Amplitude Spectrum Transformation for
Open Compound Domain Adaptive Semantic Segmentation

Jogendra Nath Kundu[1*]**, Akshay Kulkarni**[1*]**, Suvaansh Bhambri**[1*]**,**
Varun Jampani[2]**, R. Venkatesh Babu**[1]

[1]Indian Institute of Science, Bangalore   [2]Google Research

Abstract

Open compound domain adaptation (OCDA) has emerged as a practical adaptation setting which considers a single labeled source domain against a compound of multi-modal unlabeled target data in order to generalize better on novel unseen domains. We hypothesize that an improved disentanglement of domain-related and task-related factors of dense intermediate layer features can greatly aid OCDA. Prior-arts attempt this indirectly by employing adversarial domain discriminators on the spatial CNN output. However, we find that latent features derived from the Fourier-based amplitude spectrum of deep CNN features hold a more tractable mapping with domain discrimination. Motivated by this, we propose a novel feature space Amplitude Spectrum Transformation (AST). During adaptation, we employ the AST auto-encoder for two purposes. First, carefully mined source-target instance pairs undergo a simulation of cross-domain feature stylization (AST-Sim) at a particular layer by altering the AST-latent. Second, AST operating at a later layer is tasked to normalize (AST-Norm) the domain content by fixing its latent to a mean prototype. Our simplified adaptation technique is not only clustering-free but also free from complex adversarial alignment. We achieve leading performance against the prior arts on the OCDA scene segmentation benchmarks.

1 Introduction

Deep learning has shown unprecedented success in the challenging semantic segmentation task (Long et al. 2015). In a fully supervised setting (Chen et al. 2018), most approaches operate under the assumption that the training and testing data are drawn from the same input distribution. Though these approaches work well on several benchmarks like Cityscapes (Cordts et al. 2016), their poor generalization to unseen datasets is often argued as the major shortcoming (Hoffman et al. 2016). Upon deployment in real-world settings, they fail to replicate the benchmark performance. This is attributed to the discrepancy in input distributions or domain-shift (Torralba et al. 2011). A naive solution would be to annotate the target domain samples. However, huge cost of annotation and variety of distribution shifts that could be encountered in future render this infeasible. Addressing this, unsupervised domain adaptation (DA) has emerged as a

suitable problem setup, that aims to transfer the knowledge from a labeled source domain to an unlabeled target domain.

In recent years, several unsupervised DA techniques for semantic segmentation have emerged. Such as, techniques inspired from adversarial alignment (Tsai et al. 2018), style transfer (Hoffman et al. 2018), pseudo-label self-training (Zou et al. 2019), *etc*. However, these methods assume the target domain to be a single distribution. This assumption is difficult to satisfy in practice, as test images may come from mixed or continually changing or even unseen conditions. For example, data for self-driving applications may be collected from different weather conditions (Sakaridis et al. 2018) or different time of day (Sakaridis et al. 2020) or different cities (Chen et al. 2017b). Towards a realistic DA setting, Liu et al. (2020) introduced Open Compound DA (OCDA) by incorporating mixed domains (*compound*) in the target but without domain labels. Further, *open* domains are available only for evaluation, representing unseen domains.

The general trend in OCDA (see Table 1) has been to break down the complex problem into multiple easier single-target DA problems and employ variants of existing unimodal DA approaches. To enable such a breakdown, Chen et al. (2019) rely on unsupervised clustering to obtain sub-target clusters. Post clustering, both DHA (Park et al. 2020) and MOCDA (Gong et al. 2021) embrace domain-specific learning. DHA employs separate discriminators and MOCDA uses separate batch-norm parameters (for each sub-target cluster), followed by complex adversarial alignment training for both. Though such domain-specific learning seems beneficial for compound domains, it hurts the generalization to open domains. To combat this generalization issue, MOCDA proposes online model update on encountering the open domains. Note that, such extra updates impedes the deployment-friendliness. In this work, our prime objective is to devise a simple and effective OCDA technique. To this end, we[1] aim to eliminate the requirement of **a)** adversarial alignment, **b)** sub-target clustering, **c)** incorporating domain-specific components, and **d)** online model update (refer the last row of Table 1).

To this end, we uncover key insights while exploring along the lines of disentangling domain-related and task-related cues at different layers of the segmentation archi-

---

---

[1]Project page: https://sites.google.com/view/ast-ocdaseg

	Complex adv. training (via Disc.)	Model update for open domains	Explicit domain specific learning	Clustering	Extra net
OCDA	✓	×	×	×	Disc.
DHA	✓	×	✓ (Separate Disc. for each cluster)	✓	I2I net, Disc.
MOCDA	✓	✓ (online update)	✓ (Separate BN for each cluster)	✓	Hyper net, Disc.
Ours	×	×	×	×	AST

Table 1: Characteristic comparison with OCDA works. *Cluster* refers to sub-target clusters, *I2I net* is image-to-image translation network, *Disc.* is adv. discriminator.

tecture. We perform control experiments to quantify the unwanted correlation of the deep features with the unexposed domain labels by introducing a domain-discriminability metric (DDM). DDM accuracy indicates the ease of classifying the domain label of deep features at different layers. We observe that deeper layers hold more domain-specificity and identify this as a major contributing factor to poor generalization. To alleviate this, prior-arts (Park et al. 2020) employ domain discriminators on the spatial deep features. We ask, can we get hold of a representation space that favors domain discriminability better than raw spatial deep features? Being able to do so would provide us better control to manipulate the representation space for effective adaptation.

To this end, we draw motivation from the recent surge in the use of frequency spectrum analysis to aid domain adaptation (Yang and Soatto 2020; Yang et al. 2020b; Huang et al. 2021). These approaches employ different forms of Fourier transform (FFT) to separately process the phase and amplitude components in order to carry out content-preserving image augmentations. Motivated by this, we propose to use a latent neural network mapping derived from the amplitude spectrum of the raw deep features as the desired representation space. Towards this, we develop a novel feature-space Amplitude Spectrum Transformation (AST) to auto-encode the amplitude while preserving the phase. We empirically confirm that the DDM of AST-latent is consistently higher than the same for the corresponding raw deep features. To the best of our knowledge, we are the first to use frequency spectrum analysis on spatial deep features.

