2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

ConDA: Unsupervised Domain Adaptation for LiDAR Segmentation via Regularized Domain Concatenation

Lingdong Kong[1,*], Niamul Quader[2], Venice Erin Liong[2]

***Abstract*— Transferring knowledge learned from the labeled source domain to the raw target domain for unsupervised domain adaptation (UDA) is essential to the scalable deployment of autonomous driving systems. State-of-the-art methods in UDA often employ a key idea: utilizing joint supervision signals from both source and target domains for self-training. In this work, we improve and extend this aspect. We present ConDA, a concatenation-based domain adaptation framework for LiDAR segmentation that: 1) constructs an intermediate domain consisting of fine-grained interchange signals from both source and target domains without destabilizing the semantic coherency of objects and background around the ego-vehicle; and 2) utilizes the intermediate domain for self-training. To improve the network training on the source domain and self-training on the intermediate domain, we propose an anti-aliasing regularizer and an entropy aggregator to reduce the negative effect caused by the aliasing artifacts and noisy pseudo labels. Through extensive studies, we demonstrate that ConDA significantly outperforms prior arts in mitigating domain gaps.**

I. INTRODUCTION

Large-scale annotated data are desirable as they often yield robust and generalizable models. However, annotating semantic labels for 3D data like LiDAR point clouds [1] in autonomous driving [2], [3], [4], [5], [6], [7], [8] is extremely expensive [9], [10], [11]. This motivates us to explore unsupervised domain adaptation (UDA) for transferring knowledge learned from one domain, *e.g.*, Boston, to another domain, *e.g.*, Singapore, for scalable *cross-city* deployment.

UDA aims to tackle scenarios where a model is trained on labeled data from a *source* domain and unlabeled data from a different but related *target* domain, with the goal of enabling the model to perform well during target test time. A common practice in UDA is to jointly learn from both domains [12]. Prior works fall into two lines: 1) implicit learning domain-invariant features with discriminators via adversarial training [13], [14], [15], and 2) self-training with joint supervision signals from both the source (*w/* ground-truth) and target (*w/* pseudo-labels generated by confidence thresholding) domains [16], [17]. Recent studies have shown that the latter often yields more robust models with less computation cost [18], [19]. These works, however, learn separately from source and target batches, and thus lack learning fine-grained interactions of objects and background in-between domains. Implementing an intermediate domain that can facilitate interactions to reduce the domain gap is intuitive. The ground-truth signals from the source domain can construct a "shortcut" for correcting false target predictions in the local vicinity since feature representations close to each other tend to have the same semantic label [20]. Additionally, the target "pseudo supervisions" – which often contain large amounts of ignored labels – can serve as a strong consistency regularization [21] for the source domain. This guided supervision and regularization may yield a more adaptable and robust feature learning. It is not easy, however, to directly mix domains via interpolation [22], [23] or superposition [24], [25] for semantic segmentation as such techniques can corrupt the semantic coherency [21].

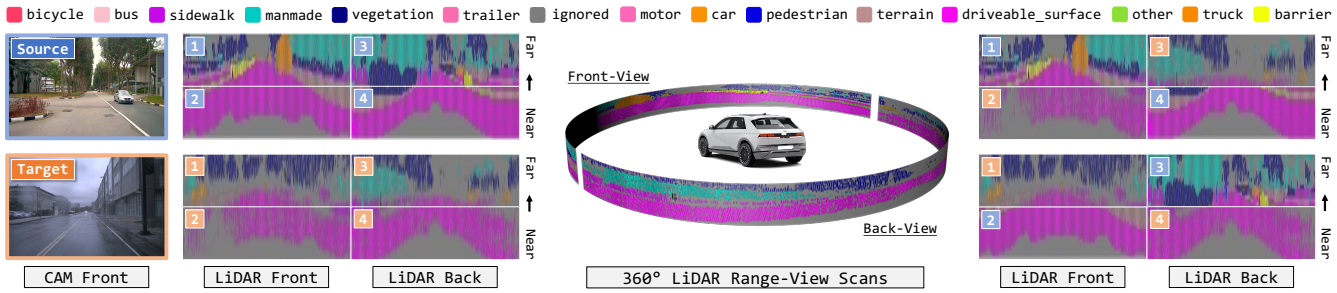
In this work, we propose a concatenation-based domain adaptation (ConDA) approach for cross-city UDA in LiDAR segmentation. ConDA enables the interactions of fine-grained semantic information in-between the source and target cities (domains) while not destabilizing the semantic coherency. We make the observations that although the overall distributions for the source and target domains are very different, there is a strong likelihood that similar objects and background tend to occupy particular regions in the LiDAR range-view (RV) around the ego-vehicle We exploit this important correlation to construct the ConDA intermediate domain that selects non-overlapping regions of point clouds from both source and target domains and concatenates them together, while maintaining the relative positions from the ego-vehicle. As shown in Fig. 1, concatenating different regions (*e.g.*, front-top, front-bottom, back-top, and back-bottom) of RV stripes from different domains can still preserve semantic consistency. We will revisit this formally in Sec. III-B.

The proposed domain concatenation strategy provides a better solution for model self-training via interchanged supervisions from both the source domain and the target domain. It is worth noting that the quality of the pseudo-labels – and in turn the generalizability of the initial training on the source domain – becomes essential since erroneous "pseudo supervision" can do evil in self-training. We counter this problem from two perspectives as follows. First, the generalizability of the model can be improved by removing high-frequency noisy aliasing artifacts [26], [27] with careful placements of low-pass anti-aliasing filters [28]. However, empirically designing such filters is challenging in the context of UDA, since the lack of target domain ground-truth prevents any empirical experiments. To reduce the detrimental effect of high-frequency aliasing artifacts [29], [30] without accessing target annotations, we propose a built-in regularization mechanism (Sec. III-C) within each convolution block to regularize high-frequency representation learning during training,

[1]Lingdong Kong is with CNRS@CREATE and the National University of Singapore. Email: lingdong@comp.nus.edu.sg.

[2]Niamul Quader and Venice Erin Liong are with Motional, Singapore. Email: {niamul.quader,venice.liong}@motional.com.

*Work done as an autonomous vehicle research intern at Motional.



