Generalize then Adapt: Source-Free Domain Adaptive Semantic Segmentation

Jogendra Nath Kundu¹* Akshay Kulkarni¹* Amit Singh¹ Varun Jampani² R. Venkatesh Babu¹
¹Indian Institute of Science, Bangalore ²Google Research

Abstract

Unsupervised domain adaptation (DA) has gained substantial interest in semantic segmentation. However, almost all prior arts assume concurrent access to both labeled source and unlabeled target, making them unsuitable for scenarios demanding source-free adaptation. In this work¹, we enable source-free DA by partitioning the task into two: a) source-only domain generalization and b) source-free target adaptation. Towards the former, we provide theoretical insights to develop a multi-head framework trained with a virtually extended multi-source dataset, aiming to balance generalization and specificity. Towards the latter, we utilize the multi-head framework to extract reliable target pseudo-labels for self-training. Additionally, we introduce a novel conditional prior-enforcing auto-encoder that discourages spatial irregularities, thereby enhancing the pseudo-label quality. Experiments on the standard GTA5→Cityscapes and SYNTHIA→Cityscapes benchmarks show our superiority even against the non-source-free prior-arts. Further, we show our compatibility with online adaptation enabling deployment in a sequentially changing environment.

1. Introduction

Almost all supervised learning systems assume that the training and testing data follow the same input distribution. However, this assumption is impractical as target scenarios often exhibit a distribution shift. For example, self-driving cars often fail to generalize when deployed in conditions different from training, such as cross-city [11] or cross-weather [66] deployment. This is because the model fails to apprehend the generic, causal factors of variations and instead, holds on to domain-specific spurious correlations [25]. Over-reliance on training data from a particular distribution can cause the model to fail even for mild domain-shifts like changes in illumination, texture, background, etc.

Unsupervised domain adaptation (DA) is one of the primary ways to address such problems. Here, the goal is to transfer the knowledge from a labeled source domain to an unlabeled target domain. The major limitation of typical DA approaches [65] is the requirement of concurrent access to both source and target domain samples. While concurrent access better characterizes the distribution shift, it is a major bottleneck for real-world deployment scenarios. Consider a modern corporate dealing where the vendor organization has access to a large-scale labeled dataset (i.e. source-data) which is used to train a source-model. The vendor finds multiple clients interested in deploying the source-model in their specific target environments. However, both parties are restrained from data sharing due to proprietary, privacy, or profit related concerns. This motivates us to seek learning frameworks where the vendor can trade only the source-model and the client can perform target adaptation without the source-data. This special case of domain adaptation [42, 36, 45] is Source-Free Domain Adaptation (SFDA).

In this work, we aim to develop an SFDA framework for semantic segmentation of urban road scenes. In a cooperative setup, both vendor and the client must adopt specialized learning strategies to benefit the end goal.

a) Vendor-side strategies. These strategies can be discussed under two broad aspects viz. source dataset and training strategy. The vendor must acquire a substantially diverse large-scale dataset aiming to subsume unknown target scenarios. In literature, Multi-Source DA (MSDA) [96, 78, 1] and domain generalization (DG) [40] works use multiple labeled source domains to improve target gener-

¹Equal contribution.
²Project page: https://sites.google.com/view/sfdaseg
alization. However, gathering annotation for more than one domain is costly and time-consuming [13]. Thus, we focus on developing a strategy to simulate multiple novel domains from samples from a single labeled domain. Carefully crafted augmentations randomly perturb the task-irrelevant factors (such as stylization, texture modulation, etc.), facilitating the learning of domain-invariant representations. Hence, we devise multiple augmentation-groups (AGs), where each group modulates the image by varying certain statistics thereby constructing virtual, labeled source-domains, to be treated as a multi-source dataset.

Next, we focus on developing an effective training strategy. The naive solution would be to train a single model on the entire multi-source dataset to learn domain-generic features. However, this can lead to sub-optimal performance if a certain AG alters the task-relevant causal factors [25]. Further, the target domain may be similar to one or a combination of AGs. In such cases, domain-specific (AG-specific) learning is more helpful. This motivates us to seek a domain-specific framework to complement the domain-generic model. Thus, we give theoretical insights to analyze domain-specific hypotheses and propose Source-only Multi-Augmentation Network (SoMAN) as shown in Fig. 1.

Going forward, we recognize that SoMAN may lack the ability to capture inductive bias, which would prevent the model from manifesting structurally consistent predictions. This is particularly important for dense prediction tasks [34, 33]. Modeling general context dependent priors encourages the prediction of plausible scene segments while discouraging common irregularities (e.g. merged-region or split-region issues [35]). To this end, we introduce a separate model namely, conditional Prior-enforcing Auto-Encoder (cPAE). cPAE is trained on segmentation maps available with the vendor, and used at the client-side to improve the source-free adaptation performance.

b) Client-side strategies. We draw motivation from pseudo-label based self-training approaches [18, 102]. The target samples are passed through the source-model to select a set of pseudo-labels which are later used to finetune the network. In the absence of source-data, effectiveness of such self-training depends on the following two aspects. First, the training must be regularized to retain the vendor-side, task-specific knowledge. We address this by allowing only a handful of weights to be updated while others are kept frozen from the vendor-side training. Second, the pseudo-label selection criteria must overcome issues related to label-noise and information redundancy. We address this by selecting the optimal prediction from the SoMAN-heads and using the pruned output after forwarding through cPAE.

In summary, we make the following main contributions:

• We propose to address source-free DA by casting the vendor-side training as multi-source learning. To this end, we provide theoretical insights to analyze different ways to aggregate the domain-specific hypotheses. It turns out that a combination of domain-generic and leave-one-out configuration performs the best.

• While accessing a single source domain, we propose a systematic way to select a minimal set of effective augmentations to resemble a multi-source scenario. The vendor uses this to develop a multi-head network, SoMAN subscribing to the leave-one-out configuration.

• Aiming to have a strong support for the spatially-structured segmentation task, we develop a conditional prior-enforcing auto-encoder. This encourages plausible dense predictions thereby enhancing the quality of pseudo-labels to aid the client-side self-training.

• Our source-free framework achieves state-of-the-art results for both GTA5 → Cityscapes and SYNTHIA → Cityscapes adaptation benchmarks, even when compared against the non-source-free prior arts.

2. Related Work

Here, we briefly review the segmentation DA literature [72].

Feature-space DA. The early works in DA for semantic segmentation are inspired from the GAN framework [17], involving training a segmentation network to confuse a domain discriminator enforcing domain invariance on the latent features [21]. Several works [9, 43, 22, 20] utilized this discriminative alignment [100, 10, 48, 15] while adding complementary modules [81, 14, 83] to improve adaptation. Another line of works [67, 8, 50, 73, 86, 88] use the same framework on low-dimensional output space [91, 31, 87, 76, 79, 80] instead of high-dimensional feature space. However, these works require cumbersome adversarial training and rely on source-target co-existence.

Image-space DA. The success of CycleGAN [99] for image-to-image translation led to several DA approaches [43, 20, 10, 16] utilizing it for input-level adaptation while also addressing semantic consistency in the transformed images. Another category of works [3, 62, 94, 12, 85] explore style-transfer techniques for input-level perceptual invariance [84, 90, 49, 39, 82] between source and target domains. However, these works also assume the co-existence of source and target domains.

Source-free DA. Bateson et al. [2] perform source-free DA for medical segmentation using entropy minimization and class-ratio alignment. Concurrent source-free works use data-free distillation, self-training, patch-level self-supervision [46] and feature corruption with entropy regularization [70] focused on target adaptation. In contrast, we develop a novel approach for vendor-side source training.

DA via self-training. Early works [102, 43, 101] use highly confident target predictions as pseudo-labels, selected using a confidence threshold. To improve the pseudo-labels, prior
works used prediction ensembling [5, 90, 98, 97], extra
networks [12], applied constraints [71], modified the confi-
dence thresholding technique [53, 41, 68], utilized image-
level pseudo-labels [59] and intra-domain (easy-hard) ad-
versarial training [56]. Most prior arts used labeled source
with self-training to retain task-specific source knowledge.

**DG and MSDA.** [96] use multiple synthetic datasets for
Multi-Source DA (MSDA) in segmentation. Restricted to
a single source setting, we use data augmentation techniques
to generate new domains. In the presence of multi-source
data, the vendor-side training is equivalent to domain general-
ization [92, 7, 57] as it does not involve training on target.

### 3. Approach

Consider a set of source image and segmentation pairs
\((x_s, y_s) \in D_s\) where the source images \(x_s\) are drawn from
a marginal distribution \(p_s\). The unlabeled target images \(x_t \in
D_t\) are drawn from \(p_t\). However, the output segmentation
maps follow a single marginal distribution \(p_y\). The goal is
to learn a mapping \(\hat{y}_s = h(x_s)\) that can generalize well for
\(x_t\). The proposed source-free domain adaptation is broadly
divided into two: vendor-side and client-side.

#### 3.1. Vendor-side Strategy

In the absence of target data, the vendor’s task effectively
reduces to domain generalization (DG) [40]. DG is shown
to be highly effective in the presence of multiple source
domains. Thus, we plan to cast the vendor-side model prep-
aration as a multi-source representation learning problem.

