

# PixMatch: Unsupervised Domain Adaptation via Pixelwise Consistency Training

Luke Melas-Kyriazi  
 Harvard University  
 Oxford University  
 Cambridge, MA

[lukemk@robots.ox.ac.uk](mailto:lukemk@robots.ox.ac.uk)

Arjun K. Manrai  
 Harvard University  
 Boston Children’s Hospital  
 Boston, MA

[Arjun.Manrai@hms.harvard.edu](mailto:Arjun.Manrai@hms.harvard.edu)

## Abstract

*Unsupervised domain adaptation is a promising technique for semantic segmentation and other computer vision tasks for which large-scale data annotation is costly and time-consuming. In semantic segmentation, it is attractive to train models on annotated images from a simulated (source) domain and deploy them on real (target) domains. In this work, we present a novel framework for unsupervised domain adaptation based on the notion of target-domain consistency training. Intuitively, our work is based on the idea that in order to perform well on the target domain, a model’s output should be consistent with respect to small perturbations of inputs in the target domain. Specifically, we introduce a new loss term to enforce pixelwise consistency between the model’s predictions on a target image and a perturbed version of the same image. In comparison to popular adversarial adaptation methods, our approach is simpler, easier to implement, and more memory-efficient during training. Experiments and extensive ablation studies demonstrate that our simple approach achieves remarkably strong results on two challenging synthetic-to-real benchmarks, GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes.*

## 1. Introduction

Deep neural network approaches for semantic image segmentation have shown widespread success in the past decade, but they remain reliant on large datasets with pixel-level annotations. Data labeling for semantic segmentation is notoriously laborious and expensive, especially in domains where experts are required (e.g. medical image segmentation). Even for annotations that can be performed by non-experts like parsing an urban scene into familiar objects, as in the Cityscapes dataset [15], it takes an estimated 90 minutes to annotate a single image [43].

The need to build generalizable models with limited data has motivated work on unsupervised domain adaptation

(UDA) approaches for semantic segmentation [63, 49, 52, 7, 23, 11], where annotated images from a simulated (source) domain, which are plentiful, are used in conjunction with unlabeled images from a real (target) domain. The simulated source domain in this “synthetic to real” translation task can be creative, such as the video game Grand Theft Auto V in the GTA5-to-Cityscapes benchmark [40] and the simulation platform SYNTHIA as in the SYNTHIA-to-Cityscapes benchmark [41].

The literature on UDA for semantic segmentation is dominated by adversarial methods, which aim to learn domain-invariant representations across multiple domains by introducing adversarial losses [23]. These methods have shown strong performance, but due to the instability of their adversarial losses, they are well-known to be highly sensitive to hyperparameters and difficult to train [54, 8, 19].

Recently, a new line of work on UDA for semantic segmentation has emerged around self-training [63, 52, 7]. These methods add loss terms to the training objective that encourage the segmentation model to make more confident predictions on the target domain (for example, by encouraging low-entropy predictions) [52, 7, 63].

This paper begins with the observation that we do not simply desire a model that makes confident predictions in target domain, rather we desire a model that makes *consistent* predictions in the target domain. That is, we desire a model for which small perturbations of inputs in the target domain lead to small, consistent changes in the output segmentation. If a model’s predictions are always confident, but they are not stable with respect to small perturbations of target images, the model is likely to be poorly-adapted to the target domain. Conversely, if a model behaves smoothly with respect to perturbations of images in the target domain, the model is likely to be better-adapted to that domain.

We propose a consistency training-based framework to directly enforce this notion of smoothness in the target domain. Our method, denoted PixMatch, adds a loss term that encourages the segmentation model’s predictions on a target domain image and a perturbed version of the same image to

be pixelwise consistent.

We experiment with four different perturbation functions, two of which are inspired by work in semi/self-supervised learning (Data Augmentations; CutMix) and two of which are inspired by work in domain adaptation (Style Transfer; Fourier Transform).

Surprisingly, we find that our baseline model, which uses heavy data augmentation as its perturbation function, performs best. This simple baseline delivers extremely strong results on GTA5-to-Cityscapes [15] and SYNTHIA-to-Cityscapes [41]. Using only a source (supervised) loss and a target consistency loss, it outperforms complex prior methods that used combinations of source, adversarial, and self-training losses.

Compared to existing adversarial approaches, PixMatch is easier to implement, more stable during training, and less memory-intensive to train. It introduces only one hyperparameter, which controls the relative weighting of the source (supervised) loss and the target consistency loss. Moreover, the simplicity of PixMatch means that it may be easily integrated into existing UDA methods and pipelines; combining PixMatch with self-training yields the state-of-the-art results.

The main contributions of this paper are:

- We introduce a novel consistency-based framework for unsupervised domain adaptation that encourages pixelwise consistency between a model’s predictions on a target image and a perturbed version of the same image.
- We investigate multiple perturbation functions, finding that a simple baseline using data augmentation performs extremely well.
- We perform extensive ablation studies on our baseline method in order to better understand its strong performance, showing that the setting of domain adaptation differs markedly from that of semi-supervised image classification.
- We find that our model may be easily combined with self-training for further performance improvements. Doing so, we achieve a new state-of-the-art on the challenging GTA5-to-Cityscapes benchmark, using a much simpler training approach than recent adversarial methods.

## 2. Background

The task of semantic image segmentation is to assign to every pixel in an image one of  $C$  semantic class labels. In the setting of unsupervised domain adaptation (UDA) for semantic segmentation, we consider a labeled source domain  $S$  and an unlabeled target domain  $T$  with the same set of semantic classes. Often, the source domain  $S$  is composed of

simulated data (e.g. GTA V) and the target  $T$  is composed of real data (e.g. Cityscapes), thus forming a “synthetic-to-real” adaptation task.

### 2.1. Related work

#### 2.1.1 Unsupervised Domain Adaptation for Semantic Segmentation

Recent work on UDA for semantic segmentation has been dominated by deep neural network methods.

Most of the literature is concerned with adversarial training [23, 49, 52, 11]. Adversarial training aims to minimize the discrepancy between source and target feature distributions by introducing a discriminator network alongside the main segmentation network. The discriminator is trained to predict an input image’s domain from the segmentation network’s intermediate feature maps, while the segmentation network is trained to fool the discriminator (and produce good segmentations on the source domain). Adversarial training generally produces strong performance but suffers from instability during training, is computationally expensive, and is highly sensitive to changes in hyperparameters.

During the past two years, a new line of work has emerged around self-training. Zou et al. [63] were the first to apply pseudolabeling to UDA; they alternatively generate pseudolabels on target data and re-train the model with these labels. They also use different pseudolabel thresholds for each class (“class-balanced self-training”) to prevent the loss from being dominated by easy classes.

Vu et al. [52] propose two entropy minimization approaches, one that operates directly (MinEnt) and one that operates adversarially (AdvEnt). Chen et al. [7] builds off AdvEnt [52] by noting that the entropy function  $H(\cdot)$  biases the loss toward well-classified pixels in an image rather than more challenging pixels. To address this, Chen et al. [7] substitute a linear function in place of the entropy function  $H$ . These methods demonstrate competitive performance with adversarial approaches on GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes.

