Continuous Pseudo-Label Rectified Domain Adaptive Semantic Segmentation with Implicit Neural Representations

Rui Gong[1] Qin Wang[1] Martin Danelljan[1] Dengxin Dai[2] Luc Van Gool[1]
[1]CVL, ETH Zürich [2]Max Planck Institute for Informatics, Saarland Informatics Campus
{gongr, qin.wang, martin.danelljan, vangool}@vision.ee.ethz.ch, ddai@mpi-inf.mpg.de

Abstract

Unsupervised domain adaptation (UDA) for semantic segmentation aims at improving the model performance on the unlabeled target domain by leveraging a labeled source domain. Existing approaches have achieved impressive progress by utilizing pseudo-labels on the unlabeled target-domain images. Yet the low-quality pseudo-labels, arising from the domain discrepancy, inevitably hinder the adaptation. This calls for effective and accurate approaches to estimating the reliability of the pseudo-labels, in order to rectify them. In this paper, we propose to estimate the rectification values of the predicted pseudo-labels with implicit neural representations. We view the rectification value as a signal defined over the continuous spatial domain. Taking an image coordinate and the nearby deep features as inputs, the rectification value at a given coordinate is predicted as an output. This allows us to achieve high-resolution and detailed rectification values estimation, important for accurate pseudo-label generation at mask boundaries in particular. The rectified pseudo-labels are then leveraged in our rectification-aware mixture model (RMM) to be learned end-to-end and help the adaptation. We demonstrate the effectiveness of our approach on different UDA benchmarks, including synthetic-to-real and day-to-night. Our approach achieves superior results compared to state-of-the-art. The implementation is available at https://github.com/ETHRuiGong/IR2F.

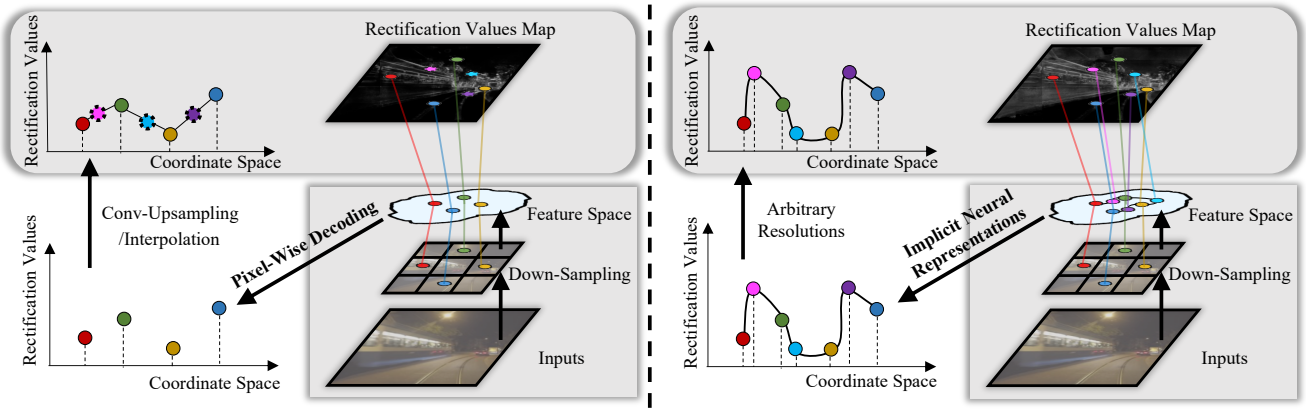
1. Introduction

Semantic segmentation, aiming at assigning the semantic label to each pixel in an image, is a fundamental problem in computer vision. Driven by the availability of large-scale datasets and the advancements in deep neural networks (DNNs), the state-of-the-art boundary has been pushed rapidly in the last decade [9, 35, 38, 51, 59, 70, 78]. However, the DNNs trained on a source domain, *e.g.* day images, generalize poorly to a different target domain, *e.g.* night images, due to the distribution shift between the domains. One straightforward idea to circumvent the issue is to annotate the images from the target domain, and then retrain the model. However, annotations for semantic segmentation are particularly costly and labor-intensive to produce, since each pixel has to be labeled. To this end, some recent works [18, 21, 61, 63, 77] resort to unsupervised domain adaptation (UDA), where the model is trained on the labeled source domain and an unlabeled target domain dataset, reducing the annotation burden.

Different from the predominant UDA methods that explicitly align the source and target distributions on the image-level [18, 21, 33, 73] or the feature-level [61–63], pseudo-labeling or self-training [23, 24, 60, 76, 82, 83] has recently emerged as a simple yet effective approach for UDA. Pseudo-labeling approaches typically first generate pseudo-labels on the unlabeled target domain using the current model. The model is then fine-tuned with target pseudo-labels in an iterative manner. However, some pseudo-labels are inevitably incorrect because of the domain shift. Therefore, pseudo-label correction, or rectification, is critical for the adaptation process. This is typically implemented in the literature by removing [82, 83] or assigning a smaller weight [24, 67, 76, 79] to pixels with low-quality and potentially incorrect pseudo-labels. The key problem is thus to formulate a *rectification function that estimates the pseudo-label quality*. We identify two important issues with current approaches.

First, most existing methods use *hard-coded heuristics* as the rectification function, *e.g.* hard thresholding of the softmax confidence [82, 83], prediction variances of different learners [79], or distance to prototypes [67, 76]. These heuristic rectification functions assume on strong correlations between the function and the pseudo-label quality, which may not be the case. For example, the rectification function that uses the variance of multiple learners [79] to suppress disagreement on the pseudo-labels can be sensitive to small objects in the adaptation [24].

The second issue is that the existing works [24] typically model the rectification function in a *discrete* spatial grid



(a) Discrete Modeling (b) Continuous Modeling

Figure 1. **Discrete *vs*. Continuous Rectification Function Modeling.** Discrete modeling suffers from the convolutional pixel-wise decoding in the fixed-grid, where some coordinates are missing (see dashed circle in (a)). Thus, the rectification values corresponding to these coordinates can only be obtained by upsampling/interpolation, which is constrained by the blurring effect and induces the inaccurate rectification values estimation in some areas, *e.g.* mask boundaries. In contrast, our continuous modeling decodes the features – in the continuous coordinate space – into rectification values, which can be generalized to arbitrary resolution and preserve finer details. (The coordinate space and rectification values are shown in 1-D axis just for better viewing.)

(see Fig.1a). Rectification values are predicted by the pixel-wise decoding from the fixed-grid feature space, which is constrained by the limited resolution. This is especially harmful when the objects in the test images are of a different scale than in the training, since the rectification function cannot generalize well on these unseen scales (see Fig.1a). Existing approaches also lose vital high-frequency information through down-/up-sampling operations [24, 25, 40, 56], which may lead to poorer pseudo-labels, in particular close to mask boundaries.

To address these two issues, we propose a novel continuous rectification-aware mixture model (RMM). **First**, instead of formulating the rectification function with heuristics and priors, we propose a principled mixture model representation, *i.e*. rectification-aware mixture model (RMM), ensuring a probabilistic end-to-end ***learnable*** formulation. **Second**, the rectification function in RMM is represented by our proposed implicit rectification-representative function (IR^{2}F), to model the pixel-wise rectification of pseudo-labels in ***continuous*** spatial coordinates, *i.e*. *continuous RMM*. The primary idea of IR^{2}F is to learn pixel-wise rectification values as latent codes, which are decoded at arbitrary continuous spatial coordinates. Given a queried coordinate, our IR^{2}F inputs latent codes around the given coordinate from the different learners (*e.g*. high-/low-resolution decoder in [24] and primary/auxiliary classifier in [79]) along with their spatial coordinates. IR^{2}F then predicts the rectification value at the queried coordinate. Our *principled* formulation is a general *plug-in* module, compatible with different rectification-aware UDA architectures.