**Non-source-free paradigm.** We assume access to \(K\)
source datasets \((x_{si}, y_{si}) \in D_{si}, \forall \alpha \in [K]\) where
images \(x_{si}\) are drawn from marginal distribution \(p_{si}\).
In non-source-free paradigm, the objective is to utilize all
the domains (including the target) to realize a hypothesis
\(h^* = \arg \min_{h \in A} \epsilon_t(h)\) with a small target error, where
\[
\epsilon_t(h) = \mathbb{E}_{(x,y) \sim p_t} [\mathcal{L}(h(x), y)] \quad h \in \mathcal{H}^* \subseteq A \quad (1)
\]
Here, \(\mathcal{L}\) is the loss and \(A\) is the hypothesis space. \(\mathcal{H}^* \subseteq A\)
can be interpreted as a hypothesis subspace spanning the hy-
potheses that can be learned using the best convex combina-
tion of sources \(\alpha^* \in \Delta = \{\alpha \in [0,1]^K : \sum_{i=1}^K \alpha[i] = 1\}\)
in the presence of concurrent access to \((D_{si})_{i=1}^K\) and \(D_t\),
i.e. \(\alpha^* = \arg \min_{\alpha} (\arg \min_{h \in A^*} \epsilon_t(h)).\)

While operating in a source-free paradigm [37, 38], let the
vendor be approached by \(M\) number of clients, each
with different target domains \(t_j, \forall j \in [M]\). For every target
\(t_j\), there exists a specific \(\alpha_j^*\) such that \(\epsilon_t(h \in \mathcal{H}^*) \leq
\epsilon_t(h \in \mathcal{H}^*) \forall \alpha \in \Delta\). However, in the absence of con-
current access to source and target domains (SFDA), it is
not possible to optimize for \(\alpha_j^*\) for any target \(t_j\). Thus, we
propose a source-free multi-domain paradigm.

**Definition 1. (Source-free multi-domain paradigm)** Consider a vendor who has access to labeled data \(\{D_{si}\}_{i=1}^K\)
from \(K\) source domains and a client who has access to una-
abeled target data \(D_{t_j}\). In the source-free paradigm, the ven-
dor prepares a prescient model with an immutable hypoth-
esis support set \(\mathcal{A}_{SF}\) (a union of certain hypothesis supports)
without any information about \(t_j\). This model is traded with
the client for target adaptation without any data sharing.

In the hypothetical scenario of source-target concu-
rent access, the client can determine the best \(\alpha_j^*\) such that
\(\epsilon_t(h \in \mathcal{H}^*) \leq \epsilon_t(h \in \mathcal{A}_{SF}).\) The proposed paradigm not
only enables adaptation without any data sharing, but also
enables the vendor to prepare a single source-model for all
future clients. Thus, the process becomes more efficient for
both vendor and client in terms of compute and storage.

#### 3.1.1 Multi-source representation learning

**Under source-free**, the vendor’s objective would be to real-
ize a learning setup that would generalize to a wide range of
unseen targets. While aiming to learn a single hypothesis,
empirical risk minimization (ERM) [77] would be the best
solution (all domains weighted equally). Consider a sce-
nario where \(p_{ij}\), i.e. marginal distribution of the target \(t_j\),
matches with the marginal of one of the source domains.
Here, the domain-specific expert for that source domain
would definitely outperform the ERM baseline. To this end,
a hypothesis support set \(\mathcal{A}_{SF}\), i.e. a union of certain hypoth-
esis supports, would provide better flexibility for SFDA.
With this intent, we discuss the following configurations.

**a) ERM.** Under ERM configuration, we set \(\mathcal{A}_{SF} = \mathcal{H}^{ERM}\)
where \(\mathcal{H}^{ERM}\) is formed with equal weightage to all the
multi-source domains i.e. \(\alpha[i] = \frac{1}{K} \forall i \in [K]\).

**b) Domain-experts+++ (DE++).** This configuration encom-
passes a set of \(K + 1\) hypothesis supports. This includes
\(K\) number of domain-specific experts alongside one ERM
support. Thus, we set \(\mathcal{A}_{SF} = \mathcal{A}_{DE++} = \mathcal{H}^{DE}\cup \mathcal{H}^{ERM}\).
For \(i\)-th support \(\mathcal{H}^{DE}_i\), \(\alpha_i[i'] = \mathbb{I}_{i'=i} \forall i' \in [K]\) where \(\mathbb{I}\) is
the indicator function (1 if input condition is true, else 0).

**c) Leave-one-out+++ (LO+++).** It may happen that using a particular
source may cause information loss that hinders optimal adap-
tation for a future target. To improve support for such targets, we introduce leave-one-out (LO) hypoth-
esis support where \(i\)-th subspace \(\mathcal{H}^{LO}_i\) is formed by leaving one
domain out, i.e. with \(\alpha_i[i'] = \frac{1}{K-1}\mathbb{I}_{i\neq i'} \forall i' \in [K].\) Similar
to DE++, LO+++ also includes \(K + 1\) hypothesis supports,
i.e. \(K\) number of LO supports with one ERM. Thus, we set
\(\mathcal{A}_{SF} = \mathcal{A}_{LO++} = \mathcal{H}^{LO}\cup \mathcal{H}^{ERM}\).

We include the ERM support, i.e. \(\mathcal{H}^{ERM}\), in both LO++
and DE++ to provide complementary domain-generic in-
formation alongside the different forms of domain-specific
information. Here, the individual hypothesis supports are
implemented as separate classifier heads trained on a com-
mum feature extractor (Sec 3.1.3). Note that we only con-

---

Note: The text continues with further details and discussions that are not fully transcribed here due to the page limit constraints. The above snippet captures the essence of the discussions on multi-source domain adaptation and the introduction of source-free paradigms with various configurations for hypothesis support sets. The text emphasizes the importance of domain-specific experts and the trade-offs between source and target domain access in the adaptation process.
consider options that require $K$ heads while other domain-specific solutions like leave-$r$-out have higher computational cost requiring $\binom{K}{r}$ heads. Next, we discuss a result comparing the target error $\epsilon(h)$ of the three configurations.

**Result 1.** Consider $DE^{++}$ hypothesis space $A^{DE^{++}}$, $LO^{++}$ hypothesis space $A^{LO^{++}}$, and unseen target data $D_t$. Then,

$$
\epsilon(h \in A^{LO^{++}}) \leq \epsilon(h \in H_{ERM})
\epsilon(h \in A^{DE^{++}}) \leq \epsilon(h \in H_{ERM})
$$

(2)

As depicted in Fig. 2, the distributed subspace constituents of $A^{DE^{++}}$ and $A^{LO^{++}}$ provides better support for a wide range of unknown target domains as compared to the same by $H_{ERM}$. Thus, in Eq. 2, $\epsilon(h \in H_{ERM})$ acts as an upper bound for the target risk, particularly in source-free paradigm. Also, the equality holds as both $A^{DE^{++}}$ and $A^{LO^{++}}$ already include $H_{ERM}$ as a constituent subspace.

**Comparison between $DE^{++}$ and $LO^{++}$.** Though, both $DE^{++}$ and $LO^{++}$ are better alternatives over ERM, it is not possible to write a general inequality involving only the target errors for $DE^{++}$ and $LO^{++}$ configurations. Note that, as shown in Fig. 2, for certain target scenarios, target error for $DE^{++}$ would be less than the same for $LO^{++}$ and vice versa. However, considering a reasonable domain-shift among the source domains, $LO^{++}$ provides lower target error over $DE^{++}$ for a wide range of practical target scenarios (see Fig. 2C). $DE^{++}$ wins particularly for cases when $p_{t_i} = p_{t_j}$, for $i \in [K]$ which is generally quite rare. $LO^{++}$ wins for a wide range of unique target scenarios.

### 3.1.2 Preparing virtual multi-source domains

Having identified $LO^{++}$ as the best option, we focus on obtaining the multi-source data. Though, we intend to expand our source-data horizon, we are restricted to a single labeled source domain. Thus, we plan to use diverse data augmentations to simulate a multi-source scenario.

**Characterizing multi-domain data.** Consider a hypothetical data generation process [61] for the source domain: A data generator $\phi$ uses the causal class factor $f_y$ and the non-causal domain-related factor $f_s$ to construct a data sample $x_s = \phi(f_y, f_s)$. Next, a set of domain-varying class-preserving augmentations $\{T_i\}_{i=1}^{K}$ are applied to obtain,

$$
x_{s_i} = T_i(x_s) = \phi(f_y, f_1 + \gamma_i f_s); \quad \gamma_i \in \mathbb{R}
$$

(3)

Here, $T_i$ modifies the original domain-specific factor $f_s$ by a weight $\gamma_i$ (without altering $f_y$) and introduces a new augmentation related domain-specific factor $f_i$. Thus, the augmentations modify the non-causal factors to simulate novel domains. The augmented datasets are realized by pairing the input with the corresponding label and are represented as $(x_{s_i} = T_i(x_s), y_s) \in D_{s_i}$, $\forall i \in [K]$.

**Effect of number of source domains $K$.** Having a very high $K$ would lead to significant overlap of the the leave-one-out subspaces with the ERM, i.e. nullify the advantage of $LO^{++}$. Further, a high $K$ would induce a higher computational complexity. Thus, it becomes essential to filter out augmentations through a principled selection criteria.

**Definition 2 (Augmentation selection criteria)** Using Eq. 3, an augmentation $T_i$ will be selected if $|\gamma_i| > 1$. We give a tractable surrogate for this condition, using a hypothesis $h_s = \arg\min_{h \in A} \epsilon(x_{s_i})$ trained only on $D_s$,

$$
\hat{\epsilon}(x_{s_i}) - \epsilon(x_{s_i}) > \tau;
$$

(4)

i.e. the gap between the empirical risks (i.e. $\hat{\epsilon}$) of $h_s$ on $D_s$, and $D_s$ should be greater than a threshold $\tau$. This ensures that $T_i$ exerts a substantial alteration in the image statistics equivalent to the style gap between two diverse domains.

Intuitively, an augmentation is selected if it can suppress ($i.e. |\gamma_i| < 1$) the original domain factor $f_s$. In practice, $\gamma_i$ is intractable in the absence of disentangled $f_y$ and $f_s$. Thus, we rely on Eq. 4 whose LHS expresses the gener-
A. Vendor-side training

B. Client-side training

Figure 3. A. SoMAN constitutes of a global-head with multiple leave-one-out heads (left). Vendor also trains cPAE to discourage prediction irregularities. B. Client receives SoMAN and cPAE from vendor, and extracts robust and meaningful pseudo-labels for target samples via the optimal head $H_f$, to perform source-free DA. The inference model uses only the optimal head $H_f$ (no computational overhead).