The AST-latent lays a suitable ground to better discern the domain-related factors which in turn allows us to manipulate the same to aid OCDA. We propose to manipulate AST-latent in two ways. First, carefully mined source-target instance pairs undergo a simulation of cross-domain feature stylization (*AST-Sim*) at a particular layer by altering the AST-latent. Second, AST operating at a later layer is tasked to normalize (*AST-Norm*) the domain content by fixing its latent to a mean prototype. As the phase is preserved in AST, both the simulated and domain normalized features retain the task-related content while altering the domain-related style. However, the effectiveness of this intervention is proportional to the quality of disentanglement. Considering DDM as a proxy for disentanglement quality, we propose to apply *AST-Sim* at a deeper layer with high DDM

followed by applying AST-Norm at a later layer close to the final task output. A post-adaptation DDM analysis confirms that the proposed *Simulate-then-Normalize* strategy is effective enough to suppress the domain-discriminability of the deeper features, thus attaining improved domain generalization compared to typical adversarial alignment. We summarize our contributions below:

- We propose a novel feature-space Amplitude Spectrum Transformation (AST), based on a thorough analysis of domain discriminability, for improved disentanglement and manipulability of domain characteristics.

- We provide insights into the usage of AST in two ways - *AST-Sim* and *AST-Norm*, and propose a novel *Simulate-then-Normalize* strategy for effective OCDA.

- Our proposed approach achieves *state-of-the-art* semantic segmentation performance on GTA5→C-Driving and SYNTHIA→C-Driving benchmarks for both compound and open domains, as well as on generalization to extended open domains Cityscapes, KITTI and WildDash.

2 Related Works

Open Compound Domain Adaptation. Liu et al. (2020) proposes to improve the generalization on compound and open domains using a memory based curriculum learning approach. DHA (Park et al. 2020) and MOCDA (Gong et al. 2021) cluster the target using k-means on convolutional feature statistics and encoder features of image-to-image translation network respectively. DHA performs adversarial alignment between the source and each sub-target cluster while MOCDA uses separate batch norm for each sub-target cluster with a single adversarial discriminator. In contrast, we simulate the target style by manipulating the amplitude spectrum of our source features in the latent space.

Stylization for DA. Several recent works that use stylization for DA can be broadly divided into - feature-statistics-based and FFT-based. First, the properties of Fourier transform (FFT), *i.e.* disentangling the input into phase spectrum (representing content) and amplitude spectrum (representing style), made it a natural choice for recent DA works (Yang and Soatto 2020). Several prior arts (Yang et al. 2020a; Huang et al. 2021) employ the FFT directly on the input RGB image in order to simulate some form of image-to-image domain translation to aid the adaptation. In contrast to these works, we utilize the FFT on the CNN feature space. Second, the feature-statistics-based methods (Kim and Byun 2020) use the first or second-order convolutional feature statistics to perform feature-space stylization (Huang and Belongie 2017) followed by reconstructing the stylized image via a decoder. The stylized images are used with other adaptation techniques like adversarial alignment. Contrary to this, we perform content-preserved feature-space stylization using the Amplitude Spectrum Transformation (AST).

3 Approach

In this section, we thoroughly analyze domain discriminability (Sec 3.1). Based on our observations, we propose the Amplitude Spectrum Transformation (Sec 3.2) and provide insights for effectively tackling OCDA (Sec 3.3).
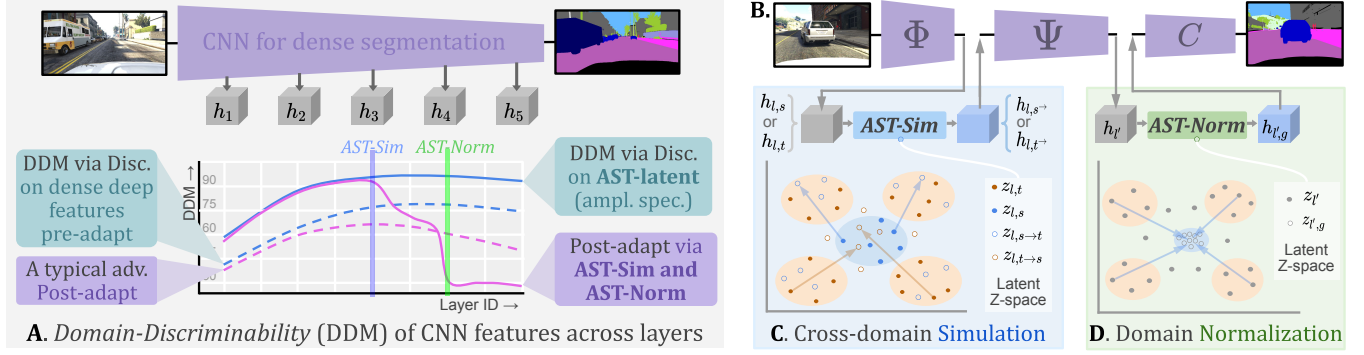
Figure 1: **A.** DDM computed on the AST-latent is consistently higher than the same on raw deep features (Sec 3.1). The post-adaptation DDM confirms the effectiveness of the proposed *Simulate-then-Normalize* strategy (Sec 3.4). **B.** Integration of AST-Sim and AST-Norm into the CNN segmentor (Sec 3.4.1). **C.** & **D.** Internal working of AST-Sim and AST-Norm (Sec 3.3).

3.1 Disentangling domain characteristics

In the literature of causal analysis (Achille and Soatto 2018), it has been shown that generally trained networks (ERM-networks) are prone to learn domain-specific spurious correlations. Here, ERM-network refers to a model trained via empirical risk minimization (ERM) on multi-domain mixture data (without the domain label). In order to test the above proposition, we introduce a metric that would capture the unwanted correlation of layer-wise deep features with the unexposed domain labels. A higher correlation indicates that the model inherently distinguishes among samples from different domains and extracts domain-specific features followed by domain-specific hypothesis (mapping from input to output space) learning. To this end, we introduce the following metric as a proportional measure of this correlation.

