) and degrade performance.

In addition to any of these supervised losses, an unsupervised domain adaptation loss can be added. For example, adversarial losses (a.k.a. GAN losses) on the features or the output space are used in prior work [9, 34] with a discriminator network $d(\cdot)$ trained in a mini-max fashion:

$$\min_{f,g}\max_{d} \mathcal{L}(f,g) + \lambda_{\text{adv}}\mathcal{L}_{\text{adv}}(f,g,d). \tag{7}$$

We refer our readers to the prior work [9, 34] for the exact formulation of $\mathcal{L}_{\text{adv}}$.

### 3.2. HEADFREEZE for preserving cross-task guidance

Building on MTL (+both anchor), we further propose our HEADFREEZE method to leverage the cross-task guidance that can be used to guide the target domain main task based on the target anchor task.

The final layers of a trained multitask network can can be seen as a decoder from its input feature space to the joint label space of the two tasks. When we train these layers on the source domain to convergence, they have only learned to predict output pairs for main and anchor tasks that are *contextually and spatially coherent*, and have never learned to output incoherent pairs (such as misaligned object edges or shapes between tasks, and contradictory outputs like vertical ceilings or flat noses). We assume that the final layers can incorporate this coherency knowledge, and are more likely to predict coherent outputs. It follows that the coherency knowledge can act as a cross-task guidance, so training on target anchor task improves the target main task by ruling out incoherent predictions.

However, it is possible that the model overfits to the target domain anchor task, ignores or forgets any cross-task guidance learned in the source domain. We force the cross-task guidance to persist across domains with HEADFREEZE. We first train the multitask network on the source domain, using Eq. 4. When it approaches convergence (or

just before it overfits to one of the tasks), we freeze the parameters of its final layers. We then train only the lower layers jointly on all available labels using Eq. 6, forcing their output to go through the pre-trained final layers. See Figure 2 (e,f) for an illustration. This procedure can be trained end-to-end by modifying the loss and optimizer's list of variables after the convergence of the first step, which is easy to do in modern frameworks such as PyTorch [23].

For implementation details such as network structure and the number of layers frozen, please see Section 4.4.

# 4. Experiment setup

We validate our methods and claims on two sets of experiments, facial images and indoor scenes, both adapting from synthetic data to real images – our motivating scenario.

## 4.1. Facial images

We perform facial surface normal estimation as the main task, and for the anchor task we choose 3D facial keypoint detection with automatically generated ground truth. Intuitively, 3D keypoints can inform surface information, and thus is a good form of guidance. As 3D keypoints can currently be reliably generated by methods that generalize well across domains, we use this to show whether *free* anchor task labels can be helpful for another label-deprived task.

We adapt from synthetic data generated by Sengupta *et al*. [27] ("SfSsyn"), using 3DMM models [1]. The dataset provides facial images with surface normal ground truth, with synthetic faces both frontal and looking to the side. We change the reference frame of the surface normal to camera coordinates to follow the definition of all other datasets.

For the target domain, we use real data from FaceWarehouse [3] ("FaceWH"). The dataset provides facial models fitted using a morphable model followed by a laplacian-based mesh deformation without any PCA reduction, so the surface normals rendered from them are both clean and faithful to the raw RGBD scan.

None of these two datasets provide an official split. We split the subjects (separated by dataset folders) into 70% for training, and 15% each for validation and test.

We obtain the anchor annotations for free. On both datasets, we use state-of-the-art Bulat *et al*. [2] to extract both 3D keypoints and 2D keypoints using their separate models. 3D keypoints are used as anchor training ground truth. We compute the facial region mask from the 2D keypoints for performing evaluation, which is a standard practice in facial surface normal estimation [33, 27].

