What can we do with *just* the model? A simple knowledge extraction framework

Sujoy Paul [1] **Ansh Khurana** [1] **Gaurav Aggarwal** [1]

Abstract

We consider the problem of adapting semantic segmentation models to new target domains, only from the trained source model, without the source data. Not only is this setting much harder than if one had access to the source data, this is necessary in many practical situations where source data is not available due to privacy and storage reasons. Our algorithm has two parts - first, we update that normalization statistics which helps to compensate for the distribution shift and second, we transfer knowledge from the source models adhering to certain equivariant and invariant transforms. The transforms helps to efficiently extract the knowledge beyond vanilla self-training. Through extensive experiments on multiple semantic segmentation tasks, we show how such a simple framework can be effective in extracting knowledge from the source model, for a variety of problem settings, and performs much better or at par with current state-of-the-art methods which are specifically tuned for the respective settings.

1. Introduction

Deep neural networks often fail to generalize across datasets even for the same task. It becomes quite challenging to acquire and annotate data for every new dataset, especially for dense prediction tasks. For example, a single image from the Cityscapes dataset required about 1.5 hours to annotate (Cordts et al., 2016). To avoid the requirement of annotated target data, Unsupervised Domain Adaptation (UDA) methods (Tzeng et al., 2017; Hoffman et al., 2018) adapt the models learned with labeled source data to the target dataset, without needing annotated target images. These approaches assume access to data from both the source and target for adaptation. Access to source data helps in grounding of source model resulting in a relatively easier adaptation task. On the other hand, we address a more restricted problem where we do not have access to any source data, but only a trained source model. This setting enables - 1) source-free adaptation (Yeh et al., 2021; Li et al., 2020b; Liu et al., 2021; Xia et al., 2021; Kundu et al., 2021; Huang et al., 2021), where privacy and storage reasons bar us from accessing the source data, 2) test-time adaptation (Sun et al., 2020; Mummadi et al., 2021; Wang et al., 2021; Khurana et al., 2021), where because of privacy as well as latency reasons we may not be able to use the source data, 3) multi-source adaptation (Gong et al., 2021; He et al., 2021; Zhao et al., 2019; 2021), where the challenge of accessing datasets increases multi-fold, and 4) semi-supervised learning (Hung et al., 2018; French et al., 2020; Mendel et al., 2020; Mittal et al., 2021; Olsson et al., 2021), where we have additional unlabeled source instances along with the source model trained only on the labeled samples. Regardless of the setting, our method is generic enough that even if we have access to the source data, it can be used in conjunction to our learning objective.

The **main contributions** of our work are:

- Our self-training based knowledge transfer approach involving equivariant and invariant transforms, works much better than vanilla self-training for source-free adaptation. It also works well for the case where we have access to multiple source models, which has never been explored in literature for semantic segmentation.
- Compensating for distribution shifts by normalization parameter updates helps in test-time adaptation as well as obtaining a better model for source-free adaptation.
- We perform extensive experiments and ablation studies across a variety of tasks to portray the generalisability and efficacy of our approach (Figure 1).

2. Related Works

Source-free adaptation for semantic segmentation has only recently gained attention. These works use pseudo-labeling (Liu et al., 2021; You et al., 2021; Ye et al., 2021; Kundu et al., 2021; Huang et al., 2021), domain alignment using

[1] Google Research, India. Correspondence to: Sujoy Paul <sujoyp@google.com>.

Presented at ICML Workshop on Principles of Distribution Shift 2022,

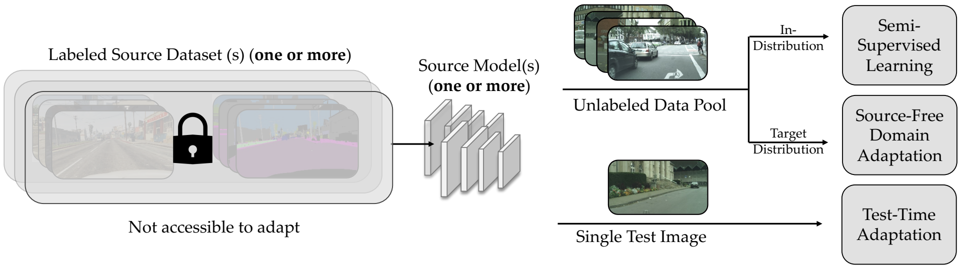

Figure 1. **Overview.** Knowledge extraction from one/more source models without accessing source data at all. This setting enables source-free adaptation from single or multiple models, test-time adaptation from a single test image and *phased* semi-supervised learning.

discriminators (Ye et al., 2021; Liu et al., 2021), target image generation (Liu et al., 2021), and contrastive learning (Huang et al., 2021). Source free adaptation is also observed in the related, but more challenging setting of test-time adaptation. Here, instead of adapting to a pool of target data, the goal is to adapt the model at test-time with only the current test samples. TENT (Wang et al., 2021) uses the entropy loss as self-supervision from the model's prediction head itself to modulate the batch normalisation affine parameters. (Mummadi et al., 2021) builds upon TENT and introduces likelihood ratio based losses as an improvement over the entropy loss. A few approaches (Nado et al., 2020; Schneider et al., 2020; Khurana et al., 2021) propose to combine source statistics with current target statistics to adapt the model at test-time. Existing works on semi-supervised learning assume access to both the labeled and unlabeled data during training as compared to our *phased* setting, where we train a source model on the labeled data, and the update it using only the unlabeled samples, assuming no access to the labeled samples. We provide a more detailed discussion of related works in Appendix A.

3. Domain Adaptation from Source Models

The setting can be formally defined as follows. Consider that we have a source model for semantic segmentation, $\mathbf{M_s} : \mathbb{R}^{H\times W\times 3} \rightarrow \mathbb{R}^{H\times W\times C}$, which takes as input an RGB image and predicts a probability mass function over the possible categories. H, W are the height and width of the image and C is the number of categories. We only have access to the source model but not the source data. The test instances originate from another domain (target), different from the source. We are given an unlabeled target dataset, $\mathcal{D}_t = \{X_i\}_{i=1}^n$, to adapt the source model $\mathbf{M_s}$ to a new model $\mathbf{M_t}$, such that it performs better on samples drawn from the target distribution, than directly using the source model. In the case of multi-source adaptation, we have multiple source models $\{\mathbf{M}_s^i\}_{i=1}^{n_s}$ to adapt from (n_s is the number of source models). For test-time adaptation, we do not have the dataset $\mathcal{D}_t$, but rather have to adapt to every test instance. We consider the extreme case of test-time adaptation, i.e., adaptation to single images (Khurana et al., 2021), in an episodic setting, where we do not update the model online on the test instances, but rather adapt to every instance starting from the given source model only. We also consider *phased* semi-supervised learning (SSL) setting, where we want to improve the given source model $\mathbf{M_s}$ using additional unlabeled samples from the source domain.