Some recent work in UDA for semantic segmentation has considered the broad notion of consistency, but they do so in a different manner from this work. Most notably, a significant body of research leverages cycle-consistency to translate images in the source domain to the target domain for domain adaptation [23, 12, 18, 32, 61, 38]. [23] employs cycle-consistent GANs for domain adaptation at both the pixel-level and feature-level. [38] trains a GAN-based image-to-image translation network and a semantic segmentation network with the same backbone, and [32] extends the idea of cross-domain cycle-consistency to the multi-source setting. [33] trains an image-to-image translation model and a semantic segmentation model in alternating stages, while the very recent work [28] considers a

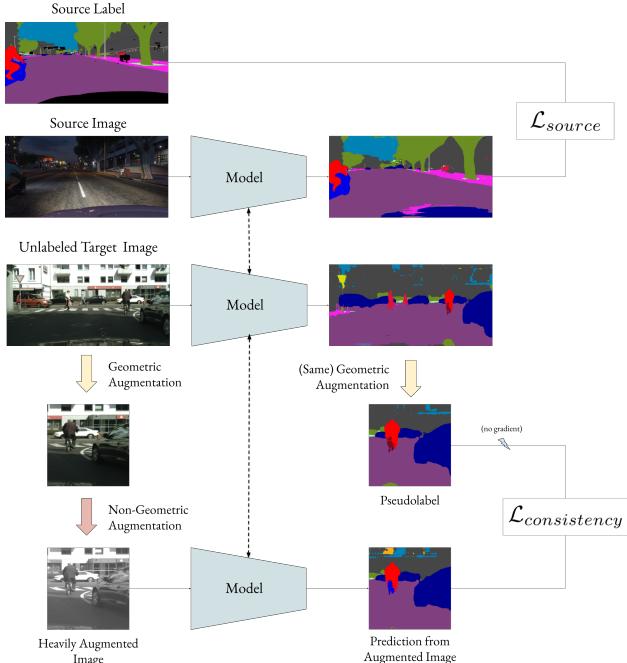


Figure 1. Our proposed pixelwise consistency training approach.

contrastive cycle-consistency loss on the level of pixels. Finally, [56] performs image-to-image translation for UDA in frequency space rather than pixel space using a Fourier transform.

Beyond cycle-consistency, [12] enforces cross-domain consistent predictions in the context of image-to-image translation for UDA for using two image-level adversarial losses and two feature-level adversarial losses. [18] enforces geometric consistency in the context of image-to-image translation, [29] aims to learn texture-invariant representations, and [26] aims to learn spatially-consistent features by segmenting images on the level of patches.

Our formulation of consistency training differs from all of these above approaches in that it directly applies a general consistency loss to the semantic segmentation network in the target domain, ensuring that the output is consistent with respect to perturbations. Our framework is flexible, supporting perturbation functions inspired by work from both the semi/self-supervised learning community (below) and the domain adaptation community. Whereas most of the previous approaches involve adversarial losses, auxiliary networks, or complex multi-stage training pipelines, our framework involves only a single segmentation network with a single non-adversarial consistency loss (in addition to the standard source cross-entropy loss). Additionally, our framework is flexible and may be used alongside any of these prior approaches.

---

### Algorithm 1: Target Consistency Loss

---

**Input:** Input image  $x \in \mathbb{R}^{H \times W \times 3}$   
**Result:** Target consistency loss  
 $y_{pseudo} = \text{model}(x) \# \text{without gradient}$   
 $x_{pert}, y_{pseudo\_pert} = \text{perturbation}_S(x, y_{pseudo})$   
 $y_{hat} = \text{model}(x_{pert})$   
 $\text{loss} = \text{cross\_entropy}(y_{pseudo\_pert}, y_{hat})$   
**return** loss

---

### 2.1.2 Consistency Training

Consistency training was first proposed for semi-supervised learning by Blum and Mitchell [4] and enjoyed widespread use throughout the 2000s under the name “co-training”. This co-training framework was slightly different than the approach we have today; Blum and Mitchell [4] trained multiple models on different views of the same set of input examples and used these models to produce pseudolabels for one another. Modern consistency training [42] produces multiple views of the same input using a stochastic data augmentation function, and uses them to train a single model.

Recently, consistency training has formed the base of multiple large advances in semi-supervised learning for image classification. Xie et al. [55] use a model to generate (soft) artificial labels and enforce strongly-augmented examples, showing strong performance on CIFAR-10 with as few as 250 examples. In MixMatch [3], Berthelot et al. mix labeled and unlabeled data while performing entropy minimization on augmented unlabeled data. In their follow-up ReMixMatch [2], they further improve the method’s sample-efficiency by optimizing the data augmentation function during training. Most recently, in FixMatch [46], Sohn et al. replace entropy minimization with pseudolabeling followed by filtering for high-confidence examples.

We denote our method “PixMatch” as a reference to this line of work, but we emphasize that it is designed for the different setting of unsupervised domain adaptation, not semi-supervised image classification. In this paper we are interested in answering the question of what forms of consistency losses are effective *for UDA specifically*.

### 2.1.3 Contrastive Learning and Data Augmentation

Consistency training is closely related to contrastive learning, which has recently led to remarkable progress in unsupervised representation learning [20, 9]. Of particular relevance for this paper is SimCLR [9]; our baseline PixMatch model uses the data augmentations proposed in SimCLR as its perturbation function for consistency training. In related work, [17] uses a student-teacher architecture with augmentations for image classification.

### 3. PixMatch

#### 3.1. Overview

PixMatch employs consistency training and pseudolabeling to enforce consistency on the target domain. Its loss function is composed of two cross-entropy loss terms. The first of these is the standard supervised cross-entropy loss on the source domain:

$$\mathcal{L}_S = -\frac{1}{n_S} \sum_{s \in S} \sum_{i=1}^{H \cdot W} H(y_s^{(i)}, p^{(i)}(y|x_s)) \quad (1)$$

where  $p^{(i)}$  is the output probability distribution at pixel  $i$  for source input  $x_s$ ,  $y$  is the ground truth semantic map, and  $n_S$  is the number of images in the source dataset  $S$ .

The second of these losses is a consistency loss on the target domain. To calculate this loss, we first run a target image  $x_t$  through the model to obtain the pseudolabel  $\hat{y}_t = \arg \max(q_t) = \arg \max(p_t(y|x_t))$ . We then perturb the (image, pseudolabel) pair using a perturbation function to yield the pair  $(x_{pert}, \hat{y}_{pert})$ . Our consistency loss function is then

$$\mathcal{L}_T = -\frac{1}{n_T} \sum_{t \in T} \sum_{i=1}^{H \cdot W} H(\hat{y}_{pert}^{(i)}, p^{(i)}(y|x_{pert})) \quad (2)$$

and the final loss is  $\mathcal{L} = \mathcal{L}_S + \lambda_T \mathcal{L}_T$ , where  $\lambda_T$  is a hyperparameter that controls the relative weighting of the source (supervised) loss and the target consistency loss.

#### 3.2. Perturbation Functions

We experiment with four perturbation functions, two inspired by the self-supervised learning community and two inspired by the domain adaptation community.

**Data Augmentation Consistency** Our baseline PixMatch model uses heavy data augmentation as the perturbation function ([Figure 3](#)). It is important to note that, unlike in image classification, when we augment an input target image using a geometric transform (e.g. cropping), we also have to perform the corresponding augmentation on the pseudolabel produced by the model.

**CutMix Consistency** Our second model uses the recently-proposed CutMix [58] regularization method as a perturbation function. Specifically, we cut-and-paste a rectangular region from a source image onto the target image and enforce that the predictions on this mixed image are consistent with the corresponding mixed labels. Note that this perturbation function takes as input both the source and target image, rather than just the target image. We use CutMix rather than other regularization methods because, in comparison to MixMatch [3] and others, [58] argues that CutMix promotes

the learning of more localizable features. This perturbation function is closely related to the approach taken in [16] for semi-supervised semantic segmentation.

