Not All Areas Are Equal: Transfer Learning for Semantic Segmentation via Hierarchical Region Selection

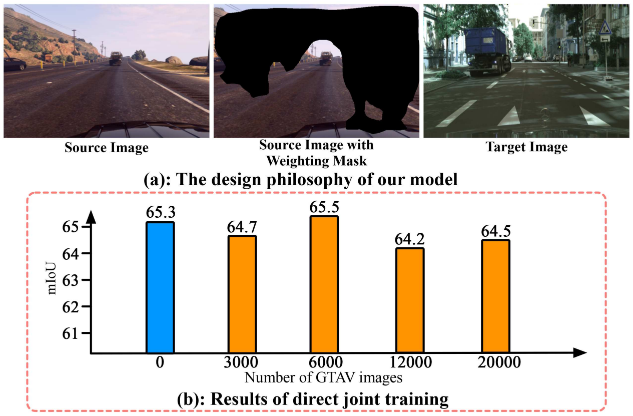
Ruoqi Sun [1]* Xinge Zhu [2]* Chongruo Wu [3] Chen Huang [4] Jianping Shi [5] Lizhuang Ma [1]
[1]Shanghai Jiao Tong University [2]The Chinese University of Hong Kong
[3]University of California, Davis [4]Carnegie Mellon University [5]SenseTime Research
ruoqisun7@sjtu.edu.cn zx018@ie.cuhk.edu.hk crwu@ucdavis.edu
chenh2@andrew.cmu.edu shijianping@sensetime.com ma-lz@cs.sjtu.edu.cn

Abstract

The success of deep neural networks for semantic segmentation heavily relies on large-scale and well-labeled datasets, which are hard to collect in practice. Synthetic data offers an alternative to obtain ground-truth labels for free. However, models directly trained on synthetic data often struggle to generalize to real images. In this paper, we consider transfer learning for semantic segmentation that aims to mitigate the gap between abundant synthetic data (source domain) and limited real data (target domain). Unlike previous approaches that either learn mappings to target domain or finetune on target images, our proposed method jointly learn from real images and selectively from realistic pixels in synthetic images to adapt to the target domain. Our key idea is to have weighting networks to score how similar the synthetic pixels are to real ones, and learn such weighting at pixel-, region- and image-levels. We jointly learn these hierarchical weighting networks and segmentation network in an end-to-end manner. Extensive experiments demonstrate that our proposed approach significantly outperforms other existing baselines, and is applicable to scenarios with extremely limited real images.

1. Introduction

The advances in deep learning have led to many breakthroughs in artificial intelligence. Various tasks in computer vision [13, 14, 32] have been revisited and have achieved the state-of-the-art performance. However, these improvements often require vast labeled data, which is prohibitively expensive for many vision tasks. Semantic segmentation is such an example, in which annotating an image pixel-wisely may take more than 90 minutes [7], resulting in a failure to scale. Alternatively, researchers [24, 25] switched to use Computer Graphics techniques to render synthetic images where pixel-wise labels are generated automatically at a much faster speed. In this paper, we consider utilizing **abundant labeled synthetic data (source domain)** and **insufficient labeled real data (target domain)** together to make a better performance on real data. It is noted that our problem setting is different from the *unsupervised domain adaptation* (*i.e.*, labeled synthetic data and unlabeled real data) and *semi-supervised domain adaptation* (*i.e.*, small amount of labeled real data and abundant unlabeled real data in one domain). Due to the labor-free annotations and superior performance, the learning scheme we focus on has much practical significance in real-world applications.



(a): The design philosophy of our model

(b): Results of direct joint training

Figure 1. (**a**): Although two images are from different domains, some regions still hold similar structures, like the car and road region. Our design philosophy is to focus on these similar regions for improving the effectiveness. (**b**): Results of joint training on *different numbers of synthetic images (from GTAV) and insufficient real images*. Blue bar indicates the result of using real images only.

Given abundant labeled synthetic images and insufficient labeled real-world images, it is natural to expect better segmentation performance from such ”augmentation” in pixel space. However, this is usually not the case and perfor-

*The first two authors contributed equally to this paper.

mance may even degrade as shown in Fig 1(b). The main reason lies in the domain gap between synthetic and real-world data, in terms of differences in textures (rendering realism), lighting conditions and so on. This domain gap can easily bias model learning towards the synthetic data distribution, causing a failure to generalize to real images [3, 23].

To alleviate this problem, data resampling methods are used to reduce the impact of data bias. Options include randomly resampling source domain images [2] or selecting those similar to target domain images based on low-level features [12]. Another family of methods uses transfer learning to apply the knowledge learned in one domain to another [1, 8, 15]. The idea is to learn the transformation to target domain in feature or output space. One common drawback of these methods is that they learn or un-learn from holistic images. However, for the pixel-wise segmentation task, it is likely to find domain knowledge or similarity in pixel regions, where transfer learning can leverage useful information locally. We show an example in Fig 1(a), in which a realistic image scene from synthetic image (source domain) contains the road region and a car that have *similar structure* to the counterpart in a real image (target domain). In other words, domain knowledge can be distributed at a fine-grained pixel level rather than only image level, and similar regions from source image make higher contributions for the joint learning.