3.1. Bridging the gap via statistics updates

BatchNorm (Ioffe & Szegedy, 2015) and InstanceNorm (Ulyanov et al., 2016) are commonly used in deep neural networks to avoid over-fitting, stabilize training, and faster convergence. For every activation layer in the network, these normalization methods estimate two parameters during training, viz., the mean and variance, which are then used in the testing phase. However, these parameters are only a good estimate for the images in the training distribution. Thus, in domain adaptation, where the target data may belong to a different distribution than the source, it may be useful to first update the mean and variance of the normalization layers, using the unlabeled target data. We choose a simple norm statistics update formulation, which is a convex combination of the source statistics and the target statistics, as follows:

$$\begin{aligned}\mu &\leftarrow \lambda\mu_s + (1-\lambda)\mu_t \\ \sigma^2 &\leftarrow \lambda\sigma_s^2 + (1-\lambda)\sigma_t^2\end{aligned} \quad (1)$$

$\{\mu_s, \sigma_s^2\}$ are the source model's normalization statistics, whereas $\{\mu_t, \sigma_t^2\}$ are the statistics which are estimated from the target data. This formulation is general enough to be used for both test-time adaptation as well as source-free domain adaptation. In **source-free adaptation**, as we have an entire dataset of unlabeled samples from the target domain, we can estimate its statistics by just forward propagating the samples through the network. Moreover, in this case, we also set the weight on the prior to be $\lambda = 0$, as we have enough samples to estimate the target statistics. In **test-time adaptation**, we only have a single test image to adapt, which may not be enough to get an accurate estimate of the statistics. Thus, we keep the prior value to be high $\lambda = 0.8$, and update it with the other parameters of the network using the losses described next.

3.2. Self-Training with Transforms

Given the normalization updated source model $\mathbf{M}_s$, we use it to label the target images via confidence-filtered pseudo labeling. Additionally, we impose consistency using certain transforms, which gives the model an opportunity to improve beyond learning from just the pseudo-labeled dataset.

Pseudo-Labeling: The pseudo-labeling function $\hat{Y} = \mathbf{PL}(\mathbf{M}(X))$ to label an image X with a model $\mathbf{M}$ can be defined as follows:

$$\hat{Y}^{x,y} = \begin{cases} j^* = \arg\max_j [\mathbf{M}(X)]^{x,y}, & \text{if} \max_j [\mathbf{M}(X)]^{x,y} > p_{j^*} \\ \text{No Label}, & \text{otherwise} \end{cases} \tag{2}$$

where x, y, represents the spatial co-ordinates, $j \in \{1, \ldots, C\}$, and $[\mathbf{M}(X)]^{x,y}$ is C-dimensional summing to 1, representing the pixel's prediction. p_j is the threshold for prediction confidence of label j, which we set to $\min(0.9, \text{median of label-wise confidence})$, where the median is computed over all pixel predictions for that label. We label the unlabeled target $\mathcal{D}_t$ using the pseudo-labeling function to obtain a new dataset $\hat{\mathcal{D}}_t = \{X_i, \hat{Y}_i\}_{i=1}^n$, where $\hat{Y}_i = \mathbf{PL}(\mathbf{M}_s(X_i))$.

Training new model $\mathbf{M}_t$ using the pseudo-labeled set $\hat{\mathcal{D}}_t$, reduces the uncertainties in the source model $\mathbf{M}_s$, offering a better hypothesis on the target. However, this only offers a limited room for improvement beyond the pseudo-label distribution. We observe that certain transforms on the target images actually lead to undesirable changes in the output. We build on top of this observation and enforce certain consistencies to transforms, as discussed next.

Equivariance. We observe that under certain transformations to the input image, the predicted segmentation maps do not appear as expected. This motivates us to use this as constraint in the learning process. For this type of transformation we want the network output to be equivariant. Formally, considering a transform $\mathbf{T} \in \mathcal{T}_e$, the consistency constraint can be expressed as,

$$\mathbf{M}_t(\mathbf{T}(X)) \approx \mathbf{T}(\mathbf{M}_t(X)) \tag{3}$$

Image mirroring and rotation fall under this transform set.

Invariance: In this case, we want the network output to be invariant to the transform. Formally, if we consider a transform $\mathbf{T} \in \mathcal{T}_i$, then the consistency constraint can be expressed as

$$\mathbf{M}_t(\mathbf{T}(X)) \approx \mathbf{M}_t(X) \tag{4}$$

Drop-block and Gaussian blur belong to this transform.

Table 1. Results of adapting GTA5 to Cityscapes. The top group are methods which use source data during adaptation, while the bottom group do not use any source data to adapt.

Source	Method	Stuff	Things	mIoU
Yes	BDL (Li et al., 2019)	61.7	38.9	48.5
	CAG (Zhang et al., 2019)	61.2	42.0	50.2
	WeakDA (Paul et al., 2020)	61.0	38.8	48.2
	Stuff (Wang et al., 2020)	62.1	39.9	49.2
	FDA (Yang & Soatto, 2020)	60.3	43.2	50.4
	SAC (Araslanov & Roth, 2021)	64.3	46.1	53.8
No	Source	45.6	31.6	37.5
	URMA (S & Fleuret, 2021)	60.3	34.0	45.1
	LD(You et al., 2021)	60.1	34.9	45.5
	SFUDA (Ye et al., 2021)	60.5	41.3	49.4
	Ours	59.0	41.3	48.8

Please refer to Appendix E for more details on the transformations used.

Learning from Collages. We further combine two images into a single collage in the spatial domain. This has similarities with Mixup (Zhang et al., 2017a) that uses weighted combination of image pairs and their labels to regularize model training. Detailed explanation for this collaging process is provided in Appendix B. In our algorithm, we use these collages in place of original images, and hence X denotes these collage images.

For details specific to each task, please refer to Appendix D.

4. Experiments

Datasets: We evaluate our framework on three combinations of source → target datasets covering both outdoor and indoor scenes. For outdoor, we evaluate on GTA5 (Richter et al., 2016) → Cityscapes (Cordts et al., 2016) and SYNTHIA (Ros et al., 2016) → Cityscapes. For indoor, we use SceneNet (McCormac et al., 2017) → SUN (Song et al., 2015). Please refer to Appendix C for more details.

Implementation Details: To have a fair comparison with the works in literature, we use the Deeplab-V2 (Chen et al., 2017a) with ResNet-101 (He et al., 2016) as the network. For more details, please refer to Appendix F.

Table 2. Results of adapting SceneNet to SUN. The top row group does not use any source data to adapt, while the bottom row uses full supervision on the target images.

Method	Stuff	Things	mIoU
Source	34.4	19.7	26.5
Ours	41.1	30.9	35.6
Full Supervision	54.9	45.5	49.8

Source-Free Adaptation: We first compare our method with state-of-the-art in literature in Table 1 for GTA5 →

Table 3. Semi-supervised learning on Cityscapes (five random splits).

Source Data →	Yes					No		
Labeled Samples	Hung et al.	CutMix	ECS	Mittal et al.	ClassMix	Source	Pseudo-Label	**Ours**
1/8	58.8	60.3 ± 1.2	60.3 ± 0.8	59.3	61.3	59.2 ± 0.8	60.7 ± 0.8	62.0 ± 1.0
1/4	62.3	63.9 ± 0.7	63.8 ± 0.7	61.9	63.6	61.5 ± 0.6	61.8 ± 0.6	63.4 ± 0.8
Full set	67.7	67.5	66.9	65.8	66.2	66.2	66.2	66.2

Cityscapes, and SceneNet → SUN in Table 2. We present the results for SYNTHIA → Cityscapes in the Appendix G. Due to limited space, instead of presenting the category-wise performances, we club them into COCO-style Stuff and Things and present the category-wise results in Appendix G. From the tables we can also see that our method outperforms other source-free adaptation methods, even with just using a batch size of 1, whereas some baselines use batch size > 1 to train.

Table 4. Source-free adaption from multiple source models (GTA5+SYNTHIA → Cityscapes). Numbers in brackets are over 16 categories of SYNTHIA. '-' indicates results not available for 19 categories.