3.1.3 Vendor-side architecture and training

Architecture. Considering the advantages of LO++, we propose a Source-only Multi-Augmentation Network, SoMAN, which is essentially a multi-head architecture with a shared CNN backbone $F$ (see Fig. 3A). Along with a global output head $H_g$ which is optimized using ERM, we employ leave-one-out heads $\{H_i\}_{i=1}^K$ trained to be sensitive towards the corresponding AG (i.e. $T_i$) while being invariant to others. Formally, the global head is trained using all the augmented datasets $i.e. \mathcal{D}_s = \bigcup_{i=1}^K \mathcal{D}_s$, and each non-global head $H_i$ is trained using a head-specific dataset $\mathcal{D}_{s_i} = \mathcal{D}_s \setminus \mathcal{D}_s$.

Training procedure. The SoMAN architecture is trained by simultaneously optimizing the spatial segmentation losses computed at the end of each output head. This encourages $F$ to extract a rich multi-source representation which retains domain-sensitive cues (as a result of the leave-one-out setup) alongside the extraction of domain-generic features. We denote the output of global head as $h_g = H_g(F(x))$. Following a similar convention, output of the leave-one-out heads are denoted by $H_i = H_i(F(x))$. Thus, the final objective for end-to-end training of SoMAN is formulated as,

$$\min_{\theta} \sum_{i=1}^K \mathbb{E}_{(x,y) \in \mathcal{D}_{s_i}} [-\langle y, \log h_i \rangle] + \mathbb{E}_{(x,y) \in \mathcal{D}_{s_g}} [-\langle y, \log h_g \rangle]$$ (5)

Here, $\theta$ denotes a set of parameters from all the heads, i.e. $\theta_F, \theta_{H_g}, \{\theta_{H_i}\}_{i=1}^K$ while $\langle \cdot, \cdot \rangle$ represents the dot product of the two inputs. In practice, the expectations are computed by sampling mini-batches from the corresponding datasets.

3.1.4 Conditional prior-enforcing autoencoder (cPAE)

In dense prediction tasks such as semantic segmentation, not all predictions are equally likely. Though the target annotations are not available during the client-side training, we aim to explicitly impart the general knowledge of scene prior to constrain the solution space. The use of scene prior would encourage plausible scene segments while discouraging irregularities (see Fig. 3A) such as “car flying in the sky”, “grass on road”, “split car shape”, “merged pedestrians”, etc. We recognize that the SoMAN may lack the ability to capture the above discussed inductive bias.

How can structural inductive bias be captured? We propose a conditional Prior-enforcing Auto-Encoder (cPAE), denoted by $Q$, that refines the predicted segmentation maps (seg-maps) conditioned on domain-generic features extracted from SoMAN. Instead of training it as a plain auto-encoder, we plan to train it as a denoising auto-encoder. The question that arises here is: how do we simulate noise for the cPAE inputs? We take advantage of sensitivity of leave-one-out heads to the corresponding AGs to simulate noisy seg-maps. Thus, the cPAE output distribution is $Q(y|F_g(x_s), \hat{y})$ where $\hat{y} = H_i(F(x_s))$. $F_g$ consists of the backbone $F$ and the first block of $H_g$ such that $F_g(x_s)$ are domain-generic features since $H_g$ is trained using all AGs. We train the cPAE to align its output distribution with the true source label distribution $p_s$ as follows

$$\min_{\theta_Q} \sum_{i=1}^K \mathbb{E}_{(x,y) \in \mathcal{D}_{s_i}} [\text{KL}(p_s(y), Q(y|F_g(x), \hat{y}))]$$ (6)

Here, KL indicates the Kullback-Leibler divergence. In practice, cross-entropy loss between the cPAE output and ground truth seg-map is used, derived from the KL term.

3.2. Client-side Strategy

Since the client can access only unlabeled target data $x^t \in \mathcal{D}^t$, we propose the use of self-training for this source-free adaptation step. However, this presents two caveats.

a) Risk of overfitting to wrong overconfident predictions. To counter this, we propose to utilize the multiple heads of
We propose to utilize the optimal head of the vendor provided SoMAN and the cPAE to generate reliable pseudo-labels.

b) Loss of task-relevant information. To avoid this, we aim to preserve the task-specific knowledge of the vendor model. While prior arts trained the entire model, we propose to train only a handful of weights belonging to the later layers of \( F \) while others are frozen from vendor-side. The frozen output heads hold useful, domain-generic, task-related inductive bias. It also constrains the optimization to operate within the hypothesis subspace of the vendor-side initialization. Thus, the client can leverage the vendor’s foresighted preparation to avoid sub-optimal solutions.

3.2.1 Pseudo-label extraction via cPAE

Since pseudo-labels are the only supervision signal in the proposed source-free self-training, it is crucial to ensure that they are highly informative and reliable. To this end, we propose to utilize the optimal head of the vendor provided SoMAN and the cPAE to obtain improved pseudo-labels. We consider the optimal head as the one that produces the lowest average self-entropy for the target training dataset. Formally, \( H_t \) is the optimal head where \( i^* = \arg \min_{i \in \{g, [K]\}} \sum_{x \in D_t} \{-h_i, \log h_i\} \) where \( h_i = H_i(F(x)) \). The optimal prediction can be represented as \( Q(h_{i^*}) \). Note that we denote the cPAE output as \( Q(h_{i^*}) \) omitting the conditional feature input for simplicity.

Using the optimal prediction, we follow [43, 102] for the confidence thresholding method. Particularly, we choose the top 33\% of the most confident pixel-level predictions per class over the entire target training set. This gives a target pseudo-labeled subset \((x_t, \hat{y}_t) \in D_t\) for self-training. Note that, the unselected pixels are assigned a separate, ‘unknown’ class which is not considered in training.

3.2.2 Source-free adaptation via self-training

We perform three rounds of self-training, following [90], where each round consists of pseudo-label extraction in an offline manner followed by supervised training on the extracted pseudo-labels. Entropy minimization is used as a regularizer during self-training. Further, we use the shared backbone \( F \) along with the optimal head, \( H_{i^*} \), for both self-training and test-time inference. Formally,

\[
\min_{\theta_F} \mathbb{E}_{(x_t, \hat{y}_t) \in D_t} \left[ -\langle \hat{y}_t, \log H_{i^*}(F(x_t)) \rangle \right]
\]

3.2.3 Test-time inference

As we propose only optimal head (i.e. \( H_{i^*} \)) self-training, our inference-stage model is \( H_{i^*}(F(x_t)) \) as shown in Fig. 3B. However, cPAE provides a further improvement in performance if used during inference. But, unless otherwise specified, the experiments use only \( H_{i^*}(F(x_t)) \) for self-training and evaluation, for a fair comparison. Note that, ‘w/ cPAE’ means that cPAE was used only for pseudo-label extraction.

4. Experiments

We perform a thorough evaluation of our approach against state-of-the-art prior works across multiple settings.

4.1. Experimental Settings

a) Network architectures. Following [43, 90], we employ 2 widely-used network architectures for the DA setting on semantic segmentation, DeepLabv2 [4] with ResNet101 [19] backbone and FCN8s [47] with VGG16 [69] backbone.

b) Datasets. We extensively evaluate the proposed approach on two popular synthetic-to-real benchmarks i.e. GTA5\( \rightarrow \)Cityscapes and SYNTHIA\( \rightarrow \)Cityscapes. We provide the complete implementation details in the Suppl.

c) Evaluation metric. Following [43, 90], we compute per-class IoUs as well as mean IoU (mIoU) over all 19 classes for the GTA5\( \rightarrow \)Cityscapes task. For SYNTHIA\( \rightarrow \)Cityscapes, we report the same for 13 and 16 classes because of the lower number of overlapping classes. Following [53, 81, 95], we use multi-scale testing. Due to space limitations, we report mean IoUs for class-groups\(^2\) instead of reporting IoUs for each individual class.

d) Augmentations. We select the following \( K = 5 \text{ AGs} \) (see Fig. 2D) using Definition 2 with the mIoU metric.

\begin{itemize}
  \item **Aug-A** (FDA [90]): This uses Fourier transform to transfer style from a reference image while retaining the semantic features [89] of the input. While FDA [90] transfers the style from target images, we do not access target data for vendor-side training. We use a small subset from style transfer dataset [24] and random noise as reference images.
  \item **Aug-B** (Style augmentation [26]): This technique uses a deep style transfer network for style randomization by randomly sampling a style embedding from a multivariate normal distribution instead of using reference style image. This provides practically infinite number of stylization options.
  \item **Aug-C** (AdaIN [24]): This uses Adaptive Instance Normalization (AdaIN) layers to inject style from a given reference image. In contrast to Aug-B, this provides a way to stylize images using a desired style image. We use a small subset from style transfer dataset [24] as reference images.
  \item **Aug-D** (Weather augmentation) [28, 54]: We use realistic weather augmentations to generate varying levels of snow and frost in the images. Compared to other AGs, this simulates realistic variations in the road scene images.
  \item **Aug-E** (Cartoon augmentation) [28]: This technique generates cartoonized versions of input images. This augmentation is diverse and useful as it produces almost texture-less images as in cartoons or comic books.
\end{itemize}

\(^2\)Background (BG) - building, wall, fence, vegetation, terrain, sky; Minority Class (MC) - rider, train, motorcycle, bicycle; Road Infrastructure Vertical (RIV) - pole, traffic light, traffic sign; Road Infrastructure Ground (RIG) - road, sidewalk; and Dynamic Stuff (DS) - person, car, truck, bus.
Table 1. Quantitative evaluation on GTA5→Cityscapes. Performance on different segmentation architectures: A (DeepLabv2 ResNet-101), B (FCN8s VGG16). SF indicates source-free adaptation. See Suppl. for the extended table with per-class IoUs. Ours (V) indicates use of our vendor-side AGs with prior art, * indicates results produced using the released code of prior arts.