Motivated by these findings, we propose a hierarchical transfer learning framework to learn the real image segmentation by combining information in synthetic images at three levels: pixel-, region- and image-levels. Three weighting networks are learned together to assign higher weights to such synthetic image granularities that are similar to real ones(target domain). We are hence able to learn from both real images and selected synthetic pixels for domain adaptation purposes, which follows the nature of pixel-wise task and is much more flexible than training image selection. Note that our weighting networks are jointly trained with segmentation network in an end-to-end manner. The entire training framework takes any combination of real and synthetic datasets as input, with no assumptions about their distributions — the common domain knowledge is automatically mined locally.

The main contributions of this paper can be summarized as follows:

- Our studies reveal a practical and cost-free learning scheme to improve the performance of real image segmentation with abundant synthetic images. It is also a step towards the generic learning setting with multiple datasets (sources).
- We develop a hierarchical transfer learning method for semantic segmentation, with the ability of learning from insufficient real images and auto-mined similar synthetic pixels.
- Extensive experiments are conducted on various datasets. The proposed method achieves state-of-the-art performance, while still stays strong with extremely insufficient (about only 50%) real images for training.

2. Related Work

Semantic Segmentation Semantic segmentation is an important task in a large variety of fields, like autonomous vehicles, remote sensing, etc. Recently, the revolution of deep neural network has pushed this task to a new stage [5, 31, 32, 21]. Unfortunately, training such deep models usually requires a large amount of well-labeled images, which is expensive and time-consuming. To save time and cost for annotation, researchers attempted to obtain data and corresponding free labels from the video game GTAV [24] or their own simulation environment [9]. Although collecting them is much faster and cheaper, the use of synthetic images does not necessarily generalize to real images due to the domain gap.

Transfer Learning Transfer learning aims to apply the knowledge learned from one domain to improve the learning in another. It is a popular approach to address issues caused by insufficient data in one of domains or the data gap between different domains [4, 11].

There are two main folds of transfer learning methods: data selection and domain transformation. 1). For the data selection methods, the random selection strategy [1, 8] can be seen as a simplest option, then [12] improved by scoring the source data according to the low-level feature-based similarity. 2). For the domain transformation methods, Sarafianos et al. [27] applied the Adaptive-SVM+ algorithm to extract useful information from source domain.

Many recent works have been trying to apply GANs for domain alignment. For example, [10, 11, 30, 34, 33] use adversarial training to obtain domain-invariant representations and reduce the domain gap. Volpi et al. [29] trained the encoder of the source domain to augment features via auxiliary training. Chen et al. [6] exploited spatial structure to transfer region-level knowledge from source to target domain. [16] enhanced the semi-supervised learning by combining the discriminator with the weighting map prediction on the pixel-level only, in which all data from same domain is used.

The common drawback of methods mentioned above is that they are applied at single level, neglecting the fact that domain knowledge/similarity can be distributed at multiple levels (*i.e.*, pixel, region and image-level) across different images. In this paper, we propose a transfer learning method to transfer knowledge from hierarchical levels in source domain, jointly training with semantic segmentation. Note that our problem setting is different from other transfer learning settings. Unlike the unsupervised [4, 6, 34] and

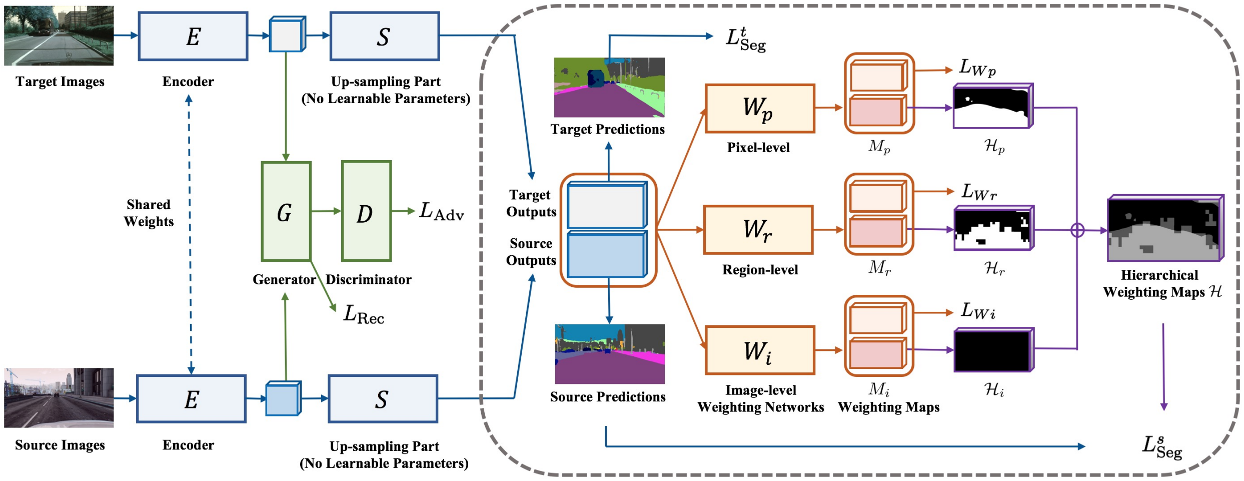


Figure 2. An overview of our model architecture. A pair of source and target domain images go through the encoder E and a segmentation classifier S (only consists of upsample operation) to predict segmentation maps under loss L^s_{Seg} and L^t_{Seg}. For source image predictions, they are re-weighted by 3 weighting networks W_p, W_r and W_i at pixel-, region- and image levels before computing the loss L^s_{Seg}. We also improve encoder E's expressiveness by attaching a generator G and a discriminator D to assess the reconstructed image quality (L_{Rec}) and fidelity (L_{Adv}). For each pair of source and target images, we alternatively optimize encoder E, networks G+D, and weighting networks (by L_{W_p}, L_{W_r}, L_{W_i}) via back-propagation.