Source	Yes				No	
Method	DALU	MSDA-CL	MADAN	MADAN+	Pseudo-Label	**Ours**
mIoU	43.1 (46.8)	- (54.0)	- (45.4)	- (48.5)	45.6 (49.5)	48.5 (52.6)

Multi-source Adaptation: In this case, we have two source models trained on GTA5 and SYNTHIA, and adapt the model to Cityscapes (Table 4). We compare with DALU (Gong et al., 2021), MSDA-CL (He et al., 2021), MADAN (Zhao et al., 2019) and MADAN+ (Zhao et al., 2021), which also use DeepLabv2+ResNet-101. In multi-source domain adaptation, the challenge is to extract the useful information from the multiple models, and not getting negatively affected by some of the models which provide incorrect information. So, the performances are expected to be close to the best source model. E.g., in SYNTHIA, there is limited data on three categories - terrain, truck, and train and thus when adapted to Cityscapes, the model is not able to recognize those categories at all, and thus the performance on 19 categories is low. However, in multi-source domain adaptation, with the GTA5 model, our framework is able to identify which model to use for what information, and performs close to the best source model. When compared with state-of-the-art methods which uses source data to adapt, the performance obtained by our method is quite close, even without using any source data.

Table 5. Results for test-time adaptation with a single iteration of optimization at test time.

Datasets	Source	PTN	BN	TENT	Entropy	Likelihood	**Ours**
GTA5 → Cityscapes	37.5	36.7	40.4	37.6	38.4	38.3	43.5
SYNTHIA → Cityscapes	32.1	31.8	32.9	32.7	32.6	32.5	34.9
SceneNet → SUN	26.5	25.2	28.2	26.6	27.3	27.2	28.6

Test Time Adaptation (TTA): We perform test-time adaptation using the same source models as above on all the three datasets. We compare with TENT (Wang et al., 2021), prediction time normalization (PTN) (Nado et al., 2020), batch-normalization statistics update (BN) (Schneider et al., 2020) with their recommended prior value, as well as using the losses proposed in (Wang et al., 2021; Mummadi et al., 2021) (Table 5). We observe that a simple change in the batchnorm parameterization along with fine-tuning using the pseudo-labeling loss is enough to achieve much better performance than the source model and all other methods.

***Phased* Semi-Supervised Learning (SSL):** In this setting, we build source models on $1/8^{th}$ and $1/4^{th}$ split of the Cityscapes dataset, and then adapt the model on the the remaining unlabeled data using our framework. We compare with (Hung et al., 2018), CutMix (French et al., 2020), ECS (Mendel et al., 2020), (Mittal et al., 2021) and ClassMix (Olsson et al., 2021), which also use DeepLabv2+ResNet-101. As can be seen in Table 3, our method with the consistency constraints performs better than vanilla pseudo-labeling, and even better than many methods in literature which are designed specifically for SSL and use the labeled source data in the learning process. The last row presents the performance of the backbone in the fully supervised setting for indicating the upper-bound.

5. Conclusion

We propose a framework for adapting semantic segmentation models from source to target without access to the labeled source data. We present a self-training framework by enforcing consistencies with certain transforms to efficiently extract information from the source model. With only a few existing works in this challenging setting, our approach compares favorably against strong baselines. Empirical evaluation shows that our method performs comparable to many state-of-the-art methods that use source data to adapt. We further show the usefulness of our approach for fully test-time adaptation in which adaptation is done individually for each test image. Future works can explore extracting information such as shape priors, from the source and infusing them into the target model, for better adaptation.

References

Araslanov, N. and Roth, S. Self-supervised augmentation consistency for adapting semantic segmentation. In *CVPR*, 2021.

Chang, W.-L., Wang, H.-P., Peng, W.-H., and Chiu, W.-C. All about structure: Adapting structural information across domains for boosting semantic segmentation. In *CVPR*, 2019.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 2017a.

Chen, Y., Li, W., and Gool, L. V. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In *CVPR*, 2018.

Chen, Y.-H., Chen, W.-Y., Chen, Y.-T., Tsai, B.-C., Wang, Y.-C. F., and Sun, M. No more discrimination: Cross city adaptation of road scene segmenters. In *ICCV*, 2017b.

Choi, J., Kim, T., and Kim, C. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In *ICCV*, 2019.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

Dong, J., Cong, Y., Sun, G., Liu, Y., and Xu, X. Cscl: Critical semantic-consistent learning for unsupervised domain adaptation. In *ECCV*, 2020.

Du, L., Tan, J., Yang, H., Feng, J., Xue, X., Zheng, Q., Ye, X., and Zhang, X. Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In *ICCV*, 2019.

French, G., Laine, S., Aila, T., Mackiewicz, M., and Finlayson, G. D. Semi-supervised semantic segmentation needs strong, varied perturbations. In *BMVC*, 2020.

Gong, R., Dai, D., Chen, Y., Li, W., and Van Gool, L. mdalu: Multi-source domain adaptation and label unification with partial datasets. In *ICCV*, pp. 8876–8885, 2021.

He, J., Jia, X., Chen, S., and Liu, J. Multi-source domain adaptation with collaborative learning for semantic segmentation. In *CVPR*, pp. 11008–11017, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hoffman, J., Wang, D., Yu, F., and Darrell, T. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016.

Hoffman, J., Tzeng, E., Park, T., Zhu, J.-Y., Isola, P., Saenko, K., Efros, A. A., and Darrell, T. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018.

Huang, J., Lu, S., Guan, D., and Zhang, X. Contextual-relation consistent domain adaptation for semantic segmentation. In *ECCV*, 2020.

Huang, J., Guan, D., Xiao, A., and Lu, S. Model adaptation: Historical contrastive learning for unsupervised domain adaptation without source data. In *NeurIPS*, 2021.

Hung, W.-C., Tsai, Y.-H., Liou, Y.-T., Lin, Y.-Y., and Yang, M.-H. Adversarial learning for semi-supervised semantic segmentation. In *BMVC*, 2018.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Khurana, A., Paul, S., Rai, P., Biswas, S., and Aggarwal, G. SITA: single image test-time adaptation. *arXiv preprint arXiv:2112.02354*, 2021.

Kim, M. and Byun, H. Learning texture invariant representation for domain adaptation of semantic segmentation. In *CVPR*, 2020.

Kundu, J. N., Kulkarni, A., Singh, A., Jampani, V., and Babu, R. V. Generalize then adapt: Source-free domain adaptive semantic segmentation. In *ICCV*, 2021.

Li, G., Kang, G., Liu, W., Wei, Y., and Yang, Y. Content-consistent matching for domain adaptive semantic segmentation. In *ECCV*, 2020a.

Li, R., Jiao, Q., Cao, W., Wong, H.-S., and Wu, S. Model adaptation: Unsupervised domain adaptation without source data. In *CVPR*, pp. 9641–9650, 2020b.

Li, Y., Yuan, L., and Vasconcelos, N. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019.

Lian, Q., Lv, F., Duan, L., and Gong, B. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In *ICCV*, 2019.

Liang, J., Hu, D., and Feng, J. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In *ICML*, 2020.

Liu, Y., Zhang, W., and Wang, J. Source-free domain adaptation for semantic segmentation. In *CVPR*, 2021.

Lv, F., Liang, T., Chen, X., and Lin, G. Cross-domain semantic segmentation via domain-invariant interactive relation transfer. In *CVPR*, 2020.

McCormac, J., Handa, A., Leutenegger, S., and Davison, A. J. Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. *arXiv preprint arXiv:1612.05079*, 2016.

McCormac, J., Handa, A., Leutenegger, S., and Davison, A. J. Scenenet rgb-d: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *ICCV*, 2017.

Mei, K., Zhu, C., Zou, J., and Zhang, S. Instance adaptive self-training for unsupervised domain adaptation. *ECCV*, 2020.

Mendel, R., Souza, L. A. D., Rauber, D., Papa, J. P., and Palm, C. Semi-supervised segmentation based on error-correcting supervision. In *ECCV*, 2020.

