CrDoCo: Pixel-level Domain Transfer with Cross-Domain Consistency

Yun-Chun Chen[1,2]   Yen-Yu Lin[1]   Ming-Hsuan Yang[3,4]   Jia-Bin Huang[5]

[1]Academia Sinica   [2]National Taiwan University   [3]UC Merced   [4]Google   [5]Virginia Tech
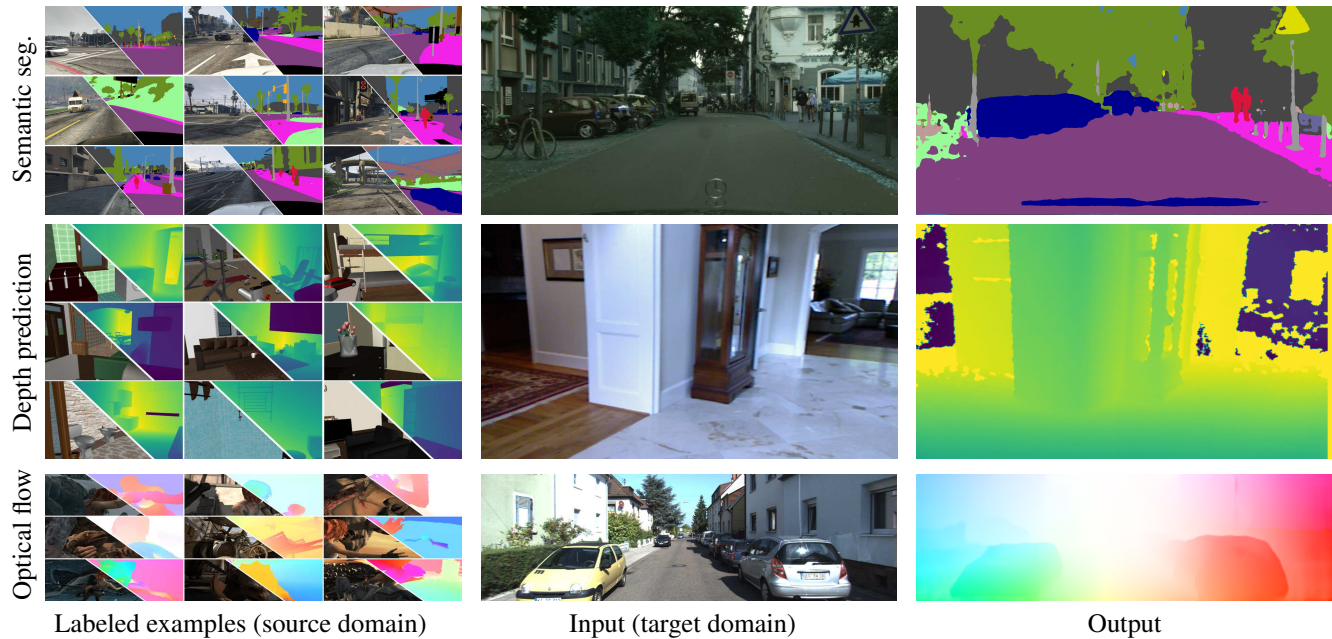
Figure 1: **Applications of the proposed method.** Our method has the applications ranging from semantic segmentation (top row), depth prediction (middle row), to optical flow estimation (bottom row).

Abstract

Unsupervised domain adaptation algorithms aim to transfer the knowledge learned from one domain to another (e.g., synthetic to real images). The adapted representations often do not capture pixel-level domain shifts that are crucial for dense prediction tasks (e.g., semantic segmentation). In this paper, we present a novel pixel-wise adversarial domain adaptation algorithm. By leveraging image-to-image translation methods for data augmentation, our key insight is that while the translated images between domains may differ in styles, their predictions for the task should be consistent. We exploit this property and introduce a cross-domain consistency loss that enforces our adapted model to produce consistent predictions. Through extensive experimental results, we show that our method compares favorably against the state-of-the-art on a wide variety of unsupervised domain adaptation tasks.

1. Introduction

Deep convolutional neural networks (CNNs) are extremely data hungry. However, for many dense prediction tasks (e.g., semantic segmentation, optical flow estimation, and depth prediction), collecting large-scale and diverse datasets with pixel-level annotations is difficult since the labeling process is often expensive and labor intensive (see Figure 1). Developing algorithms that can transfer the knowledge learned from one labeled dataset (i.e., source domain) to another unlabeled dataset (i.e., target domain) thus becomes increasingly important. Nevertheless, due to the domain-shift problem (i.e., the domain gap between the source and target datasets), the learned models often fail to generalize well to new datasets.

To address these issues, several unsupervised domain adaptation methods have been proposed to align data distributions between the source and target domains. Existing methods either apply feature-level [39, 26, 44, 42, 15, 14]

or pixel-level [1, 36, 7, 14] adaptation techniques to minimize the domain gap between the source and target datasets. However, aligning marginal distributions does not necessarily lead to satisfactory performance as there is no explicit constraint imposed on the predictions in the target domain (as no labeled training examples are available). While several methods have been proposed to alleviate this issue via curriculum learning [34, 6] or self-paced learning [53], the problem remains challenging since these methods may only learn from cases where the current models perform well.

Our work. In this paper, we present CrDoCo, a pixel-level adversarial domain adaptation algorithm for dense prediction tasks. Our model consists of two main modules: 1) an image-to-image translation network and 2) two domain-specific task networks (one for source and the other for target). The image translation network learns to translate images from one domain to another such that the translated images have a similar distribution to those in the translated domain. The domain-specific task network takes images of source/target domain as inputs to perform dense prediction tasks. As illustrated in Figure 2, our core idea is that while the original and the translated images in two different domains may have different styles, their predictions from the respective domain-specific task network should be exactly the same. We enforce this constraint using a *cross-domain consistency loss* that provides additional supervisory signals for facilitating the network training, allowing our model to produce consistent predictions. We show the applicability of our approach to multiple different tasks in the unsupervised domain adaptation setting.