**Style Consistency** Our third perturbation function is inspired by the line of work in domain adaptation that tries to transfer the style of the target images to the source prior to or during training [23, 49, 12, 38]. Here, we transfer the style of the source to the target and enforce consistency between the model’s predictions on the original target image and the source-stylized target image. To perform the target-to-source stylization, we use the zero-shot style transfer model from [45] (AvatarNet).

**Fourier Consistency** Our fourth perturbation function, inspired by but different from [56], modifies the target image in frequency space using a Fourier transform. [56] replaces the low-frequency part of source images with that of target images, creating source images with low-frequency target styles; they train using a cross-entropy loss between these modified source images and the source labels, alongside entropy minimization on the target images. Differently, we replace the low-frequency part of the target images with that of the source, creating target images with low-frequency source styles; we train with a consistency loss between the original images and these Fourier-stylized target images, alongside the standard cross-entropy loss on the source.

#### 3.3. Relationship between PixMatch and Self-Training

As a general framework, PixMatch is closely related to self-training methods. In particular, if the perturbation function is the identity function, then the pseudolabels produced during consistency training are the standard pseudolabels from self-training. It is also possible to perform PixMatch in a soft manner, analogous to the way in which entropy minimization is a soft variant of pseudolabeling. In our ablation studies on our baseline model, we find that a hard approach gives better empirical results.

However, our framework is different from other self-training methods in that it enables domain-adaptation-specific self-training; the perturbation function may incorporate information from the source data. For example, our Fourier-based consistency function uses source-domain Fourier frequencies to perturb the target images. By contrast, other pure self-training approaches [63, 7, 52] use only target-domain information.

## 4. Experiments

**Datasets** We evaluate our method on the two standard large-scale UDA segmentation benchmarks, GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes. For evaluation, we

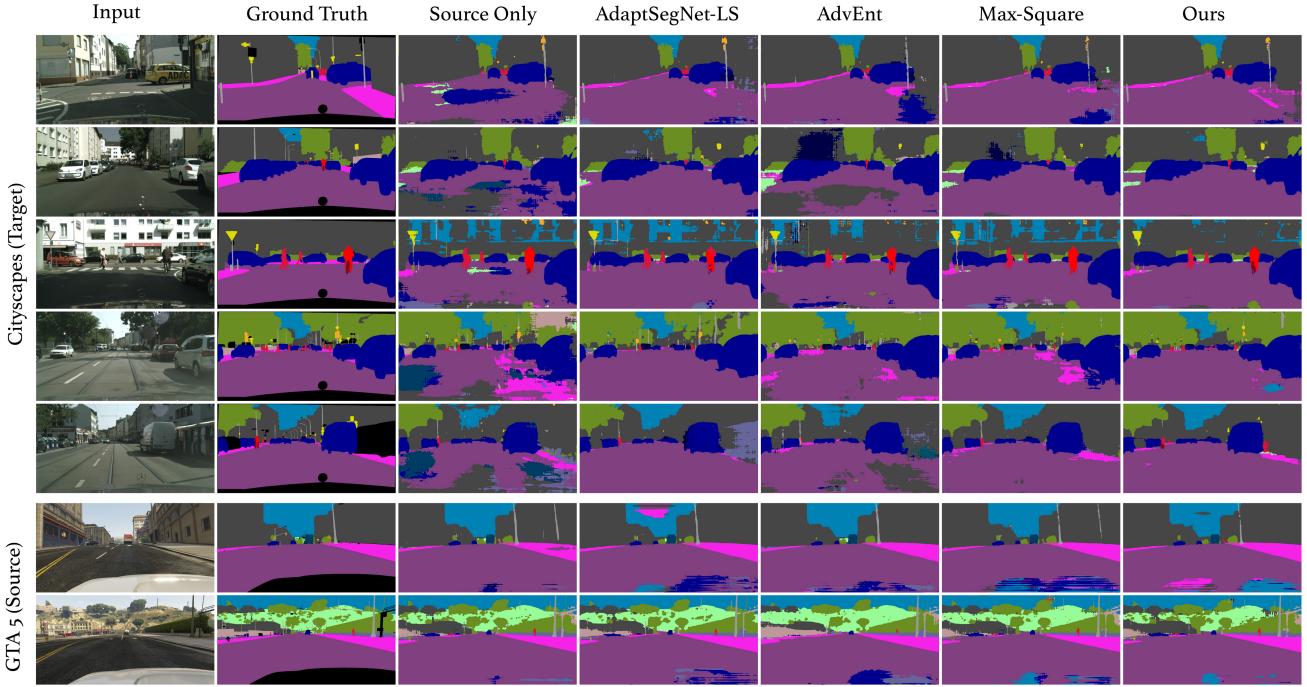


Figure 2. Qualitative examples of our consistency training method and prior methods on GTA5-to-Cityscapes. The column "Ours" refers to our baseline model with augmentation-based perturbations. Note that these images are not hand-picked; they are the first 5 images in the Cityscapes validation set. Visually, our method yields better segmentations than prior techniques. On the third image, for example, it is the only method that properly segments the bicycle (shown in red) and the only method that does not mistake a large proportion of the building for sky.

measure per-class and mean Intersection-over-Union (IoU). As is standard, we evaluate using all 19 classes for GTA5-to-Cityscapes, and using both 16 and 13 classes for SYNTHIA-to-Cityscapes (because SYNTHIA shares only 16 classes with Cityscapes).

**Model** For purposes of comparison with previous methods, we use the DeepLab-v2 segmentation model [6] with a ResNet-101 [21] backbone. We train using SGD with learning rate  $1.0 \cdot 10^{-4}$ , momentum 0.9, weight decay  $5 \cdot 10^{-4}$ , and a polynomial learning rate decay:  $lr = lr * (1 - \frac{iter}{max\_iter})^{0.9}$ . All experiments are conducted on a single NVIDIA RTX 2080TI GPU with 11GB of VRAM. The exact parameters used for each perturbation function are included in the appendix.

**Results** We present our results on the highly competitive GTA5-to-Cityscapes and SYNTHIA-to-Cityscapes benchmarks in Tables 1 and 2. For comparison, we have included 27 other methods from the past two years, including all top results from the SoTa benchmark list on [PapersWithCode](#). Our simple consistency training method performs extremely well relative to other methods.

We also show qualitative results in Figures 2 and 5.

## 5. Analysis

**Overall** On GTA5-to-Cityscapes (Table 1), our baseline augmentation-based model performs best, followed by CutMix-, Fourier-, and finally style-based consistency. When combined with entropy minimization (in the form of the max-square loss), our baseline method attains an mIoU of 50.3, outperforming all other methods with the exception of a single method using a distillation of multiple models that performs 0.1 mIoU better (50.4).

On SYNTHIA-to-Cityscapes (Table 2), CutMix-based training method performed best, followed by Fourier-augmentation-, and again style-based consistency. Apart from style-based consistency, all are within 0.4 mIoU; they perform very competitively with more complex domain adaptation approaches.

**Fourier** It is natural to compare the results of our Fourier-based model to those from FDA [56], which modifies the source images directly. On GTA5-to-Cityscapes, our method outperforms FDA (45.4 vs 44.6 mIoU). On SYNTHIA-to-Cityscapes, [56] only presents results for a distillation of an ensemble of models, not a single model, but nonetheless our single model only slightly underperforms the distilled ensemble (51.1 vs 52.5 mIoU).