(a) (b) (c)

Fig. 1. Illustrative examples for domain concatenation. (a) Visual RGB and LiDAR range-view (RV) projections of the source (ground-truth) and target (pseudo-labels) domains. Images adopted from nuScenes [2]. (b) Cylindrical representation of LiDAR RV. (c) Concatenated examples. Mixing domains using our ConDA strategy yields semantically realistic intermediate domain samples for self-training.

as well as to mitigate possible aliasing caused by region re-grouping. Furthermore, as the pseudo-labels are "guessed" by the model trained on the source domain, it is intuitive to leverage the uncertainty [31] of such "guesses" for filtering non-confident selections. Entropy [32] – a measure of choice freedom – has been proven conducive for estimating prediction uncertainty [33], [34]. Different from prior arts that implicitly minimize entropy in an adversarial way [35], [36], we design an entropy aggregator (Sec. III-D) which explicitly eliminates high entropy target predictions and thus improves the overall quality for the intermediate domain supervisions.

Overall, this work has the following key contributions:

- We propose ConDA, a novel framework that facilitates fine-grained interactive learning in-between domains. To the best of our knowledge, we are the first to explore cross-city UDA for uni-modal LiDAR segmentation, which can serve as a baseline for future research.
- We design two efficient regularization techniques to reduce detrimental aliasing artifacts and uncertain target predictions during model pre-training and self-training.
- We conduct comprehensive studies on the effects of our technical contributions on two challenging cross-city UDA scenarios. Our methods provide significant performance gains over state-of-the-art approaches.

II. Related Work

UDA on Visual RGB. Adversarial training [13], [15] and self-training [16], [17] dominate almost all kinds of 2D scene adaptation scenarios [37], [38], [39], [40]. Methods based on adversarial training adopt domain discriminators [14], [41] to implicitly search for domain-invariant features via distance measurements at different levels, *i.e.*, input-level [42], [43], [44], [45], [46], feature-level [47], [48], [49], and output-level [50], [35], [51]. However, such methods suffer from high computational costs and tend to be sensitive to hyperparameters and target domain changes [52], [53]. Self-training, on the other hand, offers a lighter option by jointly learning from both source and target domain supervisions [54], where the latter can be generated via confidence thresholding [55], [18], [56]. Evidence shows that these methods – albeit powerful in 2D – become less effective in 3D [53], [57], which motivates us to design new methods w.r.t. the characteristics of the LiDAR data, such as the spatial consistency of the range-view representation.

UDA on LiDAR Data. Adaptations are more challenging in 3D as point clouds are sparse, unstructured, and have limited visual cues compared to images [58], [59], [60]. ePointDA [61] learns a dropout noise rendering from real-world data to match synthetic data. Complete&Label [62] targets on cross-sensor UDA where sparsity rather than city discrepancies serves as the major domain gap, which is beyond the scope of this work. xMUDA [53] and DsCML [57] employ self-training alongside multi-modality learning for cross-city adaptations. However, the assumption of having access to synchronized RGB and LiDAR data in both source and target domains is not always practical, which limits such methods. Also, they use point/voxel backbones which require huge memory consumptions and suffer from low inference speed. Our framework is built upon mature 2D CNNs and does not require data from multiple modalities thus maintaining both simplicity and efficiency for practitioners.

Domain Mixed Inputs. Mixing-based strategies [22], [24], [25] have been widely adopted in fully- [63], [64], [65] and semi- [66], [67], [23] supervised learning tasks, but very few touched UDA. SimROD [68] proposed to create 2x2 collages with two source and two target images for object detection. DACS [44] cuts source objects out and pastes them onto the target images, alongside mixing corresponding source labels and target pseudo-labels. While the former [68] can corrupt the semantic consistency, the latter [44] requires extra costs for the "copy-paste" operation and lacks mixing background and target objects. Our ConDA provides more fine-grained interactions in-between domains and the concatenations are performed on the fly during training (via simple `torch.cat` or `tf.concat`) with almost zero cost.

Regularization. Anti-aliasing filters are popularly applied to reduce the aliasing artifacts in networks [26], [27], [28]. Such regularizers, however, have not been used in UDA. We conjecture that this is because the empirical observations required in filter design become impractical on the annotation-free target domain. In this work, we adopt a learning-based approach as an alternative to regularize aliasing artifacts. Another regularization is uncertainty estimation, which has been proved useful in semi-supervised learning [31]. For UDA, IntraDA [36] uses a discriminator to close the gap between the low-entropy and high-entropy samples. We design an entropy aggregator to disable the usage of non-confident

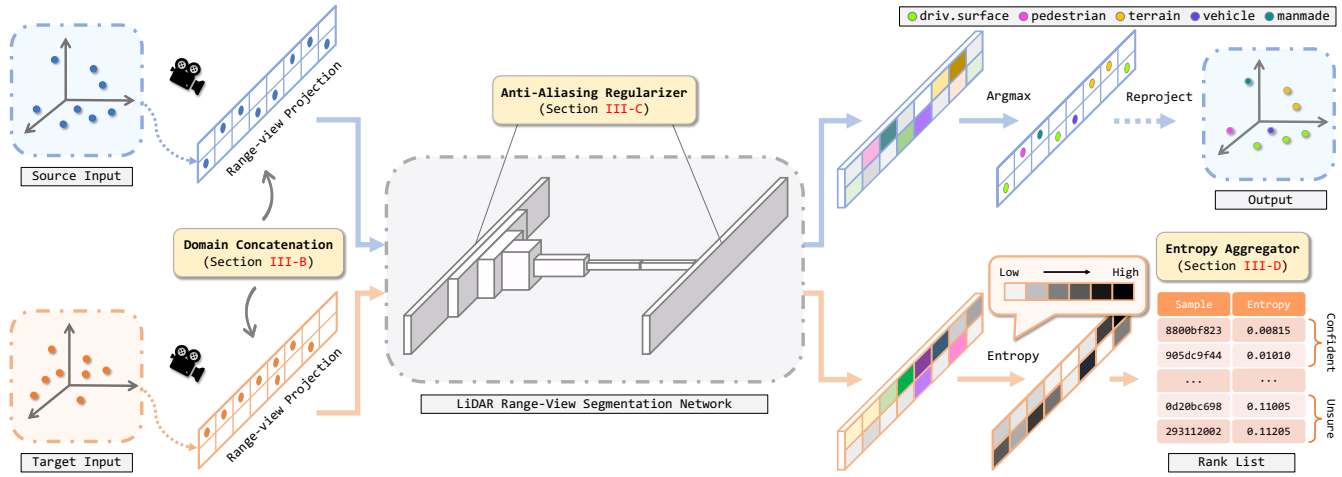


Fig. 2. Overview of our concatenation-based domain adaptation (ConDA) framework. After preprocessing (Sec. III-A), sample stripes from both domains are mixed via RV concatenation (Sec. III-B). The concatenated inputs are fed into the segmentation network for feature extraction. We include anti-aliasing regularizers inside convolution operations (Sec. III-C) to suppress the learning of high-frequency aliasing artifacts. The segmented RV cells are then projected back to the point clouds. Here the target prediction part is omitted for simplicity. To mitigate the impediment caused by erroneous target predictions, we design an entropy aggregator (Sec. III-D) which splits samples into a confident set and an unsure set and disables the usage of samples from the latter set.