Our contributions. First, we present an adversarial learning approach for unsupervised domain adaptation which is applicable to a wide range of dense prediction tasks. Second, we propose a cross-domain consistency loss that provides additional supervisory signals for network training, resulting in more accurate and consistent task predictions. Third, extensive experimental results demonstrate that our method achieves the state-of-the-art performance against existing unsupervised domain adaptation techniques. Our source code is available at https://yunchunchen.github.io/CrDoCo/

2. Related Work

Unsupervised domain adaptation. Unsupervised domain adaptation methods can be categorized into two groups: 1) feature-level adaptation and 2) pixel-level adaptation. Feature-level adaptation methods aim at aligning the feature distributions between the source and target domains through measuring the correlation distance [39], minimizing the maximum mean discrepancy [26], or applying adversarial learning strategies [44, 42] in the feature
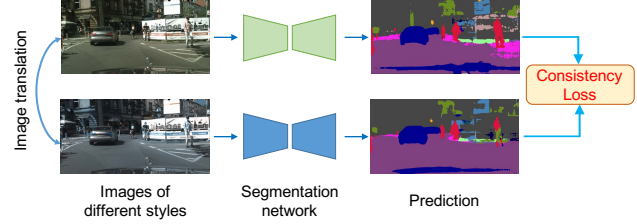

Figure 2: **Main idea.** While images may have different appearances/styles in different domains, their task predictions (e.g., semantic segmentation as shown in this example) should be exactly the same. Our core idea in this paper is to impose a cross-domain consistency loss between the two task predictions.

space. In the context of image classification, several methods [10, 11, 26, 27, 43, 44] have been developed to address the domain-shift issue. For semantic segmentation tasks, existing methods often align the distributions of the feature activations at multiple levels [15, 17, 42]. Recent advances include applying class-wise adversarial learning [4] or leveraging self-paced learning policy [53] for adapting synthetic-to-real or cross-city adaptation [4], adopting curriculum learning for synthetic-to-real foggy scene adaptation [34], or progressively adapting models from daytime scene to nighttime [6]. Another line of research focuses on pixel-level adaptation [1, 36, 7]. These methods address the domain gap problem by performing data augmentation in the target domain via image-to-image translation [1, 36] or style transfer [7] methods.

Most recently, a number of methods tackle joint feature-level and pixel-level adaptation in image classification [14], semantic segmentation [14], and single-view depth prediction [48] tasks. These methods [14, 48] utilize image-to-image translation networks (e.g., the CycleGAN [51]) to translate images from source domain to target domain with pixel-level adaptation. The translated images are then passed to the task network followed by a feature-level alignment.

While both feature-level and pixel-level adaptation have been explored, aligning the marginal distributions without enforcing explicit constraints on target predictions would not necessarily lead to satisfactory performance. Our model builds upon existing techniques for feature-level and pixel-level adaptation [14, 48]. The key difference lies in our cross-domain consistency loss that explicitly penalizes inconsistent predictions by the task networks.

Cycle consistency. Cycle consistency constraints have been successfully applied to various problems. In image-to-image translation, enforcing cycle consistency allows the network to learn the mappings without paired data [51, 22].

In semantic matching, cycle or transitivity based consistency loss help regularize the network training [50, 49, 3]. In motion analysis, forward-backward consistency check can be used for detecting occlusion [28, 20, 52] or learning visual correspondence [45]. Similar to the above methods, we show that enforcing two domain-specific networks to produce consistent predictions leads to substantially improved performance.

Learning from synthetic data. Training the model on large-scale synthetic datasets has been extensively studied in semantic segmentation [41, 42, 15, 14, 7, 17, 34, 35, 53], multi-view stereo [18], depth estimation [48], optical flow [40, 19, 21], amodal segmentation [16], and object detection [7, 30]. In our work, we show that the proposed cross-domain consistency loss can be applied not only to synthetic-to-real adaptation but to real-to-real adaptation tasks as well.

3. Method

In this section, we first provide an overview of our approach. We then describe the proposed loss function for enforcing cross-domain consistency on dense prediction tasks. Finally, we describe other losses that are adopted to facilitate network training.

3.1. Method overview

We consider the task of unsupervised domain adaptation for dense prediction tasks. In this setting, we assume that we have access to a source image set X_S, a source label set Y_S, and an unlabeled target image set X_T. Our goal is to learn a task network F_T that can reliably and accurately predict the dense label for each image in the target domain.

To achieve this task, we present an end-to-end trainable network which is composed of two main modules: 1) the image translation network $G_{S \to T}$ and $G_{T \to S}$ and 2) two domain-specific task networks F_S and F_T. The image translation network translates images from one domain to the other. The domain-specific task network takes input images to perform the task of interest.

As shown in Figure 3, the proposed network takes an image I_S from the source domain and another image I_T from the target domain as inputs. We first use the image translation network to obtain the corresponding translated images $I_{S \to T} = G_{S \to T}(I_S)$ (in the target domain) and $I_{T \to S} = G_{T \to S}(I_T)$ (in the source domain). We then pass I_S and $I_{T \to S}$ to F_S, I_T and $I_{S \to T}$ to F_T to obtain their task predictions.

3.2. Objective function

The overall training objective $\mathcal{L}$ for training the proposed network consists of five loss terms. First, the image-level adversarial loss $\mathcal{L}_{\text{adv}}^{\text{img}}$ aligns the image distributions between the translated images and the images in the corresponding domain. Second, the reconstruction loss $\mathcal{L}_{\text{rec}}$ regularizes the image translation network $G_{S \to T}$ and $G_{T \to S}$ to perform self-reconstruction when translating an image from one domain to another followed by a reverse translation. Third, the feature-level adversarial loss $\mathcal{L}_{\text{adv}}^{\text{feat}}$ aligns the distributions between the feature representations of the translated images and the images in the same domain. Fourth, the task loss $\mathcal{L}_{\text{task}}$ guides the two domain-specific task networks F_S and F_T to perform dense prediction tasks. Fifth, the cross-domain consistency loss $\mathcal{L}_{\text{consis}}$ enforces consistency constraints on the task predictions. Such a cross-domain loss couples the two domain-specific task networks F_S and F_T during training and provides supervisory signals for the unlabeled target domain image I_T and its translated one $I_{T \to S}$. Specifically, the training objective $\mathcal{L}$ is defined as

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{consis}} \cdot \mathcal{L}_{\text{consis}} + \lambda_{\text{rec}} \cdot \mathcal{L}_{\text{rec}} \\ + \lambda_{\text{img}} \cdot \mathcal{L}_{\text{adv}}^{\text{img}} + \lambda_{\text{feat}} \cdot \mathcal{L}_{\text{adv}}^{\text{feat}}, \end{aligned} \tag{1}$$

where λ_{consis}, λ_{rec}, λ_{img}, and λ_{feat} are the hyper-parameters used to control the relative importance of the respective loss terms. Below we outline the details of each loss function.