We thoroughly analyze our continuous RMM on different UDA benchmarks, including *synthetic-to-real* and *day-to-night* settings. Extensive experimental results demonstrate the effectiveness of continuous RMM, outperforming the previous state-of-the-art (SOTA) methods by a large margin, including on SYNTHIA→Cityscapes (**+1.9%** mIoU), Cityscapes→Dark Zurich (**+3.0%** mIoU) and ACDC-Night (**+3.4%** mIoU). Overall, continuous RMM reveals the significant potential of modeling pseudo-labels rectification for UDA in the learnable and continuous manner, inspiring further research in this field.

2. Related Work

Unsupervised Domain Adaptation (UDA). UDA for semantic segmentation aims at adapting the model from the labeled source domain to the unlabeled target domain. To this end, different strategies are proposed, which can be generally categorized into two classes: 1) *adversarial learning* based algorithms make use of domain discriminator to align the domain distributions on the images inputs space [14, 43, 48], features space [22] and outputs space [39, 61, 64]; 2) *pseudo-labeling (or self-training)* based algorithms typically generate pseudo-labels on the unlabeled target domain. To avoid the error accumulation caused by noisy pseudo-label drift, different approaches have been developed for pseudo-label rectification, *e.g*. confidence thresholding [82,83], uncertainty estimation [67,79] and pseudo-label prototypes [67, 76]. These methods formulate the pseudo-label rectification function as hard-coded heuristics, while our method formulates the rectification function in the end-to-end learnable manner.

Implicit Neural Representations (INR). Implicit neural

representations are originally proposed for 3D reconstruction, where object shapes [2, 11, 19, 45, 74], scene surfaces [27,47,58,75] and structure appearances [3,41,42,80] are represented as a multi-layer perceptron (MLP). The core idea is to map coordinates to signals with MLP. Very recently, the vast success of implicit neural representations in 3D reconstruction motivates the further exploration in 2D tasks, *e.g.* image representations [11, 57], image super-resolution [10, 72], and feature alignment [25]. Different from previous methods that explore the in-domain learning, we focus on leveraging implicit neural representations to rectify pseudo-labels to help the cross-domain adaptation.

3. Method

3.1. Preliminary

In UDA problem, we are given the well-labeled source domain, $\mathcal{D}_s = \{(\mathbf{x}_i^s, \mathbf{y}_i^s)\}_{i=1}^{n_s}$, and the unlabeled target domain, $\mathcal{D}_t = \{\mathbf{x}_i^t\}_{i=1}^{n_t}$, where $\mathbf{x}^s, \mathbf{x}^t \in \mathbb{R}^{H\times W\times 3}$ are RGB images while $\mathbf{y}^s \in \{0,1\}^{H\times W\times C}$ is the C-class semantic label map associated with $\mathbf{x}^s$. The goal of UDA is to train the semantic segmentation model $\mathcal{F}_\theta$ on $\mathcal{D}_s, \mathcal{D}_t$ and evaluate $\mathcal{F}_\theta$ on the target domain testing data.

Since the ground truth label $\mathbf{y}^t$ corresponding to $\mathbf{x}^t$ is not available, the pseudo-labeling (or self-training) strategy for UDA generates pseudo-labels by, $\hat{\mathbf{y}}^{t(i,j,c)} = [c = \arg\max \mathcal{F}_\theta(\mathbf{x}^t)^{(i,j)}]$, where (i, j, c) represents (row, column, class) index and $[\cdot]$ is the Iverson bracket. Then $\mathcal{F}_\theta$ is trained by, $\mathcal{L}_{ce} = CE(\mathcal{F}_\theta(\mathbf{x}^t), \hat{\mathbf{y}}^t) + CE(\mathcal{F}_\theta(\mathbf{x}^s), \mathbf{y}^s)$, where $CE(\cdot)$ denotes the cross-entropy loss. As pseudo labels $\hat{\mathbf{y}}^t$ are not necessarily correct, different schemes are advocated to rectify pseudo labels, where the rectification function is denoted as $\mathcal{H}(\cdot)$. Most existing pseudo-label rectifying methods can be categorized into one of the following three types, 1) weighting pseudo-label based cross-entropy loss with the estimated rectification values $\mathcal{H}(\mathbf{x}^t)$ [79], *i.e.* $\mathcal{L}_{ce}^t = \mathcal{H}(\mathbf{x}^t) \odot CE(\mathcal{F}_\theta(\mathbf{x}^t), \hat{\mathbf{y}}^t)$; 2) weighting soft pseudo-labels with the estimated rectification values $\mathcal{H}(\mathbf{x}^t)$ [67,76], *i.e.* $\hat{\mathbf{y}}^{t(i,j,c)} = [c = \arg\max(\mathcal{H}(\mathbf{x}^t)^{(i,j)} \odot \mathcal{F}_\theta(\mathbf{x}^t)^{(i,j)})]$; 3) averaging pseudo-labels from multiple K learners (*e.g.* decoders) [4, 26, 66] to rectify pseudo labels of each single learner, *i.e.* $\hat{\mathbf{y}}^{t(i,j)} = \frac{1}{K}\sum_{k=1}^{K} \mathcal{F}_{\theta^k}(\mathbf{x}^t)^{(i,j)}$, where $\odot$ denotes the element-wise multiplication.

In general, such pseudo-labeling-based approaches can be categorized into *non-ensemble* (type 1 and 2) and *ensemble* based solutions (type 3). In the domain adaptation and generalization field, numerous empirical and theoretic comparisons [1,6,26,34,39,81] between these two classes have been conducted before and after the deep learning revolution. The consensus is that ensembles can take advantage of different ensemble members (*e.g.* different data augmentation, different resolutions image and different level features as shown in Fig. 2a) to adaptively filter pseudo-label noise, and have the potential to overcome the problem of mode collapse/overfitting [6,28,50,66] in non-ensemble methods. Thus, the *ensemble* method is particularly remarkable and taken as the test-bed in this work.

3.2. Rectification-Aware Mixture Model

The key is *how to formulate a rectification function to estimate the pseudo-labels quality*. Instead of utilizing hard-coded heuristics and priors as the rectification function, we propose a *principled end-to-end learnable* formulation. Based on the fact that existing methods make use of multiple members (auxiliary classifiers/decoders, prototypes, different images resolutions/augmentations) to rectify the models [24, 79], we reformulate the pseudo-labels rectification problem in principled manner as learning a rectification-aware mixture model (RMM), drawing inspiration from mixture density networks (MDN) [5] and deep ensembles [29, 44]. In RMM, each mixture member is weighted by the rectification function, which is the measurement of pseudo-labels quality of the corresponding member, formulated as,

$$p(\hat{\mathbf{y}}^t|\mathbf{x}^t) = \sum_{k=1}^{K} \mathbf{r}_k \phi_k(\hat{\mathbf{y}}^t|\theta_k, \mathbf{x}^t)\,, \tag{1}$$

where K is the number of mixture members, $\phi_k(\cdot|\theta_k)$ denotes an arbitrary parametric distribution conditioned on parameters θ_k, and $\mathbf{r}_k = \mathcal{H}(\mathbf{x}^t)$ are the estimated rectification values by the rectification function $\mathcal{H}(\cdot)$, satisfying $\sum_{k=1}^{K} \mathbf{r}_k = 1$. Specifically, the primary/auxiliary decoders in [79], the high-resolution/low-resolution image decoders in [24] and the different data augmentation techniques in [1] can be seen as $\phi(\cdot|\theta_k)$ in Eq. (1), as shown in Fig. 2a. Benefiting from RMM, the rectification function $\mathcal{H}(\cdot)$ can be learned in the end-to-end way.

3.3. Implicit Rectification-Representative Function

In this section, we first introduce how to model the rectification function continuously with implicit neural representations, and then leverage the continuous rectification function in RMM to obtain the continuous RMM.