Year	Method	Backbone	GTA5-to-Cityscapes																																																					
			road	sdwk	bld	wall	fnc	pole	light	sign	veg.	trn.	sky	pers	rdr	car	trck	bus	tm	mtr	bike	mIoU																																		
			No Adaptation																																																					
Source only																																																								
Adversarial Methods																																																								
[49] 2018	AdaptSegNet	RN-101	86.5	25.9	79.8	22.1	20.0	23.6	33.1	21.8	81.8	25.9	75.9	57.3	26.2	76.3	29.8	32.1	7.2	29.5	32.5	41.4																																		
[49] 2018	AdaptSegNet-LS	RN-101	91.4	48.4	81.2	27.4	21.2	31.2	35.3	16.1	84.1	32.5	78.2	57.7	28.2	85.9	33.8	43.5	0.2	23.9	16.9	44.1																																		
[52] 2019	AdvEnt	RN-101	89.9	36.5	81.6	29.2	25.2	28.5	32.3	22.4	83.9	34.0	77.1	57.4	27.9	83.7	29.4	39.1	1.5	28.4	23.3	43.8																																		
[52] 2019	AdvEnt+MinEnt	RN-101	89.4	33.1	81.0	26.6	26.8	27.2	33.5	24.7	83.9	36.7	78.8	58.7	30.5	84.8	38.5	44.5	1.7	31.6	32.4	45.5																																		
[50] 2019	Patch-Disc	RN-101	92.3	51.9	82.1	29.2	25.1	24.5	33.8	33.0	82.4	32.8	82.2	58.6	27.2	84.3	33.4	46.3	2.2	29.5	32.3	46.5																																		
[60] 2019	CAG	RN-101	90.4	51.6	83.8	34.2	27.8	38.4	25.3	48.4	85.4	38.2	78.1	58.6	34.6	84.7	21.9	42.7	41.1	29.3	37.2	50.2																																		
[53] 2020	FADA	RN-101	92.5	47.5	85.1	37.6	32.8	33.4	33.8	18.4	85.3	37.7	83.5	63.2	39.7	87.5	32.9	47.8	1.6	34.9	39.5	49.2																																		
[53] 2020	FADA-MST	RN-101	91.0	50.6	86.0	43.4	29.8	36.8	43.4	25.0	86.8	38.3	87.4	64.0	38.0	85.2	31.6	46.1	6.5	25.4	37.1	50.1																																		
[25] 2020	CrCDA	RN-101	92.4	55.3	82.3	31.2	29.1	32.5	33.2	35.6	83.5	34.8	84.2	58.9	32.2	84.7	40.6	46.1	2.1	31.1	32.7	48.6																																		
Image-to-Image / Style Transfer Methods																																																								
[33] 2019	BDL (M2-F2)	RN-101	91.0	44.7	84.2	34.6	27.6	30.2	36.0	36.0	85.0	43.6	83.0	58.6	31.6	83.3	35.3	49.7	3.3	28.8	35.6	48.5																																		
[12] 2019	CrDoCo	DRN-26	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																																		
[56] 2020	FDA	RN-101	88.8	35.4	80.5	24.0	24.9	31.3	34.9	32.0	82.6	35.6	74.4	59.4	31.0	81.7	29.3	47.1	1.2	21.1	32.3	44.6																																		
[56] 2020	FDA (Ensemble)	RN-101	92.5	53.3	82.3	26.5	27.6	36.4	40.5	38.8	82.2	39.8	78.0	62.6	34.4	84.9	34.1	53.12	16.8	27.7	46.4	50.4																																		
[29] 2020	LTIR	RN-101	92.9	55.0	85.3	34.2	31.1	34.9	40.7	34.0	85.2	40.1	87.1	61.0	31.1	82.5	32.3	42.9	0.3	36.4	46.1	50.2																																		
Self-Training Methods																																																								
[63] 2018	CBST-SP	WRN-38	88.0	56.2	77.0	27.4	22.4	40.7	47.3	40.9	82.4	21.6	60.3	50.2	20.4	83.8	35.0	51.0	15.2	20.6	37.0	46.2																																		
[52] 2019	MinEnt	RN-101	86.2	18.6	80.3	27.2	24.0	23.4	33.5	24.7	83.3	31.0	75.6	54.6	25.6	85.2	30.0	10.9	0.1	21.9	37.1	42.3																																		
[7] 2019	MaxSquare (MS)	RN-101	88.1	27.7	80.8	28.7	19.8	24.9	34.0	17.8	83.6	34.7	76.0	58.6	28.6	84.1	37.8	43.1	7.2	32.2	34.2	44.3																																		
[7] 2019	MS+IW+Multi	RN-101	89.4	43.0	82.1	30.5	21.3	30.3	34.7	24.0	85.3	39.4	78.2	63.0	22.9	84.6	36.4	43.0	5.5	34.7	33.5	46.4																																		
[62] 2019	CRST (MRENT)	RN-101	91.8	53.4	80.6	32.6	20.8	34.3	29.7	21.0	84.0	34.1	80.6	53.9	24.6	82.8	30.8	34.9	16.6	26.4	42.6	46.1																																		
[62] 2019	CRST (MRKLD)	RN-101	91.0	55.4	80.0	33.7	21.4	37.3	32.9	24.5	85.0	34.1	80.8	57.7	24.6	84.1	27.8	30.1	26.9	26.0	42.3	47.1																																		
[62] 2019	CRST (LRENT)	RN-101	91.8	53.5	80.5	32.7	21.0	34.0	29.0	20.3	83.9	34.2	80.9	53.1	23.9	82.7	30.2	35.6	16.3	25.9	42.8	45.9																																		
[34] 2019	PyCDA	RN-101	90.5	36.3	84.4	32.4	28.7	34.6	36.4	31.5	86.8	37.9	78.5	62.3	21.5	85.6	27.9	34.8	18.0	22.9	49.3	47.4																																		
[26] 2019	MLSL (SISC) ST	RN-101	91.0	49.3	79.9	24.4	27.9	37.9	45.1	45.1	81.3	19.0	61.7	63.9	28.0	86.5	23.9	42.3	41.9	33.1	44.4	48.7																																		
[26] 2020	MLSL (+PWL) ST	RN-101	89.0	45.2	78.2	22.9	27.3	37.4	46.1	43.8	82.9	18.6	61.2	60.4	26.7	85.4	35.9	44.9	36.4	37.2	49.3	49.0																																		
Other Methods																																																								
[30] 2019	SWD	RN-101	92.0	46.4	82.4	24.8	24.0	35.1	33.4	34.2	83.6	30.4	80.9	56.9	21.9	82.0	24.4	28.7	6.1	25.0	33.6	44.5																																		
[31] 2020	CCM	RN-101	93.5	57.6	84.6	39.3	24.1	25.2	35.0	17.3	85.0	40.6	86.5	58.7	28.7	85.8	49.0	56.4	5.4	31.9	43.2	49.9																																		
[28] 2020	PLCA	RN-101	84.0	30.4	82.4	35.3	24.8	32.2	36.8	24.5	85.5	37.2	78.6	66.9	32.8	85.5	40.4	48.0	8.8	29.8	41.8	47.7																																		
[39] 2020	IntraDA	RN-101	90.6	37.1	82.6	30.1	19.1	29.5	32.4	20.6	85.7	40.5	79.7	58.7	31.1	86.3	31.5	48.3	0.0	30.2	35.8	46.3																																		
<i>Ours (CutMix)</i>		RN-101	90.9	45.2	81.3	26.4	19.2	21.7	33.2	18.3	84.3	41.1	78.8	61.2	21.3	87.2	43.6	50.2	5.8	29.3	21.8	45.4																																		
<i>Ours (Style)</i>		RN-101	83.7	34.5	81.5	18.9	11.0	17.6	24.3	24.0	84.7	31.5	77.9	59.8	21.2	81.3	42.5	36.2	2.51	6.30	5.23	39.2																																		
<i>Ours (Fourier)</i>		RN-101	83.0	34.4	81.1	27.9	14.4	22.3	35.1	15.7	85.3	39.4	77.3	59.1	18.2	86.6	43.3	49.5	5.46	30.2	26.8	44.0																																		
<i>Ours (Augmentations)</i>		RN-101	81.0	33.4	84.3	32.9	27.6	25.7	38.3	47.0	86.5	36.9	84.9	64.6	28.7	85.8	42.3	40.2	1.5	33.7	41.8	48.3																																		
<i>Ours (Augmentations) + MS</i>		RN-101	91.6	51.2	84.7	37.3	29.1	24.6	31.3	37.2	86.5	44.3	85.3	62.8	22.6	87.6	38.9	52.3	0.65	37.2	50.0	<b>50.3</b>																																		