Mittal, S., Tatarchenko, M., and Brox, T. Semi-supervised semantic segmentation with high- and low-level consistency. *PAMI*, 2021.

Mummadi, C. K., Hutmacher, R., Rambach, K., Levinkov, E., Brox, T., and Metzen, J. H. Test-time adaptation to distribution shift by confidence maximization and input transformation. *arXiv preprint arXiv:2106.14999*, 2021.

Murez, Z., Kolouri, S., Kriegman, D., Ramamoorthi, R., and Kim, K. Image to image translation for domain adaptation. In *CVPR*, 2018.

Musto, L. and Zinelli, A. Semantically adaptive image-to-image translation for domain adaptation of semantic segmentation. *BMVC*, 2020.

Nado, Z., Padhy, S., Sculley, D., D'Amour, A., Lakshminarayanan, B., and Snoek, J. Evaluating prediction-time batch normalization for robustness under covariate shift. *CoRR*, 2020.

Olsson, V., Tranheden, W., Pinto, J., and Svensson, L. Classmix: Segmentation-based data augmentation for semi-supervised learning. In *WACV*, 2021.

Pan, F., Shin, I., Rameau, F., Lee, S., and Kweon, I. S. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In *CVPR*, 2020.

Paul, S., Tsai, Y.-H., Schulter, S., Roy-Chowdhury, A. K., and Chandraker, M. Domain adaptive semantic segmentation using weak labels. *ECCV*, 2020.

Richter, S. R., Vineet, V., Roth, S., and Koltun, V. Playing for data: Ground truth from computer games. In *ECCV*, 2016.

Ros, G., Sellart, L., Materzynska, J., Vazquez, D., and Lopez, A. M. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016.

S, P. T. and Fleuret, F. Uncertainty reduction for model adaptation in semantic segmentation. In *CVPR*, 2021.

Saleh, F. S., Aliakbarian, M. S., Salzmann, M., Petersson, L., and Alvarez, J. M. Effective use of synthetic data for urban scene semantic segmentation. In *ECCV*, 2018.

Schneider, S., Rusak, E., Eck, L., Bringmann, O., Brendel, W., and Bethge, M. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020.

Shin, I., Woo, S., Pan, F., and Kweon, I. S. Two-phase pseudo label densification for self-training based domain adaptation. In *ECCV*, 2020.

Song, S., Lichtenberg, S. P., and Xiao, J. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

Subhani, M. N. and Ali, M. Learning from scale-invariant examples for domain adaptation in semantic segmentation. *ECCV*, 2020.

Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., and Hardt, M. Test-time training with self-supervision for generalization under distribution shifts. In *ICML*, 2020.

Tsai, Y.-H., Hung, W.-C., Schulter, S., Sohn, K., Yang, M.-H., and Chandraker, M. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018.

Tsai, Y.-H., Sohn, K., Schulter, S., and Chandraker, M. Domain adaptation for structured output via discriminative patch representations. In *ICCV*, 2019.

Tzeng, E., Hoffman, J., Saenko, K., and Darrell, T. Adversarial discriminative domain adaptation. In *CVPR*, 2017.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

Vu, T.-H., Jain, H., Bucher, M., Cord, M., and Pérez, P. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. In *ICLR*, 2021.

Wang, Z., Yu, M., Wei, Y., Feris, R., Xiong, J., Hwu, W.-m., Huang, T. S., and Shi, H. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In *CVPR*, 2020.

Wu, Z., Han, X., Lin, Y.-L., Uzunbas, M. G., Goldstein, T., Lim, S. N., and Davis, L. S. Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation. In *ECCV*, 2018.

Xia, H., Zhao, H., and Ding, Z. Adaptive adversarial network for source-free domain adaptation. In *ICCV*, 2021.

Yang, Y. and Soatto, S. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020.

Ye, M., Zhang, J., Ouyang, J., and Yuan, D. Source data-free unsupervised domain adaptation for semantic segmentation. In *ACM-MM*, pp. 2233–2242, 2021.

Yeh, H.-W., Yang, B., Yuen, P. C., and Harada, T. Sofa: Source-data-free feature alignment for unsupervised domain adaptation. In *WACV*, 2021.

You, F., Li, J., Zhu, L., Chen, Z., and Huang, Z. Domain adaptive semantic segmentation without source data. In *ACM-MM*, 2021.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. *ICLR*, 2017a.

Zhang, Q., Zhang, J., Liu, W., and Tao, D. Category anchor-guided unsupervised domain adaptation for semantic segmentation. *NeurIPS*, 2019.

Zhang, Y., David, P., and Gong, B. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017b.

Zhang, Y., Qiu, Z., Yao, T., Liu, D., and Mei, T. Fully convolutional adaptation networks for semantic segmentation. In *CVPR*, 2018.

Zhao, S., Li, B., Yue, X., Gu, Y., Xu, P., Hu, R., Chai, H., and Keutzer, K. Multi-source domain adaptation for semantic segmentation. *NeurIPS*, 32, 2019.

Zhao, S., Li, B., Xu, P., Yue, X., Ding, G., and Keutzer, K. Madan: multi-source adversarial domain aggregation network for domain adaptation. *IJCV*, 129(8):2399–2424, 2021.

Zou, Y., Yu, Z., Kumar, B. V. K. V., and Wang, J. Domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018.

A. Related Works

In this paper, we delve into three main problem settings where our generic knowledge transfer method would be useful - source-free adaptation, test-time adaptation, and semi-supervised learning. We discuss works related to these problems below.

Source-Free Domain Adaptation. In semantic segmentation, existing UDA methods can be categorized primarily into three groups: output alignment (Tsai et al., 2018; Chen et al., 2018; 2017b; Hoffman et al., 2016; Zhang et al., 2017b; Araslanov & Roth, 2021), pixel-adaptation (Chang et al., 2019; Choi et al., 2019; Hoffman et al., 2018; Murez et al., 2018; Wu et al., 2018; Zhang et al., 2018; Yang & Soatto, 2020) and pseudo-labeling (Saleh et al., 2018; Zou et al., 2018; Lian et al., 2019; Zhang et al., 2019; Li et al., 2020a; Pan et al., 2020; Shin et al., 2020; Mei et al., 2020; Dong et al., 2020). There are also several methods that try to combine these strategies (Du et al., 2019; Li et al., 2019; Tsai et al., 2019; Vu et al., 2019; Paul et al., 2020; Wang et al., 2020; Musto & Zinelli, 2020; Kim & Byun, 2020; Lv et al., 2020; Huang et al., 2020; Subhani & Ali, 2020). Compared to these approaches, we assume no access to the source dataset, which is a more realistic setting but makes the task much more challenging. Unlike the above methods, there have been a few works for classification tasks which do not use source data, but only the source model for adaptation. The methods involve - entropy minimization with divergence maximization (Liang et al., 2020), pseudo-labeling with self-reconstruction (Yeh et al., 2021), generating additional target images (Li et al., 2020b) and self-supervision (Xia et al., 2021). Source-free adaptation for semantic segmentation has only recently gained attention. These works use pseudo-labeling (Liu et al., 2021; You et al., 2021; Ye et al., 2021; Kundu et al., 2021; Huang et al., 2021), domain alignment using discriminators similar to UDA (Ye et al., 2021; Liu et al., 2021), target image generation (Liu et al., 2021), and contrastive learning (Huang et al., 2021). Some methods modify the pseudo-labelling process by introducing negative labelling for pixels (You et al., 2021) and by using robustness to dropout to regularise them (S & Fleuret, 2021). (Kundu et al., 2021) attacks the problem differently by making the source model itself robust, which assumes access to the source data and the training process. In comparison, we do not generate images, do not use any additional networks, and do not assume access to the source data or training process but only the trained source model.