pseudo-labels during domain adaptation, which does not rely on extra discriminator network and directly reduces the uncertainty for the intermediate domain supervisions.

III. Technical Approach

A. Preliminaries

The proposed ConDA framework consists of three major components as shown in Fig. 2. Let A denote a LiDAR point cloud. We project each point $\mathbf{o}=(x,y,z)$ on the 360° scan via a mapping $\Pi:\mathbb{R}^3\mapsto\mathbb{R}^2$ to a cylindrical RV image $\mathbf{a}\in\mathbb{R}^{6,h,w}$ with height h and width w. h is set based on the beam number of the sensor and w is determined by the horizontal angular resolution. Each pixel in $\mathbf{a}$ consists of the point coordinates (x,y,z), intensity, range $||\mathbf{o}||_2$, and a binary occupancy mask. Note that the RV projection preserves the range information and the spatial correspondence for the LiDAR scans, *e.g.*, the near-front and far-front points are projected onto the left-bottom and left-top of the RV images, respectively, which is a unique feature of the LiDAR representations (Fig. 1). To extract RV features efficiently, we design a seven-stage fully-convolutional network G with strided convolutions [69] and skip-connections [70] as our backbone. Since h is much smaller than w, we only downsample the height at later stages. The segmentation head combines the upsampled outputs from the last four stages for multi-scale feature aggregations. Compared to previous RV networks [71], [72], [73], [74], our proposed G offers a better trade-off between accuracy and speed, which is essential for LiDAR UDA.

B. Domain Concatenation

In the context of UDA for LiDAR segmentation, we denote samples $\mathbf{a}$ from the labeled source domain and unlabeled target domain as $A_s=\{(\mathbf{a}_{s,m},q_{s,m})\}_{m=1}^{M}$ and $A_t=\{(\mathbf{a}_{t,n})\}_{n=1}^{N}$, respectively, where M and N are the total number of source and target samples. For segmentation with C classes, the network G can be optimized in a supervised way with source samples by minimizing the cross-entropy loss as follows:

$$\min_{\mathbf{w}} L_s=-\frac{1}{M}\sum_{m=1}^{M}\sum_{c=1}^{C} q_{s,m}^{(c)}\log p(c|\mathbf{a}_{s,m},\mathbf{w}), \tag{1}$$

where $\mathbf{w}$ denotes the weights of G; $p(\cdot)$ is the probability of class c in the softmax output. Due to the lack of ground-truth in the target domain, we consider the target labels as hidden variables [16] and select the most confident target predictions on the existing model as one-hot "pseudo-labels" $\hat{q}_t$. The learning objective for the target domain is:

$$\min_{\mathbf{w}} L_t=-\frac{1}{N}\sum_{n=1}^{N}\sum_{c=1}^{C}\hat{q}_{t,n}^{(c)}\log p(c|\mathbf{a}_{t,n},\mathbf{w}),$$
$$\text{s.t. } \hat{q}_{t,n}=\begin{cases}\arg\max_c p(c|\mathbf{a}_{t,n},\mathbf{w}), & \text{if}\max(p(c|\mathbf{a}_{t,n},\mathbf{w}))\geq\theta\\ \text{ignored}, & \text{otherwise}\end{cases} \tag{2}$$

where θ is a threshold for filtering non-confident pseudo-labels. Similar to [18], [56], we set a proportion parameter k to determine the class-wise thresholds θ_c for each class c to balance the class distributions. Now given a source batch $\{(\mathbf{a}_s,q_s)\}$ and a target batch $\{(\mathbf{a}_t,\hat{q}_t)\}$, our intermediate domain construction mechanism $M(\cdot)$ follows three steps. First, define a template for the total number of segregation regions along the near-far dimension (m) in the RV projections and around the ego-vehicle (n). Fig. 1a shows an example of a domain concatenation with $m=2$ and $n=2$, having front-near, front-far, back-near, and back-far regions, respectively. Second, slice regions based on the above template for every sample in both batches. This gives $((b_s+b_t)\times m\times n)$ sliced stripes, where b_s and b_t are the batch sizes. Third, concatenate the stripes while keeping their spatial locations consistent, resulting in (b_s+b_t) intermediate domain samples $\mathbf{a}_\pi$. Their labels q_π can be obtained via the same arrangement of the original labels and pseudo-labels. The segmentation loss L_π for this intermediate domain can be computed using the mixed batch $\{(\mathbf{a}_\pi,q_\pi)\}$ in a way similar to Eq. 1. The overall objective for self-training is to minimize $L=L_s(\mathbf{w},q_s)+\sigma\cdot L_\pi(\mathbf{w},q_\pi)$, where σ is a coefficient that controls the probability of accessing the

intermediate domain. As shown in Fig. 1c, this simple concatenation approach mixes objects and background from both domains, while still preserving the overall consistency. This stability in semantic coherence comes from the priors that the LiDAR point clouds are unstructured and extremely sparse even after RV projections. Take nuScenes [3] as an example. We find that on average 59.93% of RV cells are empty, which could downplay the negative impact of region rearrangement since the degree of continuity is low.