Table 1. GTA5-to-Cityscapes results. RN-101 and WRN-38 refer to ResNet-101 and Wider-ResNet-38 architectures.

**CutMix** The CutMix-based model, which has no domain-adaptation-specific components, gives very strong results. This is notable because most highly-competitive recent methods employ domain-adaptation-specific components in the form of adversarial training (e.g. [53]), domain-specific priors (e.g. [26]), style transfer (e.g. [29]), or distribution alignment (e.g. [30]). Compared to other pure self-training

methods (e.g. MinEnt [52], MaxSquare [7], CRST [62], PyCDA [34]) our method performs substantially better.

**Style** The style-based variant performs badly relative to our other models and to other approaches that use style transfer directly. We hypothesize that this underperformance may be due to our choice of style transfer model, as we

		SYNTHIA-to-Cityscapes																		
Year	Method	Backbone	road	sidewalk	building	wall*	fence*	pole*	light	sign	veg.	sky	person	rider	car	bus	motor	bike	mIoU-16	mIoU-13
		No Adaptation																		
Source only		RN-101	17.7	15.0	74.3	10.1	0.1	25.5	6.3	10.2	75.5	77.9	57.1	19.2	31.2	31.2	10.0	20.1	30.1	34.3
Adversarial Methods																				
[49] 2018	AdaptSegNet	RN-101	79.2	37.2	78.8	10.5	0.3	25.1	9.9	10.5	78.2	80.5	53.5	19.6	67.0	29.5	21.6	31.3	39.5	45.9
[49] 2018	AdaptSegNet-LS	RN-101	84.0	40.5	79.3	10.4	0.2	22.7	6.5	8.0	78.3	82.7	56.3	22.4	74.0	33.2	18.9	34.9	40.8	47.6
[52] 2019	AdvEnt	RN-101	87.0	44.1	79.7	9.6	0.6	24.3	4.8	7.2	80.1	83.6	56.4	23.7	72.7	32.6	12.8	33.7	40.8	47.6
[52] 2019	AdvEnt+MinEnt	RN-101	85.6	42.2	79.7	8.7	0.4	25.9	5.4	8.1	80.4	84.1	57.9	23.8	73.3	36.4	14.2	33.0	41.2	48.0
[50] 2019	Patch-Disc	RN-101	82.4	38.0	78.6	8.7	0.6	26.0	3.9	11.1	75.5	84.6	53.5	21.6	71.4	32.6	19.3	31.7	40.0	46.5
[60] 2019	CAG	RN-101	84.7	40.8	81.7	7.8	0.0	35.1	13.3	22.7	84.5	77.6	64.2	27.8	80.9	19.7	22.7	48.3	44.5	52.6
[53] 2020	FADA	RN-101	84.5	40.1	83.1	4.8	0.0	34.3	20.1	27.2	84.8	84.0	53.5	22.6	85.4	43.7	26.8	27.8	45.2	52.5
[25] 2020	CrCDA R	RN-101	86.2	44.9	79.5	8.3	0.7	27.8	9.4	11.8	78.6	86.5	57.2	26.1	76.8	39.9	21.5	32.1	42.9	50.0
Image-to-Image / Style Transfer Methods																				
[33] 2019	BDL (M2-F2)	RN-101	86.0	46.7	80.3	-	-	-	14.1	11.6	79.2	81.3	54.1	27.9	73.7	42.2	25.7	45.3	-	51.4
[12] 2019	CrDoCo	DRN-26	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	38.5
[56] 2020	FDA (Ensemble)	RN-101	79.3	35.0	73.2	-	-	-	19.9	24.0	61.7	82.6	61.4	31.1	83.9	40.8	38.4	51.1	-	52.5
[29] 2020	LTIR	RN-101	92.6	53.2	79.2	-	-	-	1.6	7.5	78.6	84.4	52.6	20.0	82.1	34.8	14.6	39.4	-	49.3
Self-Training Methods																				
[63] 2018	CBST-SP	WRN-38	53.6	23.7	75.0	12.5	0.3	36.4	23.5	26.3	84.8	74.7	67.2	17.5	84.5	28.4	15.2	55.8	42.5	48.4
[52] 2019	MinEnt	RN-101	73.5	29.2	77.1	7.7	0.2	27.0	7.1	11.4	76.7	82.1	57.2	21.3	69.4	29.2	12.9	27.9	38.1	44.2
[7] 2019	MaxSquare (MS)	RN-101	77.4	34.0	78.7	5.6	0.2	27.7	5.8	9.8	80.7	83.2	58.5	20.5	74.1	32.1	11.0	29.9	39.3	45.8
[7] 2019	MS + IW + Multi	RN-101	82.9	40.7	80.3	10.2	0.8	25.8	12.8	18.2	82.5	82.2	53.1	18.0	79.0	31.4	10.4	35.6	41.4	48.2
[62] 2019	CRST (MRENT)	RN-101	69.6	32.6	75.8	12.2	1.8	35.3	23.3	29.5	77.7	78.9	60.0	28.5	81.5	25.9	19.6	41.8	43.4	49.6
[62] 2019	CRST (MRKLD)	RN-101	67.7	32.2	73.9	10.7	1.6	37.4	22.2	31.2	80.8	80.5	60.8	29.1	82.8	25.0	19.4	45.3	43.8	50.1
[62] 2019	CRST (LRENT)	RN-101	65.6	30.3	74.6	13.8	1.5	35.8	23.1	29.1	77.0	77.5	60.1	28.5	82.2	22.6	20.1	41.9	42.7	48.7
[34] 2019	PyCDA	RN-101	75.5	30.9	83.3	20.8	0.7	32.7	27.3	33.5	84.7	85.0	64.1	25.4	85.0	45.2	21.2	32.0	46.7	53.3
[26] 2020	MLSL (SISC)	RN-101	73.7	34.4	78.7	13.7	2.9	36.6	28.2	22.3	86.1	76.8	65.3	20.5	81.7	31.4	13.9	47.3	44.4	50.8
[26] 2020	MLSL (+PWL)	RN-101	59.2	30.2	68.5	22.9	1.0	36.2	32.7	28.3	86.2	75.4	68.6	27.7	82.7	26.3	24.3	52.7	45.2	51.0
Other Methods																				
[30] 2019	SWD	RN-101	82.4	33.2	82.5	-	-	-	22.6	19.7	83.7	78.8	44.0	17.9	75.4	30.2	14.4	39.9	-	48.1
[31] 2020	CCM	RN-101	79.6	36.4	80.6	13.3	0.3	25.5	22.4	14.9	81.8	77.4	56.8	25.9	80.7	45.27	29.9	52.0	45.2	52.9
[28] 2020	PLCA	RN-101	82.6	29.0	81.0	11.2	0.2	33.6	24.9	18.3	82.8	82.3	62.1	26.5	85.6	48.9	26.8	52.2	<b>46.8</b>	54.0
[39] 2020	IntraDA	RN-101	84.3	37.7	79.5	5.3	0.4	24.9	9.2	8.4	80.0	84.1	57.2	23.0	78.0	38.1	20.3	36.5	41.7	48.9
<i>Ours (CutMix)</i>		RN-101	84.8	42.1	81.0	6.98	0.25	27.9	15.6	16.6	82.3	80.7	53.9	21.8	83.1	39.3	21.0	43.1	43.8	51.2
<i>Ours (Augmentations)</i>		RN-101	73.0	38.9	70.8	6.0	0.1	27.0	17.0	20.3	83.0	84.2	59.1	27.0	80.1	37.4	17.8	52.4	43.4	50.8
<i>Ours (Fourier)</i>		RN-101	82.9	36.2	81.7	9.74	0.11	29.7	16.7	19.2	84.3	84.2	62.3	16.9	84.6	39.4	3.04	52.5	44.0	51.1
<i>Ours (Style)</i>		RN-101	82.1	38.0	76.2	3.97	0.12	26.4	14.2	11.0	75.5	70.7	54.3	20.6	75.8	36.9	19.4	39.3	40.3	47.3
<i>Ours (Fourier + CutMix)</i>		RN-101	92.5	54.6	79.8	4.78	0.08	24.1	22.8	17.8	79.4	76.5	60.8	24.7	85.7	33.5	26.4	54.4	46.1	<b>54.5</b>

