ADA-AT/DT: An Adversarial Approach for Cross-Domain and Cross-Task Knowledge Transfer

Ruchika Chavhan Ankit Jha Biplab Banerjee Subhasis Chaudhuri
{chavhanruchika2801, ankitjha16, getbiplab}@gmail.com, sc@ee.iitb.ac.in
Indian Institute of Technology Bombay, India

Abstract

We deal with the problem of cross-task and cross-domain knowledge transfer in the realm of scene understanding for autonomous vehicles. We consider the scenario where supervision is available for a pair of tasks in a source domain while it is available for only one of the tasks in the target domain. Given that, the goal is to perform inference for the task in the target which is devoid of any training information. We argue that the only reported work in learning across tasks and domains (AT/DT) [26] faces the problem of domain shift between the source and target domains, hindering predictions on the target domain when the transfer of knowledge is learned on a statistically different yet related source domain. As a remedy, we develop a novel framework called ADA-AT/DT based on the adversarial training strategy to ensure that the domain-gaps are minimized for the common cross-domain supervised task. This, in effect, helps in realizing a domain-independent task-transfer function that eventually helps in performing improved inference in the target domain. We demonstrate that our proposed method significantly outperforms [26] by using models with 81% fewer trainable parameters. In addition, we perform experiments on a transformation mapping similar to U-Net to ensure maximum exploitation of features for task transfer. Extensive experiments have been performed on four different domains (Synthia, CityScapes, Carla, and KITTI) for two visual tasks (depth estimation and semantic segmentation) to confirm the superiority of our method.

1. Introduction

Deep learning frameworks have played a significant role in creating revolutionary technologies such as self-driving vehicles, chat-bots, recommendation systems, etc. Most technologies that function on deep learning models are required to perform multiple tasks in order to understand semantics and geometry of the environment in which they operate. Within the paradigm of computer vision, scene understanding by means of monocular depth estimation, semantic segmentation of visual stream, etc. are crucial for many visual inference tasks, notably the effective operation of self-driving cars. Generally speaking, majority of the vision-related tasks performed by self-driving cars can be realized by training dense and complex convolutional networks on synthetic and real-life datasets simultaneously. Moreover, recent works have proven that many visual tasks are highly correlated and performance for these tasks can be enhanced by training them together instead of designing stand-alone models for each of them.

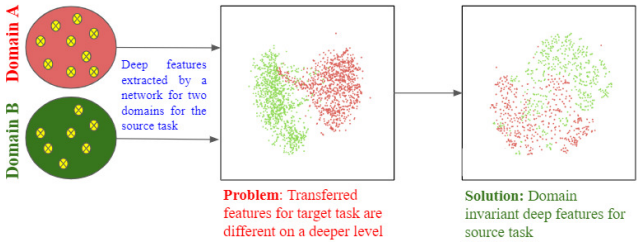

Figure 1: We describe the motivation for generating a domain-invariant representation for domain A (red) and domain B (green). Domain shift in abstract features needs to be addressed for smoother domain transfer. Our framework ADA-AT/DT tackles this problem by generating a domain indistinguishable deep representations.

Multi-task learning [28] is the area of machine learning involving improvement of generalization capacity of a task by leveraging domain-specific information from other closely related tasks. In [39], the taxonomical structure of a certain set of visual tasks has been studied to reuse supervision among strongly related tasks. Semantic segmentation and monocular depth estimation are both examples of supervisory tasks that perform dense pixel to pixel mapping to represent both the semantic and geometrical information of a scene. Rather than addressing these tasks separately, it was studied that incorporating knowledge from these two strongly correlated tasks reciprocally promotes the performance of both the tasks [3, 25, 16, 4]. Most multi task learning studies presume that all tasks use a single representation

before learning task-specific parameters. Therefore a configuration consisting of a single encoder and multiple task-specific decoders is allowed to mutually refine the results of all the tasks [23, 12, 17, 22, 19, 21, 35].

Transfer learning [41] aims to enhance the learners' output on target domains by transferring the information found in different but linked source domains. A domain is defined as $\mathbb{D} = \{\mathcal{X}, \mathcal{P}(\mathcal{X})\}$ where $\mathcal{X}$ denotes the feature space with a marginal probability distribution $P(\mathcal{X})$. We define a source domain $\mathcal{A}$, consisting of a sufficient amount of annotated data $\mathcal{A} = \{x_i, y_i\}$, where $x_i \in \mathcal{X}_\mathcal{A}, P(\mathcal{X}_\mathcal{A})$ and $y_i \in \mathcal{Y}_\mathcal{A}$. The target domain $\mathcal{B} = \{\mathcal{X}_\mathcal{B}, \mathcal{P}(\mathcal{X}_\mathcal{B})\}$ consists of unlabeled data sampled from a marginal distribution closely related to the source domain distribution such that $P(\mathcal{X}_\mathcal{A}) \neq P(\mathcal{X}_\mathcal{B})$. For a specific domain, a task $\mathcal{T}$ comprises of a label space $\mathcal{Y}$ and an objective function $f(\mathcal{X})$, which learns the conditional probability distribution $P(\mathcal{Y}|\mathcal{X})$ in a supervised manner from available labeled data. Deep Domain adaptation [36] is a newly emerging field which aims to abate the compulsion of large amounts of labeled data for a particular domain. The core principle of domain adaptation is to obtain a transferable representation over multiple domains and learning to extract domain-invariant features to reduce the shift in the domains on a deeper abstract level [38, 31, 34, 29]. Several studies [18, 2, 14] have focused on employing generative models as a combination of a generator and discriminator trained in an adversarial setup to predict the domain labels of input images through a domain confusion loss [33, 32, 7, 30, 8].