C. Anti-Aliasing Regularizer

We formulate a regularizer that is built within each convolution filter in G to reduce learning from aliasing artifacts. Our goal is to impose regularization on high-frequency representation learning since they are more susceptible to aliasing artifacts [27], [28]. We achieve this via $\mathbf{f}_{c,r} = \mathbf{f}_r \odot \mathbf{f}_c$, where $\mathbf{f}_r$ denotes our regularizer which consists of learnable parameters having the same size as the convolution filter $\mathbf{f}_c$; $\mathbf{f}_{c,r}$ is the regularized filter kernel of each convolution; $\odot$ is the Hadamard multiplication. Note that during the earlier stages of training, the network tends to learn low-frequency representations [30], [29] that are robust to aliasing artifacts. This will thereby update $\mathbf{f}_r$ such that it becomes more suited for low-frequency representation learning of $\mathbf{f}_{c,r}$. In later phases of training or UDA self-training, however, networks are more inclined to learn increasingly higher frequency representations and thus become more susceptible to aliasing artifacts [26]. The modulation of our $\mathbf{f}_r$ on $\mathbf{f}_c$ regularizes gradient updates corresponding to these high-frequency representations and in particular, regularizes the ones that are considerably different from the earlier network learning. This implicit regularization mechanism at later stages of training makes $\mathbf{f}_{c,r}$ more resistant to aliasing artifacts than the plain $\mathbf{f}_c$. Notably, since $\mathbf{f}_c$ and $\mathbf{f}_r$ are both constants during inference, the regularized kernel $\mathbf{f}_{c,r}$ only needs to be computed once at the end of training and can be used at inference without adding any additional computational cost or structural changes to the network – this makes our regularizer unique among all previous anti-aliasing mechanisms.

D. Entropy Aggregator

Given the fact that the pseudo-labels generated from the source pre-trained model tend to be noisy [34], we design an entropy aggregator to disable the access of non-confident target predictions and thus improve the overall quality of the intermediate domain supervisions. More formally, given a target sample $\mathbf{a}_{t,n}$, its entropy map E_n composed of the normalized pixel-wise entropies can be calculated as follows:

$$E_n = \frac{-1}{\log(C)} \sum_{c=1}^{C} p(c|\mathbf{a}_{t,n}, \mathbf{w}) \log p(c|\mathbf{a}_{t,n}, \mathbf{w}). \quad (3)$$

The median value $\mathbf{e}_n$ of E_n for the target sample $\mathbf{a}_{t,n}$ is used as the global-level indicator of uncertainty, which is more robust than the average value due to the large "noisy" predictions for the empty RV cells. The target set $A_t = \{(\mathbf{a}_{t,n})\}_{n=1}^{N}$ is reorganized based on the entropy ranking and only the pseudo-labels from the top ϖ most confident target samples are included as the supervisions for the intermediate domain.

TABLE I

EFFECTIVENESS FOR EACH COMPONENT IN CONDA. EVALUATED ON THE BOSTON → SINGAPORE ADAPTATION SETTING.

Δ	mIoU	Configuration
-12.0	34.9	No adaptation (source-only)
-6.2	40.7	Baseline (vanilla self-training)
-5.0	41.9	+= Anti-aliasing filters (Sec. III-C)
-1.6	45.3	+= Domain concatenation (Sec. III-B)
-1.0	45.9	+= Entropy aggregator (Sec. III-D)
0.0	46.9	Full ConDA framework

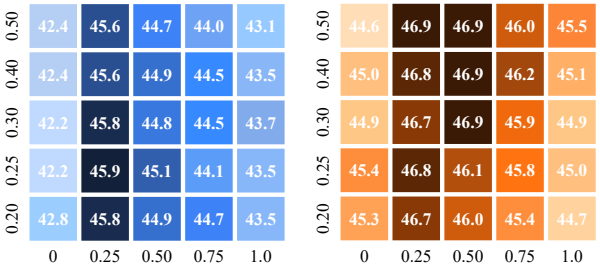


Fig. 3. Sensitivity analysis for the concatenation probability σ (horizontal axis) and proportion threshold k (vertical axis) in self-training round 1 (left) and round 2 (right). The darker the color, the higher the mIoU score.

IV. EXPERIMENTS AND ANALYSIS

A. Settings

Data. We construct two RV-based cross-city UDA scenarios with nuScenes [2], [3] – a large-scale autonomous driving database widely adopted in academia. We split samples based on their geographic locations. This gives 15695 and 12435 training samples and 3090 and 2929 evaluation samples for Boston and Singapore, respectively. All training samples are used as the source/target for adaptations. Different from xMUDA [53] which only assigns semantic labels to points inside bounding boxes with 4 object and 1 background classes, we adopt the *lidarseg* subset in nuScenes which contains 16 classes and fine-grained point-level annotations. **Implementation Details**. We project point clouds into RV images of size 32×1920 as the inputs for G (*cf.* Sec. III-A). It is first trained from scratch with only source samples for 80 epochs and then fine-tuned under our entropy aggregator-guided self-training procedure. Both rounds are trained for 20 epochs. We denote results for the supervised learning and direct adaptation as "oracle" and "source-only". Since this is a new benchmark, we could only compare ConDA with state-of-the-art adversarial [50], [35], [51], self-training [18], [55], [36], and consistency training [75], [76] methods originally tested for 2D adaptations. For fairness, we replace their backbones with our RV network and keep other configurations in default. For methods based on self-training, we generate their pseudo-labels offline as in [55], [18]. All methods are implemented using PyTorch on NVIDIA Tesla V100 GPUs. **Evaluation Metrics**. We follow the conventional reporting of the intersection-over-union (IoU) scores (%) over each class, the mean IoU (mIoU) and the frequency-weighted IoU (FIoU) scores (%) over all classes in our experiments.

TABLE II

ADAPTATION RESULTS FOR DIFFERENT RV CONCATENATION STRATEGIES (*cf.* FIG. 4) AND OTHER MIXING TECHNIQUES.

Method	w/o	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	[68]	[22]	[25]	[24]	[63]
mIoU	41.9	43.4	43.5	43.7	44.5	44.6	43.9	44.5	44.6	**45.3**	45.0	45.2	**45.3**	44.9	44.6	44.2	33.7	41.1	43.1	42.6	40.1

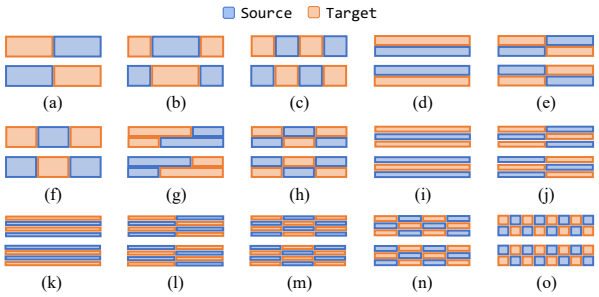


Fig. 4. ConDA RV concatenation strategies with various granularities.