semi-supervised domain adaptation work [16], our method takes the labeled source and labeled target images together to automatically mine the similarity between them to do segmentation adaptation.

3. Methodology

Our goal is to perform transfer learning from insufficient real data (target domain) and abundant synthetic data (source domain) to improve the performance of semantic segmentation. Concretely, we have the dataset $X_s = \{(x^s, y^s)\}$ and $X_t = \{(x^t, y^t)\}$ drawn from a labeled source domain s and a labeled target domain t, sharing the set of categories for segmentation. During training, we take both source and target data as the inputs, while we only test on target images. Note that there is no overlap between the training and testing images, and our setting is different from the semi-supervised learning which has some labeled data and more unlabeled data but both are in one domain.

Under our problem setting, the learning difficulty lies in the data gap between source and target domains. To address such gap adaptively, we propose weighting networks to favor regions from source images that are highly similar to the target ones, and leverage them to benefit joint learning from both source and target domains. In order to take both local and global information into account, we learn hierarchical weighting networks to score similarity at pixel level, region level and the entire image level. The weighting networks are learned together with the segmentation network in an end-to-end manner. Motivated by the effectiveness of adversarial adaptation methods [11, 26], we also incorporate the GAN into our model, whose discriminator drives the source distribution towards the target one, to further help domain adaptation. Fig 2 illustrates our model architecture, which will be detailed in the following sections.

3.1. Hierarchical Weighting Networks

We aim to address the data gap by learning with target real images and only similar synthetic image regions. We propose weighting networks for such fine-grained region selection rather than image-level selection. The weighting networks should assign higher weights to synthetic image regions that are similar to real image regions from target domain. Due to significant variations in both texture and appearance (such as color and lighting) from two domains, we encourage the segmentation network to predict the same for similar structured regions. Therefore, it is effective to define the similarity in segmentation label space (no textures exist in label space), thus give more weight on those pixels with similar label structures. This essentially robustifies segmentation to data variance across domains in a transfer learning framework. We also see the link to data augmentation, but in a much more flexible way to augment with arbitrary image regions from source domain.

To enrich the transfer learning process, we propose hierarchical weighting networks to find between-domain similarity at pixel (network W_p), region (network W_r) and image (network W_i) levels. Their objective functions are:

$$L_{W_k} = \|M_k(x) - d_k\|_1, k \in \{p, r, i\}, \tag{1}$$

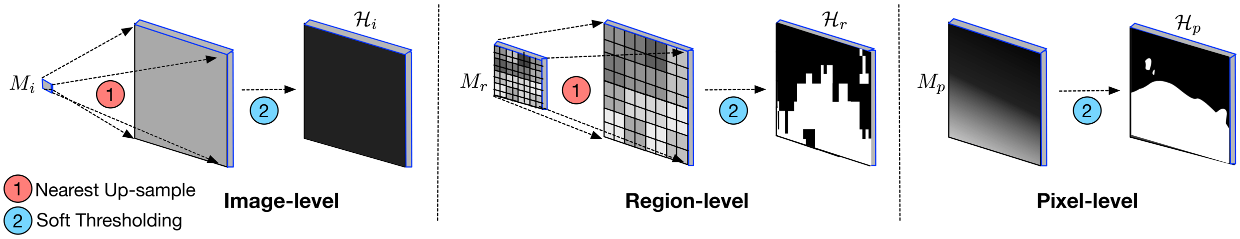


Figure 3. The workflow of the weighing map generation.

where $M_k(x)$ denotes the outputs of weighting network W_k. Here d_k denotes the ground-truth domain labels at different levels. Specifically, d_k is an all-zero map for source label, and all-one map for target. It has sizes of 512×1024, 64×128 at pixel- and region- level respectively, while it is a scalar at image level.

Semantic Segmentation Losses. As mentioned above, we aim to give more weights on source regions with similar label structures. We thus set different segmentation losses for target and source domain, respectively. For the target domain, we apply the normal cross-entropy as follows:

$$L^t_{\text{Seg}} = -\sum y^t \log \mathcal{F}(x^t). \tag{2}$$

where $\mathcal{F}$ is the segmentation part, including the Encoder E and up-sampling part S. $\mathcal{F}(x^t)$ denotes the output of the segmentation network with the input of target image x^t.