During training, we use the standard losses for both tasks: for surface normal estimation, cosine loss (see [33]); for 3D keypoint detection, a heatmap regression for the 2D positions, and a vector regression for depth (see [2]). During evaluation of surface normal, we use five metrics in the literature. Specifically, the angular difference between predicted 3D surface normal and the ground truth is treated as the error and computed for each pixel. Then we aggregate the root mean square angular error (RMSE), mean of the error (Mean), median of the error (Median), and percentages of pixels with errors below 11.25° and 30°. Only valid regions are considered, so we ignore pixels outside the face or where there is no ground truth (e.g. where depth is missing and surface normal cannot be correctly estimated).

## 4.2. Indoor scenes

We again perform surface normal estimation as the main task, but use semantic segmentation as the anchor task to demonstrate, since semantic segmentation has annotations available across many datasets. The semantic boundaries can inform discontinuities in surface normal space, and some categories such as ceilings have very constrained normal directions. Other categories with no fixed shape or expected direction can be hard to improve.

We adapt from the SUNCG dataset [31] with physically-based rendering [41], which provides images, semantic segmentation, and surface normal ground truth. We use NYUdv2 [29] as the target domain, with additional surface normal estimated from depth by Ladicky *et al*. [16]. We only use the labeled portion of the dataset.

SUNCG is large, so we use a 90%-5%-5% split for train, validation, and test. We use NYUdv2's official split. Normal estimation loss and metrics are the same as before, and semantic segmentation is trained using cross-entropy.

## 4.3. Compared methods

Since we address the domain adaptation problem, we compare to unsupervised adaptation methods that applies either a multi-level version of Ganin *et al*. [9] or state-of-the-art Tsai *et al*. [34], either on the single task model (DA), or on our method for further improvement (HEADFREEZE +DA). Adversarial training is brittle and not all configurations work too well. We implement our own version and perform hyperparameter tuning, and omit some of the underperforming combinations. We also compare to an oracle method that uses both tasks labels on both domains, including the target domain main task. This gauges how far each method is from fully successful adaptation.

For facial surface normal, we compare to a recent and popular intrinsic decomposition method SfSNet [27], which produces surface normal based on extra domain knowledge (lighting model for unsupervised learning). We use their released model trained on synthetic data and on unsupervised CelebA [20], a much larger dataset. This comparison only serves to prove that our method is effective instead of being a controlled experiment, since neither our network structure or external knowledge is similar.

For ablation studies, we compare to baselines shown in Fig. 2: single task baseline, multitask with only one domain (MTL (+src anchor)), multitask with source main task and target anchor task (MTL (+tgt anchor)) as used in prior work such as Liu *et al*. [19], and MTL (+both anchor).

| | | | | | Faces: SfSsyn→FaceWH | | | | | Indoor: SUNCG→NYUdv2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $y_{\mathcal{S}m}$ | $y_{\mathcal{S}a}$ | $y_{\mathcal{T}m}$ | $y_{\mathcal{T}a}$ | $< 11.25^\circ$ | $< 30^\circ$ | RMSE | Mean | Median | $< 11.25^\circ$ | $< 30^\circ$ | RMSE | Mean | Median |
| Baseline | ✓ | | | | 0.424 | 0.929 | 17.8 | 14.8 | 12.8 | 0.298 | 0.683 | 33.5 | 25.8 | 18.8 |
| Baseline+DA | ✓ | | | | 0.456 | 0.937 | 17.2 | 14.2 | 12.1 | **0.316** | 0.703 | 33.3 | 25.2 | **17.6** |
| HEADFREEZE (ours) | ✓ | ✓ | | ✓ | **0.519** | **0.954** | **15.8** | **12.9** | **10.9** | 0.301 | 0.708 | **31.8** | 24.6 | 18.0 |
| HEADFREEZE (ours)+DA | ✓ | ✓ | | ✓ | 0.455 | 0.935 | 17.2 | 14.2 | 12.1 | **0.316** | **0.715** | 32.0 | **24.4** | **17.4** |
| Oracle | ✓ | ✓ | ✓ | ✓ | 0.907 | 0.995 | 7.8 | 6.2 | 5.2 | 0.340 | 0.734 | 30.4 | 23.1 | 16.5 |
| SfSNet [27] | – | – | – | – | 0.495* | 0.965 | 15.2 | 12.9* | 11.3* | | | | | |