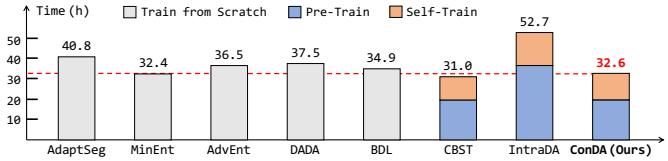


Fig. 5. Training time (in hours) needed for each method during adaptations.

B. Ablation Studies

Q1: ***What is the effect for each component in ConDA?***

A1: We adopt the Boston → Singapore setting in our ablation studies without the loss of generality. We stratify the three major components in our framework and show their impacts in Tab. I. Specifically, the anti-aliasing regularizer offers an improvement of 1.2% mIoU over the baseline and surpasses the source-only case by 7.0% mIoU. Comparing the frequency spectrum of 3x3 kernels from the network, we also find that the ratio of *'average low-frequency amplitudes'* to *'average high-frequency amplitudes'* is 23.2% higher (statistically significant with t-test: $p < 0.001$) for the network trained with our regularizer. On top of that, our domain concatenation further improves 3.4% mIoU. Another boost of 1.6% mIoU is achieved under the two-round guidance self-training of our entropy aggregator. Overall, our framework significantly improves the adaptation results from 34.9% mIoU to 46.9% mIoU, which corresponds to nearly a 34.4% relative improvement over the source-only.

Q2: ***What are the optimal hyperparameters for ConDA?***

A2: We conduct extensive experiments to show the best possible selections for the hyperparameters. Specifically, the vertical and horizontal axes of Fig. 3 show the impact of the proportion parameter k and the concatenation parameter σ during the two-round self-training. We find that lower values for k (*e.g.* 0.25) in round 1 and relatively higher values (*e.g.* 0.5) in round 2 tend to give higher mIoU. We conjecture that this relatively conservative choice (or lower value) at round 1 is helpful since pseudo-labels tend to be noisy at this stage. The quality of pseudo-label gets much better at round 2 and including more of them gives a positive impact on the performance. As for σ, we observe that the best possible scores are achieved between 0.25 and 0.50. Training *w/o* ($\sigma = 0$) or *w/* all ($\sigma = 1$) concatenated samples does not perform well. For ϖ (after setting both k and σ as 0.25 without the loss of generality), we find that large ϖ involves more false positives while small ϖ limits the diversity. A compromise value like 0.50 gives the best possible scores.

Q3: ***What is the best practice for RV concatenation?***

A3: Besides the intuitive front-back RV concatenation, we also consider other scenarios as in Fig. 4 and show their results in Tab. II. As shown, strategies *i* and *l* perform the best while *j* and *k* offer competitive results. We note that: 1) increasing the granularity of the interactions between source and target tends to improve performance (strategies *a* to *l*); 2) increasing the granularity beyond a certain limit (strategies *m*, *n*, and *o*) can deteriorate performance, which is likely due to the instability in semantic coherence of the objects and background in the concatenated stripes; and 3) fine-grained interactions along the vertical axes, *i.e.,* near to far regions, perform better than interactions along the horizontal axes, *i.e.,* bearing around the ego-vehicle (strategies *i* and *f*), suggesting that the former likely yields better domain interactions while better maintaining the semantic consistency.

Q4: ***How is domain concatenation superior to others?***

A4: We compare five popular mixing techniques in Tab. II. SimROD [68] stitched samples from both domains as inputs for adaptation while MixUp [22], CutMix [24], CutOut [25], and Mix3D [63] are general regularization methods adopted for fully- and semi-supervised learning. It can be seen that they have shown sub-par performance in the RV representation for UDA in LiDAR segmentation. Differently, our approach is able to effectively leverage the spatial context of RV and combine both domains into an intermediate domain for fine-grained interactive learning and regularization.

C. Comparison to the State of the Art

Benchmarking Results. We compare ConDA with eight state-of-the-art methods on the Boston → Singapore and Singapore → Boston scenarios in Tab. III and Tab. IV. In both cases, ConDA substantially outperforms other competitors in terms of mIoU and FIoU. Notably, in contrast to prior approaches that tend to improve performance on relatively easier classes (*i.e.*, classes on which the source-only has already performed well), ConDA yields considerable performance gains on almost all classes. This strongly supports our findings that fine-grained objects/background interactions in-between domains are conducive for closing the domain gap and thus improving the adaptations.

Qualitative Assessment. Fig. 6 presents visualization results for different methods, *i.e.*, self-training (CBST [18]), adversarial training (DADA [51]), and both (IntraDA [36]). We observe that while the prior arts only give limited gains in certain areas, ConDA mitigates the false predictions holistically in most regions around the ego-vehicle. We accredit this to both the generalization ability provided by domain