Continuous Rectification Function Modeling with IR^{2}F. *Representing rectification function* $\mathcal{H}(\cdot)$ is the core part of building a rectification-aware mixture model. Current approaches essentially model rectification function in a *discrete* way. They compute rectification values on a predefined discrete grid, often using convolutional decoders and disregarding intermediate locations. For example, as shown in Fig. 2b, [24] introduces an additional convolutional decoder, as $\mathcal{H}(\cdot)$, to predict $\mathbf{r}_k$ on the discrete fixed-grid. However, this leads to coarse and over-smoothed outputs due to the low resolution and up/down-sampling stages in the decoder. On the other had, spatially detailed rectification values are important in order to achieve high-quality

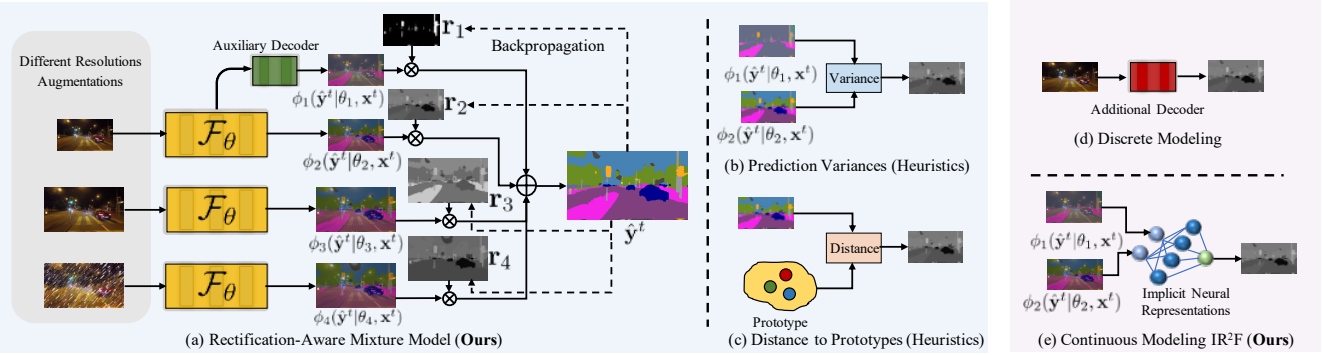


Figure 2. **Rectification-Aware Mixture Model (RMM) and Different Rectification Function Modeling.** Our rectification function is learned end-to-end by our proposed RMM as shown in (a), without relying on the predefined heuristics in (b) and (c). Moreover, rectification function in our RMM is modeled in the continuous manner, by the proposed implicit rectification-representative function (IR^{2}F) in (e), to overcome the resolution limitation of the fixed-grid discrete modeling in (d).

pseudo labels, especially at mask boundaries [7, 24]. To overcome the problems and get spatially accurate rectification values, the key idea of this work is to employ the *continuous* rectification function modeling mechanism, which is *learnable* and then decoded at continuous spatial coordinates in *arbitrary* resolution.

To this end, our proposed implicit rectification-representative function (IR^{2}F) views the pixel-wise rectification value $\mathbf{r}_k$ as a *continuous* signal in the 2D coordinate space. Inspired by implicit neural representations [42, 58] for 3D shape reconstruction and 2D image super-resolution [10, 72], our implicit rectification-representative function (IR^{2}F) aims at learning the implicit function $f_{\theta'}$ to decode the feature map $\mathcal{G}(\mathbf{x}^t)$ into the pixel-wise rectification values $\mathbf{r}_k$. That is, $\mathcal{H}(\cdot)$ in Sec. 3.2 is represented by $f_{\theta'}$. $\mathbf{r}_k$ is continuously decoded in the 2D coordinate space $\mathcal{O}$, formulated as,

$$\mathbf{r}^{\mathbf{o}_q} = (\mathbf{r}_k^{\mathbf{o}_q})_{k=1}^{K} = f_{\theta'}(\mathcal{G}(\mathbf{x}^t)^*, \mathbf{o}_q - \mathbf{o}^*), \tag{2}$$

where $\mathbf{o}_q \in \mathcal{O}$ is a queried 2D coordinate in the continuous coordinate space $\mathcal{O}$, and $(\mathbf{r}_k^{\mathbf{o}_q})_{k=1}^{K} = (\mathbf{r}_1^{\mathbf{o}_q}, \dots, \mathbf{r}_K^{\mathbf{o}_q})$ is the predicted rectification values for all ensamble members at location $\mathbf{o}_q$. $f_{\theta'}$ is parameterized by θ' as a multi-layer perceptron (MLP). $\mathcal{G}(\mathbf{x}^t)^*$ is the nearest feature vector from $\mathbf{o}_q$ in $\mathcal{G}(\mathbf{x}^t)$, and $\mathbf{o}^*$ is the 2D coordinate of $\mathcal{G}(\mathbf{x}^t)^*$ in $\mathcal{O}$. IR^{2}F can be seen as the mapping from the coordinate space to the rectification value space, *i.e.* $f_{\theta'}(\mathcal{G}(\mathbf{x}^t), \cdot) : \mathcal{O} \rightarrow \mathcal{R}$.

Spatial Encoding. As noticed by previous works [57, 72], directly inputting the spatial coordinates to an MLP of the implicit neural representation leads to a loss of high-frequency content. However, the high-frequency information, *e.g.* the edge information between the objects, is crucial to UDA for semantic segmentation as pointed out in [7, 24, 36]. In order to overcome this shortcoming, following [25, 72], we employ a spatial encoding of the spatial coordinates, before it is fed into the MLP of our IR^{2}F in Eq. (2). We use a sinusodal positional encoding,

$$\psi(\mathbf{o}) = (\sin(\omega_1 \mathbf{o}), \cos(\omega_1 \mathbf{o}), \dots, \sin(\omega_n \mathbf{o}), \cos(\omega_n \mathbf{o})), \tag{3}$$

$$\mathbf{r}^{\mathbf{o}_q} = f_{\theta'}(\mathcal{G}(\mathbf{x}^t)^*, \psi(\mathbf{o}_q - \mathbf{o}^*), \mathbf{o}_q - \mathbf{o}^*). \tag{4}$$

where the frequencies $\omega_1, \omega_2, \dots, \omega_n$ are learnable during training and n is the spatial encoding dimension.

Continuous RMM based on IR^{2}F. Benefiting from the continuous rectification function modeling with IR^{2}F in Sec. 3.3, rectification values of our proposed RMM in Sec. 3.2 are predicted in the continuous coordinate space, and can be generalizable to arbitrary resolution. Moreover, to take advantage of multiple learners in our RMM, the input representation $\mathcal{G}(\mathbf{x}^t)$ in Eq. (4) is obtained by stacking the feature information from different ensemble members,

$$\mathcal{G}(\mathbf{x}^t) = \text{Concat}(\mathcal{G}_1(\mathbf{x}^t), \mathcal{G}_2(\mathbf{x}^t), \dots, \mathcal{G}_K(\mathbf{x}^t)), \tag{5}$$

Then rectification values for RMM are obtained by substituting Eq. (5) into Eq. (4). Therefore, considering Eq. (4) and Eq. (1), the continuous RMM can be formulated as,

$$p(\hat{\mathbf{y}}^t | \mathbf{x}^t, \mathbf{o}_q) = \sum_{k=1}^{K} \mathbf{r}_k^{\mathbf{o}_q} \phi_k(\hat{\mathbf{y}}^t | \theta_k, \mathbf{x}^t). \tag{6}$$

Here, $p(\hat{\mathbf{y}}^t | \mathbf{x}^t, \mathbf{o}_q)$ is the predicted class distribution at spatial location $\mathbf{o}_q$. The rectification values $\mathbf{r}_k^{\mathbf{o}_q}$ can thus be queried at any pixel coordinate, by the continuous implicit neural representations $f_{\theta'}$.

3.4. IR^{2}F-RMM Rectified Self-Training

Our proposed continuous RMM based on IR^{2}F can be used as a *plug-in* strategy, to promote and rectify the pseudo-labels used for self-training in UDA. In this section, we introduce how our continuous RMM can be plugged into two popular UDA frameworks.