3.3. Cross-domain consistency loss $\mathcal{L}_{\text{consis}}$

Since we do not have labeled data in the target domain, to allow our model to produce accurate task predictions on unlabeled data, we first generate a translated version of I_T (i.e., $I_{T \to S}$) by passing I_T to the image translation network $G_{T \to S}$ (i.e., $I_{T \to S} = G_{T \to S}(I_T)$). Our key insight is that while I_T (belongs to the target domain) and $I_{T \to S}$ (belongs to the source domain) may differ in appearance or styles, these two images should have the same task prediction results (i.e., $F_T(I_T)$ and $F_S(I_{T \to S})$ should be exactly the same). We thus propose a cross-domain consistency loss $\mathcal{L}_{\text{consis}}$ that bridges the outputs of the two domain-specific task networks (i.e., F_S and F_T). The loss enforces the consistency between the two task predictions $F_T(I_T)$ and $F_S(I_{T \to S})$. For semantic segmentation task, we compute the bi-directional KL divergence loss and define the cross-domain consistency loss for semantic segmentation $\mathcal{L}_{\text{consis}}$ task as

$$\begin{aligned} & \mathcal{L}_{\text{consis}}(X_T; G_{S \to T}, G_{T \to S}, F_S, F_T) \\ = & - \mathbb{E}_{I_T \sim X_T} \sum_{h,w,c} f_{T \to S}(h, w, c) \log \left(f_T(h, w, c) \right) \\ & - \mathbb{E}_{I_T \sim X_T} \sum_{h,w,c} f_T(h, w, c) \log \left(f_{T \to S}(h, w, c) \right), \end{aligned} \tag{2}$$

where $f_T = F_T(I_T)$ and $f_{T \to S} = F_S(I_{T \to S})$ are the task predictions for I_T and $I_{T \to S}$, respectively, while c denotes the number of classes.
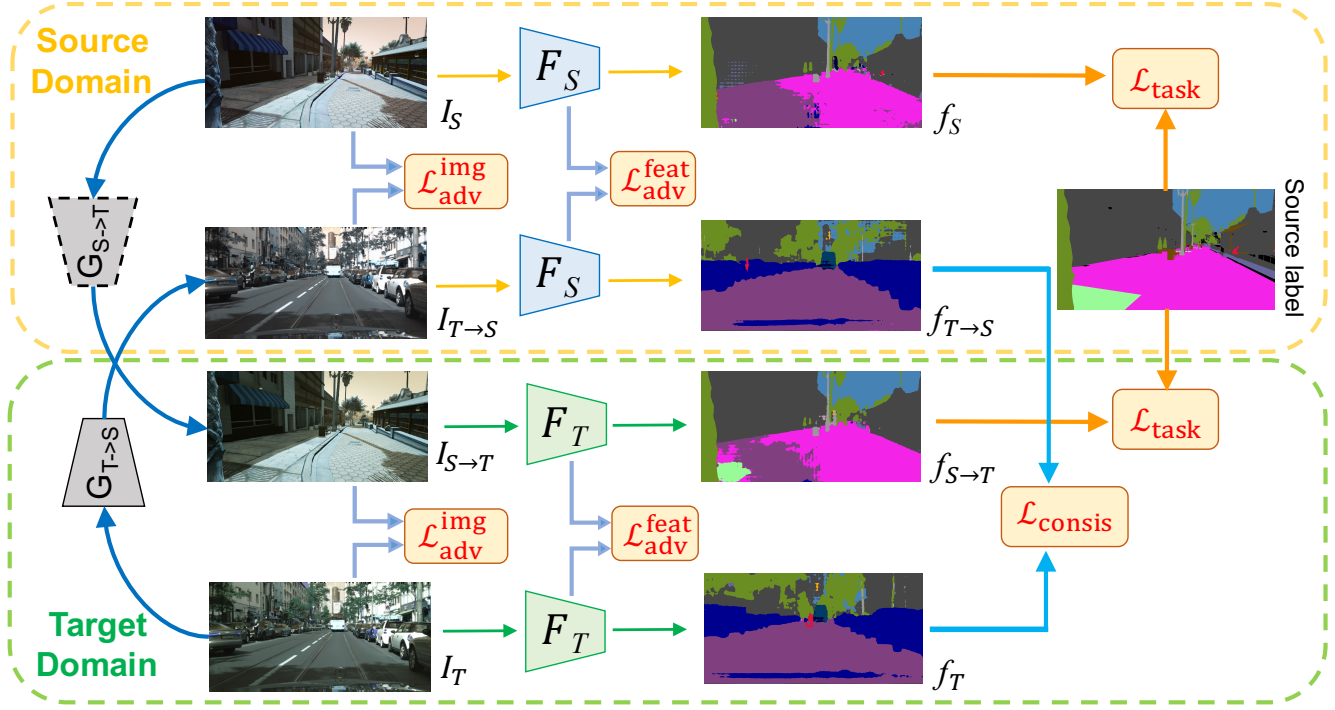
Figure 3: **Overview of the proposed method.** Our model is composed of two main modules: an image translation network (highlighted in gray) and two domain-specific task networks (highlighted in blue and green, respectively). The image translation network learns to translate input images from one domain to the other. The input and the translated images are then fed to their corresponding domain-specific task networks to perform task predictions. Our main contribution lies in the use of cross-domain consistency loss $\mathcal{L}_{\text{consis}}$ for regularizing the network training.

As our task models produce different outputs for different tasks, our cross-domain consistency loss $\mathcal{L}_{\text{consis}}$ is *task-dependent*. For depth prediction task, we use the ℓ_1 loss for the cross-domain consistency loss $\mathcal{L}_{\text{consis}}$. For optical flow estimation task, the cross-domain consistency loss $\mathcal{L}_{\text{consis}}$ computes the endpoint error between the two task predictions.

3.4. Other losses

In addition to the proposed cross-domain consistency loss $\mathcal{L}_{\text{consis}}$, we also adopt several other losses introduced in [14, 48, 51] to facilitate the network training.

Task loss $\mathcal{L}_{\text{task}}$. To guide the training of the two task networks F_S and F_T using labeled data, for each image-label pair (I_S, y_s) in the source domain, we first translate the source domain image I_S to $I_{S \to T}$ by passing I_S to $G_{S \to T}$ (i.e., $I_{S \to T} = G_{S \to T}(I_S)$). Similarly, images before and after translation should have the same ground truth label. Namely, the label for $I_{S \to T}$ is identical to that of I_S which is y_s.

We can thus define the task loss $\mathcal{L}_{\text{task}}$ for training the two domain-specific task networks F_S and F_T using *labeled* data. For semantic segmentation, we calculate the

cross-entropy loss between the task predictions and the corresponding ground truth labels as our task loss $\mathcal{L}_{\text{task}}$. Likewise, the task loss $\mathcal{L}_{\text{task}}$ is also task dependent. We use ℓ_1 loss for depth prediction task and endpoint error for optical flow estimation.