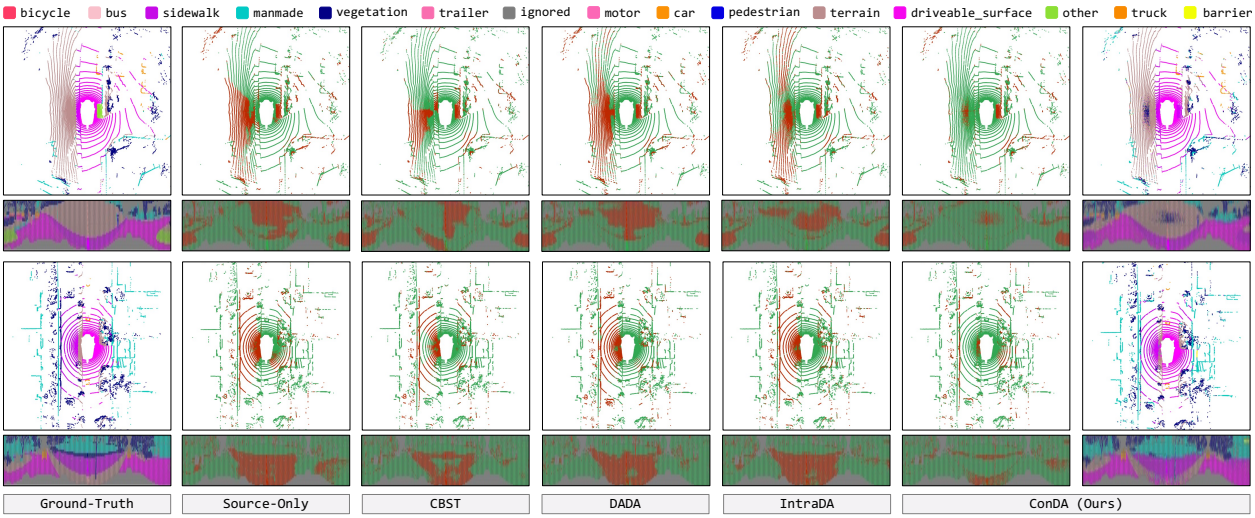


Fig. 6. Qualitative results from both the bird's eye view and range-view. To highlight the difference between the predictions and ground-truth, the **correct** and **incorrect** points/pixels are painted in **green** and **red**, respectively. Best viewed in color and zoom-ed in for details.

TABLE III

ADAPTATION RESULTS FOR **BOSTON** → **SINGAPORE**. THE METHODS ARE GROUPED, FROM TOP TO BOTTOM, AS ADVERSARIAL TRAINING, SELF-TRAINING, BOTH, AND OURS. ALL IOU SCORES ARE GIVEN IN PERCENTAGE (%). BEST SCORE FOR EACH CLASS IS HIGHLIGHTED IN **BOLD**.

Method	barr	bicy	bus	car	const	moto	ped	cone	trail	truck	driv	othe	walk	terr	manm	veg	FIoU ↑	mIoU ↑
Oracle	79.5	33.6	87.5	88.9	37.6	75.6	70.5	50.3	0.0	76.2	95.1	53.7	60.2	74.4	83.4	85.5	84.2	65.8
Source-only	29.3	1.3	52.0	71.4	7.2	11.7	42.6	12.2	0.0	30.4	85.9	12.7	32.6	41.0	62.5	65.9	64.3	34.9
AdaptSeg [50]	28.0	**7.2**	60.9	70.7	7.7	17.4	45.5	14.3	0.0	36.4	88.1	28.4	36.0	43.1	63.0	66.7	66.1	38.3
MinEnt [35]	31.7	4.0	63.7	70.6	5.8	15.9	47.7	13.7	**0.1**	34.9	87.9	22.4	37.5	41.5	59.9	62.2	64.3	37.5
AdvEnt [35]	28.7	5.9	59.4	76.4	7.2	18.2	50.6	**16.7**	0.0	32.6	87.0	28.1	36.6	44.0	63.9	67.1	66.3	38.9
DADA [51]	27.4	4.9	60.0	67.7	7.3	15.9	44.4	14.7	0.0	33.9	87.1	21.2	34.9	42.1	62.2	64.9	64.8	36.8
BDL_{PL} [55]	39.2	0.3	53.0	73.2	6.8	16.0	40.2	8.5	0.0	29.8	88.7	21.3	39.7	48.5	67.1	67.9	68.3	37.5
CBST [18]	39.4	5.3	66.1	75.6	9.3	20.7	47.8	14.9	0.0	34.1	88.4	25.5	38.1	49.9	66.7	68.5	68.6	40.7
$AdaptSeg_{PL}$ [50]	29.9	0.3	47.9	64.4	4.9	7.4	28.4	4.6	0.0	24.8	83.1	21.8	38.3	46.5	67.1	68.9	66.0	33.7
IntraDA [36]	28.0	5.6	57.8	76.1	6.2	18.6	47.4	13.8	0.0	32.1	87.3	27.6	37.0	44.4	63.4	66.5	66.2	38.3
ConDA (Ours)	**54.1**	6.8	**67.4**	**77.2**	**12.1**	**38.7**	**51.8**	16.0	0.0	**44.0**	**90.4**	**38.7**	**44.0**	**62.9**	**70.7**	**75.0**	**74.1**	**46.9**

TABLE IV

ADAPTATION RESULTS FOR **SINGAPORE** → **BOSTON**. THE METHODS ARE GROUPED, FROM TOP TO BOTTOM, AS ADVERSARIAL TRAINING, SELF-TRAINING, BOTH, AND OURS. ALL IOU SCORES ARE GIVEN IN PERCENTAGE (%). BEST SCORE FOR EACH CLASS IS HIGHLIGHTED IN **BOLD**.

Method	barr	bicy	bus	car	const	moto	ped	cone	trail	truck	driv	othe	walk	terr	manm	veg	FIoU ↑	mIoU ↑
Oracle	71.3	35.5	71.5	86.9	41.6	35.4	69.7	61.2	57.6	68.0	95.9	70.7	79.7	58.7	89.9	83.9	88.5	67.3
Source-only	15.5	7.9	20.6	70.5	16.1	3.6	41.9	11.4	0.5	40.6	90.2	10.7	41.7	19.1	77.4	74.5	73.4	33.9
AdaptSeg [50]	15.9	2.4	40.4	73.9	15.2	5.5	48.3	8.3	0.4	46.3	92.2	18.7	54.5	19.0	79.0	70.9	76.2	36.9
MinEnt [35]	**19.2**	0.2	36.1	73.2	15.7	6.2	50.3	10.8	0.8	45.0	91.5	24.1	54.8	21.7	78.7	71.8	76.0	37.5
AdvEnt [35]	12.5	**9.0**	43.0	74.1	14.7	7.0	51.4	12.7	0.5	47.0	91.4	14.5	53.6	19.2	80.1	73.4	76.2	37.8
DADA [51]	18.5	2.9	35.5	73.0	15.0	6.5	49.3	11.0	1.6	43.6	91.8	12.2	52.7	19.7	79.8	73.4	76.1	36.7
BDL_{PL} [55]	18.9	2.7	30.8	75.8	13.3	3.8	45.4	7.4	1.8	45.4	92.8	19.7	58.4	18.7	80.1	76.3	77.6	37.0
CBST [18]	17.7	1.4	33.6	75.0	13.3	6.4	52.3	12.3	**1.9**	46.9	92.5	22.8	57.2	19.7	80.2	77.3	77.5	38.1
$AdaptSeg_{PL}$ [50]	10.5	0.6	33.5	71.3	**17.2**	5.2	41.9	11.4	1.0	43.5	90.4	18.5	**60.1**	20.0	80.0	74.6	76.1	36.3
IntraDA [36]	12.3	6.2	41.2	73.4	14.1	5.6	43.5	13.4	0.7	**48.1**	91.2	16.4	54.1	18.8	79.2	70.6	75.7	36.8
ConDA (Ours)	11.8	**9.0**	**49.1**	**76.3**	7.2	**18.0**	**62.6**	**15.2**	0.0	47.9	**92.3**	**34.9**	59.4	**27.9**	**83.2**	**82.3**	**79.3**	**42.3**

concatenation and the regularization enhancement offered by our anti-aliasing regularizer and entropy aggregator.