Multi-task learning and domain adaptation are fields in machine learning that focus on inter-task and inter-domain knowledge transfer respectively. In several cases, supervision is available in a domain for limited number of tasks. A cross domain and cross task knowledge transfer problem was first addressed in [26]. A technique to remove the necessity of labeled data for a particular task in the target domain, while utilizing information from available data and related domains was introduced. Two domains and two highly correlated tasks (depth estimation and semantic segmentation) are considered: the source domain consisting of data corresponding to both tasks and a target domain for which annotations for one of the tasks is unavailable. The training procedure introduced in [26] involves training two task-specific base models on respective domains. The key idea behind [26] is that the features required to obtain the output of one task can be extracted from the features corresponding to a strongly correlated task. This has been demonstrated by training a mapping function to transform deep features extracted from depth estimation to those extracted from semantic segmentation and vice versa for suitable domains. The details of the training procedure is discussed in subsequent sections. The base models in [26] are concurrently trained on the union of two domains, which may result in a profound difference in deep features of the two domains. This becomes an issue when the datasets involved are from synthetic and real life sources, as the transformation mapping is learnt on data from a single domain that provides supervision for both tasks. Therefore, if the extracted features are highly disjointed, the transformation mapping might not provide favorable results in the target domain. In our work, we address the issues faced by the method in [26] when applied for two domains comprising data from dissimilar sources.

As aforesaid, we study the problem of cross task cross domain learning fused with adversarial domain adaptation along with a brief study of different transfer function architectures. As discussed earlier, domain shift in intermediate deep features can impede the transfer function to generalise knowledge mapping across tasks on the target domain as shown in Figure 1. With an adversarial domain adaptation framework, we tackle this problem to produce deep features that are immune to domain shift. We introduce a binary domain classifier to the base model trained on both the domains to promote the generation of domain-invariant deep features. We conduct several experiments to select the optimal domain classifier and architecture of transfer functions to obtain domain-invariant and task-discriminative deep features from which knowledge can be exploited and transferred on the target domain. In subsequent sections, we show that employing a network with fewer number of parameters along with adversarial domain adaptation and more proficient transfer functions can outperform the existing method in most of the standard evaluation metrics. The contributions and results of the paper are summarized as follows:

- To the best of our knowledge, we are the first to employ adversarial domain adaptation in the cross task and cross domain setting to address the problem of domain divergence of intermediate features. The addition of this framework plays a key role in bridging the gap deep features for two different domains, thus providing superior results in generalization to target domains.
- We also perform extensive experiments to select the optimal adversarial training setup for the domain classifier on the criteria of performance and domain-invariance of features based on t-SNE visualization.
- We also conduct multiple experiments on the architecture of different transfer functions designed in the form of a U-Net with spatial attention and report an significant improvement in performance for both depth estimation and semantic segmentation.

2. Related Work

Minimizing domain discrepancy through generative models: Adversarial training based generative models in-

troduce domain-confusion by producing domain independent embeddings. Extensive studies [34, 31, 7, 33, 32], have been performed to obtain a representation that is both domain-invariant and discriminative of the primary task. The main focus is to minimize the task-specific loss while maximising domain classification loss. This has been implemented in the form of a model that consists of an encoder G, a task-specific decoder D, and a domain classifier model ψ that predicts a binary domain label for inputs [31]. The loss function is designed in a way that backpropagating its gradients [7, 8] through the model leads to G providing an intermediate representation indistinguishable for both the domains similar to a min-max optimization. An adversarial domain adaptation methodology was adopted by [33, 32], employing a domain discriminator and a source representation mapping trained in a min-max fashion to reduce the distance between the source and target mapping distributions. Several other studies in the same spirit are [11, 10, 24, 8, 40, 2, 37]. In our work, we employ a similar method resulting in a domain-invariant deep feature representation.

Learning across Tasks and Domains: In AT/DT (Across Tasks Domain Transfer) [26], a transfer function $G_{1\to 2}$ is learnt in the domain for which supervision is available for all tasks and partial supervision for a target domain. The knowledge transfer between two tasks is realised by learning this transformation mapping between task-specific abstract representations. Their approach aims to boost performance on a single task by extracting information from related tasks and similar domains. The authors perform their experiments two synthetic datasets: Synthia and Carla, and two datasets from real life sources: CityScapes and KITTI. Their contribution supplements existing domain adaptation methods as their focus is on task transfer across domains, instead of classical domain adaptation frameworks. Our work uses a similar methodology, integrating adversarial domain adaptation along with a extensive study of transfer function architectures.

3. Proposed Methodology

3.1. Preliminaries

To ensure notation consistency, we adopt the naming conventions introduced in [26]. Apparently, we consider two dense predictions tasks for scene understanding: monocular depth estimation and semantic segmentation respectively. The two tasks are denoted by $\mathcal{T}_1$ and $\mathcal{T}_2$ interchangeably. Besides, the source domain is denoted by $\mathcal{A}$, for which supervision is available for both $\mathcal{T}_1$ and $\mathcal{T}_2$. On the other hand, we have access to the supervised information only for $\mathcal{T}_1$ in the target domain $\mathcal{B}$. Under this setup, the goal is to predict the outputs for $\mathcal{T}_2$ in $\mathcal{B}$. In this regard, let $\{\mathcal{X}_j^{\mathcal{A}/\mathcal{B}}, \mathcal{Y}_j^{\mathcal{A}/\mathcal{B}}\}$ $(j \in 1, 2)$, denote the domain and task-specific training samples (for $\mathcal{T}_1$ and $\mathcal{T}_2$ for $\mathcal{A}$ and $\mathcal{T}_1$ for $\mathcal{B}$).