3.1.1 Quantifying domain discriminability (DDM).
Domain-Discriminability Metric (DDM) is an accuracy metric that measures the discriminativeness of the features for domain classification. Given a CNN segmentor (see Fig. 1A), $\{h_k\}_{k=1}^K$ denote the 3D tensor of neural activities at the output of convolutional layers with k as the layer depth. Here, $h_k \in \mathcal{H}_k$ with $\mathcal{H}_k$ denoting the raw spatial CNN feature space. Following this, DDM is denoted as λ_k.

Empirical analysis. We obtain a multi-domain source variant with 4 sub-domains by performing specific domain-varying image augmentations such as weather augmentation, cartoonization, *etc*. (refer Suppl.). Following this, a straightforward procedure to compute DDM for each layer k would be to report the accuracy of fully-trained domain discriminators operating on the intermediate deep features of the frozen ERM-network. In Fig. 1A, the dashed blue curve shows the same in a plot. A peculiar observation is that DDM increases while traversing from input towards output.

Observation 1. *An ERM-network trained on multi-domain data for dense semantic segmentation tends to learn increasingly more domain-specific features, in the deeper layers.*

Remarks. This is in line with the observation of increasing memorization in the deeper layers against the early layers as studied by Stephenson et al. (2021). This is because the increase in feature dimensions for deeper layers allows

more room to learn unregularized domain-specific hypotheses. This effect is predominant for dense prediction tasks, as domain-related information is also retained as higher-order clues along the spatial dimensions, implying more room to accommodate domain-specificity. A pool of OCDA or UDA methods (Park et al. 2020; Tsai et al. 2018), that employ adversarial domain alignment, aim to minimize the DDM of deeper layer features as a major part of the adaptation process (dashed magenta curve in Fig. 1A).

3.1.2 Seeking latent representation for improved DDM.
We ask ourselves, can we get hold of a representation space that favors domain discriminability better than the raw spatial deep features? Let, $\mathcal{Z}_k$ be a latent representation space where the multi-domain samples are easily separable based on their domain label. We expect the DDM value to increase if the corresponding discriminator operates on this latent representation instead of the raw spatial deep features. Essentially, we seek to learn a mapping $\mathcal{H}_k \rightarrow \mathcal{Z}_k$ which would encourage a better disentanglement of domain-related factors from the spatial deep features which are known to hold entangled domain-related and task-related information.

To this end, we draw motivation from the recent surge in the use of frequency spectrum analysis to aid domain adaptation (Yang and Soatto 2020; Yang et al. 2020a; Huang et al. 2021). These approaches employ different forms of Fourier transform (FFT) to separately process the phase and amplitude components in order to carry out content-preserving image augmentations. Though both amplitude and phase are necessary to reconstruct the spatial map, it is known that magnitude-only reconstructions corrupt the image making it unrecognizable while that is not the case for phase-only reconstructions. In other words, changes in the amplitude spectrum alter the global style or appearance, i.e., the domain-related factors while the phase holds onto the crucial task-related information. Based on these observations, we hypothesize that the domain-related latent, $\mathcal{Z}_k$ can be derived from the amplitude spectrum.

Empirical analysis. Based on the aforementioned discussion, we develop a novel Amplitude Spectrum Transformation (AST). Similar to an auto-encoding setup, AST involves the encoding ($\mathcal{H} \rightarrow \mathcal{Z}$) and decoding ($\mathcal{Z} \rightarrow \mathcal{H}$) transfor-

mations operating on the amplitude spectrum of the feature maps. Refer Sec 3.2 for further details. Following this, DDM is obtained as the accuracy of a domain discriminator network operating on latent $\mathcal{Z}$ space (repeated for each layer k). In Fig. 1A, the solid blue curve shows the same in a plot.

Observation 2. *Domain discriminability (and thus DDM) is easily identifiable and manipulatable in the latent $\mathcal{Z}_k$ space, i.e. a latent neural network mapping derived from the amplitude spectrum of the raw spatial deep features ($\mathcal{H}_k$ space).*

Remarks. Several prior works (Gatys et al. 2016; Huang et al. 2017) advocate that spatial feature statistics capture the style or domain-related factors. Following this, style transfer networks propose to normalize the channel-wise first or second-order statistics, *e.g.*, instance normalization. One can relate the latent AST representation as a similar measure to represent complex domain discriminating clues that are difficult to extract via multi-layer convolutional discriminators.

3.2 Amplitude Spectrum Transformation (AST)

Several prior arts (Yang et al. 2020; Huang et al. 2021) employ frequency spectrum analysis directly on the input RGB image to simulate some form of image-to-image domain translation. To the best of our knowledge, we are the first to use frequency spectrum analysis on spatial deep features.

3.2.1 AST as an auto-encoding setup. Broadly, AST involves an encoder mapping and a decoder mapping. As shown in Fig. 2, the encoder mapping involves a Fourier transform $\mathcal{F}$ of the input feature h to obtain the amplitude spectrum $\mathcal{F}_A(h)$ and the phase spectrum $\mathcal{F}_P(h)$. The amplitude spectrum is passed through a transformation $\mathcal{T}$ and a fully-connected encoder network Q_e to obtain the AST-latent $z = Q_e \circ \mathcal{T} \circ \mathcal{F}_A(h)$. Consequently, the decoder mapping involves the inverse transformations to obtain $\hat{h} = \mathcal{F}^{-1}(\mathcal{T}^{-1} \circ Q_d(z), \mathcal{F}_P(h))$. It is important to note that the encoding side phase spectrum is directly forwarded to the decoding side to be recombined with the reconstructed amplitude spectrum via inverse Fourier transform $\mathcal{F}^{-1}$.

Here, the vectorization ($\mathcal{T}$) and inverse-vectorization ($\mathcal{T}^{-1}$) of the amplitude spectrum must adhere to its symmetric about origin (mid-point) property. To this end, $\mathcal{T}$ only vectorizes the first and fourth quadrants and $\mathcal{T}^{-1}$ reconstructs the full spectrum from the reshaped first and fourth quadrants by mirroring about the origin (see Fig. 2).