Training Complexity. Fig. 5 shows the training time of different methods. Note that the pseudo-label generation time is excluded since this operation is conventionally conducted offline [53], [18], [55]. We find that while the inference speeds for these methods are similar (since they are sharing the same range-view segmentation backbone), ConDA is still faster than most adversarial methods [50], [35], [51] which rely on additional discriminators for learning domain-invariant features during the adaptation. Our work is comparable with MinEnt [35] and CBST [18] in terms of speed but provides much better adaptation performance.

V. CONCLUSION

We presented ConDA, a concatenation-based UDA framework that leverages the spatial coherency in-between the source and target domains for fine-grained interactive learning. Extensive experiments showed that ConDA can substantially improve the segmentation performance over baselines and competitive approaches. The robustness of our framework has shed light on its utility and potential for flexible deployment in the autonomous driving perception system.

Acknowledgements. This research is part of the programme DesCartes and is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. This work is affiliated with the WP4 of the DesCartes programme, with an identity number: A-8000237-00-00.

REFERENCES

[1] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020.

[3] W. K. Fong, R. Mohan, J. V. Hurtado, L. Zhou, H. Caesar, O. Beijbom, and A. Valada, "Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking," *IEEE Robotics and Automation Letters*, pp. 3795–3802, 2022.

[4] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9297–9307, 2019.

[5] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, "One thousand and one hours: Self-driving motion prediction dataset," in *Conference on Robot Learning*, pp. 409–418, 2021.

[6] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Z. Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2446–2454, 2020.

[7] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8748–8757, 2019.

[8] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, and P. Schuberth, "A2d2: Audi autonomous driving dataset," *arXiv preprint arXiv:2004.06320*, 2020.

[9] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A lidar point cloud generator: From a virtual world to autonomous driving," in *ACM International Conference on Multimedia Retrieval*, pp. 458–464, 2018.

[10] B. Gao, Y. Pan, C. Li, S. Geng, and H. Zhao, "Are we hungry for 3d lidar data for semantic segmentation? a survey of datasets and methods," *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[11] I. Achituve, H. Maron, and G. Chechik, "Self-supervised learning for domain adaptation on point clouds," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 123–133, 2021.

[12] M. Toldo, A. Maracani, U. Michieli, and P. Zanuttigh, "Unsupervised domain adaptation in semantic segmentation: A review," *Technologies*, vol. 8, no. 2, pp. 1–35, 2020.

[13] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *International Conference on Machine Learning*, pp. 1180–1189, 2015.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, pp. 2672—2680, 2014.

[15] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, pp. 1–35, 2016.

[16] D.-H. Lee, "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks," in *International Conference on Machine Learning Workshop*, pp. 896–901, 2013.

[17] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. V. Le, "Rethinking pre-training and self-training," in *Advances in Neural Information Processing Systems*, 2020.

[18] Y. Zou, Z. Yu, B. V. K. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *European Conference on Computer Vision*, pp. 289–305, 2018.

[19] N. Araslanov and S. Roth, "Self-supervised augmentation consistency for adapting semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15384–15394, 2021.

[20] Y. Luo, J. Zhu, M. Li, Y. Ren, and B. Zhang, "Smooth neighbors on teacher graphs for semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8896–8905, 2018.

[21] G. French, S. Laine, T. Aila, M. Mackiewicz, and G. Finlayson, "Semi-supervised semantic segmentation needs strong, varied perturbations," in *British Machine Vision Conference*, 2020.

[22] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018.

[23] V. Verma, K. Kawaguchi, A. Lamb, J. Kannala, A. Solin, Y. Bengio, and D. Lopez-Paz, "Interpolation consistency training for semi-supervised learning," *Neural Networks*, 2021.

[24] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6023–6032, 2019.

[25] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[26] C. Vasconcelos, H. Larochelle, V. Dumoulin, R. Romijnders, N. L. Roux, and R. Goroshin, "Impact of aliasing on generalization in deep convolutional networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10529–10538, 2021.

[27] R. Zhang, "Making convolutional networks shift-invariant again," in *International Conference on Machine Learning*, pp. 7324–7334, 2019.

[28] X. Zou, F. Xiao, Z. Yu, and Y. J. Lee, "Delving deeper into anti-aliasing in convnets," in *British Machine Vision Conference*, 2020.

[29] S. Sinha, A. Garg, and H. Larochelle, "Curriculum by smoothing," in *Advances in Neural Information Processing Systems*, 2020.

[30] Y. Cao, Z. Fang, Y. Wu, D.-X. Zhou, and Q. Gu, "Towards understanding the spectral bias of deep learning," in *International Joint Conference on Artificial Intelligence*, pp. 2205–2211, 2021.

[31] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Proceedings of the International Conference on Neural Information Processing Systems*, pp. 529–536, 2004.

[32] C. E. Shannon, "A mathematical theory of communication," *he Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[33] Z. Zheng and Y. Yang, "Unsupervised scene adaptation with memory regularization in vivo," in *International Joint Conference on Artificial Intelligence*, pp. 1076–1082, 2020.

[34] Y. Wang, J. Peng, and Z. Zhang, "Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9092–9101, 2021.

[35] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2517–2526, 2019.

[36] F. Pan, I. Shin, F. Rameau, S. Lee, and I. S. Kweon, "Unsupervised intra-domain adaptation for semantic segmentation through self-supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3764–3773, 2020.

[37] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European Conference on Computer Vision*, pp. 102–118, 2016.

[38] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3234–3243, 2016.

[39] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3213–3223, 2016.