Test-Time Adaptation. A more challenging setting than source-free adaptation has recently been proposed where instead of adapting to a pool of target data, the goal is to adapt the model at test-time with only the current test samples. TTT (Sun et al., 2020) adds an auxiliary self-supervision branch while training the network to tune the encoder at test-time. This requires modifications to the source model training procedure and thus access to the source data. To counter this, TENT (Wang et al., 2021) uses the entropy loss as self-supervision from the model's prediction head itself to modulate the batch normalisation affine parameters. (Mummadi et al., 2021) builds upon TENT and introduces likelihood ratio based losses as an improvement over the entropy loss. A few approaches (Nado et al., 2020; Schneider et al., 2020; Khurana et al., 2021) propose to combine source statistics with current target statistics to adapt the model at test-time. Instead of assuming access to a batch of test samples, we operate in the episodic setting, assuming access to only a single test instance (Wang et al., 2021; Khurana et al., 2021).

Semi-Supervised Learning. Semi-supervised approaches assume access to both the labeled and unlabeled data during training as compared to our *phased* setting, where we train a source model on the labeled data, and the update it using only the unlabeled samples, assuming no access to the labeled samples. Existing methods are based on employing multiple branches for adversarial training to differentiate between the predicted and ground truth segmentation distribution (Hung et al., 2018) or to add collaborative training components to ensure low-high level consistency (Mittal et al., 2021), or correction of pseudo-labels (Mendel et al., 2020). Also, (Olsson et al., 2021; French et al., 2020) suggest creating artificial images by combining image pairs, and generating corresponding pseudo-labels.

B. Image Collages

Figure 2 illustrates the creation of image collages which are used to regularise model training. Given a pair of images X_i, X_j, the collage image is constructed by concatenating the first half of X_i with the second half of X_j along the width. Now, to obtain the pseudo-labels for the collage image, we also concatenate the pseudo-labels $\mathbf{PL}(\mathbf{M}_s(X_i))$ and $\mathbf{PL}(\mathbf{M}_s(X_j))$. The variations generated by collaging helps the model learn better decision boundaries as is evident from ablation study discussed in Appendix K.

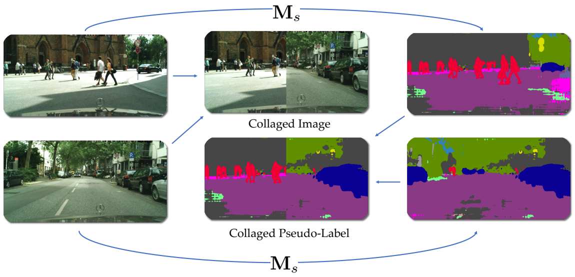

Figure 2. Process of image and pseudo-label collage creation. We use such collages to learn the target model.

C. Datasets

GTA5 → Cityscapes: In this setting, we consider GTA5 (Richter et al., 2016) as the source and Cityscapes (Cordts et al., 2016) as the target dataset. The source images are at 760×1280 resolution, while the target images are used at 512×1024. The datasets have 19 categories. The source dataset has 24966 training images and the target has 2975 training images on which we perform adaptation, and 500 validation images. We report the performance on these 500 validation images as is the usual practice in the literature.

SYNTHIA → Cityscapes: In this setting, we consider SYNTHIA (Ros et al., 2016) as the souce and Cityscapes (Cordts et al., 2016) as the target dataset. SYNTHIA contains 9400 training images. However, unlike GTA5, due to lack of proper annotations for a few categories in SYNTHIA, we remove them from evaluation and report the results for 16 categories, following the literature.

SceneNet → SUN: Both of the above two settings are for outdoor scenes, and in this setting we consider indoor scenes with SceneNet (McCormac et al., 2016) as the source and SUN (Song et al., 2015) as the target. The SceneNet dataset has around 5 million simulated images. However, a lot of the images are rather simple with only a few categories in them. Thus, to train the source model, we only choose the top $50,000$ images having the highest number of categories. The SUN dataset contains 5285 training images on which we perform adaptation and 5050 test images which are used for evaluation. Both of these datasets contain 13 categories. Specifically, we use the label transformation available with the SceneNet dataset [1] to map the labels in the SUN dataset such that it matches with the label space of SceneNet.

D. Methodology for different tasks

Source-Free Adaptation. Using the recipes discussed above, the steps for source-free adaptation are as follows: first, compute the norm statistics over the entire dataset using only forward passes through the network. Then use Eqn 1 to replace the network's norm parameters, setting $\lambda = 0$, as we have enough samples to get a low variance estimate of the statistics. After we obtain the norm updated network, we learn a new network using self-training along with the transforms by optimizing for the following objective:

$$\min_{\mathbf{M}_t} \quad \mathcal{L}_c = \sum_{X \in \mathcal{D}_t} l_c\Big(\mathbf{M}_t(X), \mathbf{PL}(\mathbf{M}_s(X))\Big) \tag{5}$$

$$\text{s.t.} \quad \mathbf{M}_t(\mathbf{T}(X)) \approx \mathbf{M}_t(X) \qquad \forall X \in \mathcal{D}_t, \mathbf{T} \in \mathcal{T}_i \tag{6}$$

$$\mathbf{M}_t(\mathbf{T}(X)) \approx \mathbf{T}(\mathbf{M}_t(X)) \quad \forall X \in \mathcal{D}_t, \mathbf{T} \in \mathcal{T}_e \tag{7}$$

l_c is the cross-entropy loss. Note that we enforce the consistency constraints using the outputs of the target model to allow for improvement beyond the source model, while still extracting information from the source model using the loss Eqn 5.

In practice, we compose the two sets of transforms to strengthen its power. To do so, in each iteration, we randomly choose a set of transformations $\mathcal{T} = \{\mathbf{T}_0, \ldots, \mathbf{T}_k\} \in \mathcal{P}(\mathcal{T}_i \cup \mathcal{T}_e) \setminus \emptyset$. Due to the difference in consistency required to these transforms, they need to be composed differently for input and output. The input transform is a composition of all the transforms, $\mathbf{T}_i = \mathbf{T}_0 \circ \cdots \circ \mathbf{T}_k$, but to compose the output transform, we remove those drawn from $\mathcal{T}_i$, and then compose the rest in the same manner as input transform to obtain $\mathbf{T}_o$. Then, the constraints can be simplified as $\mathbf{M}_t(\mathbf{T}_i(X)) \approx \mathbf{T}_o(\mathbf{M}_t(X))$.

[1] https://github.com/ankurhanda/sunrgbd-meta-data

We approximate the constraint using two losses, and impose it via the penalty method. For every iteration, we consider $\mathbf{T}_o(\mathbf{M}_t(X))$ i.e., the transformed outputs from the current iteration target model as ground-truths for the losses. The two losses are:

$$\mathcal{L}_r = \sum_{X \in \mathcal{D}_t} l_c\Big(\mathbf{M}_t(\mathbf{T}_i(X)), \mathbf{T}_o(\mathbf{M}_t(X))\Big) + \\ l_c\Big(\mathbf{M}_t(\mathbf{T}_i(X)), \mathbf{T}_o(\mathbf{PL}(\mathbf{M}_t(X)))\Big) \tag{8}$$

where l_c is the cross-entropy loss. The first loss is a form of soft pseudo-labeling and computed for every pixel, whereas the second loss is hard pseudo-labeling, which only computes the loss for the confident pixels. We optimize using Stochastic Gradient Descent (SGD). Note that we use the pseudo-labels using $\mathbf{M}_s$ for ground-truth in the loss function of Eqn. 5 and $\mathbf{M}_t$ in the current iteration as the ground-truth for the loss function in Eqn. 8. For clarity, the SGD updates for the k^{th} iteration, with η as the learning rate can be expressed as follows:

$$\mathbf{M}_t^{k+1} = \mathbf{M}_t^k - \eta \nabla_{\mathbf{M}_t}\Big(\mathcal{L}_c(\mathbf{M}_s) + \mathcal{L}_r(\mathbf{M}_t^k)\Big) \tag{9}$$

where the argument in the brackets are the models used to obtain the labels to compute the losses.