Training. Let $\theta_Q^{(k)} = \{\theta_{Q_e}, \theta_{Q_d}\}$ denote the trainable parameters of AST at the kth layer. Note that, AST operates independently for each channel of h_k with the same parameters. Effectively, z_k is obtained as a concatenation of z's from each channel of h_k. After obtaining the frozen *ERM-network*, AST, at a given layer, is trained to auto-encode the amplitude spectrum using the following objective,

$$\min_{\theta_Q} \mathcal{L}_{\text{AST}}(h_k, \hat{h}_k) \text{ where } \mathcal{L}_{\text{AST}}(h_k, \hat{h}_k) = \|h_k - \hat{h}_k\|_2^2 \ (1)$$

For simplicity, we denote the overall AST as a function AST and simplify the reconstruction as $\hat{h}_k = \text{AST}(h_k, z_k)$. Note that we omit the z_k term in the auto-encoding phase *i.e.* $\hat{h}_k = \text{AST}^{(ae)}(h_k)$ as z_k is passed through unmodified.
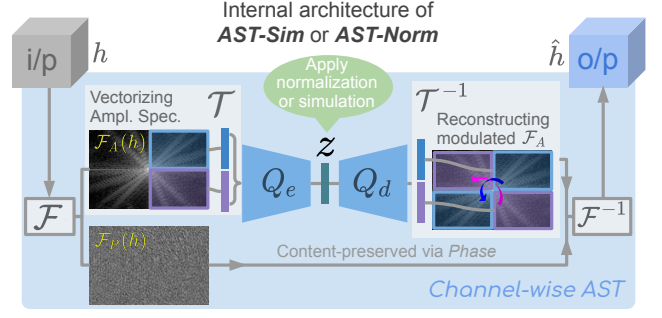

Figure 2: Internal architecture of AST auto-encoder.

3.3 Usage of AST for OCDA

AST lays a suitable ground to better identify domain-related factors (high DDM) allowing us to manipulate the same.

Insight 1. *Altering the latent z by the same obtained from a different domain or instance (i.e. $\tilde{z}$) while transforming $\tilde{h} = \text{AST}(h, \tilde{z})$ is expected to imitate feature space cross-domain translation (h-to-$\tilde{h}$) as the crucial task-related information remains preserved via the unaltered phase spectrum.*

3.3.1 AST for cross-domain simulation (*AST-Sim*). One of the major components in prior OCDA approaches is to discover sub-target clusters as it is used either for domain-translation of source samples (Park et al. 2020) or to realize the domain-specific batch-normalization (Gong et al. 2021). In order to obtain reliable sub-target clusters, both works rely on unsupervised clustering of the compound target data. In contrast, we aspire to avoid such discovery and do away with introducing an additional hyperparameter (no. of clusters) in order to realize a clustering-free OCDA algorithm.

Cross-domain pairing. How to select reliable source-target pairs for cross-domain simulation? In the absence of sub-target clusters, a naive approach would be to form random pairs of source and target instances. Aspiring to formalize a better source-target pairing strategy, we propose to pair instance with the most distinct domain style. This is motivated from the hard negative mining as used in deep metric learning approaches (Suh et al. 2019). Consider a source dataset with instances $\{s_i\}_{i=1}^{N_s}$ and a target dataset with instances $\{t_j\}_{j=1}^{N_t}$. For each target instance t_j, we mine a source instance s_i such that these are maximally separated in the AST-latent space of the lth CNN layer. Formally, we obtain a set of cross-domain pairs as,

$$\mathcal{U} = \{(s_i, t_j) : \forall t_j, i = \arg\max_{i'} \zeta(z_{l,s_{i'}}, z_{l,t_j})\} \quad (2)$$

where z_{l,s_i} and z_{l,t_j} are the lth layer AST-latent for the s_i and t_j instances respectively. Here, ζ is a L2 distance function.

Cross-domain simulation. Utilizing Insight 1, we simulate the style of t_j in the source s_i by replacing its AST-latent *i.e.*, $\hat{h}_{l,s_{i \to j}} = \text{AST}(h_{l,s_i}, z_{l,t_j})$. Similarly, $\hat{h}_{l,t_{j \to i}}$ represents the source stylized target feature. We illustrate the cross-domain simulation in Fig. 1C. For notation simplification, $\hat{h}_{l,s_{i \to j}}$ and $\hat{h}_{l,t_{i \to j}}$ are denoted as $h_{l,s^{\to}}$ and $h_{l,t^{\to}}$ respectively.

Note that, $h_{l,s^{\rightarrow}}$ and $h_{l,t^{\rightarrow}}$ hold the same task-related content as $h_{l,s}$ and $h_{l,t}$ respectively. Thus, both original and stylized features are expected to have the same segmentation label.

3.3.2 AST for domain normalization (*AST-Norm*). We draw motivation from batch normalization (BN) (Ioffe et al. 2015), which normalizes the features with first and second order statistics during forward pass. Along the same lines, we aim to normalize the AST-latent of intermediate deep features by altering it with a fixed mean prototype. Let $\mathcal{V}$ be the set of l'^{th}-layer deep features obtained from all the four variants, *i.e.*, $h_{l',s}, h_{l',t}, h_{l',s^{\rightarrow}}, h_{l',t^{\rightarrow}}$. We compute the fixed domain prototype $z_{l',g}$ as follows,

$$z_{l',g} = \mathop{\mathbb{E}}_{h \in \mathcal{V}} \left[Q_e \circ \mathcal{T} \circ \mathcal{F}_A(h) \right] \tag{3}$$

Following Insight 1, the domain normalization is performed as $h_{l',s_g} = \text{AST}(h_{l',s}, z_{l',g})$. We illustrate the domain normalization in Fig. 1D. The normalized versions of $h_{l',t}, h_{l',s^{\rightarrow}}, h_{l',t^{\rightarrow}}$ are $h_{l',t_g}, h_{l',s_g^{\rightarrow}}, h_{l',t_g^{\rightarrow}}$ respectively.

3.4 Simulate-then-Normalize for OCDA

The prime objective in OCDA is to adapt the task-knowledge from a labeled source domain $(x_s, y_s) \in \mathcal{D}_s$ to an unlabeled compound target, $x_t \in \mathcal{D}_t$ towards better generalization to unseen open domains. The advantages of AST (*i.e.* identifying and manipulating domain-related factors) well cater to the specific challenges encountered in OCDA against employing it for single-source single-target settings.