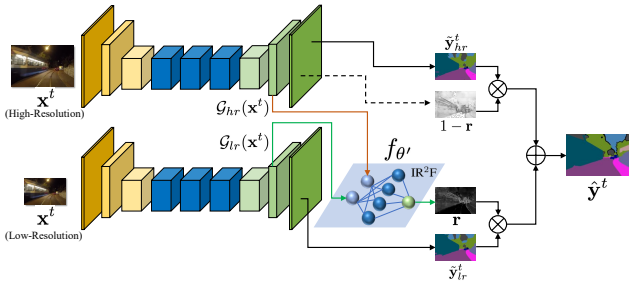


Figure 3. **Plugging continuous RMM into HRDA.**

HRDA. HRDA [24] is a multi-resolution inputs framework for UDA semantic segmentation, fusing the predictions of low-/high-resolution (LR/HR) inputs to capture both the long-range context from LR and the detailed knowledge from HR. Our continuous RMM module can be plugged into the HRDA framework by considering the two resolution branches as two mixture members, as shown in Fig. 3. Rectified pseudo-labels $\hat{\mathbf{y}}^t$ can then be formally written as,

$$\begin{aligned}
&\mathbf{r}^{\mathbf{o}_q} = f_{\theta'}(\text{Concat}(\mathcal{G}_{lr}(\mathbf{x}^t), \mathcal{G}_{hr}(\mathbf{x}^t))^*, \psi(\mathbf{o}_q - \mathbf{o}^*), \mathbf{o}_q - \mathbf{o}^*),\\
&\tilde{\mathbf{y}}^t_{lr} = \phi_1(\hat{\mathbf{y}}^t|\theta_k, \mathbf{x}^t), \quad \tilde{\mathbf{y}}^t_{hr} = \phi_2(\hat{\mathbf{y}}^t|\theta_k, \mathbf{x}^t),\\
&\tilde{\mathbf{y}}^t = \mathbf{r}\tilde{\mathbf{y}}^t_{lr} + (1-\mathbf{r})\tilde{\mathbf{y}}^t_{hr},\\
&\hat{\mathbf{y}}^{t(i,j,c)} = [c = \arg\max \tilde{\mathbf{y}}^{t(i,j)}],
\end{aligned} \tag{7}$$

where $\tilde{\mathbf{y}}^t_{lr}, \tilde{\mathbf{y}}^t_{hr}$ are soft pseudo-labels predicted by low-/high-resolutions branches, resp. $\mathcal{G}_{lr}(\mathbf{x}^t), \mathcal{G}_{hr}(\mathbf{x}^t)$ are feature maps from low-/high-resolutions branches, resp. Concat is realized by firstly up-sampling with bi-linear interpolation, and then pixel-wise concatenation. (i, j, c) are the (row, column, class) index, and $[\cdot]$ is the Iverson bracket.

MRNet. MRNet [79] is a rectification-aware UDA framework, where there are primary and auxiliary classifiers. In MRNet, the variances between the primary and auxiliary classifiers are used as the rectification values. Our continuous RMM can be used to replace this rule and instead learn the rectification. By inserting the continuous RMM into MRNet, the pseudo-labels $\hat{\mathbf{y}}^t$ can be written as,

$$\begin{aligned}
&\mathbf{r}^{\mathbf{o}_q} = f_{\theta'}(\text{Concat}(\mathcal{G}_{pr}(\mathbf{x}^t), \mathcal{G}_{aux}(\mathbf{x}^t))^*, \psi(\mathbf{o}_q - \mathbf{o}^*), \mathbf{o}_q - \mathbf{o}^*),\\
&\tilde{\mathbf{y}}^t_{pr} = \phi_1(\hat{\mathbf{y}}^t|\theta_k, \mathbf{x}^t), \quad \tilde{\mathbf{y}}^t_{aux} = \phi_2(\hat{\mathbf{y}}^t|\theta_k, \mathbf{x}^t),\\
&\tilde{\mathbf{y}}^t = \mathbf{r}\tilde{\mathbf{y}}^t_{pr} + (1-\mathbf{r})\tilde{\mathbf{y}}^t_{aux},\\
&\hat{\mathbf{y}}^{t(i,j,c)} = [c = \arg\max \tilde{\mathbf{y}}^{t(i,j)}],
\end{aligned} \tag{8}$$

where $\tilde{\mathbf{y}}^t_{pr}, \tilde{\mathbf{y}}^t_{aux}$ are the soft pseudo-labels from the primary and auxiliary classifiers, resp. $\mathcal{G}_{pr}(\mathbf{x}^t), \mathcal{G}_{aux}(\mathbf{x}^t)$ are feature maps from primary and auxiliary classifiers, resp.

Rectified Pseudo-Labels based Self-Training Loss. With pseudo-labels $\hat{\mathbf{y}}^t$ rectified by our continuous RMM, the semantic segmentation network $\mathcal{F}_\theta$ and our implicit neural representations $f_{\theta'}$ are trained jointly in the end-to-end manner, through the standard cross-entropy loss written as,

$$\mathcal{L} = CE(\mathcal{F}_\theta(\mathbf{x}^t), \hat{\mathbf{y}}^t). \tag{9}$$

4. Experiments

In this section, we demonstrate the effectiveness of our continuous RMM for UDA semantic segmentation on different benchmarks, synthetic-to-real and day-to-night. We compare our continuous RMM to other heuristics-based and/or discrete rectification modeling methods, to show the benefits of our learnable and continuous rectification function modeling based on RMM and IR^{2}F.

4.1. Experimental Setup

Datasets. We use the conventional notation A→B do describe the domain adaptation task, where A is the labeled source domain and B is the unlabeled target domain. We consider four different tasks in two categories. *Syntheic-to-Real*: There are two settings, GTA [49] → Cityscapes [12] and SYNTHIA [52] → Cityscapes [12]. *Day-to-Night*: There are also two tasks, Cityscapes [12] → Dark Zurich [54] and Cityscapes [12] → ACDC-Night [55]. Details of different datasets are put in the supplementary.

Implementation Details. *Framework and Backbone*: Our default framework is based on HRDA [24] with the MiT-B5 [70] backbone. In addition, the method is also evaluated with other backbones such as MRNet [79] (in Table 4), and ResNet-101 [20] (in Table 3). For all experiments, we simply insert our IR^{2}F based continuous RMM into the decoder without modifying the backbone architecture. *Implicit Neural Representations*: f'_θ in IR^{2}F is implemented with 4-layer MLP, with ReLU activation and hidden dimension as 256. *Training Details*: By default, we follow the training details of HRDA. In Table 4, we follow the training details of MRNet. The framework is implemented with PyTorch [46], and all the experiments are conducted on a TITAN RTX GPU.

4.2. Experimental Results

Comparison with SOTA UDA Methods. In Table 1 and Table 2, we compare our proposed IR^{2}F-based continuous RMM with other existing UDA semantic segmentation methods, under the synthetic-to-real and day-to-night benchmarks, respectively. As observed in Table 1, our IR^{2}F-based continuous RMM outperforms other SOTA methods for UDA semantic segmentation on the synthetic-to-real benchmark, especially by 1.9% mIoU under SYNTHIA → Cityscapes setting. As shown in Table 2, on the challenging day-to-night benchmark with a larger domain gap, our IR^{2}F-based continuous RMM shows a stronger performance improvement over existing SOTA methods for UDA nighttime segmentation, by 3.0% and 3.4% mIoU over the previous state-of-the-art under the Cityscapes → Dark Zurich and Cityscapes → ACDC-Night settings, respectively. Note that, the existing SOTA methods for UDA nighttime segmentation always require the day images in the target domain as the reference for adaptation (see Table 2). Instead, our IR^{2}F-based continuous RMM method

Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	MC	Bike	mIoU
GTA → Cityscapes																				
CBST [82]	91.8	53.5	80.5	32.7	21.0	34.0	28.9	20.4	83.9	34.2	80.9	53.1	24.0	82.7	30.3	35.9	16.0	25.9	42.8	45.9
MRNet [79]	90.4	31.2	85.1	36.9	25.6	37.5	48.8	48.5	85.3	34.8	81.1	64.4	36.8	86.3	34.9	52.2	1.7	29.0	44.6	50.3
DACS [60]	89.9	39.7	87.9	30.7	39.5	38.5	46.4	52.8	88.0	44.0	88.8	67.2	35.8	84.5	45.7	50.2	0.0	27.3	34.0	52.1
TACS [17]	93.0	55.9	87.9	38.2	38.8	40.4	42.1	54.5	87.5	46.7	87.8	66.3	33.7	90.2	47.5	54.2	0.0	41.2	53.3	55.8
CorDA [65]	94.7	63.1	87.6	30.7	40.6	40.2	47.8	51.6	87.6	47.0	89.7	66.7	35.9	90.2	48.9	57.5	0.0	39.8	56.0	56.6
BAPA [36]	94.4	61.0	88.0	26.8	39.9	38.3	46.1	55.3	87.8	46.1	89.4	68.8	40.0	90.2	60.4	59.0	0.0	45.1	54.2	57.4
ProDA [76]	87.8	56.0	79.7	46.3	44.8	45.6	53.5	53.5	88.6	45.2	82.1	70.7	39.2	88.8	45.5	59.4	1.0	48.9	56.4	57.5
EHTD [31]	95.4	68.8	88.1	37.1	41.4	42.5	45.7	60.4	87.3	42.6	86.8	67.4	38.6	90.5	66.7	61.4	0.3	39.4	56.1	58.8
UndoUDA [37]	92.9	52.7	87.2	39.4	41.3	43.9	55.0	52.9	89.3	48.2	91.2	71.4	36.0	90.2	67.9	59.8	0.0	48.5	59.3	59.3
CPSL [32]	92.3	59.9	84.9	45.7	29.7	52.8	61.5	59.5	87.9	41.5	85.0	73.0	35.5	90.4	48.7	73.9	26.3	53.8	53.9	60.8
DDB [8]	95.3	67.4	89.3	44.4	45.7	38.7	54.7	55.7	88.1	40.7	90.7	70.7	43.1	92.2	60.8	67.6	34.2	48.7	63.7	62.7
DAFormer [23]	95.7	70.2	89.4	53.5	48.1	49.6	55.8	59.4	89.9	47.9	92.5	72.2	44.7	92.3	74.5	78.2	65.1	55.9	61.8	68.3
HRDA [24]	96.4	74.4	**91.0**	**61.6**	51.5	**57.1**	**63.9**	69.3	91.3	48.4	**94.2**	**79.0**	**52.9**	93.9	84.1	85.7	75.9	**63.9**	67.5	73.8
IR^{2}F-RMM (Ours)	**97.5**	**80.0**	**91.0**	60.0	**53.3**	56.2	**63.9**	**72.4**	**91.7**	**51.0**	**94.2**	**79.0**	51.1	**94.3**	**84.7**	**86.7**	**75.9**	62.6	**67.8**	**74.4**
SYNTHIA → Cityscapes																				
CBST [82]	68.0	29.9	76.3	10.8	1.4	33.9	22.8	29.5	77.6	–	78.3	60.6	28.3	81.6	–	23.5	–	18.8	39.8	42.6
MRNet [79]	87.6	41.9	83.1	14.7	1.7	36.2	31.3	19.9	81.6	–	80.6	63.0	21.8	86.2	–	40.7	–	23.6	53.1	47.9
DACS [60]	80.6	25.1	81.9	21.5	2.9	37.2	22.7	24.0	83.7	–	90.8	67.6	38.3	82.9	–	38.9	–	28.5	47.6	48.3
TACS [17]	91.5	60.4	82.5	21.8	1.5	31.7	31.6	28.0	84.7	–	89.0	66.7	35.8	81.0	–	42.8	–	28.5	45.9	51.5
BAPA [36]	91.7	53.8	83.9	22.4	0.8	34.9	30.5	42.8	86.6	–	88.2	66.0	34.1	86.6	–	51.3	–	29.4	50.5	53.3
CorDA [65]	**93.3**	**61.6**	85.3	19.6	5.1	37.8	36.6	42.8	84.9	–	90.4	69.7	41.8	85.6	–	38.4	–	32.6	53.9	55.0
ProDA [76]	87.8	45.7	84.6	37.1	0.6	44.0	54.6	37.0	88.1	–	84.4	74.2	24.3	88.2	–	51.1	–	40.5	45.6	55.5
UndoUDA [37]	82.5	37.2	81.1	23.8	0.0	45.7	57.2	47.6	87.7	–	85.8	74.1	28.6	88.4	–	66.0	–	47.0	55.3	56.7
EHTD [31]	93.0	69.8	84.0	36.6	9.1	39.7	42.2	43.8	**88.2**	–	88.1	68.3	29.0	85.5	–	54.1	–	37.1	56.3	57.8
CPSL [32]	87.2	43.9	85.5	33.6	0.3	47.7	57.4	37.2	87.8	–	88.5	79.0	32.0	90.6	–	49.4	–	50.8	59.8	57.9
DAFormer [23]	84.5	40.7	88.4	41.5	6.5	50.0	55.0	54.6	86.0	–	89.8	73.2	48.2	87.2	–	53.2	–	53.9	61.7	60.9
HRDA [24]	85.2	47.7	88.8	**49.5**	4.8	57.2	**65.7**	60.9	85.3	–	92.9	79.4	52.8	89.0	–	64.7	–	63.9	64.9	65.8
IR^{2}F-RMM (Ours)	90.4	54.9	**89.4**	48.0	**7.4**	**59.0**	65.5	**63.2**	87.8	–	**94.1**	**80.5**	**55.8**	**90.0**	–	**65.9**	–	**64.5**	**66.8**	**67.7**

Table 1. **Synthetic-to-Real**: GTA → Cityscapes, SYNTHIA → Cityscapes. Best results are denoted in bold.

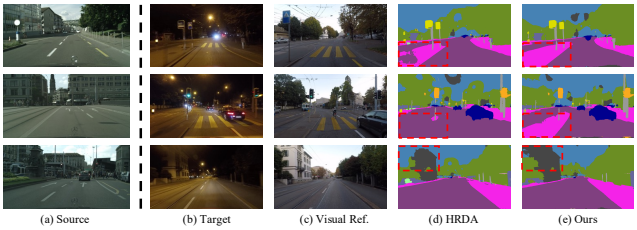


Figure 4. **Qualitative Comparisons for UDA Semantic Segmentation,** under Cityscapes → Dark Zurich. (a) shows the example of Cityscapes images. (b) includes the Dark Zurich images. (c) covers the day images corresponding to the night images in (b) for better visual references. Note that, the day images in (c) are only used for visualization references, but are not used for training and testing. (d) and (e) are the segmentation results for (b) from HRDA [24] and our method, respectively.

does not need these auxiliary data, and still outperforms the SOTA methods by a large margin. It verifies the strong generalization ability of our proposed IR^{2}F-based continuous RMM compared to the existing SOTA UDA semantic segmentation methods, under different scenarios.

Different Backbones. Besides the experimental results in Table 1 and Table 2, we show more quantitative comparisons between our method and the existing SOTA UDA method HRDA in Table 3, with the ResNet101 [20] backbone, to further verify the advantage of modeling rectification function in a continuous manner. As reported in Table 3, by simply plugging our proposed learnable continuous rectification model, our method outperforms HRDA in the GTA, SYNTHIA → Cityscapes benchmarks. Moreover, as the reference, the highest performance with ResNet-101 backbone, other than HRDA and our method, for GTA, SYNTHIA → Cityscapes are 62.7% in [8] and 57.9% in [32], resp. It means both HRDA and our IR^{2}F, *learnable rectification function modeling methods*, outperform other heuristics-based rectification function modeling methods under the ResNet-101 backbone, and supporting the validity and rationality of modeling the rectification function in the learnable manner as done by our RMM in Sec. 3.2.

Insertion of IR^{2}F-based Continuous RMM into MRNet. Our proposed IR^{2}F-based continuous RMM is in principle a plug-in module, which can be inserted into different UDA frameworks. In order to prove its compatibility with other UDA frameworks, we insert our IR^{2}F-based continuous RMM into MRNet [79]. In MRNet, pseudo-labels are originally rectified by the uncertainty measurement, which is formulated as prediction variances between the primary and auxiliary classifiers. The inputs into the primary and auxiliary classifiers are different-level features. In Table 4, it is shown that our IR^{2}F-RMM improves MRNet by 2.0% and 1.8% under GTA, SYNTHIA → Cityscapes, resp.