To implement knowledge transfer across tasks [26], the first step is to train task-specific models on corresponding domains in order to learn optimal intermediate features. In terms of the proposed model architecture, we consider a deep neural network $\mathcal{N}_j^k$ trained for task $\mathcal{T}_j$ and domain $k = \{\mathcal{A}, \mathcal{B}, \mathcal{A} \cup \mathcal{B}\}$, consisting of an encoder $E_{\theta_j}^k$ and a decoder $\mathcal{D}_{\phi_j}^k$. The output of the neural network for a domain k is denoted by $\hat{y}_j^k = \mathcal{D}_{\phi_j}(E_{\theta_j}(x^k))$. The main aim of this paper is to learn the parameters θ_j such that the encoder becomes domain-invariant and ϕ_j such that the decoded output resembles the ground truth. For practical notations, we refer to the encoder and decoder as E_j^k and $\mathcal{D}_j^k$. The networks are trained on task-specific losses on available annotated samples. The next step is to learn a knowledge transformation mapping across tasks. We learn a transfer function $G_{1\to 2}$, which will transform deep features for $\mathcal{T}_1$ to deep features corresponding to $\mathcal{T}_2$.

3.2. Training Procedure

In this section, we will discuss the steps to train all the components on our proposed model. The entire training procedure in [26] along the modifications proposed in this paper are mentioned below in brief in three steps:

1. In [26], two neural networks are trained to solve $\mathcal{T}_1$ and $\mathcal{T}_2$ on their respective domains. The task $\mathcal{T}_1$ is solved by a neural network that is trained on the union of the two domains. Uncomplimentary results can be obtained on the target domain if the intermediate features belonging to respective domains are disjoint. Our work focuses on altering the training procedure of the network for which supervision is available for both domains $\mathcal{A}$ and $\mathcal{B}$ with the addition of a domain classifier resulting in domain-invariant features at an intermediate level.

2. Training a transformation mapping across tasks for $\mathcal{A}$. In addition to a transfer function performing normal reconstruction [26], we have also experimented with the architecture of the transfer function. We train a mapping architecturally similar to U-Net and promote finer extraction of features by the addition of spatial attention.

3. Obtaining output on target domain $\mathcal{B}$ from the learnt transfer function.

Training the base models: The first step is to train the neural networks $\mathcal{N}_1$ and $\mathcal{N}_2$ on their respective tasks. The base network $\mathcal{N}_1^{A\cup B}$ is trained for task $\mathcal{T}_1$ on both domains $\mathcal{A}$ and $\mathcal{B}$, while the network $\mathcal{N}_2^{\mathcal{A}}$ is trained for task $\mathcal{T}_2$ only on domain $\mathcal{A}$.

In contrast to [26], our work makes modifications in the training procedure of $\mathcal{N}_1$ trained on both the domains. We introduce a binary domain classifier $\mathcal{C}$ to the base model $\mathcal{N}_1^{A\cup B}$ that performs classification on the abstract representations $E_1^{A\cup B}(x^k)$ where x_k is an image from either $\mathcal{A}$ or

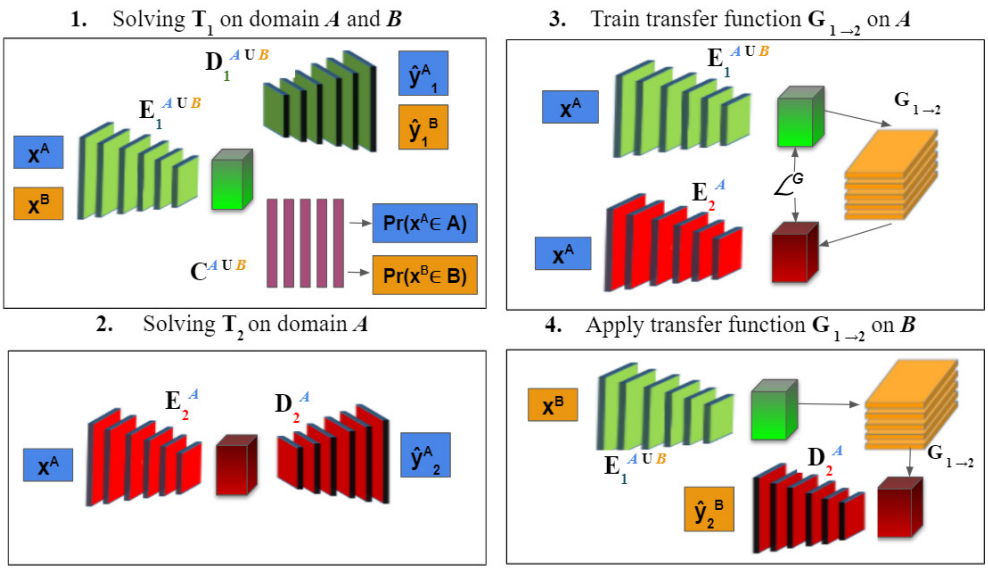

Figure 2: Overview of the ADA-AT/DT framework. (1) We introduce a novel architecture for $\mathcal{N}_1^{A\cup B}$ consisting of a binary domain classifier $\mathcal{C}^{A\cup B}$, trained in a min-max optimization setup. (2) We train $\mathcal{N}_2^A$ on domain A using available labeled data. (3) The transfer function $G_{1\to 2}^{\mathcal{A}}$ is utilized to transfer deep features from task $\mathcal{T}_1$ (green) to deep features for task $\mathcal{T}_2$ (red). (4) The transformed features provided by $G_{1\to 2}^{\mathcal{A}}$ are decoded to obtain data for task $\mathcal{T}_2$ on domain $\mathcal{B}$ without any supervision.