Table 1. Comparison in our two experimental settings. Unsupervised domain adaptation with Tsai *et al*. [34] is shown for indoor scenes, and with Ganin *et al*. [9] shown for faces, whereas the other combinations underperform (see supplemental). Our HEADFREEZE method is comparable to surface normal estimated from SfSNet, without the use of a lighting model. HEADFREEZE +DA performs closest to oracle in indoor scene, but domain adaptation methods fail to improve HEADFREEZE for faces. Statistical significance computed from 3 runs. (*) denotes a method with domain knowledge performs *equal to or worse* than our best performing method.

| | | | | | Faces: SfSsyn→FaceWH | | | | | Indoor: SUNCG→NYUdv2 | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $y_{\mathcal{S}m}$ | $y_{\mathcal{S}a}$ | $y_{\mathcal{T}m}$ | $y_{\mathcal{T}a}$ | $< 11.25^\circ$ | $< 30^\circ$ | RMSE | Mean | Median | $< 11.25^\circ$ | $< 30^\circ$ | RMSE | Mean | Median |
| Baseline | ✓ | | | | 0.424 | 0.929 | 17.8 | 14.8 | 12.8 | 0.298 | 0.683 | 33.5 | 25.8 | 18.8 |
| MTL (+src anchor) | ✓ | ✓ | | | 0.409 | 0.935 | 17.7 | 14.9 | 13.1 | 0.280 | 0.666 | 34.1 | 26.6 | 19.8 |
| MTL (+tgt anchor) | ✓ | | | ✓ | 0.162 | 0.791 | 24.3 | 21.8 | 20.4 | 0.260 | 0.662 | 32.9 | 26.2 | 20.6 |
| MTL (+both anchor) | ✓ | ✓ | | ✓ | 0.492 | **0.953** | 16.0 | 13.3 | 11.4 | 0.275 | 0.675 | 32.4 | 25.7 | 19.8 |
| HEADFREEZE (ours) | ✓ | ✓ | | ✓ | **0.519** | **0.954** | **15.8** | **12.9** | **10.9** | **0.301** | **0.708** | **31.8** | **24.6** | **18.0** |

Table 2. Ablation studies. See Figure 2 for each method's formulation. Other MTL baselines underperform while MTL (+both anchor) outperforms, indicating the importance of shared anchor tasks. Our HEADFREEZE technique further boosts MTL (+both anchor) performance. Statistical significance computed from 3 runs.

### 4.4. Implementation details

Code will be released. We use a ResNet50 [11] with FPN [18] for our network backbone, with the ResNet pretrained on ImageNet [24]. We use the variant with 3 upsampling layers with skip connection, and used a deconvolution layer as the output layer for both tasks, making the output 50% of the input resolution. For HEADFREEZE, we freeze the layers after the second upsampling layer, including any skip connection weights. Some tasks require additional non-spatial outputs. A common practice of 3D keypoint estimation [2] is to output a heatmap for projected 2D positions, and a vector for 3D depth. We add a fully-connected branch of 2 layers with 256 hidden units after the global average pooling over the second upsampling layer's output. Batches of the same size are sampled from each domain for each iteration. We choose $\lambda$ so losses from different domains and tasks have similar magnitudes. For adversarial training and dataset processing, please refer to our supplemental material.

Hyperparameter tuning is hard in TADA, just like in any unsupervised domain adaptation, due to the lack of target domain main task ground truth in validation. Although by evaluating against available ground truth we can tune most hyperparameters (e.g. stop criteria, learning rate, layers to freeze), some parameters critical to target main task (e.g. discriminator network complexity, its learning rate and loss weights) may barely cause any change. We empirically find that the discriminator accuracy being very frequently lower than 55% and good qualitative results (absence of artifacts) are good indicators of successful adaptation, and tune the parameters accordingly.