Feature-level adversarial loss $\mathcal{L}_{\text{adv}}^{\text{feat}}$. In addition to imposing cross-domain consistency and task losses, we apply two feature-level discriminators D_S^{feat} (for source domain) and D_T^{feat} (for target domain) [51]. The discriminator D_S^{feat} helps align the distributions between the feature maps of I_S (i.e., f_S) and $I_{T \to S}$ (i.e., $f_{T \to S}$). To achieve this, we define the feature-level adversarial loss in the source domain as

$$
\begin{aligned}
\mathcal{L}_{\text{adv}}^{\text{feat}} &(X_S, X_T; G_{T \to S}, \mathcal{F}_S, D_S^{\text{feat}}) \\
&= \mathbb{E}_{I_S \sim X_S}[\log(D_S^{\text{feat}}(f_S))] \\
&\quad + \mathbb{E}_{I_T \sim X_T}[\log(1 - D_S^{\text{feat}}(f_{T \to S}))].
\end{aligned}
\tag{3}
$$

Similarly, D_T^{feat} aligns the distributions between f_T and $f_{S \to T}$. This corresponds to another feature-level adversarial loss in the target domain as $\mathcal{L}_{\text{adv}}^{\text{feat}}(X_T, X_S; G_{S \to T}, \mathcal{F}_T, D_T^{\text{feat}})$.

Image-level adversarial loss $\mathcal{L}_{\text{adv}}^{\text{img}}$. In addition to feature-level adaptation, we also consider image-level adaptation between the translated images and those in the corresponding domain. Similar to Zhu *et al.* [51], we deploy two image-level discriminators D_S^{img} (for source domain) and D_T^{img} (for target domain). The D_S^{img} aims at aligning the distributions between the image I_S and the translated one $I_{T \to S}$. To accomplish this, we define the image-level adversarial loss in the source domain as

$$
\begin{aligned}
\mathcal{L}_{\text{adv}}^{\text{img}} &\left(X_S, X_T; G_{T \to S}, D_S^{\text{img}}\right) \\
&= \mathbb{E}_{I_S \sim X_S}[\log(D_S^{\text{img}}(I_S))] \\
&\quad + \mathbb{E}_{I_T \sim X_T}[\log(1 - D_S^{\text{img}}(I_{T \to S}))].
\end{aligned}
\tag{4}
$$

Similarly, we have another image-level adversarial loss in the target domain as $\mathcal{L}_{\text{adv}}^{\text{img}}(X_T, X_S; G_{S \to T}, D_T^{\text{img}})$.

Reconstruction loss $\mathcal{L}_{\text{rec}}$. Finally, we use an image reconstruction loss $\mathcal{L}_{\text{rec}}$ to regularize the training of the image translation network. We exploit the property that when translating an image from one domain to another followed by performing a reverse translation, we should obtain the same image. Namely, $G_{T \to S}(G_{S \to T}(I_S)) \approx I_S$ for any I_S in the source domain and $G_{S \to T}(G_{T \to S}(I_T)) \approx I_T$ for any I_T in the target domain hold.

More precisely, we define the reconstruction loss $\mathcal{L}_{\text{rec}}$ as

$$
\begin{aligned}
\mathcal{L}_{\text{rec}} &\left(X_S, X_T; G_{S \to T}, G_{T \to S}\right) \\
&= \mathbb{E}_{I_S \sim X_S}[\|G_{T \to S}(G_{S \to T}(I_S)) - I_S\|_1] \\
&\quad + \mathbb{E}_{I_T \sim X_T}[\|G_{S \to T}(G_{T \to S}(I_T)) - I_T\|_1].
\end{aligned}
\tag{5}
$$

Following Zhu *et al.* [51], we use the ℓ_1 norm to define the reconstruction loss $\mathcal{L}_{\text{rec}}$.

Based on the aforementioned loss functions, we aim to solve for a target domain task network F_T^* by optimizing the following min-max problem:

$$
F_T^* = \underset{F_T}{\arg\min} \min_{\substack{F_S, \\ G_{S \to T} \\ G_{T \to S}}} \max_{\substack{D_S^{\text{img}}, D_T^{\text{img}} \\ D_S^{\text{feat}}, D_T^{\text{feat}}}} \mathcal{L}.
\tag{6}
$$

Namely, to train our network using labeled source domain images and unlabeled target domain images, we minimize the cross-domain consistency loss $\mathcal{L}_{\text{consis}}$, the task loss $\mathcal{L}_{\text{task}}$, and the reconstruction loss $\mathcal{L}_{\text{rec}}$. The image-level adversarial loss $\mathcal{L}_{\text{adv}}^{\text{img}}$ and the feature-level adversarial loss $\mathcal{L}_{\text{adv}}^{\text{feat}}$ are optimized to align the image and feature distributions *within the same domain*. The proposed cross-domain consistency loss, in contrast, aligns the task predictions in two *different domains*.

3.5. Implementation details

We implement our model using PyTorch. We use the CycleGAN [51] as our image-to-image translation network $G_{S \to T}$ and $G_{T \to S}$. The structure of the image-level discriminators D_S^{img} and D_T^{img} consists of four residual blocks, each of which is composed of a convolutional layer followed by a ReLU activation. For the feature-level discriminators D_S^{feat} and D_T^{feat}, we use the same architecture as Tsai *et al.* [42]. The image-to-image translation network $G_{S \to T}$ and $G_{T \to S}$, and the discriminators D_S^{img}, D_T^{img}, D_S^{feat}, and D_T^{feat} are all randomly initialized. We have a batch size of 1, a learning rate of 10^{-3} with momentum 0.9, and set the weight decay as 5×10^{-4}. Our hyper-parameters setting: $\lambda_{\text{consis}} = 10$, $\lambda_{\text{rec}} = 10$, $\lambda_{\text{img}} = 0.1$, and $\lambda_{\text{feat}} = 0.001$. We train our model on a single NVIDIA GeForce GTX 1080 GPU with 12 GB memory.

4. Experimental Results

4.1. Semantic segmentation

We present experimental results for semantic segmentation in two different settings: 1) *synthetic-to-real*: adapting from synthetic GTA5 [32] and SYNTHIA [33] datasets to real-world images from Cityscapes dataset [5] and 2) *real-to-real*: adapting the Cityscapes dataset to different cities [4].