Multi-source Adaptation. In this case, we have multiple source models, and would want to extract the knowledge from these source models to a target model. For each of the source models, we update the norm as above. Let us denote the norm updated source models as $\{\mathbf{M}_s^i\}_{i=1}^{n_s}$. Now, if we can formulate a function $\mathbf{M}_s$, which gathers the knowledge from these multiple source models, then the rest of the learning mechanism can be used as described above. Now, in multi-source adaptation, the challenge is to identify the best model from which we should transfer the knowledge, and avoid negative transfer. We use the entropy of every pixels as a confidence measure to weigh the predictions. Formally, the composite source model which combines the predictions from these multiple source models can be expressed as follows:

$$[\mathbf{M}_s(X)]^{x,y} = \sum_{i=1}^{n_s} \frac{[e_i]^{x,y}}{\sum_i [e_i]^{x,y}} [\mathbf{M}_s^i(X)]^{x,y} \tag{10}$$

where $[e_i]^{x,y} = {}^{1}\!/_{\mathbb{E}[-\log[\mathbf{M}_s^i(X)]^{x,y}]}$.

This formulation is simple, allows category-wise knowledge extraction across models depending on their confidence, and also fits in well with the proposed single source approach. Note that in this case, we still learn a single target model, which extracts knowledge from all the source models.

Test-Time Adaptation. This setting has been introduced in recent works (Sun et al., 2020; Wang et al., 2021; Mummadi et al., 2021; Khurana et al., 2021). We consider the most realistic scenario - given only the source model, we need to adapt to a single image at a time, and reverting back to the source model after every sample. This is termed as the episodic setting in literature. Here, as discussed in Section 3.1, we replace the batch norm layer with the formulation in Eqn 1, where μ_t, σ_t is estimated using a single test image. We initialize $\lambda = 0.8$, treating it as a learnable parameter. We then optimize the pseudo-labeling loss in Eqn 5, for only the prior and the classifier layer of the network, keeping the other parameters of the network frozen. To maintain efficient inference during test-time, we limit the optimization to only a single backward pass.

***Phased* Semi-Supervised Learning (SSL):** Our knowledge transfer method also works for semi-supervised learning, where we have a small amount of labeled data and large amount of unlabeled data, both from the same domain. We learn the source model $\mathbf{M}_s$ using the labeled dataset, and then solve the same optimization problem as in Eqn 5, 6, 7, using only the additional unlabeled data from the same domain.

E. Details for various transforms

Illustrations of various invariant and equivariant transforms used is shown in 3. For image mirroring, we first randomly choose a column, vertically splitting the image, and mirror the larger side of the image onto the smaller side. For rotation, we randomly choose the rotation degree between $[-5°, +5°]$. In drop-block, we randomly choose k blocks of size $b \times b$ and set them to 0, where $\frac{kb^2}{HW} \approx p$, with two free parameters b, p. H and W are the image height and width. For Gaussian blur, we choose a kernel size and filter variance. We perform a hyper-parameter analysis for these parameters in Sections I and J.

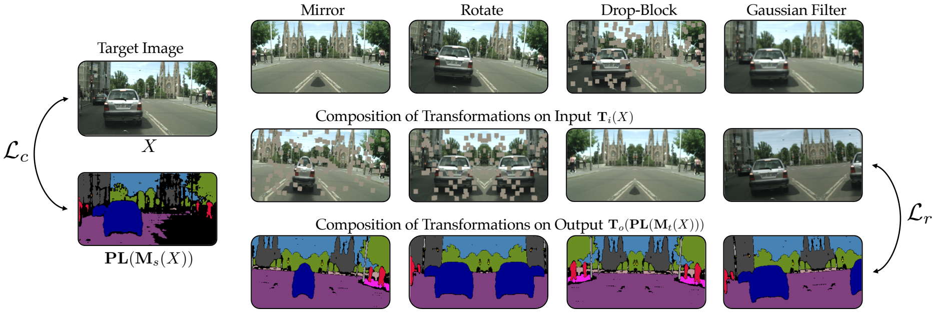
Mirror
Rotate
Drop-Block
Gaussian Filter
Target Image
X
$\mathcal{L}_c$
$\mathbf{PL}(\mathbf{M}_s(X))$
Composition of Transformations on Input $\mathbf{T}_i(X)$
Composition of Transformations on Output $\mathbf{T}_o(\mathbf{PL}(\mathbf{M}_t(X)))$
$\mathcal{L}_r$

Figure 3. **Transforms.** Different transforms used in our algorithm. The left-most side shows an image with the pseudo-labels obtained from the source model $\mathbf{M}_s$, which are used to compute the loss $\mathcal{L}_c$. The top row shows the transformations individually, followed by their random compositions in the second row, and the last row shows the transformations on the output of the target model for a certain iteration, which are used as ground-truth for computing loss $\mathcal{L}_r$.

F. Implementation Details

We use one GPU to train our models with a batch size of 1 in all experiments. We use SGD with an initial learning rate of 2.5×10^{-4} with polynomial decay of power 0.9 (Chen et al., 2017a). We use the standard metric of mean intersection over union (mIoU) (Chen et al., 2017a) to evaluate all algorithms.

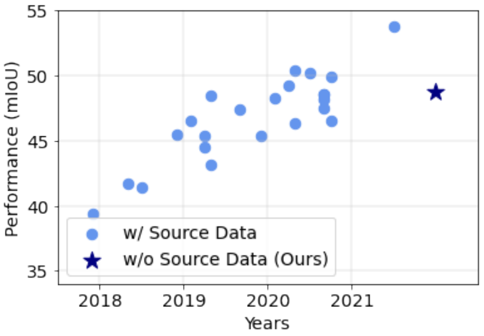

Figure 4. Comparing our approach with the methods that use source data, plotted on the time axis when the works were published (on GTA5 → Cityscapes). Note that our method performs much better than many recent methods even without using any source data.

G. Category-wise Segmentation Results

In this section, we present the category-wise segmentation results for all three dataset settings, GTA5→Cityscapes, SYNTHIA→Cityscapes and SceneNet→SUN in Table 6, 7 and 8 respectively. Stuff includes road, sidewalk, building, wall, fence, veg, terrain and sky; while Things include sign, person, rider, car, truck, bus, train, mbike, bike, light and pole. Additionally, we also plot the performances methods in the literature with the time when they were published in Figure 4 on GTA5→Cityscapes.

H. Semi-Supervised Learning Results on SUN

Table 9 presents results of semi-supervised learning on the SUN dataset. As we can see that vanilla pseudo-labeling does not improve the performance beyond the source model. However, our method performs much better than the source. Moreover, the amount of improvement is more for lower labeled data regime, which shows that using the unlabeled data actually helps.

Table 6. Results of adapting GTA5 to Cityscapes. The top row group are methods which use source data during adaptation, while the bottom row group do not use any source data to adapt.