| #   | Method               | Arch. | SF | BG | MC | RIV | RIG | DS | mIoU |
|-----|----------------------|-------|----|----|----|-----|-----|----|------|
| 1   | PLCA [29]            | A     | 57.3| 28.3|31.1|57.2 |60.2 |47.7|
| 2   | CrCDA [23]           | A     | 57.5| 24.5|33.8|73.9 |57.6 |48.6|
| 3   | RPT [95]             | A     | 62.5| 34.9|42.0|67.3 |59.4 |53.2|
| 4   | DACS [74]            | A     | 63.1| 24.2|45.9|64.7 |61.8 |52.1|
| 5   | FADA [81]            | A     | 61.9| 26.7|35.0|70.8 |56.7 |50.1|
| 6   | IAST [53]            | A     | 60.4| 32.6|34.1|76.5 |60.7 |52.2|
| 7   | Ours (V) + FADA*     | A     | 62.8| 27.1|35.3|71.1 |57.2 |50.6|
| 8   | Ours (V) + IAST*     | A     | 61.0| 33.1|34.6|77.1 |61.2 |52.8|
| 9   | Uurma [70]           | A     | ✓   |55.8|23.8|22.3|73.7 |52.8 |45.1|
| 10  | SRDA* [2]            | A     | ✓   |57.1|20.2|33.5|68.8 |51.9 |45.8|
| 11  | Ours (w/o cPAE)      | A     | ✓   |61.8|30.3|35.1|69.2 |60.8 |51.6|
| 12  | Ours (w/cPAE)        | A     | ✓   |62.8|33.4|36.2|72.0 |66.4 |53.4|
| 13  | LTIR [30]            | B     | ×   |58.6|14.0|26.5|73.5 |42.5 |42.3|
| 14  | FADA [81]            | B     | ×   |57.7|16.3|25.8|71.7 |50.1 |43.8|
| 15  | PCEDA [89]           | B     | ×   |56.4|20.5|31.2|67.5 |49.5 |44.6|
| 16  | SPDA [46]            | B     | ✓   |51.6|7.8 |15.9|56.6 |35.8 |35.8|
| 17  | Ours (w/o cPAE)      | B     | ✓   |54.7|19.9|27.3|66.2 |50.3 |43.4|
| 18  | Ours (w/cPAE)        | B     | ✓   |49.9|30.3|32.9|74.9 |50.8 |45.9|

Table 2. Ablation study for GTA5→Cityscapes. * indicates 3 rounds of self-training after the mentioned method. The client-side ablations begin from the best vendor-side model.

| Method               | mIoU |
|----------------------|------|
| Vendor-side          |      |
| Standard single-source* | 44.4 |
| Multi-source ERM*    | 47.6 |
| Domain-experts+++ (DE+++)* | 48.0 |
| Leave-one-out+++ (LO+++)* | 51.6 |
| w/o cPAE              | 51.6 |
| + Inference via cPAE | 52.5 |
| w/cPAE                | 53.4 |
| + Inference via cPAE | 54.2 |

| Client-side          |      |
|----------------------|------|
| w/o cPAE              | 51.6 |

4.2. Discussion

We provide an extensive ablation study of both the vendor-side and the client-side preparation. Further, we show that our approach generalizes across novel target scenarios and is compatible to online domain adaptation.

4.2.1 Comparison with prior arts.

We compare our proposed approach with prior arts in Table 1 and 3. We also compare our vendor-side approach with prior DG works in Table 4. Our method achieves state-of-the-art performance across all benchmarks. We also present the qualitative evaluation of our approach in Fig. 4.

Our proposed client-side adaptation is more scalable compared to prior works like PCEDA [89], RPT [95], IAST [53] in two ways. First, our method does not require image-to-image translation networks (PCEDA) or adversarial training (RPT, IAST) thereby reducing the adaptation complexity. Also note that the frozen cPAE is used only to obtain better pseudo-labels and is not involved in backpropagation for adaptation training. Second, the client can perform adaptation to multiple different target domains without the complex vendor-side training and without access to the source data. We study the second aspect further in the paper. 

a) Comparison with source-free prior arts. We implemented [2] for GTA5→Cityscapes (see #10-12 in Table 1) since they only report results for single object segmentation. We outperform their approach by a significant margin (8.1%). We also compare with concurrent source-free works [46, 70] (see #9 vs. #12, #16 vs. #18 in Table 1 and #7 vs. #9 in Table 3) and outperform them by ~12%.

b) Disentangling the gains from use of augmented data. We show the results for 2 prior arts [53, 81] using our vendor-side AGs during training (#5-8 in Table 1). While the performance improves compared to that originally reported, our proposed method (#12 in Table 1) still outperforms them. Thus, the improvement of our proposed method depends not only on the use of AGs but also on the multi-head, leave-one-out SoMAN framework and the cPAE.

4.2.2 Ablation study

Table 2 reports a detailed ablation to independently analyse the components of the vendor and client side strategies.

First, we evaluate the effectiveness of the proposed vendor-side strategies. For a fair comparison, we use a consistent client-side training for all the vendor-side ablations. As a baseline, we employ a standard (unaugmented) single-
source-trained model. The ERM model gives an improvement of 3.2% over the baseline. Next, we evaluate $DE^{++}$ and observe an improvement of 0.4%. $LO^{++}$ gives a further improvement of 3.6% over $DE^{++}$. This shows the clear superiority of $LO^{++}$ over both ERM and $DE^{++}$.

Second, under client-side ablation, cPAE for pseudo-label extraction gives a boost of 1.8%. Further, using cPAE for inference gives an additional 0.8-0.9% improvement.

### 4.2.3 Analyzing cross-dataset generalization

Unlike prior arts which assume concurrent access to source and target (inculcates target-bias), our target-free vendor-side model is expected to generalize well to unseen targets. To this end, Table 5 shows our generalizability to other road-scene datasets, such as Foggy-Cityscapes and NTHU-Cross-City [11]; before (#1-3) and after (#4-8) self-training on the related real domain, i.e. Cityscapes. Among different variants, we achieve a superior average generalization even without concurrent access to samples from the related domain, Cityscapes. Note that concurrent access is beneficial to better characterize the domain gap.

### 4.2.4 Compatibility to online domain adaptation

Online adaptation [27, 52] refers to a deployment setting where the model is required to continuously adapt to the current working conditions. The current state of the model may overcome its past domain-biases to perform the best at a given scenario. The proposed client-side training can be seen as an online adaptation algorithm. Here, the frozen parameters of the multi-head SoMAN helps to retain task-specific knowledge while allowing adaptation to unlabeled samples from the new environment. In the last section of Table 5, the initial Cityscapes adapted SoMAN is independently adapted to different secondary domains under Foggy-Cityscapes and NTHU-Cross-City. We also compare our results with recent Cityscapes→NTHU-Cross-City works (#9-12) that concurrently access labeled Cityscapes and unlabeled NTHU-Cross-City datasets. The improved performance shows our compatibility to online adaptation.

### 5. Conclusion

We introduced a source-free DA framework for semantic segmentation, recognizing practical scenarios where source and target data are not concurrently accessible. We cast the vendor-side training as multi-source learning. Based on theoretical insights, we proposed SoMAN that balances generalization and specificity using the systematically selected AGs without access to the target. To provide a strong support for the dense prediction task, cPAE is trained to denoise segmentation predictions and improve pseudo-label quality for client-side source-free self-training. Extending this framework to more DA scenarios involving category-shift can be a direction for future research.

**Acknowledgements.** This work was supported by MeitY (Ministry of Electronics and Information Technology) project (No. 4(16)2019-ITEA), Govt. of India.
Supplementary Material

In this supplementary, we provide detailed theoretical insights, extensive implementation details with additional qualitative and quantitative performance analysis. Towards reproducible research, we have released our code and trained network weights at our project page: https://sites.google.com/view/sdfaseg. This supplementary is organized as follows:

- Section A: Notations (Table 6)
- Section B: Extended theoretical insights
  - Discussion on Result 1 (Sec. B.1)
  - Augmentation selection criteria (Sec. B.2)
- Section C: Implementation details
  - Experimental settings (Sec. C.1, Table 8)
  - Vendor-side training (Sec. C.2, Algo. 1)
  - Client-side training (Sec. C.3, Algo. 2)
  - Optimization details (Sec. C.4)
- Section D: Analysis
  - Optimal choice of AGs (Sec. D.1, Fig. 5B, Table 7)
  - Empirical evaluation of Result 1 (Sec. D.2, Table 9)
  - Impact of cPAE (Sec. D.3, Fig. 5C)
  - Time complexity analysis (Sec. D.4, Table 10)
  - Qualitative analysis (Sec. D.5, Fig. 6, 7, 8, 9)
  - Quantitative analysis (Sec. D.6, Table 11, 12)

A. Notations

We summarize the notations used in the paper in Table 6. The notations are listed under 5 groups viz., Distributions, Datasets, Networks, Samples/Outputs and Theoretical.