For the source domain, we perform the weighted segmentation loss with hierarchical weighting map $\mathcal{H}$. From the equation 1, the outputs of weighting networks $(M_k(x^s), k \in \{p, r, i\})$ are obtained. Since the size of $M_k(x)$ is not compatible with ground truth, we perform the nearest up-sampling and soft thresholding to get corresponding weighting maps $(\mathcal{H}_k, k \in \{p, r, i\})$. The detailed workflow is shown in Fig 3. Soft thresholding is defined as follows:

$$\mathcal{H}_k(x^s) = I(M_k(x^s) > mean(M_k(x^s))), k \in \{p, r, i\}, \tag{3}$$

where $I(\cdot)$ is an indication function. The soft thresholding using the adaptive mean value as the threshold works better than hard thresholding (*i.e.*, using a fixed value as the threshold) since it adapts to the current score range and guarantees to select some relatively similar regions. Note that the image level weighting map is a scalar, we set $mean(M_i(x^s))$ as 0.5.

After getting $\mathcal{H}_k(x^s)$, we average them to obtain the hierarchical weighing map $\mathcal{H}(x^s)$:

$$\mathcal{H}(x^s) = \frac{1}{3}(\mathcal{H}_p(x^s) + \mathcal{H}_r(x^s) + \mathcal{H}_i(x^s)), \tag{4}$$

which actually takes into account the local and global information for similarity-based transfer learning. Hence, the segmentation loss L^s_{Seg} for source domain is formulated as a weighted cross-entropy loss:

$$L^s_{\text{Seg}} = -\sum y^s(\mathcal{H}(x^s) \odot \log \mathcal{F}(x^s)), \tag{5}$$

where y^s is the ground truth label and $\odot$ denotes the element-wise multiplication. The total segmentation loss L_{Seg} is the sum of L^s_{Seg} and L^t_{Seg}, *i.e.* $L_{\text{Seg}} = L^t_{\text{Seg}} + L^s_{\text{Seg}}$.

Shared Weighting Map vs. Multi-channel Weighting Map. Note that the weighting map mentioned above is derived from the segmentation label map that has various channels (corresponding to class categories — 19 channels in this paper). Since the label structures vary a lot for different classes, their weighting mechanisms can also differ. Thus besides learning a shared weighting map by $W^1_{k\in p,r,i}$ for all classes, we can also learn separate weighting maps by $W^{19}_{k\in p,r,i}$ for each class channel. We implement both types of weighting maps and investigate the effectiveness of them under various settings.

Improving Expressiveness across Domains In our hierarchical transfer learning framework, we rely on a good feature encoder $E(\cdot)$ to generate the domain-variant features for potential different domain images. To improve the expressiveness of $E(\cdot)$ for better transfer learning, we inherit adversarial adaptation methods [4, 11, 34] to attach a generative adversarial network (GAN) to the encoded features. The goal is to drive the representations of source image close to the distribution of target image, which is supervised by a reconstruction loss and an adversarial loss shown as follows. This part does aid the domain adaptation and ablation study indicates its effectiveness.

3.2. Network Optimization

There are several learnable components in our model, including encoder $E(\cdot)$, discriminator $D(\cdot)$, generator $G(\cdot)$ and weighting networks $W(\cdot)$ (Note that $S(\cdot)$ has no learnable parameters). An alternative update is applied during the network optimization, which is illustrated in Algorithm 1. Except L_{Seg} and $L_{W_{k\in p,r,i}}$, reconstruction loss L_{Rec}

and adversarial loss L_{Adv} are also used during training and shown as follow.

Reconstruction loss L_{Rec} We use Conv5 features from the encoder $E(\cdot)$, and attach a generator $G(\cdot)$ to reconstruct each input image. The reconstruction loss L_{Rec} is defined as an L_1 loss in pixel space. The detailed architecture of the encoder and generator are shown in Section 3.3.

Adversarial loss L_{Adv} We also follow the adversarial strategy [20] to use a discriminator $D(\cdot)$ to promote the fidelity of reconstructed images. The generator $G(\cdot)$ and discriminator $D(\cdot)$ are alternately trained by adversarial loss L_{Adv} as in a min-max game. In this way, we encourage the Encoder $E(\cdot)$ to generate domain-invariant feature representations which could fool the discriminator.

Algorithm 1 The proposed hierarchical transfer learning method

Input: source domain X_s and target domain X_t; N is the number of iterations.
Initialization: Initialize hierarchical weighting networks W, generator $G(\cdot)$ and discriminator $D(\cdot)$ from scratch. Encoder $E(\cdot)$ is initialized with ImageNet-pretrained model.

```
repeat
    $\{x^s, y^s\} \leftarrow$ random image pair from source domain
    $\{x^t, y^t\} \leftarrow$ random image pair from target domain
    Generate predictions for both $x^s$ and $x^t$
    $\mathcal{H}(x^s) \leftarrow$ generate the hierarchical weighting map for source image by Eq. (4)
    $L_{\text{Seg}} \leftarrow$ compute segmentation loss for target and source image by Eq. (2) and Eq. (5)
    $E \leftarrow \min L_{\text{Seg}} + L_{\text{Adv}}$
    $W_{k\in p,r,i} \leftarrow \min L_{W_{k\in p,r,i}}$, Eq. (1)
    $G \leftarrow \min L_{\text{Rec}} + L_{\text{Adv}}$
    $D \leftarrow \min L_{\text{Adv}}$
until $N$
```

3.3. Network Architecture

Hierarchical weighting networks It consists of 5 convolution layers with kernel 4×4 and stride of 2 followed by a Leaky-ReLU with parameter 0.2 except for the last layer. The number of channels is 64, 128, 256, 512, for the respective convolutional layer. An up-sampling layer is attached after the pixel-level weighting network to resize the output to the original dimension.