Empirical analysis. After obtaining the frozen ERM-network and the frozen layer-wise AST modules, we employ *AST-Sim* (Fig. 1C) and *AST-Norm* (Fig. 1D) to analyze its behavior towards the task performance. We observe that performing *AST-Sim* or *AST-Norm* at the earlier layer with low DDM hurts the task performance. We infer that a low DDM value indicates inferior disentanglement of domain-related cues at the corresponding AST-latent. In other words, the task-related cues tend to leak through the amplitude spectrum thereby distorting task-related content.

Insight 2. *Applying AST-Sim at a layer with a high DDM value followed by applying AST-Norm at a later layer close to the final task output helps to effectively leverage the advantages of AST modules for the challenging OCDA setting.*

Remarks. From the perspective of causal analysis literature, the use of *AST-Sim* can be thought of as an intervention (He et al. 2008) that implicitly encourages the network to focus on the uninterrupted task-specific cues. However, the effectiveness of this intervention is proportional to the quality of disentanglement. Here, DDM can be taken as a proxy to measure the degree of disentanglement quality. Thus, we propose to perform *AST-Sim* at a layer l, with high DDM, as indicated in Fig. 1B. However, we realize that the uninterrupted phase pathway still leaks domain-related information (relatively high DDM for $k > l$) hindering domain generalization. To circumvent this, we propose to perform *AST-Norm* at a later layer as a means to further fade away the domain-related information. A post-adaptation DDM analysis (solid magenta curve in Fig. 1A) confirms that the proposed *Simulate-then-Normalize* strategy is effective enough

---

Algorithm 1: Pseudo-code for the proposed approach

1: **Input:** source data $(x_s, y_{gt}) \in \mathcal{D}_s$, target data $x_t \in \mathcal{D}_t$. Initialize θ (*i.e.* parameters of Φ, Ψ, and C) from standard source training. Initialize θ_{cs} and θ_{dn} (*i.e.* parameters of ASM_{cs} and ASM_{dn}) from auto-encoder training discussed in Sec 3.2.1.
2: *Pre-adaptation*: Integrate ASTs into the CNN segmentor and finetune θ, θ_{cs}, and θ_{dn} following Eq. 4 and 5.

    Adaptation via Simulate-then-Normalize

3: Freeze θ_{cs}, and θ_{dn} from the pre-adaptation training. Recompute fixed domain prototype $z_{l',g}$ using Eq. 3 and pseudo-labels y_{pgt} using Eq. 8, after every epoch.
4: **for** $iter < MaxIter$ **do**:
5:    $x_s, y_{gt}, x_t, y_{pgt} \leftarrow$ batch sampled from $\mathcal{D}_s, \mathcal{D}_t$
6:    Obtain predictions following Fig. 3
7:    **update** θ by optimizing Eq. 6 and 7
8: **end for**

---

to suppress the domain-discriminability of the later deep features, thus attaining improved domain generalization.

Architecture. As shown in Fig. 1B, we divide the CNN segmentor into three parts; *i.e.* Φ, Ψ, and C. Following this, the *AST-Sim* (denoted as AST_{cs}) and *AST-Norm* (denoted as AST_{dn}) are inserted after Φ and Ψ respectively (see Fig. 3B, D). Next, we discuss the adaptation training.

3.4.1 Pre-adaptation training. In this stage, we aim to prepare (or initialize) the network components before the proposed *Simulate-then-Normalize* procedure (*i.e.* adaptation). We start from the pre-trained standard source model and the pre-trained auto-encoder based AST-networks at layer l and l', *i.e.* $\text{AST}_{cs}^{(ae)}$, and $\text{AST}_{dn}^{(ae)}$. Here, the superscript *(ae)* indicates that these networks are not yet manipulated as *AST-Sim*, or *AST-Norm* (*i.e.*, without manipulating AST-latent).

Following this, the pre-adaptation finetuning involves two pathways. First, x_s is forwarded to obtain $y_s = C \circ \text{AST}_{dn}^{(ae)} \circ \Psi \circ \text{AST}_{cs}^{(ae)} \circ \Phi(x_s)$. The following source-side supervised cross-entropy objective is formed as;

$$\min_{\theta} \mathcal{L}_s \quad \text{where} \quad \mathcal{L}_s = L_{CE}(y_s, y_{gt}) \tag{4}$$

Here, θ denotes the collection of parameters of Φ, Ψ, and C. Note that, after integrating ASTs into the CNN segmentor, the content-related gradients flow unobstructed through the frozen AST networks while finetuning Φ and Ψ.

In the second pathway, both x_s and x_t are forwarded to obtain input-output pairs to finetune θ_{cs}, and θ_{dn}, *i.e.*

$$\min_{\theta_{cs}} \mathcal{L}_{\text{AST}}(h_{l,s}, \hat{h}_{l,s}) + \mathcal{L}_{\text{AST}}(h_{l,t}, \hat{h}_{l,t})$$
$$\min_{\theta_{dn}} \mathcal{L}_{\text{AST}}(h_{l',s}, \hat{h}_{l',s}) + \mathcal{L}_{\text{AST}}(h_{l',t}, \hat{h}_{l',t}) \tag{5}$$

Here, $(h_{l,s}, \hat{h}_{l,s})$ and $(h_{l,t}, \hat{h}_{l,t})$ are the input-output pairs corresponding to x_s and x_t to update $\text{AST}_{cs}^{(ae)}$. Similarly, $(h_{l',s}, \hat{h}_{l',s})$ and $(h_{l',t}, \hat{h}_{l',t})$ are the input-output pairs corresponding to x_s and x_t to update $\text{AST}_{dn}^{(ae)}$.

3.4.2 Adaptation via *Simulate-then-Normalize*. The finetuned networks from pre-adaptation are adapted via the
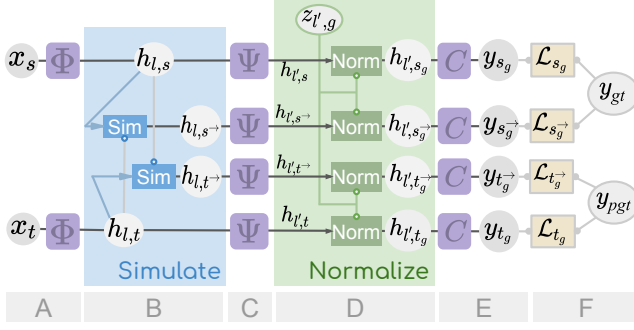
Figure 3: Data flow during adaptation training (Sec 3.4.2).

simulate-then-normalize procedure. We manipulate the AST-latent, thus denoting $\mathtt{AST}_{cs}^{(ae)}$, and $\mathtt{AST}_{dn}^{(ae)}$ as $\mathtt{AST}_{cs}$ and $\mathtt{AST}_{dn}$ respectively. Note that, latent manipulation is independent of the network weights, θ_{cs} and θ_{dn}. Thus, we freeze θ_{cs} and θ_{dn} from pre-adaptation to preserve their auto-encoding behaviour, and only update θ during adaptation.