Ablation Study. In order to prove the effectiveness of different components in our proposed IR^{2}F-based continuous RMM, we conduct a set of ablation studies under the synthetic-to-real benchmarks. In Table 5, we ablate different ways of estimating rectification values $\mathbf{r}_k$ in Eq. (1),

Method	Road	SW	Build	Wall	Fence	Pole	TL	TS	Veg	Terrain	Sky	Person	Rider	Car	Truck	Bus	Train	MC	Bike	mIoU
Cityscapes → Dark Zurich																				
ADVENT [63]	85.8	37.9	55.5	27.7	14.5	23.1	14.0	21.1	32.1	8.7	2.0	39.9	16.6	64.0	13.8	0.0	58.8	28.5	20.7	29.7
AdaptSeg [61]	86.1	44.2	55.1	22.2	4.8	21.1	5.6	16.7	37.2	8.4	1.2	35.9	26.7	68.2	45.1	0.0	50.1	33.9	15.6	30.4
BDL [33]	85.3	41.1	61.9	32.7	17.4	20.6	11.4	21.3	29.4	8.9	1.1	37.4	22.1	63.2	28.2	0.0	47.7	39.4	15.7	30.8
DMAda [13]*	75.5	29.1	48.6	21.3	14.3	34.3	36.8	29.9	49.4	13.8	0.4	43.3	50.2	69.4	18.4	0.0	27.6	34.9	11.9	32.1
DACS [60]	83.1	49.1	67.4	33.2	16.6	42.9	20.7	35.6	31.7	5.1	6.5	41.7	18.2	68.8	76.4	0.0	61.6	27.7	10.7	36.7
GCMA [53]*	81.7	46.9	58.8	22.0	20.0	41.2	40.5	41.6	64.8	31.0	32.1	53.5	47.5	75.5	39.2	0.0	49.6	30.7	21.0	42.0
MGCDA [54]*	80.3	49.3	66.2	7.8	11.0	41.4	38.9	39.0	64.1	18.0	55.8	52.1	53.5	74.7	66.0	0.0	37.5	29.1	22.7	42.5
CDAda [71]*	90.5	60.6	67.9	37.0	19.3	42.9	36.4	35.3	66.9	24.4	79.8	45.4	42.9	70.8	51.7	0.0	29.7	27.7	26.2	45.0
DANNet [68]*	90.4	60.1	71.0	33.6	22.9	30.6	34.3	33.7	**70.5**	31.8	80.2	45.7	41.6	67.4	16.8	0.0	73.0	31.6	22.9	45.2
GLASS [30]*	91.6	63.1	71.2	34.7	**26.7**	41.4	39.7	38.4	68.6	34.8	**83.7**	41.3	40.8	69.6	21.5	0.0	63.5	32.1	19.4	46.4
DANIA [69]*	91.5	62.7	**73.9**	39.9	25.7	36.5	35.7	36.2	71.4	35.3	82.2	48.0	44.9	73.7	11.3	0.1	64.3	36.7	22.7	47.0
CCDistill [16]*	89.6	58.1	70.6	36.6	22.5	33.0	27.0	30.5	68.3	33.0	80.9	42.3	40.1	69.4	58.1	0.1	72.6	47.7	21.3	47.5
DAFormer [23]	93.5	65.5	73.3	39.4	19.2	53.3	44.1	44.0	59.5	34.5	66.6	53.4	52.7	82.1	52.7	9.5	89.3	50.5	38.5	53.8
HRDA [24]	90.4	56.3	72.0	39.5	19.5	57.8	**52.7**	43.1	59.3	29.1	70.5	60.0	**58.6**	84.0	**75.5**	11.2	90.5	51.6	40.9	55.9
IR^{2}F-RMM (Ours)	**94.7**	**75.1**	73.2	**44.4**	25.7	**60.6**	39.0	**47.4**	70.2	**41.6**	77.3	**62.4**	55.5	**86.4**	55.5	**20.0**	**92.0**	**55.3**	**42.8**	**58.9**
Cityscapes → ACDC-Night																				
DMAda [13]*	74.7	29.5	49.4	17.1	12.6	31.0	38.2	30.0	48.0	22.8	0.2	47.0	25.4	63.8	12.8	46.1	23.1	24.7	24.6	32.7
MGCDA [54]*	74.5	52.5	69.4	7.7	10.8	38.4	40.2	43.3	61.5	36.3	37.6	55.3	25.6	71.2	10.9	46.4	32.6	27.3	33.8	40.8
GCMA [53]*	78.6	45.9	58.5	17.7	18.6	37.5	43.6	43.5	58.7	39.2	22.5	57.9	29.9	72.1	21.5	56.3	41.8	35.7	35.4	42.9
DANNet [68]*	90.7	61.2	75.6	35.9	28.8	26.6	31.4	30.6	**70.8**	39.4	78.7	49.9	28.8	65.9	24.7	44.1	61.1	25.9	34.5	47.6
DANIA [69]*	91.0	60.9	**77.7**	40.3	**30.7**	34.3	37.9	34.5	70.0	37.2	79.6	45.7	32.6	66.4	11.1	37.0	60.7	32.6	37.9	48.3
GALSS [30]*	91.8	**65.0**	76.4	38.1	30.0	35.8	38.5	37.6	69.2	41.4	**79.8**	45.8	31.2	69.6	38.0	59.9	45.7	24.9	37.2	50.3
HRDA [24]	87.3	46.2	76.0	35.7	17.5	**52.0**	**50.3**	**53.6**	53.1	44.0	41.7	**64.8**	**40.9**	76.3	49.1	64.8	83.1	**36.0**	**51.5**	53.9
IR^{2}F-RMM (Ours)	**92.8**	64.8	74.5	**42.4**	15.0	51.7	36.7	52.4	66.6	**46.7**	62.7	64.1	36.3	**80.3**	**59.8**	**72.1**	**87.7**	32.0	50.5	**57.3**

Table 2. **Day-to-Night**: Cityscapes → Dark Zurich, Cityscapes → ACDC-Night. * indicates auxiliary daytime/ twilight images corresponding to night images on the target domain are needed for training. But our IR^{2}F-RMM does not need. Best results are denoted in bold.

Method	GTA → Cityscapes	SYNTHIA → Cityscapes
HRDA-ResNet	64.6	60.0
IR^{2}F-ResNet (Ours)	**65.4**	**61.4**

Table 3. **Comparisons to HRDA,** with ResNet-101 backbone. As the reference, the highest performance with ResNet-101 backbone, other than HRDA and our method, for GTA, SYNTHIA → Cityscapes are 62.7% in [8] and 57.9% in [32], respectively.

Method	GTA → Cityscapes	SYNTHIA → Cityscapes
MRNet	50.3	47.9
IR^{2}F-RMM (Ours)	**52.3**	**49.7**

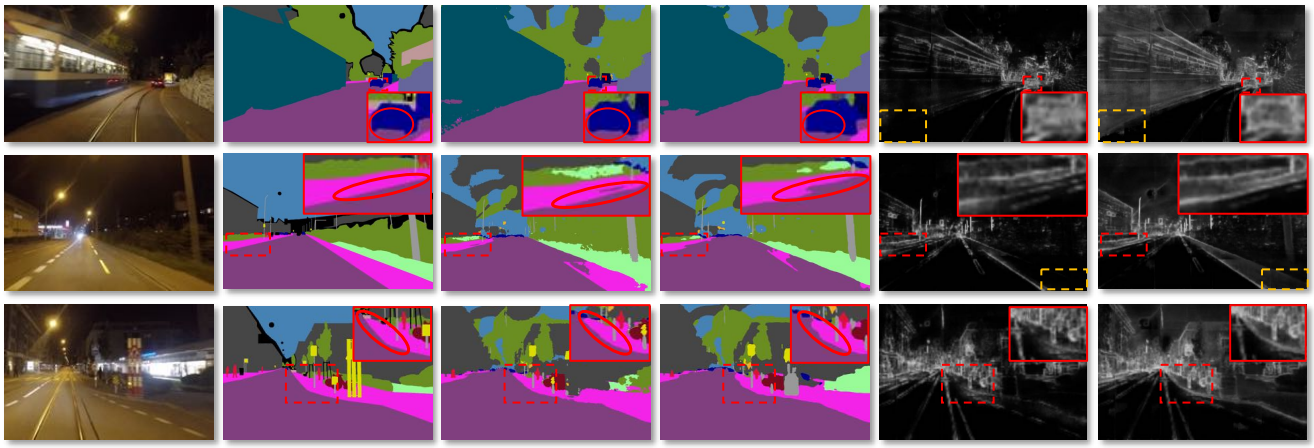
Table 4. **Combination with MRNet.** Our IR^{2}F-based continuous RMM is inserted into MRNet, to replace the original uncertainty based pseudo-labels rectification adopted by MRNet.