Segmentation network We use FCN8s [22] as the semantic segmentation model. The backbone is VGG16 [28] which is pretrained on the ImageNet dataset. We divide the network into encoder $E(\cdot)$ and segmentation classifier $S(\cdot)$ ($S(\cdot)$ has no learnable parameters).

Generative adversarial network We apply the PatchGAN [18] as the discriminator, which tries to classify overlapping image patches as real or fake. The generator is composed of 2 Residual blocks and 7 convolutional layers. The kernel size, stride and padding of the first 6 convolutional layers are respectively 3×3, 2, and 1, while the last layer has 1×1, 1, and 1. The discriminator contains 7 convolutional layers with kernel size, stride and padding being 3×3, 2, and 1, respectively. Leaky-ReLU layers with parameterized 0.01 are adopted for the first 6 convolutional layers.

4. Experiments

In this paper, three datasets are employed in our experiments, including two synthetic datasets GTAV [24] and SYNTHIA [25], and one real-world dataset CITYSCAPES [7].

GTAV has 24,966 urban scene images rendered by the gaming engine GTAV. The semantic categories are compatible with the CITYSCAPES dataset. We take the whole GTAV dataset with labels as the source domain data.

SYNTHIA is a large dataset which contains different video sequences rendered from a virtual city. We take SYNTHIA-RAND-CITYSCAPES as the source domain data which provides 9,400 images from all the sequences with CITYSCAPES-compatible annotations.

CITYSCAPES is a real-world image dataset focused on the urban scene, which consists of 2,975 images in training set and 500 images for validation. The resolution of images is 2048 × 1024 and 19 semantic categories are provided with pixel-level labels. We take the whole training set as the target domain data. The results of our transfer learning scheme are reported on the validation set.

It can be found that both synthetic datasets consist of a large amount of images and the real-world dataset is much small. Thus it is well motivated that synthetic data provides an appealing option for the issue of insufficient data.

Training Details Adam [17] optimization is applied with β_1=0.9 and β_2=0.999. The initial learning rate is 1e-4 and is decreased with polynomial decay with power of 0.9. Due to the GPU memory limitation, images used in our experiments are resized to 1024×512 and batch size is 1. Since the discriminator is easier to converge than generator, we slightly perturb the labels of discriminator during training.

4.1. Experimental Results

In this section, we provide a quantitative evaluation by performing multiple joint learning experiments, i.e., GTAV + CITYSCAPES → CITYSCAPES, SYNTHIA + CITYSCAPES → CITYSCAPES and GTAV + SYNTHIA + CITYSCAPES → CITYSCAPES. More experimental experiments is contained in the supplementary material.

Method	Backbone	Setting	Mean IoU
Swami *et al.* [26]	VGG16	Un-	37.1%
CL [34]	VGG16	Un-	38.1%
ROAD [6]	VGG16	Un-	35.9%
Hung *et al.* [16]	ResNet-101	Semi-	67.7%
FCN	VGG16	-	65.3%
Direct Joint Training	VGG16	Joint-	64.6%
Target Finetuning	VGG16	Joint-	66.0%
FCN+GAN	VGG16	Joint-	64.0%
FCN+0-1 Conf. Mask	VGG16	Joint-	63.7%
FCN+Focal Loss [19]	VGG16	Joint-	66.2%
FCN+W^1	VGG16	Joint-	66.5%
PixelDA [4]	VGG16	Joint-	66.1%
Ours with W^1	VGG16	Joint-	**67.6%**
Ours with W^{19}	VGG16	Joint-	**68.1%**

Table 1. Experimental results of transfer learning using GTAV and CITYSCAPES (GTAV + CITYSCAPES → CITYSCAPES). W^1 and W^{19} denote our shared and multi-channel weighting mechanisms, respectively. Un-, Semi- and Joint- are the abbreviations of unsupervised domain adaptation, semi-supervised learning and joint learning. * means the model is trained on CITYSCAPES dataset without source datasets.

Several baseline methods are defined in the following: **1) Direct Joint Training:** we directly combine both synthetic and real-world data. **2) Target Finetuning:** the model is pretrained with the synthetic data and then finetuned using the real-world data. **3) FCN+GAN:** to verify the effect of GAN, we design a model only containing FCN segmentation part and GAN part. The VGG16 is adopted as the backbone. **4) PixelDA [4]:** since this work is an unsupervised domain adaptation method which is not compatible with our setting, we extend it to our problem by giving the label of both synthetic and real-world data. The segmentation network uses FCN8s with the backbone of VGG16. **5) FCN+W^1:** to verify the effect of weighting networks, we design a model only containing FCN segmentation part and W^1 part. The VGG16 is adopted as the backbone.

We also compare with other methods, which focus on the unsupervised domain adaptation [6, 26, 34] and semi-supervised learning [16], to demonstrate the superiority of our learning scheme.