The adaptation training involves four data-flow pathways as shown in Fig 3. The *Simulate* step at layer-l outputs the following features; $(h_{l,s}, h_{l,s^{\rightarrow}}, h_{l,t^{\rightarrow}}, h_{l,t})$. Moving forward, the *Normalize* step at layer-l' outputs the following: $(h_{l',s_g}, h_{l',s_g^{\rightarrow}}, h_{l',t_g^{\rightarrow}}, h_{l',t_g})$. Finally, we obtain the following four predictions; $(y_{s_g}, y_{s_g^{\rightarrow}}, y_{t_g^{\rightarrow}}, y_{t_g})$.

As shown in Fig. 3F, we apply supervised losses against ground-truth (GT) y_{gt} for the source-side predictions *i.e.*

$$\min_{\theta}(\mathcal{L}_{s_g}{=}L_{CE}(y_{s_g}, y_{gt})); \ \min_{\theta}(\mathcal{L}_{s_g^{\rightarrow}}{=}L_{CE}(y_{s_g^{\rightarrow}}, y_{gt})) \quad (6)$$

Following self-training DA works (Zou et al. 2019), we use pseudo-labels (pseudo-GT) y_{pgt} for target predictions *i.e.*

$$\min_{\theta}(\mathcal{L}_{t_g}{=}L_{CE}(y_{t_g}, y_{pgt})); \ \min_{\theta}(\mathcal{L}_{t_g^{\rightarrow}}{=}L_{CE}(y_{t_g^{\rightarrow}}, y_{pgt})) \quad (7)$$

Pseudo-label extraction. In order to obtain reliable pseudo-labels y_{pgt}, we prune the target predictions by performing a consistency check between y_{t_g} and $y_{t_g^{\rightarrow}}$. Thus, $y_{pgt} = \arg\max_c y_{t_g}$ is obtained only for reliable pixels *i.e.* for pixels x_t which satisfy the following criteria,

$$\mathcal{D}_t^{(pgt)}{=}\{x_t : x_t \in \mathcal{D}_t \text{ and } \arg\max_c y_{t_g} = \arg\max_c y_{t_g^{\rightarrow}}\} \quad (8)$$

Note that, only the target pixels in $\mathcal{D}_t^{(pgt)}$ contribute in Eq. 7. The overall training procedure is summarized in Algo. 1.

4 Experiments

We thoroughly evaluate the proposed approach against *state-of-the-art* prior works in the Open Compound DA setting.

Datasets. Following Gong et al. (2021), we used the synthetic GTA5 (Richter et al. 2016) and SYNTHIA (Ros et al. 2016) datasets as the source. In C-Driving (Liu et al. 2020), *rainy*, *snowy*, and *cloudy* sub-targets form the compound target domain, and *overcast* sub-target forms the open domain. Further, we use Cityscapes (Cordts et al. 2016), KITTI

Method	Compound (C)			Open (O)	Average	
	Rainy	Snowy	Cloudy	Overcast	C	C+O
Source only	16.2	18.0	20.9	21.2	18.9	19.1
ASN (Tsai et al. 2018)	20.2	21.2	23.8	25.1	22.1	22.5
CBST (Zou et al. 2018)	21.3	20.6	23.9	24.7	22.2	22.6
IBN-Net (Pan et al. 2018)	20.6	21.9	26.1	25.5	22.8	23.5
PyCDA (Lian et al. 2019)	21.7	22.3	25.9	25.4	23.3	23.8
OCDA (Liu et al. 2020)	22.0	22.9	27.0	27.9	24.5	25.0
DHA (Park et al. 2020)	27.0	26.3	30.7	32.8	28.5	29.2
Ours (AST-OCDA)	28.2	27.8	31.6	34.0	**29.2**	**30.4**
Source only†	23.3	24.0	28.2	30.2	25.7	26.4
ASN (Tsai et al. 2018)†	25.6	27.2	31.8	32.1	28.8	29.2
MOCDA (Gong et al. 2021)†	24.4	27.5	30.1	31.4	27.7	29.4
DHA (Park et al. 2020)†	27.1	30.4	35.5	36.1	32.0	32.3
Ours (AST-OCDA)†	32.7	32.2	38.9	39.2	**34.6**	**35.7**

Table 2: Comparison on GTA5→C-Driving benchmark. † indicates 150k training iterations, otherwise 5k iterations.

Method	Compound (C)			Open (O)	Average	
	Rainy	Snowy	Cloudy	Overcast	C	C+O
Source only	16.3	18.8	19.4	19.5	18.4	18.5
CBST (Zou et al. 2018)	16.2	19.6	20.1	20.3	18.9	19.1
CRST (Zou et al. 2019)	16.3	19.9	20.3	20.5	19.1	19.3
ASN (Tsai et al. 2018)	17.0	20.5	21.6	21.6	20.0	20.2
AdvEnt (Vu et al. 2019)	17.7	19.9	20.2	20.5	19.3	19.6
DHA (Park et al. 2020)	18.8	21.2	23.6	23.6	21.5	21.8
Ours (AST-OCDA)	20.3	22.6	24.9	25.4	**22.6**	**23.3**
Source only∗	16.5	18.2	21.4	20.6	19.2	19.8
AdvEnt (Vu et al. 2019)∗	21.8	22.6	26.2	25.7	23.9	24.7
ASN (Tsai et al. 2018)∗	24.9	26.9	30.7	30.3	28.0	29.0
MOCDA (Gong et al. 2021)∗	26.6	27.9	32.4	31.8	29.1	30.3
Ours (AST-OCDA)∗	27.9	28.8	33.9	34.2	**30.2**	**31.2**

Table 3: Comparison on SYNTHIA→C-Driving benchmark. All methods employ the long-training strategy. ∗ indicates mIoU over 11 classes otherwise over 16 classes.