## 5. Results

Table 1, 2 shows our results and ablation studies.

**Facial images**. On SfSsyn to FaceWH adaptation, HEADFREEZE outperforms the non-adaptation baseline. Adding domain adaptation [9] does improve baseline results, but still underperforms our HEADFREEZE method. HEADFREEZE is comparable to the popular SfSNet [27], which underperforms on Median and $11.25^\circ$ but outperforms on RMSE and $30^\circ$. However, all methods are still quite far from the oracle method that uses target domain main task annotations.

Perhaps a very surprising observation is that the unsupervised domain adaptation methods added to HEADFREEZE would *hurt* performance instead of improving them. In fact, HEADFREEZE *without adaptation* is the best method apart from the oracle (and SfSNet). The adaptation puts HEADFREEZE at the same level with DA [9], eliminating any advantage brought by the anchor task. We have vigorously tuned the adversarial loss hyperparameters, yet still cannot find a configuration that would not hurt performance. In comparison, HEADFREEZE (and even MTL (+both anchor) in Table 2) work naturally. We analyze the reason for our robustness in Section 5.1.

In the ablation study, the baseline and MTL with anchor

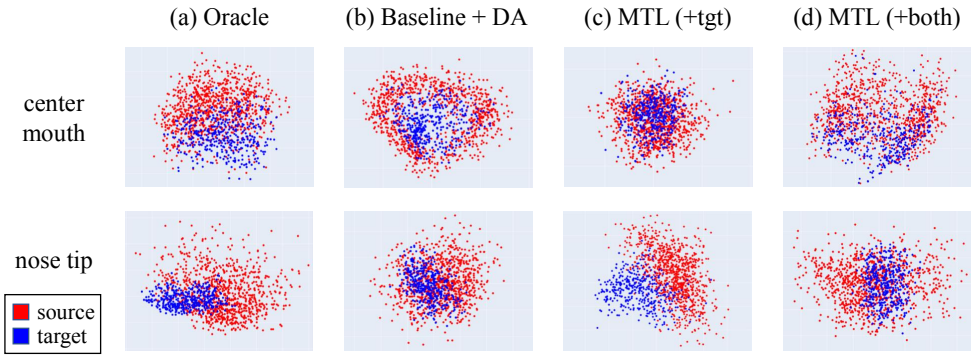


Figure 3. PCA visualization of the subtle differences between methods' feature space at different facial keypoint locations. (a) Oracle does not have fully overlapping domain due to systematic distribution differences. (b) Forcing domains' distributions to be similar can deviate the features from the oracle and hurt performance (top row). (c) Training MTL with one task per domain may encourage using separate feature space regions for different domains (bottom row). (d) MTL (+both anchor) produces feature distributions slightly more visually similar to the oracle. Disclaimer: the baseline's visualization (not shown) is also similar to the oracle, so this cannot indicate higher performance. Other facial locations may not exhibit observed behaviors as clearly. Best viewed in color.