GTAV + CITYSCAPES → CITYSCAPES. In this experiment, we use the GTAV as our source dataset, and CITYSCAPES as our target dataset. As shown in Table 1, our model achieves better performance (mIoU = **68.1**) compared with baselines. Several conventional methods for joint learning and finetuning perform comparable results with original FCN, which indicates that direct training with synthetic data does not benefit the model on real domain. Compared with other transfer learning setting, including unsupervised domain adaptation [6, 26, 34] and semi-supervised learning [16], our weighting network learning from both domains achieves better performance without introducing any extra cost. By comparing **FCN+W^1** to **FCN+GAN**, we can find that the proposed hierarchical weighting network is more crucial than GAN, which indicates the effectiveness of weighting network for transfer learning. PixelDA [4] learns a pixel-space transformation, which achieves an improvement of 2.1 compared with FCN+GAN. Comparing to two weighting baselines, including 0-1 Confidence Mask and Focal loss [19], our method achieves better performance. By incorporating the weighting network to selectively learn from the synthetic pixel, our proposed method is more effective to mine the knowledge from both domains. The results demonstrate that it performs better than existing methods, with an improvement of 4.1 over FCN+GAN and 2.0 over [4].

Method	Setting	Mean IoU
Swami *et al.* [26]	Un-	34.8%
CL [34]	Un-	34.2%
ROAD [6]	Un-	36.2%
FCN	-	65.3%
Direct Joint Training	Joint-	62.9%
Target Finetuning	Joint-	64.8%
FCN+GAN	Joint-	62.6%
PixelDA [4]	Joint-	64.0%
Ours with W^1	Joint-	**66.3%**
Ours with W^{19}	Joint-	**66.8%**

Table 2. Experimental results of joint learning using SYNTHIA and CITYSCAPES.

Method	Setting	Mean IoU
FCN	-	65.3%
Direct Joint Training	Joint-	64.2%
Target Finetuning	Joint-	66.5%
FCN+GAN	Joint-	64.9%
PixelDA [4]	Joint-	65.3%
Ours with W^1	Joint-	**68.2%**
Ours with W^{19}	Joint-	**68.8%**

Table 3. Experimental results of joint learning using GTAV, SYNTHIA and CITYSCAPES.

SYNTHIA + CITYSCAPES → CITYSCAPES. We follow the experiment setting as previous and choose 19 classes as the label in both SYNTHIA and CITYSCAPES datasets. We report the results of joint learning using SYNTHIA and CITYSCAPES in Table 2. We find that similar conclusions could be reached from results. It is noted that due to the large domain gap, direct joint training worsens the results of original FCN. The multi-channel weighting map (Ours with W^1) shows better performance than the shared weighting maps, while both methods have a significant improvement than the baseline method.

Method	Single	Hier	Mean IoU
FCN			65.3%
Ours with W^1	$\surd$		66.7%
Ours with W^1		$\surd$	67.6%
Ours with W^{19}	$\surd$		66.9%
Ours with W^{19}		$\surd$	68.1%

Table 4. Ablation experiment of single VS hierarchical weighting network using GTAV + CITYSCAPES → CITYSCAPES.

GTAV + SYNTHIA + CITYSCAPES → CITYSCAPES. To verify the robustness of our model, we design this joint learning experiments with multiple synthetic datasets and single real-world dataset. In this experiment, the model is trained by fist using GTAV as the source dataset and then using GTAV + SYN as the source dataset. As shown in Table 3, with multiple synthetic datasets, our proposed model is able to consistently achieve the better performance, which demonstrates its robustness and high flexibility in the complicated settings. In such setting, PixelDA [4] even delivers a worse result (mIoU=65.3) compared with the result in Table 1 (mIoU=66.1) , which indicates that learning from synthetic data without selection might bias to the source domain and possess limited robustness when processing large amount of labeled synthetic data from multiple sources. Therefore, compared with PixelDA [4], our method has a performance gain of 3.5 points, validating the effectiveness of the proposed weighting networks. By combining the GTAV and SYNTHIA datasets as the source domain, our model obtains 0.6 and 0.7 points improvement respectively comparing with the performance of the model trainined on the single dataset, which shows that the knowledge mined by our method focuses on the similarity of the target domain and they can promote each other when they are combined together.

4.2. Ablation Studies

In this section, we perform the ablation experiment to verify the effect of hierarchical-level weighting network. The ablation experiment is conducted on the GTAV dataset and CITYSCAPES dataset. We compare the hierarchical weighting networks with single-level (pixel-level only) weighting network. In the Table 4, it can be observed that the hierarchical mechanism performs consistently better (0.9 and 1.2 gain for W^1 and W^{19}, respectively) than the single pixel-wise mechanism, which demonstrates that the hierarchical weighting network possessing both local and global information enhances the semantic segmentation.

4.3. Discussion

In this section, we design several experiments to verify the capability of our model. We first visualize the weighting maps generated by hierarchical weighting network to display how our model measures the similar regions and then we provide the visualization of segmentation results using different methods. Finally, we randomly sample 1/4, 1/2 target images to investigate the effectiveness of our method.

