PixMatch: Unsupervised Domain Adaptation via Pixelwise Consistency Training

Luke Melas-Kyriazi
Harvard University
Cambridge, MA
lmelaskyriazi@college.harvard.edu

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

$$L_S = -\frac{1}{n_S} \sum_{s \in S} \sum_{i=1}^{H \cdot W} H\left(y_s^{(i)}, p^{(i)}(y|x_s)\right)$$  \hspace{1cm} (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 $\hat{y}_t = \text{arg max}(q_t) = \text{arg max}(p_t(y|x_t))$. 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

$$L_T = -\frac{1}{n_T} \sum_{t \in T} \sum_{i=1}^{H \cdot W} H\left(y_{pert}^{(i)}, p^{(i)}(y|x_{pert})\right)$$  \hspace{1cm} (2)$$

and the final loss is $L = L_S + \lambda_T L_T$, 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 style-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.

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 $1.0 \times 10^{-4}$, momentum 0.9, weight decay $5 \times 10^{-4}$, and a polynomial learning rate decay: $lr = lr \times (1 - \frac{iter}{max\_iter})^{0.9}$. 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).
This is notable because most highly-competitive recent methods employ domain-adaptation-specific components in the adaptation-specific components, giving very strong results.

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...
<table>
<thead>
<tr>
<th>Year</th>
<th>Method</th>
<th>Backbone</th>
<th>road</th>
<th>sidewalk</th>
<th>building</th>
<th>wall*</th>
<th>fence*</th>
<th>pole*</th>
<th>light</th>
<th>sign</th>
<th>veg*</th>
<th>sky</th>
<th>person</th>
<th>rider</th>
<th>car</th>
<th>bus</th>
<th>motor</th>
<th>bike</th>
<th>mIoU-16</th>
<th>mIoU-13</th>
</tr>
</thead>
<tbody>
<tr>
<td>2018</td>
<td>AdaptSegNet</td>
<td>RN-101</td>
<td>79.2</td>
<td>37.2</td>
<td>78.8</td>
<td>10.5</td>
<td>0.3</td>
<td>25.1</td>
<td>9.9</td>
<td>10.5</td>
<td>78.2</td>
<td>80.5</td>
<td>53.5</td>
<td>19.6</td>
<td>67.0</td>
<td>29.5</td>
<td>21.6</td>
<td>31.3</td>
<td>39.5</td>
<td>45.9</td>
</tr>
<tr>
<td>2018</td>
<td>AdaptSegNet-LS</td>
<td>RN-101</td>
<td>84.0</td>
<td>40.5</td>
<td>79.3</td>
<td>10.4</td>
<td>0.2</td>
<td>22.7</td>
<td>6.5</td>
<td>8.0</td>
<td>78.3</td>
<td>82.7</td>
<td>56.3</td>
<td>22.4</td>
<td>74.0</td>
<td>33.2</td>
<td>18.9</td>
<td>34.9</td>
<td>40.8</td>
<td>47.6</td>
</tr>
<tr>
<td>2019</td>
<td>AdvEnt</td>
<td>RN-101</td>
<td>87.0</td>
<td>41.1</td>
<td>79.7</td>
<td>9.6</td>
<td>0.6</td>
<td>24.3</td>
<td>4.8</td>
<td>7.2</td>
<td>80.1</td>
<td>83.6</td>
<td>56.4</td>
<td>23.7</td>
<td>72.7</td>
<td>32.6</td>
<td>12.8</td>
<td>33.7</td>
<td>40.8</td>
<td>47.6</td>
</tr>
<tr>
<td>2019</td>
<td>AdvEnt+MinEnt</td>
<td>RN-101</td>
<td>85.6</td>
<td>42.2</td>
<td>79.7</td>
<td>8.7</td>
<td>0.4</td>
<td>25.9</td>
<td>5.4</td>
<td>8.1</td>
<td>80.4</td>
<td>84.1</td>
<td>57.9</td>
<td>23.8</td>
<td>73.3</td>
<td>36.4</td>
<td>14.2</td>
<td>33.0</td>
<td>41.2</td>
<td>48.0</td>
</tr>
<tr>
<td>2019</td>
<td>Patch-Disc</td>
<td>RN-101</td>
<td>82.4</td>
<td>38.0</td>
<td>78.6</td>
<td>8.7</td>
<td>0.6</td>
<td>26.0</td>
<td>3.9</td>
<td>11.1</td>
<td>75.5</td>
<td>84.6</td>
<td>53.5</td>
<td>21.6</td>
<td>71.4</td>
<td>32.6</td>
<td>19.3</td>
<td>31.7</td>
<td>40.0</td>
<td>46.5</td>
</tr>
<tr>
<td>2019</td>
<td>CAG</td>
<td>RN-101</td>
<td>84.7</td>
<td>40.8</td>
<td>81.7</td>
<td>7.8</td>
<td>0.0</td>
<td>35.1</td>
<td>13.3</td>
<td>22.7</td>
<td>84.5</td>
<td>77.6</td>
<td>64.2</td>
<td>27.8</td>
<td>80.9</td>
<td>19.7</td>
<td>22.7</td>
<td>48.3</td>
<td>44.5</td>
<td>52.6</td>
</tr>
<tr>
<td>2020</td>
<td>FDA</td>
<td>RN-101</td>
<td>84.5</td>
<td>40.1</td>
<td>83.1</td>
<td>4.8</td>
<td>0.0</td>
<td>34.3</td>
<td>20.1</td>
<td>27.2</td>
<td>84.8</td>
<td>84.0</td>
<td>53.5</td>
<td>22.6</td>
<td>85.4</td>
<td>43.7</td>
<td>26.8</td>
<td>27.8</td>
<td>45.2</td>
<td>52.5</td>
</tr>
<tr>
<td>2020</td>
<td>CrCDA R</td>
<td>RN-101</td>
<td>86.2</td>
<td>44.9</td>
<td>79.5</td>
<td>8.3</td>
<td>0.7</td>
<td>27.8</td>
<td>9.4</td>
<td>11.8</td>
<td>78.6</td>
<td>86.5</td>
<td>57.2</td>
<td>26.1</td>
<td>76.8</td>
<td>39.9</td>
<td>21.5</td>
<td>32.1</td>
<td>42.9</td>
<td>50.0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$\lambda_T$</th>
<th>0.00 (Source)</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
<th>0.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>36.9</td>
<td>48.7</td>
<td>49.4</td>
<td>49.8</td>
<td>48.0</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>$\tau$</th>
<th>0.00</th>
<th>0.50</th>
<th>0.90</th>
<th>0.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>49.9</td>
<td>48.7</td>
<td>47.1</td>
<td>47.0</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>$\lambda_{MSL}$</th>
<th>0.00</th>
<th>0.05</th>
<th>0.10</th>
<th>0.15</th>
</tr>
</thead>
<tbody>
<tr>
<td>mIoU</td>
<td>49.9</td>
<td>50.7</td>
<td>50.5</td>
<td>50.4</td>
</tr>
</tbody>
</table>

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 $\tau = 0$ (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