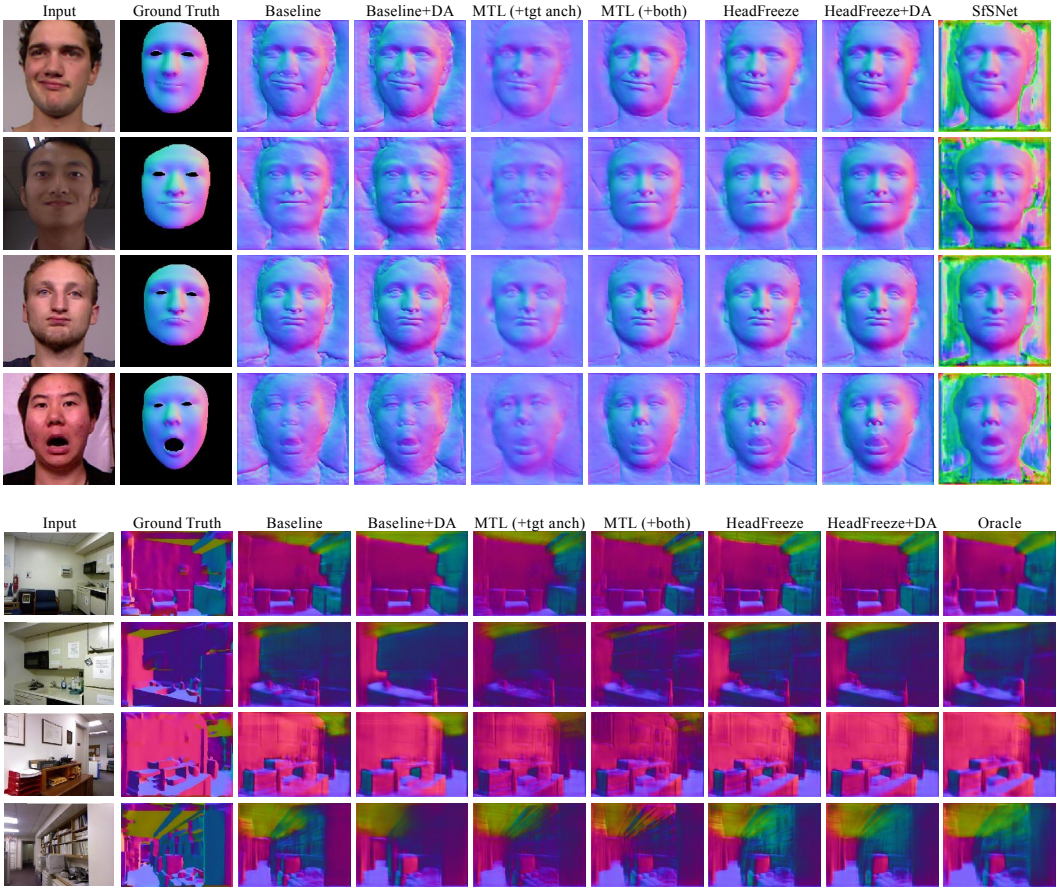


Figure 4. Qualitative results for compared methods. For domain adaptation, Ganin *et al*. [9] is shown for facial images (top), and Tsai *et al*. [34] is shown for indoor scenes (bottom). Best viewed in color.

on either one domain all underperform. Two observations are interesting: (1) MTL (+src anchor) does not perform very differently from the baseline on the target domain, indicating that the effect of multi-task learning is limited here. (2) MTL (+tgt anchor) vastly underperforms the baseline when trained with one task per domain. We hypothesize that despite the network being trained on the target $\mathcal{T}$, the task performed on $\mathcal{T}$ is too different, which *encourages* the net-

| | $< 11.25^\circ$ | $< 30^\circ$ | RMSE | Mean | Median |
|---|---|---|---|---|---|
| Baseline | 0.418 | 0.913 | 18.6 | 15.3 | 13.0 |
| Baseline+DA [9] | 0.495 | 0.944 | 16.6 | 13.5 | 11.3 |
| HEADFREEZE | 0.550 | 0.958 | 15.2 | 12.4 | 10.4 |
| HEADFREEZE +DA [9] | **0.573** | **0.963** | **14.7** | **11.9** | **10.0** |

Table 3. Facial normal estimation, with SfSsyn-frontal as the source domain, which has head pose distribution similar to FaceWH. In this experiment, domain adaptation [9] always helps performance, indicating that systematic dataset difference is the reason distribution matching adaptation fails, which HEADFREEZE is robust to.

work to learn very different features for the tasks, harming adaptation. MTL (+both anchor) outperforms other MTL methods, indicating the importance of the anchor task being trained on both domains, affirming our hypothesis. HEADFREEZE further improves all criteria by a margin, implying that the cross-task guidance learned in the source domain can be helpful for the target domain as well.