4.1.1 GTA5 to Cityscapes

Dataset. The GTA5 dataset [32] consists of $24,966$ synthetic images with pixel-level annotations of 19 categories (compatible with the Cityscapes dataset [5]). Following Hoffman *et al.* [14], we use the GTA5 dataset and adapt the model to the Cityscapes training set with $2,975$ images.

Evaluation protocols. We evaluate our model on the Cityscapes validation set with 500 images using the mean intersection-over-union (IoU) and the pixel accuracy as the evaluation metrics.

Task network. We evaluate our proposed method using two task networks: 1) dilated residual network-26 (DRN-26) [46] and 2) FCN8s-VGG16 [25]. For the DRN-26, we initialize our task network from Hoffman *et al.* [14]. For the FCN8s-VGG16, we initialize our task network from Sankaranarayanan *et al.* [35].

Results. We compare our approach with the state-of-the-art methods [41, 51, 24, 15, 14, 7, 17, 35, 47]. The top block of Table 1 presents the experimental results. Results on both feature backbones show that our method performs favorably against the state-of-the-art methods, outperforming the previous best competitors by 4.9% in mean IoU [17] when using the DRN-26 [46] and 1.0% in mean IoU [35] when using FCN8s-VGG16 [25]. We show that the proposed cross-domain consistency loss $\mathcal{L}_{\text{consis}}$ is critical for the improved performance (e.g., adding $\mathcal{L}_{\text{consis}}$ improves the mean IoU

Table 1: **Experimental results of synthetic-to-real adaptation for semantic segmentation.** We denote the top results as **bold** and underlined.

Method	Backbone	Road	Sidewalk	Building	Wall	Fence	Pole	Traffic Light	Traffic Sign	Vegetation	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	Motorbike	Bicycle	mean IoU	Pixel acc.
GTA5 → Cityscapes																						
Synth. [7]		68.9	19.9	52.8	6.5	13.6	9.3	11.7	8.0	75.0	11.0	56.5	36.9	0.1	51.3	8.5	4.7	0.0	0.1	0.0	22.9	71.9
DR [41]		67.5	23.5	65.7	6.7	12.0	11.6	16.1	13.7	70.3	8.3	71.3	39.6	1.6	55.0	15.1	3.0	0.6	0.2	3.3	25.5	73.8
CycleGAN [51]		89.3	45.1	81.6	27.5	18.6	29.0	35.7	17.3	79.3	29.4	71.5	59.7	15.7	85.3	18.2	14.8	1.4	21.9	12.5	39.6	86.6
UNIT [24]	DRN-26 [46]	90.5	38.5	81.1	23.5	16.3	30.2	25.2	18.5	79.5	26.8	77.8	59.2	17.4	84.4	22.2	16.1	1.6	16.7	16.9	39.1	87.1
FCNs ITW [15]		70.4	32.4	62.1	14.9	5.4	10.9	14.2	2.7	79.2	21.3	64.6	44.1	4.2	70.4	8.0	7.3	0.0	3.5	0.0	27.1	-
CyCADA [14]		79.1	33.1	77.9	23.4	17.3	32.1	33.3	31.8	81.5	26.7	69.0	62.8	14.7	74.5	20.9	25.6	6.9	18.8	20.4	39.5	82.3
DS [7]		89.0	43.5	81.5	22.1	8.5	27.5	30.7	18.9	84.8	28.3	84.1	55.7	5.4	83.2	20.3	28.3	0.1	8.7	6.2	38.3	87.2
GAM [17]		-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	40.2	81.1
Ours w/o $\mathcal{L}_{consis}$		89.1	44.9	80.9	27.5	18.8	30.2	35.6	17.1	79.5	27.2	71.6	59.7	16.1	84.6	18.1	14.6	1.4	22.1	10.9	39.4	85.8
Ours		95.1	49.2	86.4	35.2	22.1	36.1	40.9	29.1	85.0	33.1	75.8	67.3	26.8	88.9	23.4	19.3	4.3	25.3	13.5	45.1	89.2
Synth. [47]		18.1	6.8	64.1	7.3	8.7	21.0	14.9	16.8	45.9	2.4	64.4	41.6	17.5	55.3	8.4	5.0	6.9	4.3	13.8	22.3	-
Curr. DA [47]	FCN8s [25]	74.9	22.0	71.7	6.0	11.9	8.4	16.3	11.1	75.7	13.3	66.5	38.0	9.3	55.2	18.8	18.9	0.0	16.8	16.6	28.9	-
LSD [35]		88.0	30.5	78.6	25.2	23.5	16.7	23.5	11.6	78.7	27.2	71.9	51.3	19.5	80.4	19.8	18.3	0.9	20.8	18.4	37.1	-
Ours		89.1	33.2	80.1	26.9	25.0	18.3	23.4	12.8	77.0	29.1	72.4	55.1	20.2	79.9	22.3	19.5	1.0	20.1	18.7	38.1	86.3
SYNTHIA → Cityscapes																						
Synth. [7]		28.5	10.8	49.6	0.2	0.0	18.5	0.7	5.6	65.3	-	71.6	36.6	6.4	43.8	-	2.7	-	0.8	10.0	18.5	54.6
DR [41]		31.3	16.7	59.5	2.2	0.0	19.7	0.4	6.2	64.7	-	67.3	43.1	3.9	35.1	-	8.3	-	0.3	5.5	19.2	57.9
CycleGAN [51]		58.8	20.4	71.6	1.6	0.7	27.9	2.7	8.5	73.5	-	73.1	45.3	16.2	67.2	-	14.9	-	7.9	24.7	27.1	71.4
UNIT [24]	DRN-26 [46]	56.3	20.6	73.2	1.8	0.3	29.0	4.0	11.8	72.2	-	74.5	50.7	18.4	67.3	-	15.1	-	6.7	29.5	28.0	70.8
FCNs ITW [15]		11.5	19.6	30.8	4.4	0.0	20.3	0.1	11.7	42.3	-	68.7	51.2	3.8	54.0	-	3.2	-	0.2	0.6	17.0	-
DS [7]		67.0	28.0	75.3	4.0	0.2	29.9	3.8	15.7	78.6	-	78.0	54.0	15.4	69.7	-	12.0	-	9.9	19.2	29.5	76.5
Ours w/o $\mathcal{L}_{consis}$		58.3	17.2	64.3	2.0	0.7	24.3	2.6	5.9	72.2	-	70.8	41.9	10.3	64.2	-	12.5	-	8.0	21.3	29.8	75.3
Ours		62.2	21.2	72.8	4.2	0.8	30.1	4.1	10.7	76.3	-	73.6	45.6	14.9	69.2	-	14.1	-	12.2	23.0	33.4	79.5
Synth. [47]		5.6	11.2	59.6	8.0	0.5	21.5	0.8	5.3	72.4	-	75.6	35.1	9.0	23.6	-	4.5	-	0.5	18.0	22.0	-
Curr. DA [47]	FCN8s [25]	65.2	26.1	74.9	0.1	0.5	10.7	3.5	3.0	76.1	-	70.6	47.1	8.2	43.2	-	20.7	-	0.7	13.1	29.0	-
LSD [35]		80.1	29.1	77.5	2.8	0.4	26.8	11.1	18.0	78.1	-	76.7	48.2	15.2	70.5	-	17.4	-	8.7	16.7	36.1	-
Ours		84.9	32.8	80.1	4.3	0.4	29.4	14.2	21.0	79.2	-	78.3	50.2	15.9	69.8	-	23.4	-	11.0	15.6	38.2	84.7