B. Extended theoretical insights

B.1. Discussion on Result 1

To analyze possible solutions to the paradigm defined in Definition 1 of the paper, we introduce three configurations: 1) ERM (empirical risk minimization) i.e. weighting each multi-source domain equally, 2) domain-experts++ (DE++) i.e. an ERM subspace together with K subspaces formed from one specific domain each, 3) leave-one-out++ (LO++) i.e. an ERM subspace together with K subspaces formed by all domains except one each. We restate the result here:

\[
\begin{align*}
\epsilon_i(h \in A^{LO++}) & \leq \epsilon_i(h \in H^{ERM}) \\
\epsilon_i(h \in A^{DE++}) & \leq \epsilon_i(h \in H^{ERM})
\end{align*}
\]  

(8)

Proof. We give the proof by contradiction. Let the optimal hypothesis from LO++ have a higher target error than ERM i.e. \( \epsilon_i(h \in A^{LO++}) > \epsilon_i(h \in H^{ERM}) \). However, this implies that it cannot be the optimal hypothesis because an ERM hypothesis within LO++ has a lower target error. This is a contradiction which proves the result. The same can be shown for the DE++ case. In other words, the ERM subspace in DE++ or LO++ can provide the optimal hypothesis in the worst case and the equality in Eq. 8 will hold.

B.2. Augmentation selection criteria

From Eq. 3 of the paper, we know that the augmentation \( T_i \) modifies the non-causal domain-related factor \( f_s \) by a weight \( \gamma_i \) and introduces a new augmentation-related factor \( f_y \) without altering the causal class factor \( f_y \). For a general augmentation, we cannot restrict \( \gamma_i \) i.e. \( \gamma_i \in \mathbb{R} \). However, since we need to restrict the number of domains \( K \), augmentations need to be filtered out. The important criteria for this filtering is the ability of the augmentation to simulate new domains while suppressing the original domain. Assuming that each augmentation is capable of introducing new domain-specific features, it is crucial for it to reduce the influence of the original domain. Thus, in Definition 2 in the paper, the selection criteria is \( |\gamma_i| < 1 \).

Due to the hypothetical nature of the decomposition of a sample into class-specific and domain-specific factors, it is not feasible to use the criteria on \( \gamma_i \) directly. However, mod-
els usually rely on both class-specific and domain-specific factors for prediction [60]. Thus, the performance of a standard single-source-trained model on augmented samples can be a good measure of the residual original domain dependency. An augmentation causing low performance for the pretrained model is likely to be suppressing the original domain factors. In other words, the model is unable to latch onto the original domain factors for prediction. This gives the surrogate condition in Eq. 4 of the paper. However, the candidate augmentations must be manually filtered to not select those that perturb the class-relevant factors since it is not ensured through this criteria.

C. Implementation Details

In this section, we describe the network architectures, datasets, the training process used for vendor-side and client-side training and other implementation details.

C.1. Experimental settings

a) Network architectures. Following [43, 90], we employ 2 widely-used network architectures for the DA setting on semantic segmentation, DeepLabv2 [4] with ResNet101 [19] backbone and FCN8s [47] with VGG16 [69] backbone. For DeepLabv2-ResNet101 and FCN8s-VGG16, we define the shared backbone $F$ up to Layer3 block and Conv4 block respectively. The remaining networks are taken as output head for both. Thus, at the inference stage, our model has exactly the same amount of parameters as used in the prior-arcs. We use a fully convolutional architecture for the conditional Prior-enforcing Auto-Encoder (cPAE) with asymmetric encoder and decoder as shown in Table 8.

b) Datasets. We extensively evaluate the proposed approach on two popular synthetic-to-real benchmarks i.e., GTA5→Cityscapes and SYNTHIA→Cityscapes. For GTA5 [63], we resize the image to 1280 × 720 before randomly cropping to 1024 × 512. Whereas, for SYNTHIA [64], we resize to 1280 × 760 and random crop to 1024 × 512 following [90]. For Cityscapes [13], we resize the image to 1024 × 512. For GTA5 we use 24500 images for training and 466 images for validation. Whereas, for SYNTHIA we use 9000 images for training and 400 images for validation. For client-side evaluation, we use Cityscapes training dataset for training and the standard validation set for testing [90]. Following previous works [53, 81, 101], we use multi-scale testing to report the final performance.

c) Augmentations. We provide extra details about the AGs to enhance the reproducibility of our experiments.

Aug-A: We used images from a style transfer dataset [24] and release them with the code. For random noise, we sample uniformly from 0 to 255 for every pixel location.

Aug-B: We used this augmentation from the code release of [26] as provided. No controllable parameter available.

Aug-C: We set the strength of stylization ($\alpha$) to 0.3 which balances stylization and content preservation.

Aug-D: For snow and frost augmentation, we uniformly sample the severity between 1 and 3 (max. severity 5 possible in [28]) to balance stylization and content preservation.

Aug-E: No controllable parameter in cartoon AG [28].

C.2. Vendor-side training

The vendor-side training involves multi-head SoMAN training followed by cPAE training, as described in Algo. 1. In SoMAN training, at each iteration, an image sampled from the source dataset $D_s$ is augmented using a random AG. Since we train each head in a leave-one-out manner, the global head and $K − 1$ non-global heads are trained at each iteration (L2-L14). We update the parameters of each head using a separate optimizer (L13). The momentum parameters in the optimizers adaptively scale the gradients thereby avoiding loss-scaling hyperparameters.

In cPAE training, we use the trained non-global heads from SoMAN to generate noisy seg-maps for denoising auto-encoder training. We augment source samples with a randomly chosen $i^{th}$ AG (L17-19). Next, these are passed through the corresponding $i^{th}$ non-global head which was trained to be sensitive to that AG, thereby yielding noisy seg-maps (L20). The cPAE predictions for these noisy seg-maps are used to compute the cross-entropy loss with the ground truth seg-maps. This loss is minimized using a SGD optimizer to update the cPAE parameters (L21-L23).

C.3. Client-side training

The client-side training requires optimal head identification, pseudo-label extraction and self-training, as in Algo. 2.

In optimal head identification, for a given target, the head with the lowest $\epsilon_i(h)$ has to be chosen as per Result 1. Since the computation of $\epsilon_i(h)$ is intractable, we choose a proxy i.e. average self-entropy on the target training set (L2). Intuitively, the head closest to the target domain will be selected as it would be the most confident (lowest self-entropy).

In pseudo-label extraction, we first process the entire target training dataset (L3-L11) and store the spatial class predictions (L8) and the prediction probabilities (L9). To avoid noisy predictions, we determine class-wise thresholds (L12-L16) which are set at 33% of the most confident predictions per class. Finally, we apply the class-wise thresholds (L17-L23) and assign an unlabeled ‘unknown’ class to the pixels which do not satisfy the threshold. These unlabeled pixels are not considered in the loss computation during training.

In self-training, we use the pseudo-labeled target dataset to train in a supervised manner (L24-30). Specifically, we train a block under the shared backbone $F$ (L29) using the optimal head predictions and pseudo-labels. For DeepLabv2-ResNet101, this block is Layer3 while for FCN8s-VGG16, it is Conv3+Conv4 blocks. We find this
to perform better than training the entire backbone $F$. We perform 3 rounds of offline pseudo-label extraction and self-training, following [90]. Further, we find that the performance does not increase with more than 3 rounds.

C.4. Optimization details

We implement our framework on PyTorch [58]. Following [43], for DeepLabv2-ResNet101, we use SGD optimizer with momentum 0.9, initial learning rate 2.5e-4 with a polynomial learning rate decay with power 0.9, and weight decay 5e-4. For FCN8s-VGG16, we use Adam [32] optimizer with initial learning rate 1e-5 with step decay of 0.1 at every 20k steps. We train for 50k iterations each in vendor-side and in each round of self-training. With mixed precision, we use batch size 2 on a GTX1080Ti GPU.

D. Analysis

In this section, we analyze the choice of AGs, the empirical evaluation of Result 1 and the impact of cPAE. We show extended quantitative and qualitative evaluations of our approach and training time comparisons with prior arts.

D.1. Optimal choice of AGs

The choice of number of AGs and the AGs themselves is critical to the success of vendor-side training. We describe the AG candidates and those used by prior arts in Table 7. Firstly, it is important to choose diverse AGs to facilitate the learning of both domain-invariant and domain-specific features by the proposed SoMAN. Secondly, a higher number of AGs ($K$) and correspondingly non-global output heads incur additional computational cost in the vendor-side training. Thus, it is crucial to determine a low enough $K$ that yields a significant performance improvement.

Towards the first, Fig. 5A shows the selection of diverse AGs from a set of candidates using the performance of a standard single-source trained model. This is according to Definition 2 in the paper using mIoU (task metric) with a threshold of 25%. We observe that weaker augmentations that do not produce enough domain-shift get filtered out.

Towards the second, we analyze the effect of number of AGs on the performance in Fig. 5B. Particularly, we evaluate vendor-side trained models with varying number of AGs used during training. Since there are multiple ways to choose from the set of candidate AGs, we choose the most diverse ones first to ensure the best possible performance for a lower $K$. In other words, the AG giving the most deterioration in mIoU for a standard single-source trained model is chosen first. Fig. 5A shows that the order is E, B, C, D, A. Using this order to choose AGs for $K = \{1, 2, \ldots, 5\}$, we observe that performance saturates as $K$ reaches 5. Thus, we infer that the computational burden of adding more AGs would not result in a substantial performance improvement.

D.2. Empirical evaluation of Result 1

To empirically evaluate Result 1 of the paper, we measure the performance of each SoMAN head for a variety of target scenarios as shown in Table 9. We observe that different heads of the SoMAN give the best performance in different target scenarios. Further, in most scenarios, at least one of the leave-one-out heads performs better than ERM. This is in line with Result 1 i.e., one of the leave-one-out heads

| Method   | TF | Description                                      |
|----------|----|--------------------------------------------------|
| FDA [90] | ×  | Target images as reference for stylization in Fourier domain. |
| LTIR [30] | ×  | Uses style-swap [6] with target images as reference. |
| BDL [43] | ×  | Image-to-image translation for source→target conversion. |
| LDR [86] | ×  | Image-to-image translation for target→source conversion. |
| Ours-A   | ✓  | FDA [90] with random and style images as reference. |
| Ours-B   | ✓  | Stylization by randomly sampled style embedding [36]. |
| Ours-C   | ✓  | AdaIN [24] layers for stylization using reference images. |
| Ours-D   | ✓  | Varying levels of weather augmentations [28]: frost and snow. |
| Ours-E   | ✓  | Converts image into texture-less cartoon-like image [28]. |
| Ours-W1  | ✓  | Blurring using a $5 \times 5$ average filter. |
| Ours-W2  | ✓  | Rotating image by an angle $\in [-15, 15]$ degrees. |
| Ours-W3  | ✓  | Adding random noise to the image. |
| Ours-W4  | ✓  | Edge-preserved smoothing using bilateral filtering. |
Algorithm 1 Pseudo-code for vendor-side training

1: Input: source dataset $D_s$, standard single-source trained model $F_s, H_s$

▷ Note that CE denotes class-weighted cross-entropy.