**Indoor scenes.** We still see both HEADFREEZE and domain adaptation [34] improve over the non-adaptation baseline, but it is inconclusive whether HEADFREEZE outperforms domain adaptation [34]. But we observe that HEADFREEZE +DA further improves the adaptation-only method, closing much of the gap between baseline and the oracle.

In Table 2's ablation study, all MTL variations suffer from negative transfer, i.e. main task performance degrades as the second task is jointly learned. We still observe that MTL (+both anchor) outperforms other MTL variants, indicating that it has an adaptation effect that other variants do not possess, despite the negative transfer. We also observe that HEADFREEZE makes a larger improvement on MTL (+both anchor) than in the facial experiments.

### 5.1. Further analysis

**Failure of adaptation and face label distribution.** We analyze why the compared domain adaptation methods fail to improve HEADFREEZE in the SfSsyn-FaceWH experiment. After trials and errors, we found that the difference of head pose distributions between domains may be a major contributor. We generated a second version of the SfSsyn dataset with only frontal faces ("SfSsyn-front"), with rotation distribution closely following the estimated poses from the target dataset. We evaluate the domain adaptation methods with SfSsyn-front as the source domain in Table 3 instead, with all methods using the same hyperparameter.

The trends and conclusions are exactly the same, except that unsupervised domain adaptation always helps, making our HEADFREEZE + DA the top method. This experiment indicates that the distributional difference is indeed why adaptations [9, 34] fail. We conclude that while these prior works are effective, they would *hurt* performance when domain ground truths are differently distributed, whereas our methods are more robust to such differences. While these differences may sometimes be easily eliminated in data synthesis procedures, other times they may be expensive to eliminate, or difficult to pinpoint.

**Impact on feature space.** To better understand the impact of different methods on the feature distribution, we visualize their feature space for source and target domain. Since features at different spatial locations may encode information differently, we extract the feature at separate facial keypoint locations in the facial experiment. For each location (e.g. nose tip), we perform PCA and obtain the top two components, and visualize them in Fig. 3. Please refer to its caption for observations. This experiment resonates with our hypothesis that training MTL (+tgt anchor) with one task per domain would map source and target to different feature space regions, and that blindly matching feature distribution may be suboptimal.

**Qualitative results** are shown in Figure 4. For faces, the synthetic dataset has less facial expressions than FaceWarehouse, so baselines struggle with e.g. open mouths. Unsupervised adaptation [9] tends to erroneously force the cheeks and nose normals to the side to force the output look like side-facing faces locally. The ground truth is not extremely faithful to the image due to being fitted on RGBD scans, and both our HEADFREEZE method and SfSNet [27] capture local details better than the ground truth, although SfSNet performs better with open mouths due to their usage of a lighting model on unlabeled real faces.

For indoor scenes, HEADFREEZE improves the performance for shelves, cabinets, and ceilings more effectively than the facial datasets, possibly due to their semantic labels providing much information for their surface normal.

## 6. Summary and future work

We propose a strategy to extend prior Task-assisted Domain Adaptation methods by eliminating the need for task-specific relationship formulations. We use spatial information of a free or already available shared anchor task to align both features between domains and spatial prediction and context between tasks, and propose HEADFREEZE to further leverage the cross-task guidance to improve target domain main task. We show effectiveness and robustness of using anchor tasks against multitask baselines, and HEADFREEZE against conventional adaptation methods.

There are many open questions to answer for the effect of anchor tasks. How do we make sure main task get information from anchor task output directly? Would a design built on PAD-Net [38] work? Can we adapt multiple main tasks from only one anchor task to leverage all the rich labeling of synthetic data? How cheap can the anchor task be made? Can Taskonomy [40] help in choosing which anchor task to use? We leave these questions for future work.