$\mathcal{B}$. The classifier predicts d_k such that $d_k = 1$ if $k = \mathcal{A}$ and $d_k = 0$ if $k = \mathcal{B}$. The domain classifier $\mathcal{C}$ and the encoder $E_1^{A\cup B}$ are trained using a min-max objective while the decoder $\mathcal{D}_1^{A\cup B}$ is trained to minimise a task-specific loss. However the parameters of $\mathcal{N}_2^A$ are learnt to minimize only a task-specific loss. For $\mathcal{N}_1^{A\cup B}$, we define the domain confusion loss for abstract features and the task-specific loss as follows:

$$\begin{aligned}
\mathcal{L}_{adv}^{\mathcal{A}}(x) &= \mathbb{E}_{x\sim\mathcal{A}}[\log C(E_1^{A\cup B}(x))] \\
\mathcal{L}_{adv}^{\mathcal{B}}(x) &= \mathbb{E}_{x\sim\mathcal{B}}[\log 1 - C(E_1^{A\cup B}(x))] \\
\mathcal{L}_{adv}(x) &= \mathcal{L}_{adv}^{\mathcal{A}}(x) + \mathcal{L}_{adv}^{\mathcal{B}}(x) \\
\mathcal{L}_{task}(x,y) &= \mathcal{L}_T(x,y)
\end{aligned} \tag{1}$$

Here, x is an image from either of the two domains. $\mathcal{L}_{task}$ is the loss function specific to the task: cross entropy for semantic segmentation and the mean absolute error for depth estimation. We define a hyper-parameter α which weights the two losses for adversarial training. Therefore, the optimization setup for $\mathcal{N}_1^{A\cup B}$ is defined as:

$$\min_{E_1,\mathcal{D}_1} \max_{\mathcal{C}} (1-\alpha)\mathcal{L}_{adv} + \alpha\mathcal{L}_{task} \tag{2}$$

Figure 2 shows the sharing of $E_1^{A\cup B}$ into the domain classifier and the task-specific decoder. The aim of min-max optimization is to constraint the encoder to learn a representation of source images that are maximally task-descriptive and domain-invariant at the same time. The parameter α decides the contribution of each term in the adversarial optimization.

Training the transfer function: Once the two base models are individually trained on their respective domains and tasks, a transfer function $G_{1\to 2}$ is learnt to transfer between tasks by learning a mapping function from $\mathcal{T}_1$ to $\mathcal{T}_2$. The transfer function trained on domain $\mathcal{A}$ under the supervision of the abstract deep features of both the tasks. We define the loss function for obtaining optimal parameters for the transfer function as the mean absolute error between the transformed representation and the target task features as shown in Equation 3

$$\mathcal{L}_G = |G_{1\to 2}^{\mathcal{A}}(E_1^{\mathcal{A}\cup\mathcal{B}}(x_{\mathcal{A}})) - E_2^{\mathcal{A}}(x_{\mathcal{A}})| \tag{3}$$

In contrast to [26], our work also comprises of a detailed study on different architectures of the transfer function. The main idea behind the analysis of different types of transfer functions is to exploit the information provided by the abstract representation efficiently for better generalization on target domains and tasks. Our primary focus lies in experimenting with the U-Net architecture with spatial attention.

Testing the transfer function on $\mathcal{B}$: To solve the task $\mathcal{T}_2$ on $\mathcal{B}$, we use $G_{1\to 2}^{\mathcal{A}}$ to transform features from the abstract representation produced by $E_1^{\mathcal{A}\cup\mathcal{B}}$ for $x_{\mathcal{B}}$.

$$\hat{y}_2^{\mathcal{B}} = \mathcal{D}_2^{\mathcal{A}}(G_{1\to 2}^{\mathcal{A}}(E_1^{\mathcal{A}\cup\mathcal{B}}(x_{\mathcal{B}}))) \tag{4}$$

The output of this encoder is then passed through $G_{1\to 2}$ and the decoded by $\mathcal{D}_2^{\mathcal{A}}$ to obtain the output of the target task on $\mathcal{B}$ for which labeled data was unavailable.

4. Experimental Details

4.1. Datasets

We conduct our experiments on four datasets, with the aim of benchmarking our methodology on data sets collected from real-life sources. Regarding tasks for which labeled real-life data is scarce, the availability of a huge amount of labeled data from synthetic sources may be exploited. We consider the synthetic datasets to be the Synthia-SF dataset and the data acquired by the Carla simulator. For real datasets, we use images from the KITTI and CityScapes datasets. The Synthia dataset [15] consists of 2224 images for which labeled data for both depth estimation and semantic segmentation is available. The Carla dataset is collected from the simulator [6] for sunny, cloudy and rainy weather to ensure diversity comparable to real life conditions. For Cityscapes [5], we have used all the images in the validation and training split to benchmark our proposed method. The KITTI dataset [9] consists of 200 images for both semantic segmentation and depth estimation. We used only 11 labels of the 19 training ids used in Cityscapes and 22 training ids in Synthia because we wanted to align our tests results also with CARLA which provided only 11 semantic classes. We implement data augmentation by performing a random brightness and color augmentation with a probability of 50 % on the RGB input images.

4.2. Network Architecture

The main aim of this paper to perform adversarial domain adaptation on intermediate abstract representations by introducing a domain classifier to create confusion between the domains. Major problems in training GANs are vanishing gradients, non-convergence, mode collapse etc. The model trained by [26] for depth estimation and semantic segmentation collapses to a non-optimal mode and faces vanishing gradients when combined with the adversarial domain adaptation framework. The model proposed in [26] consists of an encoder with a large number of learnable parameters along with the usage of operations like max-pooling which is known to hinder the convergence and performance of a GAN. Therefore, we focus on employing a model that not only generates superior and meaningful representations but also tackles the challenges faced by adversarial training. We employ the model architecture used in [40] for unpaired image to image translation due to its ability to generate optimal deep features for transformation across tasks. The major merit of our proposed method is that we use fewer parameters for both base models and transfer functions while simultaneously enhancing performance on target domains. Table 1 presents the total number of parameters in our model compared to [26]. As seen from Table 1, our base model operating with the domain classifier $\mathcal{C}^{\mathcal{A}\cup\mathcal{B}}$ consists of only 16.8 million parameters which is 81 % less than the base model in [26]. We show that architecturally smaller networks face the problem of domain shift at deeper abstract levels which can be overcome using adversarial domain adaptation and more efficient transfer functions, making our framework less computationally expensive and superior in terms of performance.