by 5.7% and the pixel accuracy by 3.4% when adopting the DRN-26 [46] as the task network). Figure 4 presents an example that demonstrates the effectiveness of the proposed cross-domain consistency loss $\mathcal{L}_{consis}$. We discover that by applying the cross-domain consistency loss $\mathcal{L}_{consis}$, our model produces more consistent and accurate results before and after image translation.

4.1.2 SYNTHIA to Cityscapes

Dataset. We use the SYNTHIA-RAND-CITYSCAPES [33] set as the source domain which contains 9, 400 images compatible with the Cityscapes annotated classes. Following Dundar *et al.* [7], we evaluate images on the Cityscapes validation set with 16 classes.

Results. We compare our approach with the state-of-the-art methods [41, 51, 24, 15, 7]. The bottom block of Table 1 presents the experimental results. In either DRN-26 [46] or FCN8s [25] backbone, our method achieves state-of-the-art performance. Likewise, we show sizable improvement using the proposed cross-domain consistency loss $\mathcal{L}_{consis}$.

4.1.3 Cityscapes to Cross-City

Dataset. In addition to the *synthetic-to-real* adaptation, we conduct an experiment on the Cross-City dataset [4]

which is a *real-to-real* adaptation. The dataset contains four different cities: Rio, Rome, Tokyo, and Taipei, where each city has 3, 200 images without annotations and 100 images with pixel-level ground truths for 13 classes. Following Tsai *et al.* [42], we use the Cityscapes [5] training set as our source domain and adapt the model to each target city using 3, 200 images, and use the 100 annotated images for evaluation.

Results. We compare our approach with the Cross-City [4], the CBST [53], and the AdaptSegNet [42]. Table 2 shows that our method achieve state-of-the-art performance on two out of four cities. Note that the results in Adapt-SegNet [42] are obtained by using a ResNet-101 [13]. We run their publicly available code with the default settings and report the results using the ResNet-50 [13] as the feature backbone for a fair comparison. Under the same experimental setting, our approach compares favorably against state-of-the-art methods. Furthermore, we show that enforcing cross-domain consistency constraints, our method effectively and consistently improves the results evaluated on all four cities.

4.2. Single-view depth estimation

To show that our formulation is not limited to semantic segmentation, we present experimental results for

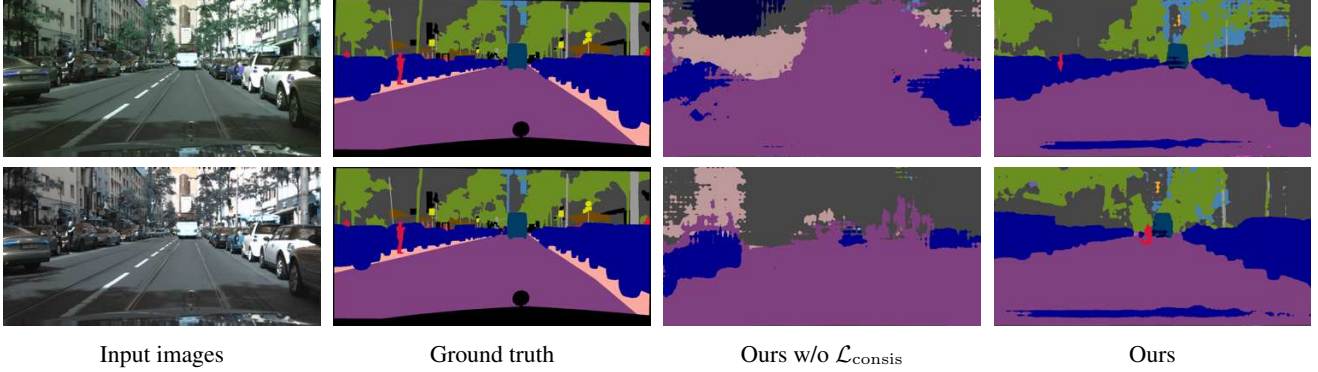| Input images | Ground truth | Ours w/o $\mathcal{L}_{consis}$ | Ours |

Figure 4: **Visual results of semantic segmentation.** We translate an image from Cityscapes to GTA5. For each input image, we present the segmentation results with and without applying the cross-domain consistency loss.

Table 2: **Experimental results of real-to-real adaptation for semantic segmentation.** Adaptation: `Cityscapes` → `Cross-City`.