(Geiger et al. 2013) and WildDash (Zendel et al. 2018) datasets as extended open domains to test generalization on diverse unseen domains. All datasets (except SYNTHIA) share 19 semantic categories. We use the mean intersection-over-union (mIoU) metric for evaluating the performance. See Suppl. for more details.

Implementation Details. Following (Park et al. 2020; Gong et al. 2021), we employ DeepLabv2 (Chen et al. 2017a) with a VGG16 (Simonyan et al. 2015) backbone as the CNN segmentor. We use SGD optimizer with a learning rate of 1e-4, momentum of 0.9 and a weight decay of 5e-4 during training. We also use a polynomial decay with power 0.9 as the learning rate scheduler. Following Park et al. (2020), we use two training schemes for GTA5 *i.e.*, short training scheme with 5k iterations and long training scheme with 150k iterations. For SYNTHIA, we use only the long training scheme following Gong et al. (2021).

4.1 Discussion

We compare our approach with prior arts on GTA5→C-Driving (Table 2) and SYNTHIA→C-Driving (Table 3) benchmarks as well as on extended open domains (Table 4). We also provide an extensive ablation study (Table 5).

GTA5 as source (Table 2, 4). For the short training scheme, our approach outperforms the SOTA method DHA (Park et al. 2020) by an average of 0.7% on compound domains and 1.2% on open domain. The improvement is en-

Source	Method	Extended Open			Avg.
		Citysc.	KITTI	WildDash	
GTA5	Source only	19.3	24.1	16.0	20.5
	ASN (Tsai et al. 2018)	22.0	23.4	17.5	22.5
	MOCDA (Gong et al. 2021)	31.1	30.9	21.6	27.8
	Ours (AST-OCDA)	32.6	31.8	23.1	**29.2**
SYN-THIA	Source only	24.7	20.7	17.3	20.8
	ASN (Tsai et al. 2018)	35.9	24.7	20.7	27.9
	MOCDA (Gong et al. 2021)	32.2	34.2	25.8	31.2
	Ours (AST-OCDA)	37.2	35.7	26.9	**33.3**

Table 4: Evaluation on extended open domains. mIoU computed over 19 classes for GTA5 as source and 11 classes for SYNTHIA as source, following Gong et al. (2021).

#	AST-Sim	Pairing		AST-Norm	Losses			GTA5→C-Driv.	
		Random	Mined		$\mathcal{L}_{s_g}, \mathcal{L}_{t_g}$	$\mathcal{L}_{s^{\rightarrow}_g}, \mathcal{L}_{t^{\rightarrow}_g}$		C	C+O
1.	×	-	-	×	-	-		25.7	26.4
2.	×	-	-	✓	✓	-		28.1	28.6
3.	✓	✓	×	×	-	-		27.9	28.3
4.	✓	✓	×	✓	×	✓		30.0	30.3
5.	✓	✓	×	✓	✓	✓		31.2	31.6
6.	✓	×	✓	×	-	-		29.0	29.8
7.	✓	×	✓	✓	×	✓		32.8	33.6
8.	✓	×	✓	✓	✓	✓		**34.6**	**35.7**

Table 5: Ablation study on GTA5→C-Driving. *Random* and *Mined* indicate random pairing and hard-mining strategy (Sec 3.3.1). *Sim* and *Norm* indicate the module usage.

hanced with the long training scheme where we outperform DHA by 2.6% and 3.1% on compound and open domains respectively. As illustrated in Table 4, our method outperforms the SOTA method MOCDA (Gong et al. 2021) on the extended open domains by 1.4% on average. This verifies the better generalization abilities of our approach for new open domains with higher domain-shift from the source data.

SYNTHIA as source (Table 3, 4). Our approach outperforms DHA (Park et al. 2020) by an average of 1.1% on compound domains and 1.8% on the open domain. Table 4 shows our superior performance on extended open domains. We outperform MOCDA (Gong et al. 2021) by 2.1% over the 3 datasets. This reinforces our better generalizability.

4.1.1 Ablation Study. Table 5 presents a detailed ablation to underline the equal importance of *AST-Sim* and *AST-Norm*.

First, we evaluate the variations proposed under *AST-Sim*. As a baseline, we use the source-only performance (#1). Deactivating *AST-Norm*, *AST-Sim* with *random* pairing strategy (#3 vs. #1) yields an average 2.1% improvement. Next, without *AST-Norm*, *AST-Sim* alongside the proposed *Mined* pairing outperforms the previous (#6 vs. #3) by 1.3%. When *AST-Norm* is applied alongside only the simulation based losses (i.e. $\mathcal{L}_{s^{\rightarrow}_g}, \mathcal{L}_{t^{\rightarrow}_g}$), using *Mined* pairing improves over *random* pairing (#7 vs. #4) by 3.0%. Finally, when *AST-Norm* is applied alongside all the four losses, *Mined* pairing outperforms *random* pairing (#8 vs. #5) by 3.5%. This verifies the consistent superiority of our well-thought-of cross-domain pairing strategy (Sec 3.3.1).

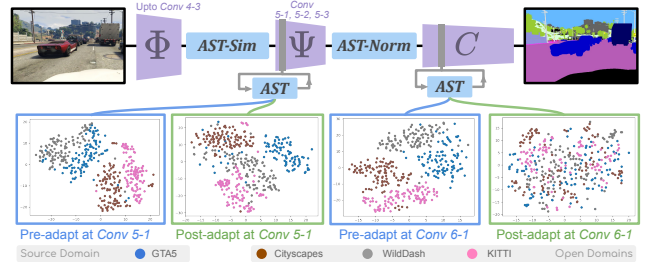Second, we evaluate the variations under *AST-Norm*. Deactivating *AST-Sim*, *AST-Norm* outperforms the base-

Figure 4: t-SNE plots at the layers following *AST-Sim* and *AST-Norm*, pre- and post-adaptation. Domains remain clustered after *AST-Sim* while they are aligned after *AST-Norm*.

line (#2 vs. #1) by 2.3% on average. *AST-Norm* alongside simulation-based losses i.e. $\mathcal{L}_{s^{\rightarrow}_g}, \mathcal{L}_{t^{\rightarrow}_g}$ (#4 vs. #3 and #7 vs. #6 for random and mined pairing respectively) yield average improvements of 2.0% and 3.8% respectively over not using *AST-Norm*. Next, *AST-Norm* used with all losses improves over the previous (#5 vs. #4 and #8 vs. #7 for random pairing and hard mining respectively) by 1.2% and 1.9% respectively. This verifies the consistent improvement of using all losses over only simulation-based losses.