Model	Number of Parameters (in million)
Base model [26]	90.1
Base model (Ours)	7.8
Base model with $\mathcal{C}^{\mathcal{A}\cup\mathcal{B}}$ (ours)	16.8
$G^{\mathcal{A}}_{1\to 2}$ [26]	226.9
$G^{\mathcal{A}}_{1\to 2}$ (U-Net)	53.2
$G^{\mathcal{A}}_{1\to 2}$ (U-Net + att.)	71.8

Table 1: A comparative study of number of parameters in networks used by our method and [26].

4.3. Study of Transfer functions

To extract and transfer adequate information from deep features of task $\mathcal{T}_1$ suitable for $\mathcal{T}_2$, our aim is to search for a network that captures contextual information relevant to the task. To accomplish this, we obtain better results by employing a transfer function with an architecture similar to U-Net introduced in [27] apart from the conventional convolution and deconvolution networks. The U-Net is known to produce better deep features that combines the localization information from downsampling network and abstract information from the upsampling path. We also experiment with attention modules combined with U-Net transfer functions which has reported superior results in most cases. Therefore we present our results for three transfer functions:

- A conventional transfer function, consisting of convolution and upconvolution layers. The network consists of three convolution and three upconvolution layers to obtain the original size of the feature space. We denote this transfer function as 'conv'.
- A U-Net like transfer function transferring features from the downsampling encoder to the decoder. The function contains three downsampling levels and and three upsampling levels. The bottleneck layers consists of two ResNet blocks. We denote this transfer function as 'U-Net'.
- A U-Net transfer with attention module. The architecture of this network is similar to the previous function except for the addition of spatial attention. This transfer function has been denoted by 'U-Net + att.'

In [26], the transformation of deep features is performed for 2048 channels, leading to a huge number of trainable pa-

rameters in the transfer function. We reduce the number of parameters by employing transfer function that utilize the extracted features proficiently at a comparably smaller dimension of 256. Please refer to Table 1 for a comparison of number of parameters in transfer functions.

4.4. Evaluation Metrics

To compare results with available ground truth, we report scores for a variety of evaluation metrics for both the tasks to compare our results with [26]. For semantic segmentation, we report pixel wise accuracy and Mean Intersection Over Union (mIoU). In order to examine the segmentation in more detail, we also report the intersection over union per class. For depth estimation, we use the metrics: Absolute Relative Error (Abs Rel), Square Relative Error (Sq Rel), Root Mean Square Error (RMSE), logarithmic RMSE. Lower values of these quantities indicate better performance and accuracy. We also report three accuracy scores: δ_α being the percentage of predictions whose maximum between ratio and inverse ratio with respect to the ground truth is lower than 1.25^α.

4.5. Training setup

We train our models on random crop of RGB images of dimension 128 x 128. The deepest features i.e the lowest spatial resolution of our model is 1/4 the size of the input image. The binary domain classifier $\mathcal{C}^{\mathcal{A}\cup\mathcal{B}}$ for depth estimation is trained using a least-square loss for stability and better predictions. Therefore, when $\mathcal{T}_1$ is depth estimation, the combined loss to train $\mathcal{N}_1^{\mathcal{A}\cup\mathcal{B}}$ from Equation 1 is the weighted sum of a L_1 loss and least-squares loss. On the other hand, binary cross entropy provides better results on domain classification when $\mathcal{T}_1$ semantic segmentation. Further details on choosing the training objective for GAN are provided in Section 4.6. We also soften labels by a factor of 0.1 to prevent the discriminator from being overconfident. For example, for the case $\mathcal{A}$ = Synthia and $\mathcal{B}$ = Carla, we use a label of 0.1 for Synthia while training the encoder and 0.9 while training the classifier using the loss defined in Equation 2. We used the Adam Optimiser with a learning rate of 5e-4 and is decreased by factor of 0.5 after every 10 epochs. We train the network for a total of 500 epochs.

4.6. Selecting the best domain classifier

Selecting the training setup for the domain classifier is a crucial step to obtain domain indistinguishable features which are also adequate for the specified task. As shown in equation 2, the encoder and the domain classifier have been trained in a min-max optimization fashion. The task specific loss has been weighted by a parameter α, therefore we have conducted extensive studies to choose the value of this hyper-parameter. In case of a GAN, there are many choices for the adversarial objective. We performed experiments on different optimization objectives namely: Binary cross entropy, Mean square error and the Wasserstein distance. Mean square error and wasserstein distance are usually employed to tackle vanishing gradients and achieve more stability while training. Therefore, it is necessary to inspect the performance of these networks for different training losses. We report our experiments for $\mathcal{N}^{A\cup B}$ for semantic segmentation trained on the Synthia CityScapes datasets. As mentioned in Section 4.5, we use soft labels while training the domain classifiers. We experiment on three choices for value of the parameter α: 0.4, 0.5 and 0.6. In Figure 3, we present the t-SNE visualizations of features at the lowest spatial resolution which are obtained by the encoder and the Mean Intersection over Union for each of the values of α and training loss. For this experiment, we train the models for only 10 epochs to judge each training setup by its capability of producing indistinguishable features and considerably better mIoU. The three columns in Figure 3 belong to the three values of $\alpha = 0.4$, 0.5 and 0.6 respectively. The mIoU has been shown in the bottom left corner of each of the plots. From Figure 3, it is clear that for semantic segmentation the best features demonstrating domain-invariance are obtained when the training objective of the domain classifier is binary cross entropy. Although the best predictions on semantic segmentation classes are obtained by a mean-square error for domain classifier. But, the features in this case are not as indistinguishable as desired. We observe that similar distinguishable t-SNE visualizations are obtained for wasserstein distance. Therefore, binary cross entropy is observed to superior compared to other objectives for semantic segmentation. In terms of mIoU, the value of α is chosen to be 0.4 as it provides better predictions on the 11 chosen classes. Similarly for depth estimation, it has been observed through experiments that mean square error has provided superior results in terms of both domain-invariance features and task-specific metrics. Extensive experiments performed on the model architecture of the domain classifier are reported in the supplementary material.