Table 2. Results on the SYNTHIA-to-Cityscapes benchmark. mIoU-16 and mIoU-13 refer to mean intersection-over-union on the standard sets of 16 and 13 classes, respectively.

used AvatarNet [45] rather than CycleGAN as in [33] or CycleGAN + StyleSwap as in [29].

**Combining Perturbation Functions** PixMatch also makes it easy to combine multiple perturbation functions. For example, Table 2 shows the result of combining the Fourier- and CutMix-based perturbations on the SYNTHIA benchmark. We give additional results for combining per-

turbation functions in the Appendix (Table 6). We find that combining two perturbation functions often improves results, while combining three or more is “too much” perturbation.

We note that we have not experimented with using different strengths/weights for different perturbation functions, nor have we experimented with stochastically applying perturbation functions. Studying the interactions of multiple perturbation functions could be a promising avenue for fu-

$\lambda_T$	0.00 (Source)	0.05	0.10	0.15	0.20
<i>mIoU</i>	36.9	48.7	49.4	<b>49.8</b>	48.0

Table 3. Results of our augmentation-based model, varying the consistency loss weight hyperparameter  $\lambda_T$ .

$\tau$	0.00	0.50	0.90	0.95
<i>mIoU</i>	<b>49.9</b>	48.7	47.1	47.0

Table 4. Results of our augmentation-based model with an additional pseudolabel threshold. A threshold of  $\tau \in [0, 1)$  corresponds to only calculating the consistency loss on pseudolabeled pixels for which the model assigns an output probability greater than  $\tau$  to one class.

$\lambda_{MSL}$	0.00	0.05	0.10	0.15
<i>mIoU</i>	49.9	<b>50.7</b>	50.5	50.4

Table 5. Results of a model combining the max-square loss [7] with our augmentation-based consistency loss, for different values of  $\lambda_{MSL}$

ture research.

## 6. Ablation Studies and Further Experiments

Below, we present an extensive set of ablation studies on our baseline (augmentation-based) model.

**Ablation: Varying  $\lambda_T$**  Our proposed consistency training method involves choosing a perturbation function and tuning the single hyperparameter  $\lambda_T$ , which trades off between the strength of the source supervised loss and the target consistency loss. We present results for  $\lambda_T = 0.00, 0.05, 0.10, 0.15$ , and  $0.20$  in Table 3, finding that  $\lambda_T = 0.10$  and  $0.15$  both yield strong results. This finding shows a stark contrast between UDA for semantic segmentation and self-supervised learning for image classification, where the consistency loss is usually much greater than 1 ( $\lambda_{unsup} > 1$ ) [46, 55].

**Ablation: Adding a Pseudolabel Threshold** It is also possible to add a pseudolabel threshold in our formulation of consistency training. This corresponds to only pseudolabeling pixels in which the target output probability exceeds some threshold  $\tau$  (other pixels are ignored from the loss). For example, [46] uses  $\tau = 0.95$ .

In Table 4, we show results for  $\tau = 0, 0.5, 0.9$ , and  $0.95$ . Surprisingly, unlike in semi-supervised learning for image classification, we find that  $\tau = 0.0$  (i.e. no thresholding) performs best. We hypothesize that this may be because when a large threshold is used, the loss is dominated by easy classes; if this is the case, future work could potentially address the issue by performing consistency training with class-wise thresholds [63].

**Combining PixMatch with Other Methods** Our consistency training method is complementary to a wide range of other domain adaptation techniques. To demonstrate this, we combine our method with the max-square loss proposed in [7]. That is, we modify our loss function to be  $\mathcal{L} = \mathcal{L}_S + \lambda_T \mathcal{L}_T + \lambda_{MSL} \mathcal{L}_{MSL}$ . In Table 5, we fix  $\lambda_T = 0.10$  and present results for different weightings  $\lambda_{MSL}$  of the max-square loss term. The across-the-board performance improvement demonstrates that our method is readily combinable with other self-training approaches. As a highlight, the variant with  $\lambda_{MSL} = 0.05$  achieves state-of-the-art performance on GTA5-to-Cityscapes.

## 7. Conclusion

We present PixMatch, an unsupervised domain adaptation approach for semantic segmentation that incorporates consistency training in the target domain. Through an extensive set of ablation studies, we sought to understand which aspects of this consistency training framework are most important to the model’s final performance. In comparison to adversarial approaches that have dominated the recent literature, our approach is faster, more stable, has superior performance, and may be readily combined with other methods. Future work may explore new perturbation functions, the combination of multiple perturbation functions, or the application of this framework to other tasks (e.g. object detection). We will release code and pretrained models to facilitate future research.