Source	Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
Yes	AdaptOutput (Tsai et al., 2018)	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
	AdvEnt (Vu et al., 2019)	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
	SSF-DAN (Du et al., 2019)	90.3	38.9	81.7	24.8	22.9	30.5	37.0	21.2	84.8	38.8	76.9	58.8	30.7	85.7	30.6	38.1	5.9	28.3	36.9	45.4
	BDL (Li et al., 2019)	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
	CAG (Zhang et al., 2019)	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
	WeakDA (Paul et al., 2020)	91.6	47.4	84.0	30.4	28.3	31.4	37.4	35.4	83.9	38.3	83.9	61.2	28.2	83.7	28.8	41.3	8.8	24.7	46.4	48.2
	Stuff (Wang et al., 2020)	90.6	44.7	84.8	34.3	28.7	31.6	35.0	37.6	84.7	43.3	85.3	57.0	31.5	83.8	42.6	48.5	1.9	30.4	39.0	49.2
	FDA (Yang & Soatto, 2020)	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	53.1	16.8	27.7	46.4	50.4
	SAC (Araslanov & Roth, 2021)	90.4	53.9	86.6	42.4	27.3	45.1	48.5	42.7	87.4	40.1	86.1	67.5	29.7	88.5	49.1	54.6	9.8	26.6	45.3	53.8
No	Source	79.7	21.8	66.8	19.3	20.6	22.8	28.9	12.9	76.3	19.5	60.9	56.2	26.6	77.8	33.3	27.7	3.9	25.0	32.5	37.5
	URMA (S & Fleuret, 2021)	92.3	55.2	81.6	30.8	18.8	37.1	17.7	12.1	84.2	35.9	83.8	57.7	24.1	81.7	27.5	44.3	6.9	24.1	40.4	45.1
	LD(You et al., 2021)	91.6	53.2	80.6	36.6	14.2	26.4	31.6	22.7	83.1	42.1	79.3	57.3	26.6	82.1	41.0	50.1	0.3	25.9	19.5	45.5
	HCL (Huang et al., 2021)	92.0	55.0	80.4	33.5	24.6	37.1	35.1	28.8	83.0	37.6	82.3	59.4	27.6	83.6	32.3	36.6	14.1	28.7	43.0	48.1
	SFUDA (Ye et al., 2021)	95.2	40.6	85.2	30.6	26.1	35.8	34.7	32.8	85.3	41.7	79.5	61.0	28.2	86.5	41.2	45.3	15.6	33.1	40.0	49.4
	Ours	89.2	37.3	82.4	29.0	23.5	31.8	34.6	28.7	84.8	45.5	80.2	62.6	32.6	86.1	45.6	43.8	0.0	34.6	54.4	48.8

Table 7. Results of adapting SYNTHIA to Cityscapes. The top group are methods which use source data during adaptation, while the bottom row do not use any source data to adapt. mIoU and mIoU* are averaged over 16 and 13 categories.

Source	Method	road	sidewalk	building	wall	fence	pole	light	sign	veg	sky	person	rider	car	bus	mbike	bike	mIoU	mIoU*
Yes	AdaptOutput (Tsai et al., 2018)	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
	AdvEnt (Vu et al., 2019)	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
	SSF-DAN (Du et al., 2019)	84.6	41.7	80.8	-	-	-	11.5	14.7	80.8	85.3	57.5	21.6	82.0	36.0	19.3	34.5	-	50.0
	CAG (Zhang et al., 2019)	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	51.4
	WeakDA (Paul et al., 2020)	92.0	53.5	80.9	11.4	0.4	21.8	3.8	6.0	81.6	84.4	60.8	24.4	80.5	39.0	26.0	41.7	44.3	51.9
	Stuff (Wang et al., 2020)	83.0	44.0	80.3	-	-	-	17.1	15.8	80.5	81.8	59.9	33.1	70.2	37.3	28.5	45.8	-	52.1
	FDA (Yang & Soatto, 2020)	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
	SAC(Araslanov & Roth, 2021)	89.3	47.2	85.5	26.5	1.3	43.0	45.5	32.0	87.1	89.3	63.6	25.4	86.9	35.6	30.4	53.0	52.6	59.3
No	Source	37.6	18.7	73.8	9.95	0.1	26.4	8.9	13.9	74.7	80.4	52.4	16.1	39.2	21.9	13.2	25.8	32.1	36.7
	URMA (S & Fleuret, 2021)	59.3	24.6	77.0	14.0	1.8	31.5	18.3	32.0	83.1	80.4	46.3	17.8	76.7	17.0	18.5	34.6	39.6	45.0
	LD(You et al., 2021)	77.1	33.4	79.4	5.8	0.5	23.7	5.2	13.0	81.8	78.3	56.1	21.6	80.3	49.6	28.0	48.1	42.6	50.1
	HCL (Huang et al., 2021)	80.9	34.9	76.7	6.6	0.2	36.1	20.1	28.2	79.1	83.1	55.6	25.6	78.8	32.7	24.1	32.7	43.5	50.2
	SFUDA (Ye et al., 2021)	90.9	45.5	80.8	3.6	0.5	28.6	8.5	26.1	83.4	83.6	55.2	25.0	79.5	32.8	20.2	43.9	44.2	51.9
	Ours	74.3	33.7	78.9	14.6	0.7	31.5	21.3	28.8	80.2	81.6	50.7	24.5	78.3	11.6	34.4	53.7	43.7	50.2

Table 8. Results of adapting SceneNet to SUN. The top row group does not use any source data to adapt, and the bottom row uses full supervision on the target images.

	SceneNet → SUN													
Method	bed	books	ceiling	chair	floor	furniture	objects	picture	sofa	table	tv	wall	window	mIoU
Source	19.6	10.1	22.2	42.5	65.4	21.3	13.4	20.9	18.2	27.1	6.2	57.1	20.2	26.5
Ours	35.3	23.5	34.1	48.7	73.6	26.7	11.1	29.9	36.9	38.1	15.0	63.2	27.2	**35.6**
Full Supervision	53.1	32.6	54.0	60.0	82.4	35.1	33.4	43.2	45.7	52.7	36.8	72.0	46.8	49.8

I. Hyper-parameter analysis of cutout

In this section, we analyse the effect of the hyper-parameters involved in the cutout transformation. Recall from the paper, in this transformation, we choose two parameters: size b of the square block, and the percentage p of the image to be removed.

Table 9. Semi-supervised learning on SUN (five random splits).

Labeled Samples	Source	Pseudo-Label	**Ours**
1/8	41.7	40.9	44.6
1/4	44.3	43.3	46.7
Full set	49.8	49.8	49.8

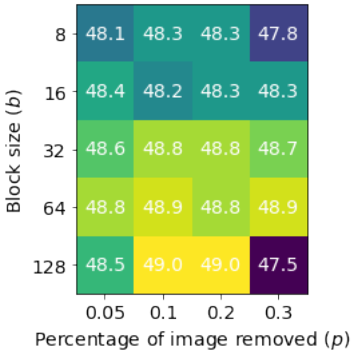

Figure 5. Ablation study of the cutout transformation.

Given these two parameters, we choose the number of blocks to be removed as $k = \frac{pHW}{b^2}$. We execute our framework for various values of b and p, while keeping the rest of the framework same, and present the performance obtained in Figure 5. By using this transform, we want the network to learn rich context information, while performing inpainting in the output space. As can be seen, with lower block size and higher percentage of image removed, the performance degrades. This is because by doing so the image becomes noisy, rather than structured removal, i.e., removed portions become well distributed throughout the image, which makes it harder to learn context information. However, with higher block size, and moderate percentage of image removed, the performance increases. We choose $b = 64, p = 0.2$, in our experiments on outdoor images, where the resolution of the image is high, and we choose $b = 32, p = 0.1$, for indoor experiments, with lower image sizes.