5. Results

In this section, we discuss the results obtained by our proposed methods and compare it with existing cross task cross domain method [26]. We have compared the results of our proposed method with five additional current state of the art methods: pixel-level domain adaptation (denoted by AT/DT (DA)), training setup introduced in [40] for unpaired image to image translation (denoted by Cycle-GAN), domain adaptation methods like CYCADA [1], FCN in the Wild [13] and Deep Adaptation networks (DAN) [20]. For the last three methods, we train the models on 11 classes to obtain metrics consistent with [26]. For each combination of $\mathcal{A}$ and $\mathcal{B}$, we present the performance of our method with

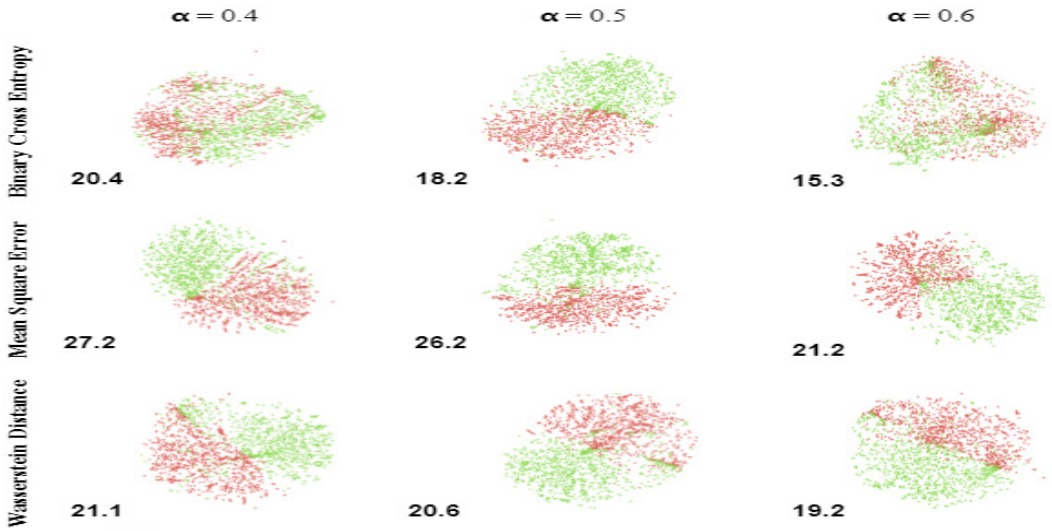

Figure 3: We conducted a study on different types of GANs to choose the most optimized network which provides domain-invariant as well as task-discriminative features. The hyper-parameter α in Equation 2 and optimization objective for domain classifier has to be chosen carefully for stable training. The Mean Intersection over Union is shown in the bottom left corner of each setup. We train the base model $\mathcal{N}^{A \cup B}$ on the Synthia(red) and CityScapes(green) datasets.

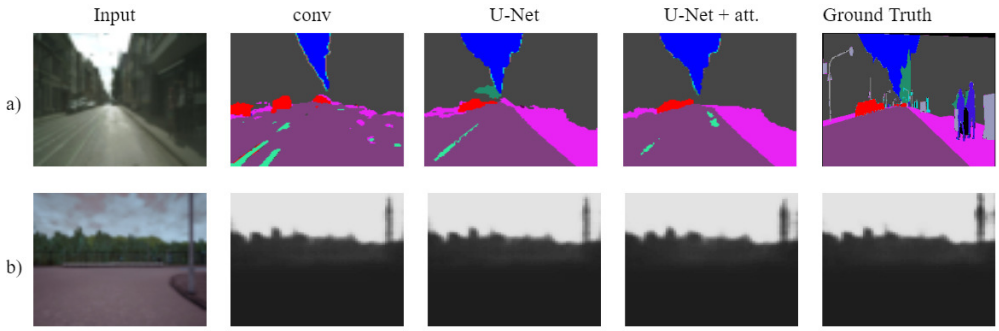

Figure 4: Qualitative results for ADA-AT/DT training setup.

three different types of transfer functions mentioned in Section 4.3. We also present qualitative results for task transfer across domains with our proposed method in Figure 4.

Depth to Semantic: In this setup, labeled data is available for both tasks in domain $\mathcal{A}$ and only for depth estimation in $\mathcal{B}$. We have compared the results of our framework with the performance metrics reported in [26]. Table 2 reports evaluation metrics as discussed in Section for domain B for monocular depth estimation. The main problem faced while learning across tasks to predict semantic segmentation is the lack of major information about small and comparably insignificant objects like Poles, Traffic lights learnt for depth estimation, hence providing inadequate information transfer for semantic segmentation. As seen from Table 2-(a), a major increment has been observed in accuracy and mIoU for U-Net and U-Net + att. transfer functions. For the classes Road, Poles, and Sky, our model is unable to provide results better than [26] by small margins. However, for classes Person, Vegetation, Vehicles and Traffic signs, our model is almost 10 times more accurate. As seen in Table 2-(b), our proposed method leads the existing methods to a higher pixel accuracy and mIoU. Only the IoU values of the classes Person, Poles, and Vegetation obtained from the pixel level domain adaptation experiments performed in [26] are greater than those obtained by our method. In this case, the best performance is recorded by our method with the U-Net transfer function. Figure 4-(b) shows the qualitative output of semantic segmentation on the Carla dataset where task transfer is learnt on the Synthia-SF dataset.