## References

- [1] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Advances in Neural Information Processing Systems*, pages 15509–15519, 2019.
- [2] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 3
- [3] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, pages 5050–5060, 2019. 3, 4
- [4] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998. 3
- [5] Alexander Buslaev, Vladimir I Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A Kalinin. Albumentations: fast and flexible image augmentations. *Information*, 11(2):125, 2020. 12
- [6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 5
- [7] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019. 1, 2, 4, 6, 7, 8
- [8] Minghao Chen, Shuai Zhao, Haifeng Liu, and Deng Cai. Adversarial-learned loss for domain adaptation, 2020. 1
- [9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 3, 12
- [10] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1841–1850, 2019.
- [11] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2018. 1, 2
- [12] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1791–1800, 2019. 2, 3, 4, 6, 7
- [13] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 1992–2001, 2017.
- [14] Jaehoon Choi, Taekyung Kim, and Changick Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 6830–6840, 2019.
- [15] 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. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016. 1, 2
- [16] Geoff French, Timo Aila, Samuli Laine, Michal Mackiewicz, and Graham Finlayson. Consistency regularization and cutmix for semi-supervised semantic segmentation. *arXiv preprint arXiv:1906.01916*, 2019. 4
- [17] Geoff French, Michal Mackiewicz, and Mark Fisher. Self-ensembling for visual domain adaptation. In *International Conference on Learning Representations*, 2018. 3
- [18] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, Kun Zhang, and Dacheng Tao. Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2427–2436, 2019. 2, 3
- [19] Ishaa Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017. 1
- [20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 3
- [21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5
- [22] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.
- [23] 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. *arXiv preprint arXiv:1711.03213*, 2017. 1, 2, 4
- [24] Weixiang Hong, Zhenzhen Wang, Ming Yang, and Junsong Yuan. Conditional generative adversarial network for structured domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, 2018.
- [25] Jiaxing Huang, Shijian Lu, Dayan Guan, and Xiaobing Zhang. Contextual-relation consistent domain adaptation for semantic segmentation. *arXiv preprint arXiv:2007.02424*, 2020. 6, 7
- [26] Javed Iqbal and Mohsen Ali. Misl: Multi-level self-supervised learning for domain adaptation with spatially independent and semantically consistent labeling. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1864–1873, 2020. 3, 6, 7

- [27] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019.
- [28] Guoliang Kang, Yunchao Wei, Yi Yang, Yueting Zhuang, and Alexander Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. *Advances in Neural Information Processing Systems*, 33, 2020. [2](#), [6](#), [7](#)
- [29] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12975–12984, 2020. [3](#), [6](#), [7](#)
- [30] Chen-Yu Lee, Tanmay Batra, Mohammad Haris Baig, and Daniel Ulbricht. Sliced wasserstein discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10285–10295, 2019. [6](#), [7](#)
- [31] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In *European Conference on Computer Vision (ECCV)*, 2020. [6](#), [7](#)
- [32] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. [2](#)
- [33] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6936–6945, 2019. [2](#), [6](#), [7](#)
- [34] Qing Lian, Lixin Duan, Fengmao Lv, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6757–6766. IEEE, 2019. [6](#), [7](#)
- [35] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2507–2516, 2019.
- [36] Geoffrey J McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975.
- [37] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, 2018.
- [38] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4500–4509, 2018. [2](#), [4](#)
- [39] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3764–3773, 2020. [6](#), [7](#)
- [40] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016. [1](#)
- [41] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. [1](#), [2](#)
- [42] Mehdi Sajjadi, Mehran Javanmardi, and Tolga Tasdizen. Regularization with stochastic transformations and perturbations for deep semi-supervised learning, 2016. [3](#)
- [43] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M Alvarez. Effective use of synthetic data for urban scene semantic segmentation. In *European Conference on Computer Vision*, pages 86–103. Springer, 2018. [1](#)
- [44] H Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.
- [45] Lu Sheng, Ziyi Lin, Jing Shao, and Xiaogang Wang. Avatar-net: Multi-scale zero-shot style transfer by feature decoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8242–8250, 2018. [4](#), [7](#), [12](#)
- [46] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*, 2020. [3](#), [8](#)
- [47] Aravind Srinivas, Michael Laskin, and Pieter Abbeel. Curl: Contrastive unsupervised representations for reinforcement learning. *arXiv preprint arXiv:2004.04136*, 2020.
- [48] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204, 2017.
- [49] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7472–7481, 2018. [1](#), [2](#), [4](#), [6](#), [7](#)
- [50] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *IEEE International Conference on Computer Vision (ICCV)*, 2019. [6](#), [7](#)
- [51] Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *arXiv preprint arXiv:1903.03825*, 2019.

- [52] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2517–2526, 2019. [1](#), [2](#), [4](#), [6](#), [7](#)
- [53] Haoran Wang, Tong Shen, Wei Zhang, L. Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In *ECCV*, 2020. [6](#), [7](#)
- [54] Jiawei Wang, Zhaoshui He, Chengjian Feng, Zhouping Zhu, Qinzhuang Lin, Jun Lv, and Shengli Xie. Domain confusion with self ensembling for unsupervised adaptation. *arXiv preprint arXiv:1810.04472*, 2018. [1](#)
- [55] Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*, 2019. [3](#), [8](#)
- [56] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4085–4095, 2020. [3](#), [4](#), [5](#), [6](#), [7](#), [12](#)
- [57] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on computer vision and pattern recognition*, pages 6210–6219, 2019.
- [58] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019. [4](#), [12](#)
- [59] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. S4l: Self-supervised semi-supervised learning. In *The IEEE International Conference on Computer Vision (ICCV)*, 10 2019.
- [60] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 435–445, 2019. [6](#), [7](#)
- [61] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source domain adaptation for semantic segmentation. In *Advances in Neural Information Processing Systems*, pages 7287–7300, 2019. [2](#)
- [62] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5982–5991, 2019. [6](#), [7](#)
- [63] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018. [1](#), [2](#), [4](#), [6](#), [7](#), [8](#)

## Appendix

### 7.1. Perturbation Function Details

Here, we describe the parameters used for each perturbation function. For the baseline model with data augmentation as the perturbation function, we use the set of augmentation in Figure 3. These are based on the augmentations from SimCLR [9] and implemented using the albumentations [5] library. For style consistency model, we obtain perturbed images by applying the Avatar-Net [45] style transfer model to the image.<sup>1</sup> We use the default Avatar-Net strength parameter of 0.1. For the Fourier consistency model, we employ the same Fourier swapping function as [56]. We use a low-frequency window size parameter of  $\beta = 0.01$ . Finally, for the CutMix consistency model, we employ the cutmix function from [58].<sup>2</sup>

### 7.2. Additional Combinations of Perturbation Functions

In Table 6, we show the result of additional combinations of perturbation functions.

---

#### Algorithm 2: Augmentations

---

```
Compose([
    RandomResizedCrop(scale=(0.2, 1)),
    Compose([
        RandomBrightnessContrast(p=1),
        HueSaturationValue(p=1)
    ], p=0.8),
    ToGray(p=0.2),
    GaussianBlur(blur_limit=5, p=0.5),
])
```

---

Figure 3. PyTorch-style pseudocode for our chosen data augmentation function.

---

<sup>1</sup>[https://github.com/tyui592/Avatar-Net\\_Pytorch](https://github.com/tyui592/Avatar-Net_Pytorch)