[40] D. Dai and L. V. Gool, "Dark model adaptation: Semantic image segmentation from daytime to nighttime," in *International Conference on Intelligent Transportation Systems*, pp. 3819–3824, 2018.

[41] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation

learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations*, 2016.

[42] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in *International Conference on Machine Learning*, pp. 1989–1998, 2015.

[43] H. Ma, X. Lin, Z. Wu, and Y. Yu, "Coarse-to-fine domain adaptive semantic segmentation with photometric alignment and category-center regularization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4051–4060, 2021.

[44] W. Tranheden, V. Olsson, J. Pinto, and L. Svensson, "Dacs: Domain adaptation via cross-domain mixed sampling," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1379–1389, 2021.

[45] Y. Yang and S. Soatto, "Fda: Fourier domain adaptation for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4085–4095, 2020.

[46] J. Yang, W. An, S. Wang, X. Zhu, C. Yan, and J. Huang, "Label-driven reconstruction for domain adaptation in semantic segmentation," in *European Conference on Computer Vision*, pp. 480–498, 2020.

[47] Y. Luo, L. Zheng, T. Guan, J. Yu, and Y. Yang, "Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2507–2516, 2019.

[48] Z. Wu, X. Han, Y.-L. Lin, M. G. Uzunbas, T. Goldstein, S. N. Lim, and L. S. Davis, "Dcan: Dual channel-wise alignment networks for unsupervised scene adaptation," in *European Conference on Computer Vision*, pp. 518–534, 2018.

[49] L. Du, J. Tan, H. Yang, J. Feng, X. Xue, Q. Zheng, X. Ye, and X. Zhang, "Ssf-dan: Separated semantic feature based domain adaptation network for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 982–991, 2019.

[50] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. Chandraker, "Learning to adapt structured output space for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7472–7481, 2018.

[51] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez, "Dada: Depth-aware domain adaptation in semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7364–7373, 2019.

[52] Z. Luo, Z. Cai, C. Zhou, G. Zhang, H. Zhao, S. Yi, S. Lu, H. Li, S. Zhang, and Z. Liu, "Unsupervised domain adaptive 3d detection with multi-level consistency," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8866–8875, 2021.

[53] M. Jaritz, T.-H. Vu, R. de Charette, E. Wirbel, and P. Pérez, "xmuda: Cross-modal unsupervised domain adaptation for 3d semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12605–12614, 2020.

[54] Z. Wang, M. Yu, Y. Wei, R. Feris, J. Xiong, W. mei Hwu, T. S. Huang, and H. Shi, "Differential treatment for stuff and things: A simple unsupervised domain adaptation method for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12635–12644, 2020.

[55] Y. Li, L. Yuan, and N. Vasconcelos, "Bidirectional learning for domain adaptation of semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6936–6945, 2019.

[56] Y. Zou, Z. Yu, X. Liu, B. V. K. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5982–5991, 2019.

[57] D. Peng, Y. Lei, W. Li, P. Zhang, and Y. Guo, "Sparse-to-dense feature matching: Intra and inter domain cross-modal learning in domain adaptation for 3d semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7108–7117, 2021.

[58] L. T. Triess, M. Dreissig, C. B. Rist, and J. M. Zöllner, "A survey on deep domain adaptation for lidar perception," in *IEEE Intelligent Vehicles Symposium Workshops*, pp. 350–357, 2021.

[59] F. Langer, A. Milioto, A. Haag, J. Behley, and C. Stachniss, "Domain transfer for semantic segmentation of lidar data using deep neural networks," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 8263–8270, 2020.

[60] P. Jiang and S. Saripalli, "Lidarnet: A boundary-aware domain adaptation model for point cloud semantic segmentation," in *IEEE International Conference on Robotics and Automation*, pp. 2457–2464, 2021.

[61] S. Zhao, Y. Wang, B. Li, B. Wu, Y. Gao, P. Xu, T. Darrell, and K. Keutze, "epointda: An end-to-end simulation-to-real domain adaptation framework for lidar point cloud segmentation," in *AAAI Conference on Artificial Intelligence*, pp. 3500–3509, 2021.

[62] L. Yi, B. Gong, and T. Funkhouser, "Complete & label: A domain adaptation approach to semantic segmentation of lidar point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15363–15373, 2021.

[63] A. Nekrasov, J. Schult, O. Litany, B. Leibe, and F. Engelmann, "Mix3d: Out-of-context data augmentation for 3d scenes," in *IEEE International Conference on 3D Vision*, pp. 116–125, 2021.

[64] M. Xu, J. Zhang, B. Ni, T. Li, C. Wang, Q. Tian, and W. Zhang, "Adversarial domain adaptation with domain mixup," in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 6502–6509, 2020.

[65] A. Sahoo, R. Panda, R. Feris, K. Saenko, and A. Das, "Select, label, and mix: Learning discriminative invariant feature representations for partial domain adaptation," *arXiv preprint arXiv:2012.03358*, 2020.

[66] V. Olsson, W. Tranheden, J. Pinto, and L. Svensson, "Classmix: Segmentation-based data augmentation for semi-supervised learning," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1369–1378, 2021.

[67] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. A. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," in *Advances in Neural Information Processing Systems*, 2019.

[68] R. Ramamonjison, A. Banitalebi-Dehkordi, X. Kang, X. Bai, and Y. Zhang, "Simrod: A simple adaptation method for robust object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3570–3579, 2021.

[69] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[70] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

[71] C. Xu, B. Wu, Z. Wang, W. Zhan, P. Vajda, K. Keutzer, and M. Tomizuka., "Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation," in *European Conference on Computer Vision*, pp. 1–19, 2020.

[72] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4213–4220, 2019.

[73] T. Cortinhal, G. Tzelepis, and E. E. Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds for autonomous driving," in *Advances in Visual Computing: 15th International Symposium*, pp. 207–222, 2020.

[74] I. Alonso, L. Riazuelo, L. Montesano, and A. C. Murillo, "3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5432–5439, 2020.

[75] A. Tarvainen and H. Valpola, "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results," in *Advances in Neural Information Processing Systems*, 2017.

[76] X. Chen, Y. Yuan, G. Zeng, and J. Wang, "Semi-supervised semantic segmentation with cross pseudo supervision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2613–2622, 2021.