Semantic to Depth: In this setup, labeled data is available for both tasks in domain $\mathcal{A}$ and only for semantic segmentation in $\mathcal{B}$. We have compared the results of our framework with the performance metrics reported in [26]. Table 3 reports evaluation metrics as discussed in Section for

	$\mathcal{A}$	$\mathcal{B}$	Method	Road	Sidewalk	Walls	Fence	Person	Poles	Vegetation	Vehicles	Tr. Sign	Building	Sky	mIoU	Acc
	Synthia	CityScapes	AT/DT [26]	**85.77**	29.40	1.23	0.00	3.72	14.55	1.87	8.85	0.38	42.79	**67.06**	23.34	64.03
(b)	Synthia	CityScapes	CYCADA [1]	72.89	21.74	0.00	0.00	0.77	**21.03**	0.00	16.66	1.34	32.67	46.71	16.23	69.03
	Synthia	CityScapes	FCN in the wild [13]	60.23	30.33	0.48	0.00	0.53	3.64	4.30	39.44	2.72	53.48	59.06	19.51	65.93
	Synthia	CityScapes	DAN [20]	54.09	3.67	0.00	0.00	0.00	0.00	2.23	18.88	0.00	1.79	41.59	9.87	51.65
	Synthia	CityScapes	**Ours (conv)**	70.21	21.78	3.71	0.00	1.36	1.79	3.69	17.31	2.85	69.82	47.04	21.77	69.93
	Synthia	CityScapes	**Ours (U-Net)**	77.30	29.51	7.67	0.00	4.74	4.63	7.37	27.36	6.01	72.44	51.08	26.19	73.99
	Synthia	CityScapes	**Ours (U-Net + att.)**	84.39	**38.56**	**10.71**	0.00	**9.11**	8.14	**11.15**	**41.86**	**8.95**	**77.94**	58.69	**31.78**	**79.68**
	Carla	CityScapes	AT/DT [26]	76.44	32.24	4.75	5.58	24.49	**24.95**	68.98	40.49	10.78	69.38	78.19	39.66	76.37
	Carla	CityScapes	CYCADA [1]	73.94	47.53	0.00	2.50	1.61	0.00	56.64	21.85	0.63	18.46	52.03	24.56	68.68
(c)	Carla	CityScapes	FCN in the wild [13]	57.26	53.76	3.72	0.42	0.65	0.12	30.17	4.411	0.00	31.11	6.177	18.16	65.06
	Carla	CityScapes	DAN [20]	68.61	23.89	0.00	0.00	0.00	0.00	40.08	42.11	0.00	3.52	52.53	17.82	62.37
	Carla	CityScapes	Cycle-GAN [40]	81.58	39.15	6.08	5.31	**30.22**	21.73	**77.71**	50.00	8.33	68.35	77.22	42.33	80.93
	Carla	CityScapes	AT/DT (DA)[26]	85.19	41.37	5.44	3.02	29.90	24.07	71.93	58.09	7.53	70.90	77.78	43.20	81.92
	Carla	CityScapes	**Ours (conv)**	62.07	28.47	5.69	2.2	2.16	1.71	41.54	23.77	0.00	27.38	39.51	21.31	54.31
	Carla	CityScapes	**Ours (U-Net)**	**85.54**	**68.54**	12.31	**6.71**	23.22	22.46	71.72	**66.66**	**13.79**	**74.45**	80.01	**47.76**	**84.92**
	Carla	CityScapes	**Ours (U-Net + att.)**	81.41	58.42	**12.45**	4.08	14.8	13.54	63.81	56.05	12.09	69.45	**81.09**	42.69	84.52

Table 2: Quantitative results obtained from depth estimation to semantic segmentation task transfer across domains

				Lower is better				Higher is better		
	$\mathcal{A}$	$\mathcal{B}$	Method	Abs Rel	Sq Rel	RMSE	RMSE log	δ_1	δ_2	δ_3
	Synthia	Carla	AT/DT [26]	0.316	5.485	11.712	0.458	0.553	0.785	0.880
(a)	Synthia	Carla	**Ours (conv)**	0.2126	1.9635	2.2695	0.3755	0.8029	0.9125	0.9538
	Synthia	Carla	**Ours (U-Net)**	0.451	1.2699	1.122	0.2483	0.878	0.9638	0.9834
	Synthia	Carla	**Ours (U-Net + att.)**	**0.0796**	**0.19125**	**0.6885**	**0.1937**	**0.9128**	**0.9732**	**0.987**
	Carla	CityScapes	AT/DT [26]	0.394	5.837	13.915	0.435	0.337	0.749	0.899
(b)	Carla	CityScapes	Cycle-GAN [40]	0.943	27.026	21.666	0.695	0.218	0.478	0.690
	Carla	CityScapes	AT/DT (DA) [26]	0.563	10.789	15.636	0.489	0.247	0.668	0.861
	Carla	CityScapes	**Ours(conv)**	1.2016	19.893	7.599	0.7528	0.6334	0.8046	0.8754
	Carla	CityScapes	**Ours (U-Net)**	0.3397	3.264	3.2385	0.5119	0.7233	0.8753	0.9275
	Carla	CityScapes	**Ours (U-Net + att.)**	**0.1894**	**0.9078**	**1.224**	**0.3169**	**0.8245**	**0.9303**	**0.9602**
	Carla	Kitti	AT/DT [26]	0.439	8.263	9.148	0.421	0.483	0.788	0.891
	Carla	Kitti	**Ours(conv)**	0.5631	7.0686	3.2385	0.5325	0.7038	0.8681	0.9268
(c)	Carla	Kitti	**Ours (U-Net)**	0.233	1.8054	1.785	0.3845	0.7886	0.9208	0.9578
	Carla	Kitti	**Ours (U-Net + att.)**	**0.1389**	**0.5381**	**0.7905**	**0.2476**	**0.8655**	**0.9566**	**0.9781**