City	Method	Feature backbone	Road	Sidewalk	Building	Light	Sign	Vegetation	Sky	Person	Rider	Car	Bus	Motorbike	Bicycle	mean IoU
Rome	Cross-City [4]	-	79.5	29.3	84.5	0.0	22.2	80.6	82.8	29.5	13.0	71.7	37.5	25.9	1.0	42.9
	CBST [53]	ResNet-38 [13]	87.1	43.9	89.7	14.8	47.7	85.4	90.3	45.4	26.6	85.4	20.5	49.8	10.3	53.6
	AdaptSegNet [42]	ResNet-101 [13]	83.9	34.2	88.3	18.8	40.2	86.2	93.1	47.8	21.7	80.9	47.8	48.3	8.6	53.8
	AdaptSegNet [42]	ResNet-50 [13]	85.4	34.6	88.1	18.9	39.1	82.3	89.1	43.2	22.4	79.9	44.6	46.0	5.3	52.2
	Ours w/o $\mathcal{L}_{consis}$	ResNet-50 [13]	84.4	31.2	87.7	18.6	38.0	80.7	85.4	43.5	19.8	79.4	45.3	44.2	5.1	51.0
	Ours	ResNet-50 [13]	90.2	37.2	91.2	22.0	41.1	86.3	91.7	47.1	25.1	83.0	48.0	47.5	6.2	55.1
Rio	Cross-City [4]	-	74.2	43.9	79.0	2.4	7.5	77.8	69.5	39.3	10.3	67.9	41.2	27.9	10.9	42.5
	CBST [53]	ResNet-38 [13]	84.3	55.2	85.4	19.6	30.1	80.5	77.9	55.2	28.6	79.7	33.2	37.6	11.5	52.2
	AdaptSegNet [42]	ResNet-101 [13]	76.2	44.7	84.6	9.3	25.5	81.8	87.3	55.3	32.7	74.3	28.9	43.0	27.6	51.6
	AdaptSegNet [42]	ResNet-50 [13]	75.8	43.9	80.7	7.7	21.1	80.8	88.0	51.2	27.4	71.1	25.6	43.7	26.9	49.5
	Ours w/o $\mathcal{L}_{consis}$	ResNet-50 [13]	74.7	44.1	81.2	5.3	19.2	80.7	86.3	52.3	27.7	69.2	24.1	45.4	25.2	48.9
	Ours	ResNet-50 [13]	77.5	43.3	81.2	10.1	23.2	79.7	88.2	57.4	31.9	72.2	29.1	38.9	22.4	50.4
Tokyo	Cross-City [4]	-	83.4	35.4	72.8	12.3	12.7	77.4	64.3	42.7	21.5	64.1	20.8	8.9	40.3	42.8
	CBST [53]	ResNet-38 [13]	85.2	33.6	80.4	8.3	31.1	83.9	78.2	53.2	28.9	72.7	4.4	27.0	47.0	48.8
	AdaptSegNet [42]	ResNet-101 [13]	81.5	26.0	77.8	17.8	26.8	82.7	90.9	55.8	38.0	72.1	4.2	24.5	50.8	49.9
	AdaptSegNet [42]	ResNet-50 [13]	76.0	25.3	78.1	15.4	22.3	81.3	91.1	45.2	34.6	69.3	2.3	20.7	48.2	46.9
	Ours w/o $\mathcal{L}_{consis}$	ResNet-50 [13]	72.3	24.9	77.6	14.3	23.1	80.9	90.7	43.6	35.2	68.9	3.1	19.8	42.4	45.9
	Ours	ResNet-50 [13]	82.1	29.3	78.2	18.2	27.5	83.1	91.2	56.4	37.8	74.3	9.5	26.0	52.1	51.2
Taipei	Cross-City [4]	-	78.6	28.6	80.0	13.1	7.6	68.2	82.1	16.8	9.4	60.4	34.0	26.5	9.9	39.6
	CBST [53]	ResNet-38 [13]	86.1	35.2	84.2	15.0	22.2	75.6	74.9	22.7	33.1	78.0	37.6	58.0	30.9	50.3
	AdaptSegNet [42]	ResNet-101 [13]	81.7	29.5	85.2	26.4	15.6	76.7	91.7	31.0	12.5	71.5	41.1	47.3	27.7	49.1
	AdaptSegNet [42]	ResNet-50 [13]	81.8	27.8	83.2	24.4	12.6	74.1	88.7	30.9	11.1	70.8	40.2	45.3	26.2	47.5
	Ours w/o $\mathcal{L}_{consis}$	ResNet-50 [13]	79.6	26.9	84.1	23.7	14.1	72.8	86.5	30.3	9.9	69.9	40.6	44.7	25.8	46.8
	Ours	ResNet-50 [13]	79.7	28.1	85.1	24.4	16.4	74.3	87.9	29.5	12.8	69.8	40.0	46.8	28.1	47.9

single-view depth prediction task. Specifically, we use SUNCG [38] as the source domain and adapt the model to the NYUDv2 [37] dataset.

Dataset. To generate the paired synthetic training data, we rendered RGB images and depth map from the SUNCG dataset [38], which contains $45,622$ 3D houses with various room types. Following Zheng *et al.* [48], we choose the camera locations, poses and parameters based on the distribution of real NYUDv2 dataset [37] and retain valid depth maps using the criteria described by Song *et al.* [38]. In to-

tal, we generate $130,190$ valid views from $4,562$ different houses.

Evaluation protocols. We use the root mean square error (RMSE) and the log scale version (RMSE log.), the squared relative difference (Sq. Rel.) and the absolute relative difference (Abs. Rel.), and the accuracy measured by thresholding ($\delta <$ threshold).

Task network. We initialize our task network from the unsupervised version of Zheng *et al.* [48].

Table 3: **Synthetic-to-real (`SUNCG → NYUv2`) adaptation for depth prediction.** The column "Supervision" indicates methods trained with NYUv2 training data. We denote the top two results as **bold** and <u>underlined</u>.

Method	Supervision	Abs. Rel. ↓	Sq. Rel. ↓	RMSE ↓	RMSE log. ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
Liu *et al.* [23]	✓	0.213	-	0.759	-	0.650	0.906	0.976
Eigen *et al.* [9] Fine	✓	0.215	0.212	0.907	0.285	0.611	0.887	0.971
Eigen *et al.* [8] (VGG)	✓	0.158	0.121	0.641	0.214	0.769	0.950	0.988
T^{2}Net [48]	✓	0.157	0.125	0.556	0.199	0.779	0.943	0.983
Synth.		0.304	0.394	1.024	0.369	0.458	0.771	0.916
Baseline (train set mean)		0.439	0.641	1.148	0.415	0.412	0.692	0.586
T^{2}Net [48]		0.257	<u>0.281</u>	0.915	<u>0.305</u>	0.540	0.832	<u>0.948</u>
Ours w/o $\mathcal{L}_{\text{consis}}$		<u>0.254</u>	0.283	<u>0.911</u>	0.306	<u>0.541</u>	<u>0.835</u>	0.947
Ours		**0.233**	**0.272**	**0.898**	**0.289**	**0.562**	**0.853**	**0.952**

Results. Table 3 shows the comparisons with prior methods [23, 9, 8, 48]. Here, the column "Supervision" indicates that the method is learned in a supervised fashion. While not directly comparable, we report their results for completeness. Under the same experimental settings, we observe that our method achieves state-of-the-art performance on all adopted evaluation metrics. Moreover, with the integration of the cross-domain consistency loss $\mathcal{L}_{\text{consis}}$, our method shows consistently improved performance.

4.3. Optical flow estimation

We show evaluations of the model trained on a synthetic dataset (i.e., MPI Sintel [2]) and test the adapted model on real-world images from the KITTI 2012 [12] and KITTI 2015 [29] datasets.