Step 1: SoMAN training

2: Initialize $F$, $H_g, \{H_i\}_{i=1}^K$ using $F_s, H_s$
3: for iter $< \text{MaxIter}$ do:
4: $x_s, y_s \leftarrow$ batch sampled from $D_s$
5: $i_1 \leftarrow \text{rand}(1, K)$
6: $\hat{x}_s \leftarrow T_{i_1}(x_s)$
7: $h_g \leftarrow H_g(F(\hat{x}_s))$
8: Compute $L_g \leftarrow \text{CE}(h_g, y_s)$
9: for $i$ in range($K$) except $\{i_1\}$ do:
10: $h_i \leftarrow H_i(F(\hat{x}_s))$
11: Compute $L_i \leftarrow \text{CE}(h_i, y_s)$
12: end for
13: update $\theta_{H_g, \{H_i\}_{i=1,i\neq i_1}}$ by minimizing $\sum_{i=1}^K L_i$ using separate optimizers
14: end for

Step 2: cPAE training

15: Randomly initialize cPAE $Q$; freeze $F, H_g, \{H_i\}_{i=1}^K$ from Step 1 training.
16: for iter $< \text{MaxIter}$ do:
17: $x_s, y_s \leftarrow$ batch sampled from $D_s$
18: $i \leftarrow \text{rand}(1, K)$  
19: $x_{s_i} \leftarrow T_i(x_s)$  
20: $h_i \leftarrow H_i(F(x_{s_i}))$
21: $\hat{h}_i \leftarrow Q(h_i, F_g(x_{s_i}))$
22: Compute $L_q \leftarrow \text{CE}(\hat{h}_i, y_s)$
23: update $\theta_Q$ by minimizing $L_q$ using SGD optimizer
24: end for

gives a lower or equal target risk than ERM.

For Foggy-Cityscapes (0.02) [66], we observe that ERM is optimal which can be considered as a worst-case scenario (large domain-shift). On the other hand, Foggy-Cityscapes (0.01) and (0.005) both have LO-E as the optimal head since they represent similar domain-shifts. Further, for NTHU-Cross-City [11], different heads are optimal for different cities since each city presents a different domain-shift.

D.3. Impact of cPAE

We train the proposed cPAE as a denoising autoencoder to encourage spatial regularities in the segmentation predictions. In Fig. 5C, we analyze the effect of inference via cPAE. Particularly, we hypothesize that the cPAE should not distort regions that were correctly predicted by the segmentation network. This is desirable to retain the inductive bias in the absence of target labels. For a given segmenta-

Algorithm 2 Pseudo-code for client-side training

1: Input: Trained SoMAN ($F, H_g, \{H_i\}_{i=1}^K$) and cPAE $(Q)$ from vendor, unlabeled target dataset $D_t$

▷ Let $[\cdot]$ denote the indexing operation, $\cdot || \cdot$ denote the append operation, $| \cdot |$ denote the cardinality, and $C$ be the number of classes.

Step 1: Optimal head identification

2: $i' \leftarrow \arg\min_{i \in \{g, |K|\}} \sum_{x \in D_t} |\{h_i, \log h_i\}|$ where $h_i = H_i(F(x)) \forall i$  
▷ Lowest average self-entropy

Step 2: Pseudo-label extraction

3: $Y_p \leftarrow \{\}$  
▷ Empty ordered list
4: $W \leftarrow \{\}$  
▷ Empty ordered list
5: $X_p \leftarrow \{\}$  
▷ Empty ordered list
6: for $x \in D_t$ do:
7: $h = Q(H_{i'}(F(x)), F_g(x))$  
8: $y_p \leftarrow \arg\max_{c \in C} h[c]$  
▷ Class predictions
9: $w \leftarrow \max_{c \in C} h[c]$  
▷ Predicted class probabilities
10: $W, Y_p, X_p \leftarrow W \parallel w, Y_p \parallel y_p, X_p \parallel x$
11: end for
12: $t \leftarrow \{\}$  
▷ List of class-wise thresholds
13: for $c \in \text{range}(C)$ do:
14: Store all prediction probabilities of class $c$ in $p_x$
15: $p_x \leftarrow W[Y_p == c]$
16: $p_x \leftarrow \text{sort}(p_x)$
17: Set threshold at top 33% most confident predictions
18: $t \leftarrow t \parallel p_x[0.66|p_x|]$
19: end for
20: $D_{t} \leftarrow \{\}$  
▷ Empty ordered list
21: for $y_p, w, x_p$ in $Y_p, W, X_p$ do:
22: for $c \in \text{range}(C)$ do:
23: Assign class-id, $C + 1$ representing ‘unknown’, to pixels with probability $< \text{class threshold } t[c]$
24: $y_p[(w < t[c]) \& (y_p == c)] \leftarrow C + 1$
25: end for
26: end for
27: $D_{t} \leftarrow D_{t} \parallel (x_p, y_p)$
28: end for

Step 3: Source-free self-training adaptation

24: Obtain $F, H_i'$ from vendor (or last self-training round)
25: for iter $< \text{MaxIter}$ do:
26: $x_t, y_t \leftarrow$ batch sampled from $D_t$
27: $\hat{h} \leftarrow H_{i'}(F(x_t))$
28: Compute $L_t \leftarrow \text{CE}(\hat{h}, y_t)$
29: update trainable parameters of $\theta_F$ by minimizing $L_t$ using SGD optimizer
30: end for

tion network, we determine these regions, denoted as +ve-regions, and the regions where the segmentation network failed, denoted as -ve-regions. Next, we use the cPAE on
the segmentation predictions and evaluate the performance in the two previously determined regions. We also evaluate repeated inference via cPAE \textit{i.e.}, passing the output of the cPAE through the cPAE again. We observe that the performance in the +ve-region is almost the same while it improves in the -ve-region. Further, we observe that repeated inference via cPAE does not give any significant improvement. Thus, we resort to inferring via cPAE only once.

D.4. Time complexity analysis

Our proposed client-side adaptation trains the shared backbone $F$ partially and does not require any additional networks like adversarial discriminators during the training. Further, we offer a simple adaptation pipeline without requiring access to source data. These factors lead to a lower training time (Table 10) for our client-side adaptation compared to prior arts while maintaining state-of-the-art adaptation performance. This makes it suitable for practical and even online adaptation scenarios.

We perform the analysis (Table 10) by measuring the average time taken for forward pass, backward pass and the optimizer step (network weights update) for each method. For FDA and PCEDA, the time per iteration is higher since they train the entire model while using FFT-based augmentation and an image-to-image translation network respectively. For a fair comparison, we use batch size 1 (without automatic mixed precision) for all methods and evaluate on a machine with Intel Xeon E3-1200 CPU, 32GB RAM and a single 11GB NVIDIA GTX1080Ti GPU using Python 3, PyTorch 1.6 and CUDA 10.2.

D.5. Qualitative analysis

We provide extended qualitative results of our proposed approach on GTA5->Cityscapes [63, 13] in Fig. 9. Further, we provide extended qualitative results of our proposed approach on GTA5->Cityscapes [63, 13] in Fig. 9.
Figure 6. Paired samples for $cPAE$ training. $cPAE$ is trained as a denoising autoencoder to encourage structural regularity in segmentation predictions and alleviate merged-region (yellow circle) and splitted-region (blue circle) problems. Best viewed in color.

Figure 7. Examples of AGs applied to GTA5 dataset images. Notice the diversity in the augmentations. Best viewed in color.
Figure 8. Qualitative evaluation of GTA5→Cityscapes and online adapted models on NTHU-Cross-City and Foggy-Cityscapes datasets. The performance generally improves from vendor-side to client-side to online-adapted model. Best viewed in color.
Figure 9. Qualitative evaluation of the proposed approach. Vendor-side model generalizes better than baseline but performs worse than client-side due to the domain gap. Inculcating prior knowledge from cPAE structurally regularizes the predictions and overcomes merged-region (yellow circle) and splitted-region (blue circle) problems. Some failure cases are also shown (white circle). Best viewed in color.

Table 10. Training time comparison of adaptation step with prior arts. AN indicates whether additional networks are involved in the adaptation training.

| Method   | AN | Training time (per iter.) (seconds) |
|----------|----|------------------------------------|
| PCEDA [89] | ✓  | 0.94                               |
| FDA [90]  | ×   | 0.90                               |
| Ours      | ×   | 0.31                               |

We also observe some failure cases in Fig. 9 (indicated by white circles) where merged-region problems occur for smaller-sized classes in the scene. More explicit ways of inculating shape priors may improve the performance further. We plan to explore this direction in our future work.