Table 3: Quantitative results obtained from semantic segmentation to depth estimation task transfer across domains

domain B for monocular depth estimation. The integration of domain adaptation at an abstract level has significantly improved the metrics. For all the three combinations of synthetic and real datasets, the U-Net transfer function along with attention boosts performance by a large margin as compared to other transfer functions. As seen from Table 3-(a), the values of errors have decreased by more than 50 %, in comparison to the existing method. The same trend is followed in the cases in Table 3-(b) and Table 3-(c), where knowledge transfer occurs across synthetic and real domains. Despite only 200 data points being available for the KITTI dataset, our proposed method is successful in learning a domain invariant representation at an abstract level, resulting in an notable enhancement in performance in Table 3-(c). In most cases, our ADA-AT/DT model with a convolution based transfer function outperforms the existing method by a significant margin, indicating that training ADA-AT/DT models is suffice to surpass depth estimation results from previous methods. Gradual increase in accuracy of predictions is observed with a U-Net transfer function. As observed from Table 3, the benefit of our proposed method is that overall prediction of depth estimation has been improved by models with substantially less number of trainable parameters. Figure 4-(a) shows the qualitative output of semantic segmentation on the CityScapes dataset where task transfer is learnt on the Carla dataset.

6. Conclusions

We proposed a novel adversarial training driven approach for cross-domain and cross-task knowledge propagation for visual scene understanding in this paper. In particular, we consider semantic segmentation and depth estimation tasks where annotations are available for both the tasks for a source domain while it is available for one of the tasks in the target domain. Given that, the goal is to obtain the outputs for the tasks which is devoid of any supervision in the target domain. The rationale behind considering the adversarial approach is to ensure that any domain-gap between the source and target can explicitly be reduced. Experimental results confirm the robustness of our model. We are currently interested in extending the model for multiple tasks where the correlation among the tasks may vary substantially.

References

[1] Cycada: Cycle consistent adversarial domain adaptation. In *International Conference on Machine Learning (ICML)*, 2018.

[2] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2017.

[3] Sumanth Chennupati, Ganesh Sistu, Senthil Yogamani, and Samir Rawashdeh. Auxnet: Auxiliary tasks enhanced semantic segmentation for automated driving. *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2019.

[4] R. Cipolla, Y. Gal, and A. Kendall. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7482–7491, 2018.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[6] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator, 2017.

[7] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation, 2014.

[8] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor S. Lempitsky. Domain-adversarial training of neural networks. *ArXiv*, abs/1505.07818, 2016.

[9] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[10] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 2551–2559, USA, 2015. IEEE Computer Society.

[11] Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *ECCV*, 2016.

[12] Robert Harb and Patrick Knöbelreiter. Efficient multi-task learning of semantic segmentation and disparity estimation. 2019.

[13] Judy Hoffman, Dequan Wang, F. Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *ArXiv*, abs/1612.02649, 2016.

[14] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

[15] Daniel Hernández Juárez, Lukas Schneider, Antonio Espinosa, David Vázquez, Antonio M. López, Uwe Franke, Marc Pollefeys, and Juan C. Moure. Slanted stixels: Representing san francisco's steepest streets. *ArXiv*, abs/1707.05397, 2017.

[16] L. Liebel and M. Körner. Multidepth: Single-image depth estimation via multi-task regression and classification. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 1440–1447, 2019.

[17] Xiao Lin, Dalila Sánchez-Escobedo, Josep R. Casas, and Montse Pardàs. Depth estimation and semantic segmentation from a single rgb image using a hybrid convolutional neural network. *Sensors (Basel, Switzerland)*, 19, 2019.

[18] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 469–477. Curran Associates, Inc., 2016.

[19] S. Liu, E. Johns, and A. J. Davison. End-to-end multi-task learning with attention. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1871–1880, 2019.

[20] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, page 97–105. JMLR.org, 2015.

[21] Ishan Misra, Abhinav Shrivastava, Abhinav Gupta, and Martial Hebert. Cross-stitch networks for multi-task learning. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3994–4003, 2016.

[22] A. Mousavian, H. Pirsiavash, and J. Košecká. Joint semantic segmentation and depth estimation with deep convolutional networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 611–619, 2016.

[23] Vladimir Nekrasov, Thanuja Dharmasiri, Andrew Spek, Tom Drummond, Chunhua Shen, and Ian D. Reid. Real-time joint semantic segmentation and depth estimation using asymmetric annotations. *2019 International Conference on Robotics and Automation (ICRA)*, pages 7101–7107, 2019.

[24] Kuan-Chuan Peng, Ziyan Wu, and Jan Ernst. Zero-shot deep domain adaptation. In *ECCV*, 2018.

[25] Pierluigi Zama Ramirez, Matteo Poggi, Fabio Tosi, Stefano Mattoccia, and Luigi di Stefano. Geometry meets semantics for semi-supervised monocular depth estimation. In *ACCV*, 2018.

[26] Pierluigi Zama Ramirez, Alessio Tonioni, Samuele Salti, and Luigi di Stefano. Learning across tasks and domains. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8109–8118, 2019.

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *ArXiv*, abs/1505.04597, 2015.

[28] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017.

[29] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *ECCV Workshops*, 2016.

[30] Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Pieter Abbeel, Sergey Levine, Kate Saenko, and Trevor Darrell. Adapting deep visuomotor representations with weak pairwise constraints, 2015.

[31] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks, 2015.

[32] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4068–4076, 2015.

[33] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017.

[34] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *ArXiv*, abs/1412.3474, 2014.

[35] Simon Vandenhende, Bert De Brabandere, and Luc Van Gool. Branched multi-task networks: Deciding what layers to share. *ArXiv*, abs/1904.02920, 2019.

[36] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey, 2018.

[37] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation, 2017.

[38] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014.

[39] Amir R Zamir, Alexander Sax, , William B Shen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2018.

[40] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[41] Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, and Qing He. A comprehensive survey on transfer learning. *ArXiv*, abs/1911.02685, 2019.