Data Amount	1/8	1/4	1/2	Full
Mean IoU	53.4%	57.7%	64.9%	68.1%

Table 5. Experimental results of GTAV + CITYSCAPES → CITYSCAPES using different amounts of real-world images. Note that we use the proposed model, *i.e.* Ours with W^{19}, and all synthetic data from GTAV are used during training.

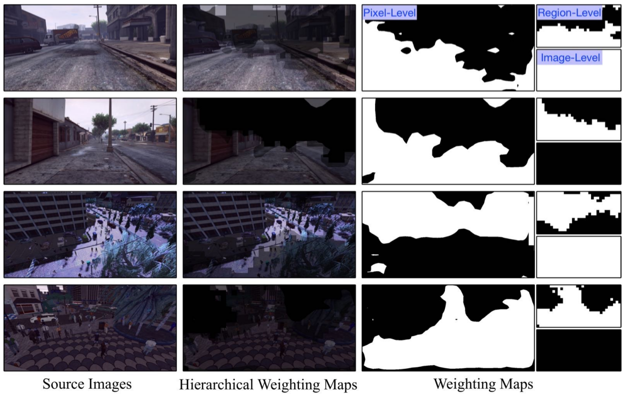


Figure 4. Shared weighting maps generated by W^1 (i.e., shared weighting mechanism). The first two rows are images sampled from the GTAV dataset, while the last two rows are from the SYNTHIA dataset. From left to right: the input image, the input image overlaid with hierarchical weighting map, the pixel-, region- and image-level weighting maps.

Visualization of Weighting Maps and Segmentation Results. As shown in Fig 4, we display the weighting maps generated by the W^1 strategy. From these weighting maps, it can be observed that weighting maps often cover the road region and ignore the building part, in which the road is most similar and dominant region between synthetic data and real-world data while the building is irrelevant and indifferent in the segmentation of driving scene.

Visualization of Segmentation Results. We show segmentation results obtained from different models in Fig 5 using GTAV+CITYSCAPES → CITYSCAPES. Compared with original FCN and FCN + GAN, our full model performs much better in terms of details and boundary, such as the lane boundary and the outline of car. The noises in the building and the road are decreased. These improvements demonstrate the effectiveness of weighting networks by focusing on learning from the most similar regions.

Analysis of Extremely Insufficient Data. To further explore the capability of our model, we design the experiment with extremely insufficient real-world data using GTAV+CITYSCAPES → CITYSCAPES. All images in the GTAV dataset are adopted, while different numbers of real-world images are randomly sampled for our transfer learn-