D.6. Quantitative analysis

We provide extended quantitative results on the GTA5→Cityscapes and SYNTHIA→Cityscapes [64] benchmarks for semantic segmentation in Tables 11, 12. We obtain state-of-the-art performance across all settings even against non-source-free approaches.
Table 11. Quantitative evaluation on GTAS→Cityscapes. Performance on different segmentation architectures: A (DeepLabv2 ResNet-101), B (FCN8s VGG-16). SF indicates whether the method supports source-free adaptation. *Ours (V) indicates use of our vendor-side AGs with prior art and * indicates reproduced by us using released code. We observe better or competitive performance on minority classes like motorcycle compared to non-source-free prior arts.

| Method          | Arch SF road sidewalk building wall fence pole bicycle | mIoU |
|-----------------|---------------------------------------------|------|
| PLCA [29]       | A × 84.0 30.4 82.4 35.3 24.8 32.2 36.8 24.5 | 85.5 |
| CrCDA [23]      | A × 92.4 55.3 82.3 31.2 29.1 32.5 33.2 35.6 | 83.5 |
| PIT [51]        | A × 87.5 43.4 78.8 31.2 30.2 36.3 39.9 42.0 | 79.2 |
| TPLD [68]       | A × 94.2 60.5 82.8 36.6 16.6 39.3 29.0 25.5 | 85.6 |
| RPT [95]        | A × 89.7 44.8 86.4 44.2 30.6 41.4 51.7 33.0 | 87.8 |
| FADA [81]       | A × 91.0 50.6 86.0 43.4 29.8 36.8 43.4 25.0 | 86.8 |
| IAST [53]       | A × 94.1 58.8 85.4 39.7 29.2 25.1 43.1 34.2 | 84.8 |
| Ours (V) FADA*  | A × 91.2 51.0 86.6 36.0 30.3 37.1 43.7 25.2 | 87.9 |
| Ours (V) IAST*  | A × 94.8 59.4 86.2 40.5 29.5 25.5 43.8 34.7 | 85.9 |
| URMA [70]       | A ✓ 92.3 55.2 81.6 30.8 18.3 37.1 17.7 12.1 | 84.2 |
| SRD* [2]        | A ✓ 90.5 47.1 82.8 32.8 28.0 29.0 35.9 34.8 | 83.3 |
| Ours (w/o cPAE)| A ✓ 90.9 48.6 85.5 35.3 31.7 36.9 34.7 34.8 | 86.2 |
| Ours (w/ cPAE)| A ✓ 91.7 53.4 86.1 36.7 32.1 37.4 38.2 35.6 | 86.7 |
| BDL [43]        | B × 89.2 40.9 81.2 29.1 19.2 14.2 29.0 19.6 | 83.7 |
| LTIR [30]       | B × 92.5 54.5 83.9 34.5 25.5 31.0 30.4 18.0 | 84.1 |
| LDR [86]        | B × 90.1 42.2 82.0 30.3 21.3 18.3 35.5 23.0 | 84.1 |
| FADA [81]       | B × 92.3 51.1 83.7 33.1 29.1 28.5 28.0 21.0 | 82.6 |
| PCEDA [89]      | B × 90.2 44.7 82.0 28.4 28.4 24.4 33.7 35.6 | 83.7 |
| SFEDA [46]      | B ✓ 81.8 35.4 82.3 21.6 20.2 25.3 17.8 4.9 | 80.7 |
| Ours (w/o cPAE)| B ✓ 90.1 44.2 81.7 31.6 19.2 27.5 29.6 26.4 | 81.3 |
| Ours (w/ cPAE)| B ✓ 92.9 56.9 82.5 20.4 6.0 30.8 34.7 33.2 | 84.6 |

Table 12. Quantitative evaluation on SYNTHIA→Cityscapes. Performance on different segmentation architectures: A (DeepLabv2 ResNet-101), B (FCN8s VGG-16). mIoU and mIoU* are averaged over 16 and 13 categories respectively. SF indicates whether the method supports source-free adaptation.

| Method          | Arch SF road sidewalk building wall fence pole bicycle | mIoU |
|-----------------|---------------------------------------------|------|
| CAG [93]        | A × 84.8 41.7 85.5 - - - 13.7 23.0 | 86.5 |
| APODA [88]      | A × 86.4 41.3 79.3 - - - 22.6 17.3 | 80.3 |
| PyCDA [44]      | A × 75.5 30.9 83.3 20.8 0.7 32.7 27.3 33.5 | 84.7 |
| TPLD [68]       | A × 80.0 44.3 82.2 19.9 0.3 40.6 20.5 30.1 | 77.2 |
| USAMR [98]      | A × 83.1 38.2 81.7 9.3 1.0 35.1 30.3 19.9 | 82.0 |
| RPL [97]        | A × 87.6 41.9 83.1 14.7 1.7 36.2 31.3 19.9 | 81.6 |
| IAST [53]       | A × 81.9 41.5 83.3 17.7 4.6 32.3 30.9 28.8 | 83.4 |
| RPT [95]        | A ✓ 89.1 47.3 84.6 14.5 0.4 39.5 34.9 35.7 | 83.8 |
| URMA [70]       | A ✓ 59.3 24.6 77.0 14.0 1.8 31.5 18.3 32.0 | 83.1 |
| Ours (w/o cPAE)| A ✓ 90.0 44.6 80.1 7.8 0.7 34.4 22.0 22.9 | 82.0 |
| Ours (w/ cPAE)| A ✓ 90.5 50.0 81.6 13.3 2.8 34.7 25.7 33.1 | 83.8 |
| PyCDA [44]      | B × 80.6 26.6 74.5 2.0 0.1 18.1 13.7 14.2 | 80.8 |
| SD [15]         | B × 87.1 36.5 79.7 - - - 13.5 7.8 | 81.2 |
| FADA [81]       | B × 80.4 35.9 80.9 2.5 0.3 30.4 7.9 22.3 | 81.8 |
| BDL [43]        | B × 72.0 30.3 74.5 0.1 0.3 24.6 10.2 25.2 | 80.5 |
| PCEDA [89]      | B × 79.7 35.2 78.2 1.4 0.6 23.1 10.0 28.9 | 79.6 |
| Ours (w/o cPAE)| B ✓ 88.5 45.4 79.8 2.8 2.2 27.4 18.4 25.4 | 82.4 |
| Ours (w/ cPAE)| B ✓ 89.9 48.8 80.9 2.9 2.5 28.1 19.5 26.2 | 83.7 |
References

[1] Surbhi Aggarwal, Jogendra Nath Kundu, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Wamda: Weighted alignment of sources for multi-source domain adaptation. In BMVC, 2020.

[2] Mathilde Bateson, Hoel Kervadec, Jose Dolz, Herve Lombaert, and Ismail Ben Ayed. Source-relaxed domain adaptation for image segmentation. In MICCAI, 2020.

[3] Wei-Lun Chang, Hui-Po Wang, Wen-Hsiao Peng, and Wei-Chen Chiu. All about structure: Adapting structural information across domains for boosting semantic segmentation. In CVPR, 2019.

[4] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4):834–848, 2017.

[5] Minghao Chen, Hongyang Xue, and Deng Cai. Domain adaptation for semantic segmentation with maximum squares loss. In ICCV, 2019.

[6] Tian Qi Chen and Mark Schmidt. Fast patch-based style transfer of arbitrary style, 2016.

[7] Wuyang Chen, Zhiding Yu, Zhangyang Wang, and Anima Anandkumar. Automated synthetic-to-real generalization. In ICML, 2020.

[8] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In CVPR, 2019.

[9] Yuhua Chen, Wen Li, and Luc Van Gool. Road: Reality oriented adaptation for semantic segmentation of urban scenes. In CVPR, 2018.

[10] Yun-Chun Chen, Yen-Yu Lin, Ming-Hsuan Yang, and Jia-Bin Huang. Crdoco: Pixel-level domain transfer with cross-domain consistency. In CVPR, 2019.

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

[12] Jaehoon Choi, Taekyung Kim, and Changikk Kim. Self-ensembling with gan-based data augmentation for domain adaptation in semantic segmentation. In ICCV, 2019.

[13] Marius Cords, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In CVPR, 2016.

[14] Jiahua Dong, Yang Cong, Gan Sun, Yuyang Liu, and Xiaowei Xu. Cscsl: Critical semantic-consistent learning for unsupervised domain adaptation. In ECCV, 2020.

[15] Liang Du, Jingang Tan, Hongye Yang, Jianfeng Feng, Xiangyang Xue, Qibao Zheng, Xiaojing Ye, and Xiaolin Zhang. SSF-dan: Separated semantic feature based domain adaptation network for semantic segmentation. In ICCV, 2019.

[16] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In CVPR, 2019.

[17] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In NeurIPS, 2014.

[18] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In NeurIPS, 2005.

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

[20] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In ICML, 2018.

[21] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. arXiv preprint arXiv:1612.02649, 2016.

[22] Haoshuo Huang, Qixing Huang, and Philipp Krahenbuhl. Domain transfer through deep activation matching. In ECCV, 2018.

[23] Jiaxing Huang, Shijian Lu, Dayan Guan, and Xiaobing Zhang. Contextual-relation consistent domain adaptation for semantic segmentation. In ECCV, 2020.

[24] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In ICCV, 2017.

[25] Maximilian Ilse, Jakub M Tomczak, and Patrick Forr´e. Selecting data augmentation for simulating interventions. In ICML, 2021.

[26] Philip T Jackson, Amir Atapour-Abarghouei, Stephen Bonner, Toby P Breckon, and Boguslaw Obara. Style augmentation: Data augmentation via style randomization. In CVPR Workshops, 2019.

[27] Vidit Jain and Erik Learned-Miller. Online domain adaptation of a pre-trained cascade of classifiers. In CVPR, 2011.

[28] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2020.

[29] Guoliang Kang, Yunchao Wei, Yi Yang, Yueeting Zhuang, and Alexander Hauptmann. Pixel-level cycle association: A new perspective for domain adaptive semantic segmentation. In NeurIPS, 2020.

[30] Myeongjin Kim and Hyeran Byun. Learning texture invariant representation for domain adaptation of semantic segmentation. In CVPR, 2020.