Method	GTA → Cityscapes	SYNTHIA → Cityscapes
HRDA	73.8	65.8
w/o. IR^{2}F *w.* AVE	71.0	61.9
w/o. IR^{2}F *w.* Conv	72.9	65.6
w/o. IR^{2}F *w.* IRE	73.3	66.3
IFA [25]	73.1	65.5
Ours	**74.4**	**67.7**

Table 5. **Ablation Study.** "AVE" means the average ensemble in Eq. (7). "Conv" means to replace the MLP structure of IR^{2}F with the convolutional neural networks. "IRE" means ensemble of the last-layer outputs instead of features from different mixture members with implicit neural representations, *i.e.* $\mathcal{G}_{lr}(\mathbf{x}^t) = \tilde{\mathbf{y}}^t_{lr}, \mathcal{G}_{hr}(\mathbf{x}^t) = \tilde{\mathbf{y}}^t_{hr}$ in Eq.(7). "IFA" leverages the INR-based semantic segmentation decoder head, as done in [25].

under the HRDA [24] framework. In our proposed IR^{2}F, $\mathbf{r}_k$ is learned by the INR from the features of different mixture members. Other ways to estimate $\mathbf{r}_k$ can be, 1) *AVE*: setting $\mathbf{r}_k = 1/K$, *i.e.* average ensemble; 2) *Conv*: replacing IR^{2}F with 5 convolutional blocks without using the coordinate information; 3) *IRE*: taking the last-layer output (before softmax) instead of features from each mixture member as input to the IR^{2}F. Besides, in Table 5, we compare to another alternative, "IFA", which leverages the INR-based segmentation decoder head as done in [25]. It is shown that "IFA" does not bring obvious benefits to UDA compared to HRDA [24], 73.1%, 65.5% *vs*. 73.8%, 65.8%, verifying the necessity and importance of rectifying incorrect pseudo-labels for UDA compared to a stronger decoder.

Comparisons to Heuristics-based/ Discrete Rectification Function Modeling. In order to showcase the advantage of our learnable and continuous rectification function modeling over the heuristics-based/ discrete one, we employ different heuristics-based/ discrete rectification function modeling methods under the HRDA framework as the baselines. As shown in Table 6, we compare our continuous IR^{2}F to different rectification function modeling methods, including the *heuristics-based* method, 1) prediction variances [79] of $\tilde{\mathbf{y}}^t_{lr}$ and $\tilde{\mathbf{y}}^t_{hr}$ in Eq. (7), 2) Monte Carlo Dropout (MC-Dropout) [15], activating dropout function during inference to obtain different predictions for ensemble, and the *discrete* method, 3) an additional convolutional decoder is exploited to estimate rectification value as done in HRDA [24]. It is shown that our learnable and continuous rectification function modeling method, IR^{2}F-RMM, outperforms all



(a) RGB (b) GT (c) HR (d) LR (e) Discrete (f) Continuous (Ours)

Figure 5. **Qualitative Comparisons between Discrete and Continuous Rectification Function Modeling.** (a) and (b) are the RGB inputs and corresponding ground truth semantic segmentation maps, respectively. (c) and (d) are the outputs of the HR, LR branches (see Sec. 3.4), *i.e.* $\arg\max \tilde{\mathbf{y}}^t_{hr}, \arg\max \tilde{\mathbf{y}}^t_{lr}$ in Eq. (7), respectively. (e) and (f) are the estimated rectification values, *i.e.* $\mathbf{r}$ in Eq. (7), by discrete modeling method (*i.e.* additional decoder in HRDA [24]) and our continuous modeling method, IR^{2}F. In (e) and (f), the brighter the part is, the ensemble result in RMM relies more on HR branch result in (c). It is shown that our continuous modeling method can rectify some areas, which are ignored by the discrete modeling method (see orange dashed boxes), and other areas, where the the discrete modeling method is affected by the blurring effect and does not perform well (see red dashed boxes). The red dashed boxes are enlarged to red solid boxes for better visualization, especially the red circle parts. *Best viewed with zooming.*

Method	GTA → Cityscapes	SYNTHIA → Cityscapes
AVE + Variance	73.7	65.1
AVE + MC-Dropout	71.8	63.9
Additional Conv Decoder	73.8	65.8
IR^{2}F	**74.4**	**67.7**

Table 6. **Comparisons to Heuristics-based/ Discrete Rectification Modeling.** "AVE" represents the average the ensemble. Heuristics-based modeling methods include, (1) "Variance": prediction variances are used to rectify pseudo-labels as done in [79], (2) "MC-Dropout": dropout is enabled during inference to get different predictions for ensemble [15], and discrete modeling method has (3) "Additional Decoder": an additional convolutional decoder is utilized to decode the rectification value as done in [24].

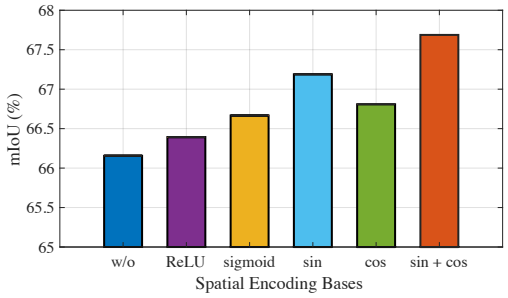


Figure 6. **Spatial Encoding Study.** Different spatial encoding bases are compared, and the combination of sin and cos reaches the highest performance.

heuristics-based and discrete modeling methods by a large margin. Furthermore, we provide the qualitative comparisons for the discrete and continuous rectification function modeling in Fig. 5. Benefiting from continuous modeling, the rectification values of IR^{2}F are more accurate and insensitive to the blurring effect of down-/up-sampling operations in DNNs (see Sec. 3.3), especially at mask boundaries.

Spatial Encoding Study. As analyzed in Sec. 3.3, the implicit neural representations are insensitive to the high-frequency signal in the image, *e.g.* boundaries in the image. To overcome the shortcomings, we introduce the spatial encoding in Eq. (3), where the combination of sin and cos is adopted as encoding basis. To study the effectiveness of spatial encoding with both sin and cos, we compare to different encoding bases in Fig. 6, including without spatial encoding, leakyReLU, sigmoid, pure sin and pure cos. It is observed that all the spatial encoding bases outperform the one without spatial encoding, proving the effectiveness of the spatial encoding. Among different spatial encoding bases, the combination of sin and cos reaches the highest performance, taken as the spatial encoding basis in IR^{2}F.

5. Conclusion

In this work, we presented continuous rectification-aware mixture model (RMM) based on implicit neural representations, which rectifies pseudo-labels for UDA in a learnable, continuous and end-to-end manner. As a principled and plug-in module, continuous RMM can be combined with different UDA frameworks, boosting the quality of pseudo-labels. Overall, our proposed continuous RMM achieves superior results compared to state-of-the-art, on synthetic-to-real and day-to-night UDA benchmarks.