<sup>2</sup><https://github.com/clovaai/CutMix-PyTorch>

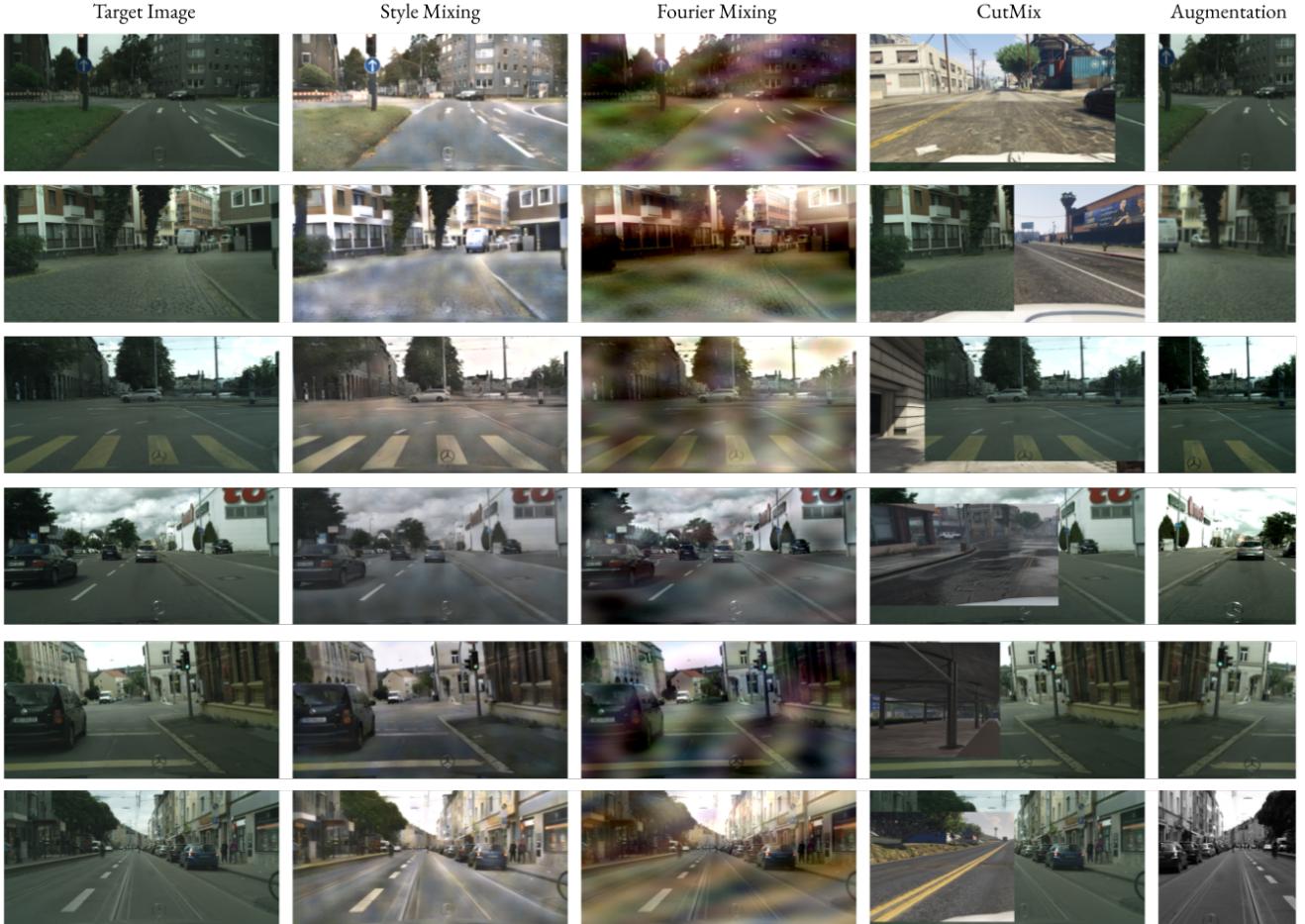


Figure 4. Examples of our data augmentation function applied to target-domain images. Note that the “augmentation” images are a different size and aspect ratio because random cropping is part of the augmentation procedure. Also note that although the “style” and “Fourier” images appear unnatural on a global scale, they appear similar to the source domain on a local/low-frequency scale.

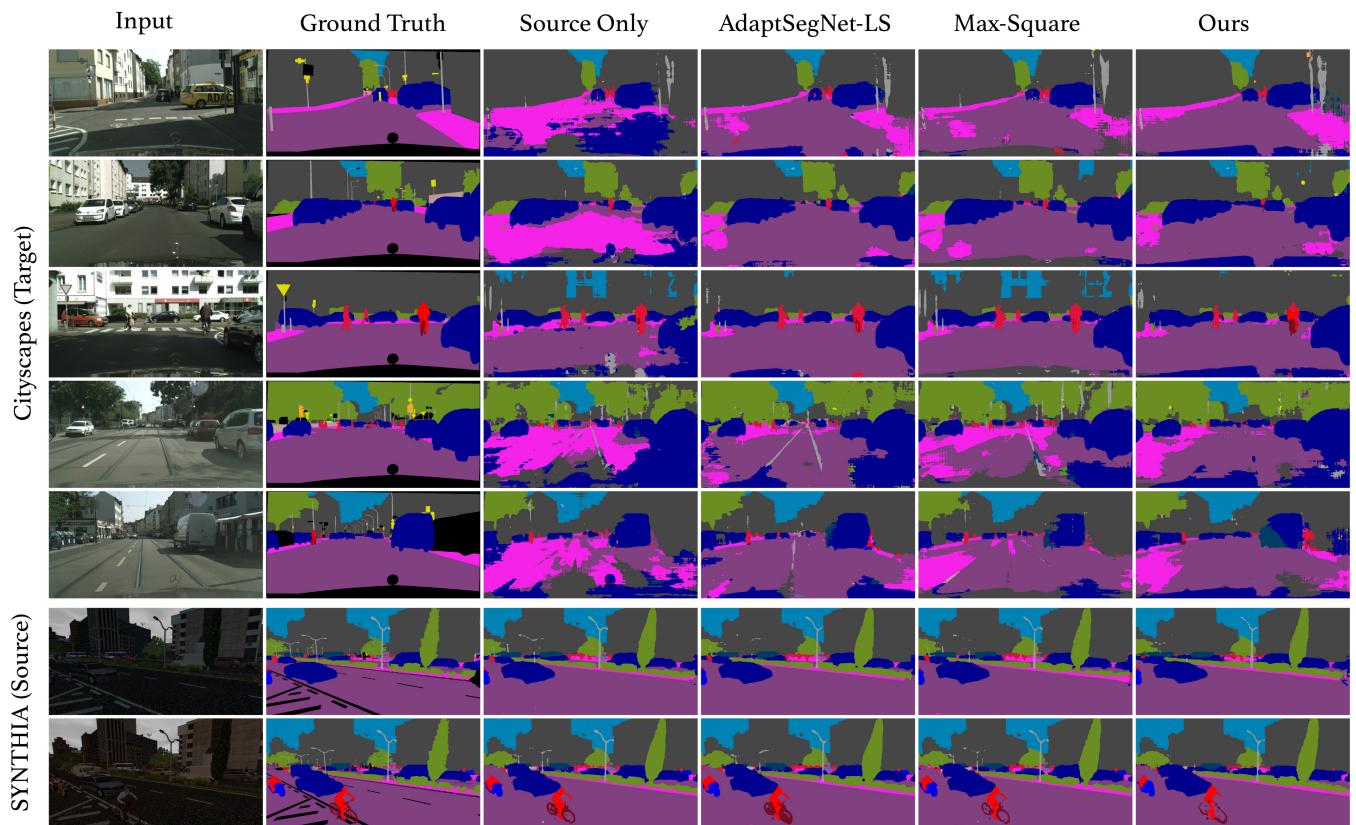


Figure 5. Qualitative examples of our consistency training method and prior methods on SYNTHIA-to-Cityscapes.

SYNTHIA-to-Cityscapes		
	mIoU-16	mIoU-13
LTIR [28]	-	49.3
SOTA (PLCA [27])	<b>46.8</b>	54.0
Ours: Fourier only	44.0	51.1
Ours: Fourier + Aug	45.6	54.0
Ours: Fourier + Cutmix	46.1	<b>54.5</b>
Ours: Fourier + Cutmix + Aug	43.7	52.1

GTA-to-Cityscapes	
	mIoU
SOTA (LTIR [28])	50.2
PLCA [27]	47.7
Ours: Aug only	48.3
Ours: Aug + Fourier	49.3
Ours: Aug + MaxSquareLoss	<b>50.4</b>

Table 6. Performance of additional combinations of perturbation functions compared to SOTA.