[31] Minsu Kim, Sung hun Joung, Seungryong Kim, Junghun Park, Ig-Jae Kim, and Kwanghoon Sohn. Cross-domain
grouping and alignment for domain adaptive semantic segmentation. In AAAI, 2021.
[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.
[33] Jogendra Nath Kundu, Phani Krishna Uppala, Anuj Pahuja, and R. Venkatesh Babu. Adadepth: Unsupervised content congruent adaptation for depth estimation. In CVPR, 2018.
[34] Jogendra Nath Kundu, Nishank Lakkakula, and R. Venkatesh Babu. Um-adapt: Unsupervised multi-task adaptation using adversarial cross-task distillation. In ICCV, 2019.
[35] Jogendra Nath Kundu, Gaurav Singh Rajput, and Venkatesh Babu Radhakrishnan. Vrt-net: Real-time scene parsing via variable resolution transform. In WACV, 2020.
[36] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In CVPR, 2020.
[37] Jogendra Nath Kundu, Naveen Venkat, Ambareesh Revanur, Rahul M V, and R. Venkatesh Babu. Towards inheritable models for open-set domain adaptation. In CVPR, 2020.
[38] Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, Rahul M V, and R. Venkatesh Babu. Class-incremental domain adaptation. In ECCV, 2020.
[39] Suhyeeon Lee, Junhyuk Hyun, Hongje Seong, and Euntai Kim. Unsupervised domain adaptation for semantic segmentation by content transfer. In AAAI, 2021.
[40] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In ICCV, 2017.
[41] Guangrui Li, Guoliang Kang, Wu Liu, Yunchao Wei, and Yi Yang. Content-consistent matching for domain adaptive semantic segmentation. In ECCV, 2020.
[42] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In CVPR, 2020.
[43] Yunshe Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In CVPR, 2019.
[44] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. In ICCV, 2019.
[45] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In ICMIL, 2020.
[46] Yang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. In CVPR, 2021.
[47] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In CVPR, 2015.
[48] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Significance-aware information bottleneck for domain adaptive semantic segmentation. In ICCV, 2019.
[49] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. Adversarial style mining for one-shot unsupervised domain adaptation. In NeurIPS, 2020.
[50] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In CVPR, 2019.
[51] Fengmao Lv, Tao Liang, Xiang Chen, and Guosheng Lin. Cross-domain semantic segmentation via domain-invariant interactive relation transfer. In CVPR, 2020.
[52] Massimiliano Mancini, Hakan Karaoguz, Elisa Ricci, Patric Jensfelt, and Barbara Caputo. Kitting in the wild through online domain adaptation. In IROS, 2018.
[53] Ke Mei, Chuang Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. In ECCV, 2020.
[54] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484, 2019.
[55] Luigi Musto and Andrea Zinelli. Semantically adaptive image-to-image translation for domain adaptation of semantic segmentation. In BMVC, 2020.
[56] Fei Pan, Inkyu Shin, Francois Rameau, Seokju Lee, and In So Kweon. Unsupervised intra-domain adaptation for semantic segmentation through self-supervision. In CVPR, 2020.
[57] Xingang Pan, Ping Luo, Jianping Shi, and Xiaoou Tang. Two at once: Enhancing learning and generalization capacities via ibn-net. In ECCV, 2018.
[58] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In NeurIPS, 2019.
[59] Sujoy Paul, Yi-Hsuan Tsai, Samuel Schulter, Amit K. Roy-Chowdhury, and Mannoharan Chandraker. Domain adaptive semantic segmentation using weak labels. In ECCV, 2020.
[60] Xingchao Peng, Zijun Huang, Ximeng Sun, and Kate Saenko. Domain agnostic learning with disentangled representations. In NeurIPS, 2019.
[61] Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Deep pointer networks for open-set domain adaptation. In NeurIPS, 2019.
[62] Fabio Pizzati, Raoul de Charette, Michela Zaccaria, and Pietro Cappellini. Playing for data: Ground truth from computer games. In ECCV, 2016.
[64] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In CVPR, 2016. 10, 16
[65] Kuniaiki Saito, Kohei Watanabe, Yoshitaka Ushiku, and Tat-suya Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In CVPR, 2018. 1
[66] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In ECCV, 2018. 1, 8, 12, 16
[67] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In CVPR, 2018. 2
[68] Inkyu Shin, Sanghyun Woo, Fei Pan, and In So Kweon. Two-phase pseudo label densification for self-training based domain adaptation. In ECCV, 2020. 3, 17
[69] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015. 6, 10
[70] Prabhu Teja Sivaprasad and Francois Fleuret. Uncertainty reduction for model adaptation in semantic segmentation. In CVPR, 2021. 2, 7, 17
[71] M Naseer Subhani and Mohsen Ali. Learning from scale-invariant examples for domain adaptation in semantic segmentation. In ECCV, 2020. 3
[72] Marco Toldo, Andrea Maracani, Umberto Michieli, and Pietro Zanuttigh. Unsupervised domain adaptation in semantic segmentation: A review. Technologies, 8(2), 2020. 2
[73] Marco Toldo, Umberto Michieli, and Pietro Zanuttigh. Unsupervised domain adaptation in semantic segmentation via orthogonal and clustered embeddings. In WACV, 2021. 2
[74] Wilhelm Tranhedan, Viktor Olsson, Juliano Pinto, and Lennart Svensson. DACS: Domain adaptation via cross-domain mixed sampling. In WACV, 2021. 7
[75] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulte, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In CVPR, 2018. 8
[76] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulte, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In ICCV, 2019. 2
[77] Vladimir Vapnik. Principles of risk minimization for learning theory. In NeurIPS, 1992. 3
[78] Naveen Venkat, Jogendra Nath Kundu, Durgesh Kumar Singh, Ambareesh Revanur, and R. Venkatesh Babu. Your classifier can secretly suffice multi-source domain adaptation. In NeurIPS, 2020. 1
[79] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. ADVENT: Adversarial entropy minimization for domain adaptation in semantic segmentation. In CVPR, 2019. 2
[80] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. DADA: Depth-aware domain adaptation in semantic segmentation. In ICCV, 2019. 2
[81] Haoran Wang, Tong Shen, Wei Zhang, Lingyu Duan, and Tao Mei. Classes matter: A fine-grained adversarial approach to cross-domain semantic segmentation. In ECCV, 2020. 2, 6, 7, 10, 17
[82] Kaihong Wang, Chenhongyi Yang, and Margrit Betke. Consistency regularization with high-dimensional non-adversarial source-guided perturbation for unsupervised domain adaptation in segmentation. In AAAI, 2021. 2
[83] Zhonghao Wang, Mo Yu, Yunchao Wei, Rogerio Feris, Jinxun Xiong, Wen-mei Hwu, Thomas S Huang, and Honghui Shi. Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation. In CVPR, 2020. 2
[84] Zuxuan Wu, Xintong Han, Yen-Liang Lin, Mustafa Gokhan Uzunbas, Tom Goldstein, Ser Nam Lim, and Larry S Davis. Dean: Dual channel-wise alignment networks for unsupervised scene adaptation. In ECCV, 2018. 2
[85] Zuxuan Wu, Xin Wang, Joseph E Gonzalez, Tom Goldstein, and Larry S Davis. Ace: adapting to changing environments for semantic segmentation. In ICCV, 2019. 2
[86] Jinyu Yang, Weizhi An, Sheng Wang, Xinliang Zhu, Chaochao Yan, and Junzhou Huang. Label-driven reconstruction for domain adaptation in semantic segmentation. In ECCV, 2020. 2, 11, 17
[87] Jinyu Yang, Weizhi An, Chaochao Yan, Peilin Zhao, and Junzhou Huang. Context-aware domain adaptation in semantic segmentation. In WACV, 2021. 2
[88] Jihan Yang, Ruijia Xu, Ruiyu Li, Xiaojuan Qi, Xiaoyong Shen, Guanbin Li, and Liang Lin. An adversarial perturbation oriented domain adaptation approach for semantic segmentation. In AAAI, 2020. 2, 17
[89] Yanchao Yang, Dong Lao, Ganesh Sundaramoorthi, and Stefano Soatto. Phase consistent ecological domain adaptation oriented domain adaptation approach for semantic segmentation. In NeurIPS, 2020. 6, 7, 16, 17
[90] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In CVPR, 2020. 2, 3, 6, 8, 10, 11, 16
[91] Fei Yu, Mo Zhang, Hexin Dong, Sheng Hu, Bin Dong, and Li Zhang. DAST: Unsupervised domain adaptation in semantic segmentation based on discriminator attention and self-training. In AAAI, 2021. 2
[92] Xiaoyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. In ICCV, 2019. 3, 7
[93] Qiming Zhang, Jing Zhang, Wei Liu, and Dacheng Tao. Category anchor-guided unsupervised domain adaptation for semantic segmentation. In NeurIPS, 2019. 7, 17
[94] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Dong Liu, and Tao Mei. Fully convolutional adaptation networks for semantic segmentation. In CVPR, 2018. 2
[95] Yiheng Zhang, Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Dong Liu, and Tao Mei. Transferring and regularizing prediction for semantic segmentation. In CVPR, 2020. 6, 7, 17
[96] Sicheng Zhao, Bo Li, Xiangyu Yue, Yang Gu, Pengfei Xu, Runbo Hu, Hua Chai, and Kurt Keutzer. Multi-source domain adaptation for semantic segmentation. In NeurIPS, 2019. 1, 3

[97] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. International Journal of Computer Vision (IJCV), 2020. 3, 7, 17

[98] Zhedong Zheng and Yi Yang. Unsupervised scene adaptation with memory regularization in vivo. In IJCAI, 2020. 3, 7, 17

[99] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In ICCV, 2017. 2

[100] Xinge Zhu, Hui Zhou, Ceyuan Yang, Jianping Shi, and Dahua Lin. Penalizing top performers: Conservative loss for semantic segmentation adaptation. In ECCV, 2018. 2

[101] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In ICCV, 2019. 2, 10

[102] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In ECCV, 2018. 2, 6, 8