**Acknowledgments** This work is supported in part by the Office of Naval Research grant ONR MURI N00014-16-1-2007 and a gift from Snap Inc.

## References


[1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH*, 1999.

[2] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017.

[3] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Transactions on Visualization and Computer Graphics*, 20:413–425, 2014.

[4] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, Jul 1997.

[5] Lluis Castrejon, Yusuf Aytar, Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Learning aligned cross-modal representations from weakly aligned data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2940–2949, 2016.

[6] Kuan Fang, Yunfei Bai, Stefan Hinterstoißer, Silvio Savarese, and Mrinal Kalakrishnan. Multi-task domain adaptation for deep learning of instance grasping from simulation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3516–3523, 2018.

[7] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

[8] Damien Fourure, Rémi Emonet, Élisa Fromont, Damien Muselet, Natalia Neverova, Alain Trémeau, and Christian Wolf. Multi-task, multi-domain learning: Application to semantic segmentation and pose regression. *Neurocomputing*, 251:68–80, 2017.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning*, pages 1180–1189, 2015.

[10] Timnit Gebru, Judy Hoffman, and Li Fei-Fei. Fine-grained recognition in the wild: A multi-task domain adaptation approach. *Proceedings of International Conference on Computer Vision (ICCV)*, pages 1358–1367, 2017.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[12] 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.

[13] Ronghang Hu, Piotr Dollár, Kaiming He, Trevor Darrell, and Ross B. Girshick. Learning to segment every thing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4241, 2018.

[14] Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. Cross-domain weakly-supervised object detection through progressive domain adaptation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5001–5009, 2018.

[15] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5454–5463, 2017.

[16] Lubor Ladicky, Bernhard Zeisl, and Marc Pollefeys. Discriminatively trained dense surface normal estimation. In *ECCV*, 2014.

[17] Jie Lei, Zhenyu Guo, and Yang Wang. Weakly supervised image classification with coarse and fine labels. *2017 14th Conference on Computer and Robot Vision (CRV)*, pages 240–247, 2017.

[18] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 936–944, 2017.

[19] Xiaodong Liu, Jianfeng Gao, Xiaodong He, Li Deng, Kevin Duh, and Ye-Yi Wang. Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In *HLT-NAACL*, 2015.

[20] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[21] Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Philip S. Yu. Learning multiple tasks with multilinear relationship networks. In *NIPS*, 2017.

[22] Mohammadreza Mostajabi, Michael Maire, and Gregory Shakhnarovich. Regularizing deep networks by modeling and predicting label structure. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5629–5638, 2018.

[23] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[24] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[25] Kuniaki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tatsuya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[26] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3752–3761, 2018.

[27] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. Sfsnet: Learning shape, reflectance and illuminance of facesin the wild'. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6296–6305, 2018.

[28] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2242–2251, 2017.

[29] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[30] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[31] Shuran Song, Fisher Yu, Andy Zeng, Angel X. Chang, Manolis Savva, and Thomas A. Funkhouser. Semantic scene completion from a single depth image. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 190–198, 2017.

[32] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 969–977, 2018.

[33] George Trigeorgis, Patrick Snape, Iasonas Kokkinos, and Stefanos P. Zafeiriou. Face normals "in-the-wild" using fully convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 340–349, 2017.

[34] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Krishna Chandraker. Learning to adapt structured output space for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018.

[35] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Krishna Chandraker. Domain adaptation for structured output via discriminative patch representations. *CoRR*, abs/1901.05427, 2019.

[36] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[37] Gül Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. In *CVPR*, 2017.

[38] Dong Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.

[39] Wei Yang, Wanli Ouyang, Xiaolong Wang, Jimmy S. J. Ren, Hongsheng Li, and Xiaogang Wang. 3d human pose estimation in the wild by adversarial learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5255–5264, 2018.

[40] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3712–3722, 2018.

[41] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas A. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5057–5065, 2017.

[42] Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5, 08 2017.