4.1.2 Analyzing domain alignment. Since *AST-Sim* (layer l) and *AST-Norm* (layer l') involve replacing the latent, their effect can be studied at the layer following it. In Fig. 4, we plot t-SNE (Maaten et al. 2008) embeddings of AST-latent at layers $l + 1$ and $l' + 1$ to examine domain alignment before and after the proposed adaptation. Though simulated cross-domain features $(h_{l,s^{\rightarrow}}, h_{l,t^{\rightarrow}})$ span the entire space, projection of the original source and sub-target domains (i.e. $h_{l,s}, h_{l,t}$) stills retains domain-specificity (first and second plots). As *AST-Norm* aims to normalize the AST-latent with a fixed mean prototype (i.e. enforcing all features to have a common AST-latent), we observe that the domains are aligned post-adaptation (third and fourth plots) at layer $l'+1$. This shows that the proposed *Simulate-then-Normalize* strategy effectively reduces the domain shift in the latent space.

5   Conclusion

In this work, we performed a thorough analysis of domain discriminability (DDM) to quantify the unwanted correlation between the deep features and domain labels. Based on our observations, we proposed the feature-space Amplitude Spectrum Transformation (AST) to better disentangle and manipulate the domain and task-related information. We provided insights into the usage of AST in two configurations - *AST-Sim* and *AST-Norm* and proposed a novel *Simulate-then-Normalize* strategy for effectively performing Open Compound DA. Extending the proposed approach for more general, realistic DA scenarios such as multi-source, multi-target DA can be a direction for future work.

References

Achille, A.; and Soatto, S. 2018. Emergence of Invariance and Disentanglement in Deep Representations. *J. Mach. Learn. Res.*, 19(1): 1947–1980.

Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; and Yuille, A. L. 2017a. 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.

Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; and Adam, H. 2018. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *ECCV*.

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

Chen, Z.; Zhuang, J.; Liang, X.; and Lin, L. 2019. Blending-target Domain Adaptation by Adversarial Meta-Adaptation Networks. In *CVPR*.

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

Gatys, L. A.; Ecker, A. S.; and Bethge, M. 2016. Image Style Transfer Using Convolutional Neural Networks. In *CVPR*.

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R. 2013. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)*.

Gong, R.; Chen, Y.; Paudel, D. P.; Li, Y.; Chhatkuli, A.; Li, W.; Dai, D.; and Gool, L. V. 2021. Cluster, Split, Fuse, and Update: Meta-Learning for Open Compound Domain Adaptive Semantic Segmentation. In *CVPR*.

He, Y.-B.; and Geng, Z. 2008. Active learning of causal networks with intervention experiments and optimal designs. *Journal of Machine Learning Research*, 9(Nov): 2523–2547.

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

Hoffman, J.; Wang, D.; Yu, F.; and Darrell, T. 2016. FCNs in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*.

Huang, J.; Guan, D.; Xiao, A.; and Lu, S. 2021. FSDR: Frequency Space Domain Randomization for Domain Generalization. In *CVPR*.

Huang, X.; and Belongie, S. 2017. Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization. In *ICCV*.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*.

Kim, M.; and Byun, H. 2020. Learning Texture Invariant Representation for Domain Adaptation of Semantic Segmentation. In *CVPR*.

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

Liu, Z.; Miao, Z.; Pan, X.; Zhan, X.; Lin, D.; Yu, S. X.; and Gong, B. 2020. Open Compound Domain Adaptation. In *CVPR*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov): 2579–2605.

Pan, X.; Luo, P.; Shi, J.; and Tang, X. 2018. Two at Once: Enhancing Learning and Generalization Capacities via IBN-Net. In *ECCV*.

Park, K.; Woo, S.; Shin, I.; and Kweon, I. S. 2020. Discover, Hallucinate, and Adapt: Open Compound Domain Adaptation for Semantic Segmentation. In *NeurIPS*.

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

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

Sakaridis, C.; Dai, D.; Hecker, S.; and Van Gool, L. 2018. Model Adaptation with Synthetic and Real Data for Semantic Dense Foggy Scene Understanding. In *ECCV*.

Sakaridis, C.; Dai, D.; and Van Gool, L. 2020. Map-Guided Curriculum Domain Adaptation and Uncertainty-Aware Evaluation for Semantic Nighttime Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Stephenson, C.; Padhy, S.; Ganesh, A.; Hui, Y.; Tang, H.; and Chung, S. 2021. On the geometry of generalization and memorization in deep neural networks. In *ICLR*.

Suh, Y.; Han, B.; Kim, W.; and Lee, K. M. 2019. Stochastic Class-Based Hard Example Mining for Deep Metric Learning. In *CVPR*.

Torralba, A.; and Efros, A. A. 2011. Unbiased look at dataset bias. In *CVPR*.

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

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

Yang, J.; An, W.; Wang, S.; Zhu, X.; Yan, C.; and Huang, J. 2020a. Label-Driven Reconstruction for Domain Adaptation in Semantic Segmentation. In *ECCV*.

Yang, Y.; Lao, D.; Sundaramoorthi, G.; and Soatto, S. 2020b. Phase consistent ecological domain adaptation. In *CVPR*.

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

Zendel, O.; Honauer, K.; Murschitz, M.; Steininger, D.; and Dominguez, G. F. 2018. WildDash - Creating Hazard-Aware Benchmarks. In *ECCV*.

Zou, Y.; Yu, Z.; Liu, X.; Kumar, B.; and Wang, J. 2019. Confidence regularized self-training. In *ICCV*.

Zou, Y.; Yu, Z.; Vijaya Kumar, B.; and Wang, J. 2018. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*.