Dataset. The MPI Sintel dataset [2] consists of $1,401$ images rendered from artificial scenes. There are two versions: 1) the final version consists of images with motion blur and atmospheric effects, and 2) the clean version does not include these effects. We use the clean version as the source dataset. We report two results obtained by 1) using the KITTI 2012 [12] as the target dataset and 2) using the KITTI 2015 [29] as the target dataset.

Evaluation protocols. We adopt the average endpoint error (AEPE) and the F1 score for both KITTI 2012 and KITTI 2015 to evaluate the performance.

Task network. Our task network is initialized from the PWC-Net [40] (without finetuning on the KITTI dataset).

Results. We compare our approach with the state-of-the-art methods [40, 31, 19]. Table 4 shows that our method achieves improved performance on both datasets. When incorporating the proposed cross-domain consistency loss $\mathcal{L}_{\text{consis}}$, our model improves the results by 1.76 in terms of average endpoint error on the KITTI 2012 test set and 10.6% in terms of F1-all on the KITTI 2015 test set.

Table 4: **Experimental results of synthetic-to-real adaptation for optical flow estimation.** Left: `MPI Sintel →` `KITTI 2012`. Right: `MPI Sintel → KITTI 2015`. The column "finetune" indicates that method is finetuned on the KITTI dataset. The bold and the underlined numbers indicate top two results, respectively.

Method	finetune	KITTI 2012			KITTI 2015		
		AEPE *train*	AEPE *test*	F1-Noc *test*	AEPE *train*	F1-all *train*	F1-all *test*
SpyNet [31]	✓	4.13	4.7	12.31%	-	-	35.05%
FlowNet2 [19]	✓	1.28	1.8	4.82%	2.30	8.61%	10.41%
PWC-Net [40]	✓	1.45	1.7	4.22%	2.16	9.80%	9.60%
FlowNet2 [19]		4.09	-	-	<u>10.06</u>	<u>30.37%</u>	-
PWC-Net [40]		<u>4.14</u>	<u>4.22</u>	**8.10%**	10.35	33.67%	-
Ours w/o $\mathcal{L}_{\text{consis}}$		4.16	4.92	13.52%	10.76	34.01%	<u>36.43%</u>
Ours		**2.19**	**3.16**	<u>8.57%</u>	**8.02**	**23.14%**	**25.83%**

4.4. Limitations

Our method is memory-intensive as the training involves multiple networks at the same time. Potential approaches to alleviate this issue include 1) adopting partial sharing on the two task networks, e.g., share the last few layers of the two task networks, and 2) sharing the encoders in the image translation network (i.e., $G_{S \to T}$ and $G_{T \to S}$).

5. Conclusions

We have presented a simple yet surprisingly effective loss for improving pixel-level unsupervised domain adaption for dense prediction tasks. We show that by incorporating the proposed cross-domain consistency loss, our method consistently improves the performances over a wide range of tasks. Through extensive experiments, we demonstrate that our method is applicable to a wide variety of tasks.

References

[1] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *CVPR*, 2017.

[2] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

[3] Yun-Chun Chen, Po-Hsiang Huang, Li-Yu Yu, Jia-Bin Huang, Ming-Hsuan Yang, and Yen-Yu Lin. Deep semantic matching with foreground detection and cycle-consistency. In *ACCV*, 2018.

[4] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. In *ICCV*, 2017.

[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 *CVPR*, 2016.

[6] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.

[7] Aysegul Dundar, Ming-Yu Liu, Ting-Chun Wang, John Zedlewski, and Jan Kautz. Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation. *arXiv*, 2018.

[8] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[9] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[10] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, 2015.

[11] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *JMLR*, 2016.

[12] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012.

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

[14] 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 *ICML*, 2018.

[15] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv*, 2016.

[16] Yuan-Ting Hu, Hong-Shuo Chen, Kexin Hui, Jia-Bin Huang, and Alexander Schwing. Sail-vos: Semantic amodal instance level video object segmentation - a synthetic dataset and baselines. In *CVPR*, 2019.

[17] Haoshuo Huang, Qixing Huang, and Philipp Krähenbühl. Domain transfer through deep activation matching. In *ECCV*, 2018.

[18] Po-Han Huang, Kevin Matzen, Johannes Kopf, Narendra Ahuja, and Jia-Bin Huang. Deepmvs: Learning multi-view stereopsis. In *CVPR*, 2018.

[19] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *CVPR*, 2017.

[20] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *ECCV*, 2018.

[21] Wei-Sheng Lai, Jia-Bin Huang, and Ming-Hsuan Yang. Semi-supervised learning for optical flow with generative adversarial networks. In *NIPS*, 2017.

[22] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *ECCV*, 2018.

[23] Fayao Liu, Chunhua Shen, Guosheng Lin, and Ian D Reid. Learning depth from single monocular images using deep convolutional neural fields. *TPAMI*, 2016.

[24] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *NIPS*, 2017.

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[26] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *ICML*, 2015.

[27] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. Unsupervised domain adaptation with residual transfer networks. In *NIPS*, 2016.

[28] Simon Meister, Junhwa Hur, and Stefan Roth. Unflow: Unsupervised learning of optical flow with a bidirectional census loss. In *AAAI*, 2018.

[29] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015.

[30] Xingchao Peng, Ben Usman, Kuniaki Saito, Neela Kaushik, Judy Hoffman, and Kate Saenko. Syn2real: A new benchmark for synthetic-to-real visual domain adaptation. In *ECCV*, 2018.

[31] Anurag Ranjan and Michael J Black. Optical flow estimation using a spatial pyramid network. In *CVPR*, 2017.

[32] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

[33] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

[34] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *ECCV*, 2018.

[35] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *CVPR*, 2018.

[36] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017.

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

[38] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017.

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

[40] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*, 2018.

[41] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 2017.

[42] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2017.

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

[44] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

[45] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019.

[46] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *CVPR*, 2017.

[47] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017.

[48] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In *ECCV*, 2018.

[49] Tinghui Zhou, Yong Jae Lee, Stella X Yu, and Alyosha A Efros. Flowweb: Joint image set alignment by weaving consistent, pixel-wise correspondences. In *CVPR*, 2015.

[50] Xiaowei Zhou, Menglong Zhu, and Kostas Daniilidis. Multi-image matching via fast alternating minimization. In *ICCV*, 2015.

[51] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[52] Yuliang Zou, Zelun Luo, and Jia-Bin Huang. Df-net: Unsupervised joint learning of depth and flow using cross-task consistency. In *ECCV*, 2018.

[53] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.