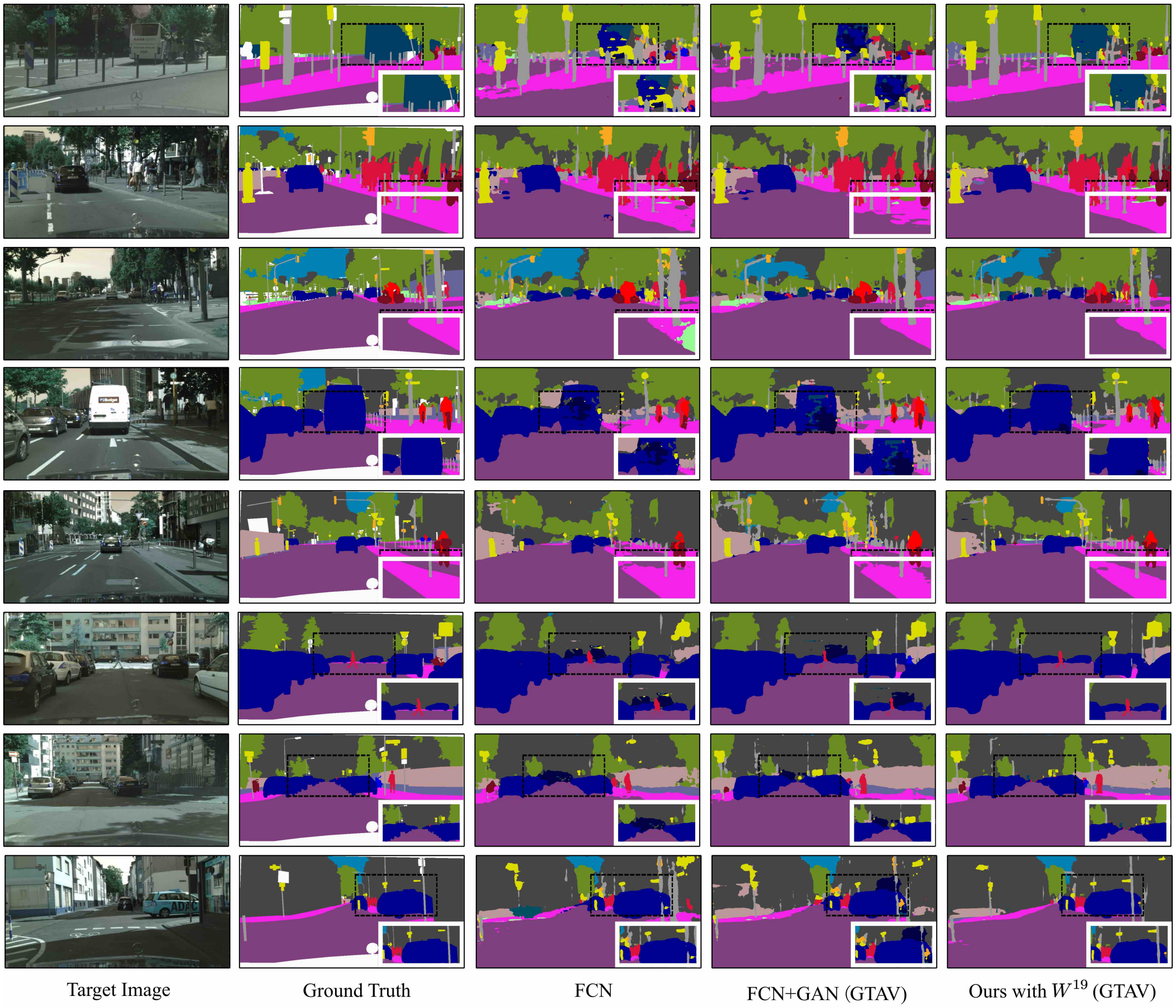


Figure 5. We show the segmentation results of different models. From left to right, the images are extracted from Target Image, Ground Truth, FCN, FCN+GAN, Ours with W^1. Our full model achieves the better results with more detailed boundary.

ing scheme. As shown in Table 5, our model using 1/2 real-world images achieves comparable performance against the FCN+GAN, and only 4% worse than the model using full real-world images, which demonstrates that our model is capable and applicable with the extremely insufficient data.

5. Conclusion

In this paper, we introduce a new transfer learning method with both real and synthetic images for semantic segmentation. We mitigate the domain gap between insufficient real data and abundant synthetic data by adaptively selecting similar synthetic pixels for learning. Hierarchical weighting networks are used to score how similar the synthetic pixels are to real ones, at pixel-, region- and image-levels respectively, which helps us adapt to target real images. Also, we learn weighting networks and segmentation network jointly in an end-to-end manner. Extensive experiments demonstrate that our proposed method outperforms other important baselines by large margins, espetially, multiple source datasets achieves more improvements in both W^1 and W^{19} strategy. Our method can also learn from extremely limited real images, and show the potential to learn from multiple data sources.

Acknowledgement

This paper is carried out at SenseTime Research in Beijing, China and is partially funded by National Social Science Foundation of China No. 18ZD22 and Science and Technology Commission of Pudong Municipality Program PKJ2018-Y46.

References

[1] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence(PAMI)*, pages 1–1, 2015.

[2] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 36–45, 2015.

[3] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1-2):151–175, 2010.

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

[5] Liang Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *arXiv preprint arXiv:1802.02611*, 2018.

[6] 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, pp. 7892-7901*, 2018.

[7] 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 *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3213–3223, 2016.

[8] Wenyuan Dai, Qiang Yang, Gui Rong Xue, and Yong Yu. Boosting for transfer learning. In *IEEE International Conference on Machine Learning(ICML)*, pages 193–200, 2007.

[9] German Ros Felipe Codevilla Antonio Lopez Dosovitskiy, Alexey and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2018.

[10] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *IEEE International Conference on Machine Learning(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. *The Journal of Machine Learning Research(JMLR)*, 17(1):2096–2030, 2016.

[12] Weifeng Ge and Yizhou Yu. Borrowing treasures from the wealthy: Deep transfer learning through selective joint finetuning. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, volume 6, 2017.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis Machine Intelligence(TPAMI)*, PP(99):1–1, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 770–778, 2016.

[15] Seunghoon Hong, Junhyuk Oh, Honglak Lee, and Bohyung Han. Learning transferrable knowledge for semantic segmentation with deep convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3204–3212, 2016.

[16] Wei-Chih Hung, Yi-Hsuan Tsai, Yan-Ting Liou, Yen-Yu Lin, and Ming-Hsuan Yang. Adversarial learning for semi-supervised semantic segmentation. *arXiv preprint arXiv:1802.07934*, 2018.

[17] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014.

[18] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, pages 702–716. Springer, 2016.

[19] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision(ICCV)*, pages 2980–2988, 2017.

[20] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016.

[21] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE international conference on computer vision*, pages 1377–1385, 2015.

[22] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015.

[23] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. *arXiv preprint arXiv:0902.3430*, 2009.

[24] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision(ECCV)*, pages 102–118. Springer, 2016.

[25] 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 *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 3234–3243, 2016.

[26] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] Nikolaos Sarafianos, Michalis Vrigkas, and Ioannis A. Kakadiaris. Adaptive svm+: Learning with privileged information for domain adaptation. In *IEEE International Conference on Computer Vision Workshop(ICCV Workshop)*, pages 2637–2644, 2017.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Riccardo Volpi, Pietro Morerio, Silvio Savarese, and Vittorio Murino. Adversarial feature augmentation for unsupervised domain adaptation. *arXiv preprint arXiv:1711.08561*, 2018.

[30] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. *IEEE International Conference on Computer Vision(ICCV)*, pages 1378–1387, 2017.

[31] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018.

[32] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 2881–2890, 2017.

[33] Xinge Zhu, Jiangmiao Pang, Ceyuan Yang, Jianping Shi, and Dahua Lin. Adapting object detectors via selective cross-domain alignment. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[34] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 568–583, 2018.