J. Hyper-parameter analysis of Gaussian filter

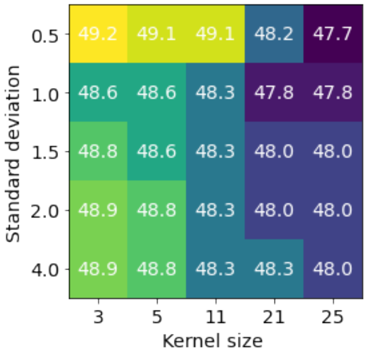

Figure 6. Ablation study of the Gaussian filtering transformation.

In this section, we analyse the effect of two hyperparameters in the Gaussian filter transformation. The two hyper-parameters are the kernel size and the standard deviation of the filter. Instead of keeping a single standard deviation for the filter, we choose it randomly between $[0.1, \sigma_{max}]$, where σ_{max} is the hyper-parameter to choose. The image becomes more blurry with higher kernel size and lower standard deviation. As the image becomes more blurry, it becomes difficult for the network to figure out the content, and thus we see a degradation in performance in the top right corner in Figure 6. Note that the standard deviation mentioned in the figure signify σ_{max}. We use a kernel size of 5 and $\sigma_{max} = 2$.

K. Ablation Studies

Table 10. Ablation of the transformations.

Collage	Mirror	Rotate	Gaussian	Cutout	GTA5 → Cityscapes	SYNTHIA → Cityscapes	SceneNet → SUN
✓					46.4	40.4	32.2
	✓				45.9	41.9	34.6
		✓			45.5	42.3	34.8
			✓		46.4	42.5	34.5
				✓	45.8	41.9	33.1
✓	✓	✓	✓	✓	**48.8**	**43.7**	**35.6**

Ablation study of transforms: In this experiment, we analyse the effect of transforms we use in our framework and the effect of learning from collage images. We use two spatial transforms, Mirror and Rotate and two augment transforms, Gaussian filter and Cutout. The results are presented in Table 10. We evaluate the model by adding only one transform at a time. As can be seen, the transforms individually improve the performance beyond just pseudo-labeling. Moreover, learning from collage images also improves the performance beyond pseudo-label learning from single images. The maximum gain is achieved when we use all of the above together. It is interesting to note that augment transforms beyond a certain parameter limit do not work as well, which is intuitive, as the content of the image may change beyond what is necessary for fine segmentation. For example, in Gaussian filtering, using a filter size more than 20 does not help with an image of size 512×1024.

Ablation study of the framework: In this experiment, we break down every part of the framework and evaluate their performance. We present the results in Table 11 for all the datasets. When we first update the normalization parameters, the performance improves by $1-4\%$. Then using the updated network for pseudo-label training using the target images offers a further $4-5\%$ improvement. Now, imposing the transform constraints using the hard and soft constraint losses as in Eqn. 8 along with the collage images further improves the performance by about 3% compared to just pseudo-labeling.

Ablation of design choices: We perform ablation of various choices involved in designing our proposed algorithm. The first one is using category-wise thresholds for pseudo-labeling, compared to uniform thresholding for all labels. To compare, we execute our framework with an uniform threshold of 0.9 for all the labels (following (Li et al., 2019)), as mentioned in the sixth row in Table 11. We observe that using label-wise thresholding ("Ours" in Table 11) performs better by $1.5-4\%$ than uniform thresholding.

In the next ablation, we study the effect of using the norm updated source model M_s instead of target model M_t in current iteration, for the constraints of Eqn. 6, 7. If we apply the transform constraints using M_s rather than M_t, then we limit its ability to improve beyond the fixed pseudo-labels, albeit with transforms applied on them. In other words, when using $\mathbf{M}_t$ for the constraints, the optimization process is allowed to learn the target model and self-improve in a way such that the constraints are satisfied on the final target model. This helps to improve the performance, as is evident by comparing the seventh row in Table 11 with the last row.

Next, we investigate whether to fine-tune the source model for adaptation or train a new target model from scratch. As can be observed from the eighth row of Table 11, fine-tuning performs worse than our method where we train a new target model from scratch. Note that the learning rate and the number of training epochs are kept the same for both cases. This can be attributed to the source model being already in a local optima in the loss landscape, thus may be less influenced by the pseudo-label losses and constraints.

Finally, we use the transforms in the standard augmentation setting (but stronger than normal augments), i.e., augment the images, pseudo-label them and learn the target model using them. We use equal portion of original and augmented images, as in our method. The results are presented as "PL+augment" in Table 11, which shows that our consistency based approach performs much better. This is because the pseudo-labels are much better on the original images rather than on the augmented ones, and it is better to transform the pseudo-labeled image, rather than pseudo-labeling the transformed image.

Table 11. Ablation of loss functions and design choices.

Datasets	Source	Update Norm	Pseudo Label	PL + soft constraints	PL + hard constraints	Uniform thresholds	Constraints using $\mathbf{M}_s$	Finetuning $\mathbf{M}_s$	PL + augment	**Ours**
GTA5 → Cityscapes	37.5	41.4	45.6	48.2	47.9	47.3	47.3	48.6	43.1	48.8
SYNTHIA → Cityscapes	32.1	35.3	40.2	43.1	43.0	41.6	42.2	43.3	39.1	43.7
SceneNet → SUN	26.5	27.4	32.1	35.0	34.8	31.9	33.2	28.9	33.8	35.6

L. Qualitative Analysis

In Figure 7 we present a visual comparison of our method with directly applying the source model on the target images, shown as "No Adapt". As can be seen in the first column, our method is able to properly label the signs, even though the source model does not have them, as shown in "No Adapt". Similar discussions can be extended to results in the second column. In the third and fourth columns, the source model is very confused with the shadows on the chairs and tables, and assigns them multiple labels. However, our method is able to predict the accurate labels with proper segmentation. In the last column, the network mistakes most portion of the road as sidewalk, a prior that pedestrians can be more often seen on sidewalks than roads. However, our method is able segment the road and sidewalk properly and is also able to segment a few of the road signs in the background.

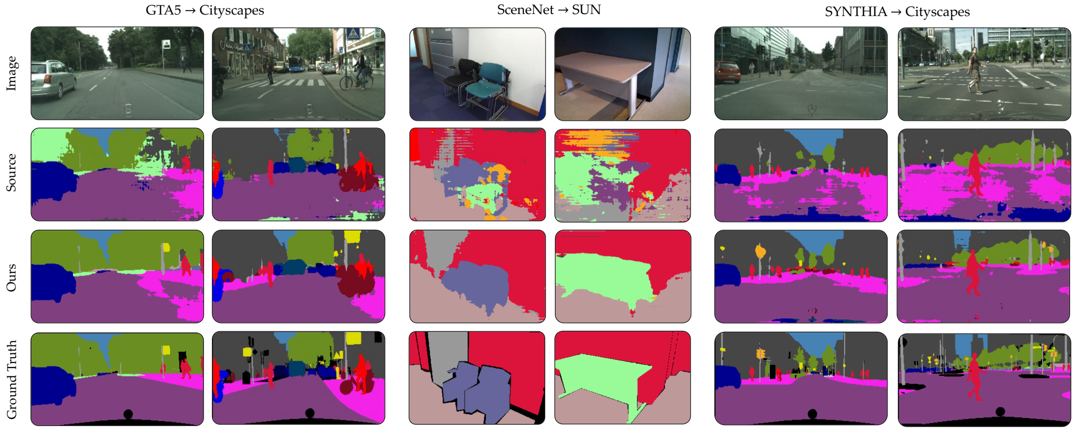

Figure 7. Qualitative visualization. The first two columns and the last two columns show results on outdoor scenes, while the middle two columns show results on indoor scenes.