References

[1] Waqar Ahmed, Pietro Morerio, and Vittorio Murino. Cleaning noisy labels by negative ensemble learning for source-free unsupervised domain adaptation. In *WACV*, 2022. 3

[2] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *CVPR*, 2020. 3

[3] Jonathan T Barron, Ben Mildenhall, Matthew Tancik, Peter Hedman, Ricardo Martin-Brualla, and Pratul P Srinivasan. Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In *ICCV*, 2021. 3

[4] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *NeurIPS*, 2019. 3

[5] Christopher M Bishop. Mixture density networks. 1994. 3

[6] Chen-Hao Chao, Bo-Wun Cheng, and Chun-Yi Lee. Rethinking ensemble-distillation for semantic segmentation based unsupervised domain adaption. In *CVPR Workshops*, 2021. 3

[7] Hongruixuan Chen, Chen Wu, Yonghao Xu, and Bo Du. Unsupervised domain adaptation for semantic segmentation via low-level edge information transfer. *arXiv preprint arXiv:2109.08912*, 2021. 4

[8] Lin Chen, Zhixiang Wei, Xin Jin, Huaian Chen, Miao Zheng, Kai Chen, and Yi Jin. Deliberated domain bridging for domain adaptive semantic segmentation. In *NeurIPS*, 2022. 6, 7

[9] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2017. 1

[10] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In *CVPR*, 2021. 3, 4

[11] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *CVPR*, 2019. 3

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 5

[13] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *ITSC*, 2018. 7

[14] Aysegul Dundar, Ming-Yu Liu, Ting-Chun Wang, John Zedlewski, and Jan Kautz. Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation. *arXiv preprint arXiv:1807.09384*, 2018. 2

[15] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, 2016. 7, 8

[16] Huan Gao, Jichang Guo, Guoli Wang, and Qian Zhang. Cross-domain correlation distillation for unsupervised domain adaptation in nighttime semantic segmentation. In *CVPR*, 2022. 7

[17] Rui Gong, Martin Danelljan, Dengxin Dai, Danda Pani Paudel, Ajad Chhatkuli, Fisher Yu, and Luc Van Gool. Tacs: Taxonomy adaptive cross-domain semantic segmentation. In *ECCV*, 2022. 6

[18] Rui Gong, Wen Li, Yuhua Chen, and Luc Van Gool. Dlow: Domain flow for adaptation and generalization. In *CVPR*, 2019. 1

[19] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. *arXiv preprint arXiv:2002.10099*, 2020. 3

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 5, 6

[21] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. In *ICML*, 2018. 1

[22] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *arXiv preprint arXiv:1612.02649*, 2016. 2

[23] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *CVPR*, 2022. 1, 6, 7

[24] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *ECCV*, 2022. 1, 2, 3, 4, 5, 6, 7, 8

[25] Hanzhe Hu, Yinbo Chen, Jiarui Xu, Shubhankar Borse, Hong Cai, Fatih Porikli, and Xiaolong Wang. Learning implicit feature alignment function for semantic segmentation. In *ECCV*, 2022. 2, 3, 4, 7

[26] Zijian Hu, Zhengyu Yang, Xuefeng Hu, and Ram Nevatia. Simple: similar pseudo label exploitation for semi-supervised classification. In *CVPR*, 2021. 3

[27] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *CVPR*, 2020. 3

[28] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, Bill Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. In *NeurIPS*, 2018. 3

[29] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017. 3

[30] Hongjae Lee, Changwoo Han, and Seung-Won Jung. Gps-glass: Learning nighttime semantic segmentation using daytime video and gps data. *arXiv preprint arXiv:2207.13297*, 2022. 7

[31] Junjie Li, Zilei Wang, Yuan Gao, and Xiaoming Hu. Exploring high-quality target domain information for unsupervised domain adaptive semantic segmentation. In *ACM MM*, 2022. 6

[32] Ruihuang Li, Shuai Li, Chenhang He, Yabin Zhang, Xu Jia, and Lei Zhang. Class-balanced pixel-level self-labeling for domain adaptive semantic segmentation. In *CVPR*, 2022. 6, 7

[33] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In *CVPR*, 2019. 1, 7

[34] Christopher Liao, Theodoros Tsiligkaridis, and Brian Kulis. Pick up the pace: Fast and simple domain adaptation via ensemble pseudo-labeling. *arXiv preprint arXiv:2205.13508*, 2022. 3

[35] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 1

[36] Yahao Liu, Jinhong Deng, Xinchen Gao, Wen Li, and Lixin Duan. Bapa-net: Boundary adaptation and prototype alignment for cross-domain semantic segmentation. In *ICCV*, 2021. 4, 6

[37] Yahao Liu, Jinhong Deng, Jiale Tao, Tong Chu, Lixin Duan, and Wen Li. Undoing the damage of label shift for cross-domain semantic segmentation. In *CVPR*, 2022. 6

[38] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1

[39] Yawei Luo, Liang Zheng, Tao Guan, Junqing Yu, and Yi Yang. Taking a closer look at domain shift: Category-level adversaries for semantics consistent domain adaptation. In *CVPR*, 2019. 2, 3

[40] Dmitrii Marin, Zijian He, Peter Vajda, Priyam Chatterjee, Sam Tsai, Fei Yang, and Yuri Boykov. Efficient segmentation: Learning downsampling near semantic boundaries. In *ICCV*, 2019. 2

[41] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 3

[42] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 3, 4

[43] Zak Murez, Soheil Kolouri, David Kriegman, Ravi Ramamoorthi, and Kyungnam Kim. Image to image translation for domain adaptation. In *CVPR*, 2018. 2

[44] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *NeurIPS*, 2019. 3

[45] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 3

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 5

[47] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020. 3

[48] Fabio Pizzati, Raoul de Charette, Michela Zaccaria, and Pietro Cerri. Domain bridge for unpaired image-to-image translation and unsupervised domain adaptation. In *WACV*, 2020. 2

[49] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *ECCV*, 2016. 5

[50] Tobias Ringwald and Rainer Stiefelhagen. Ubr ^{2} s: Uncertainty-based resampling and reweighting strategy for unsupervised domain adaptation. *arXiv preprint arXiv:2110.11739*, 2021. 3

[51] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 1

[52] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *CVPR*, 2016. 5

[53] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *ICCV*, 2019. 7

[54] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Map-guided curriculum domain adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. *TPAMI*, 2020. 5, 7

[55] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *ICCV*, 2021. 5

[56] Tiancheng Shen, Yuechen Zhang, Lu Qi, Jason Kuen, Xingyu Xie, Jianlong Wu, Zhe Lin, and Jiaya Jia. High quality segmentation for ultra high-resolution images. In *CVPR*, 2022. 2

[57] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *NeurIPS*, 2020. 3, 4

[58] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. In *NeurIPS*, 2019. 3, 4

[59] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 1

[60] Wilhelm Tranheden, Viktor Olsson, Juliano Pinto, and Lennart Svensson. Dacs: Domain adaptation via cross-domain mixed sampling. In *WACV*, 2021. 1, 6, 7

[61] Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Kihyuk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation. In *CVPR*, 2018. 1, 2, 7

[62] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Chandraker. Domain adaptation for structured output via discriminative patch representations. In *ICCV*, 2019. 1

[63] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Advent: Adversarial entropy minimization for domain adaptation in semantic segmentation. In *CVPR*, 2019. 1, 7

[64] Tuan-Hung Vu, Himalaya Jain, Maxime Bucher, Matthieu Cord, and Patrick Pérez. Dada: Depth-aware domain adaptation in semantic segmentation. In *ICCV*, 2019. 2

[65] Qin Wang, Dengxin Dai, Lukas Hoyer, Luc Van Gool, and Olga Fink. Domain adaptive semantic segmentation with self-supervised depth estimation. In *ICCV*, 2021. 6

[66] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *CVPR*, 2022. 3

[67] Yuxi Wang, Junran Peng, and ZhaoXiang Zhang. Uncertainty-aware pseudo label refinery for domain adaptive semantic segmentation. In *ICCV*, 2021. 1, 2, 3

[68] Xinyi Wu, Zhenyao Wu, Hao Guo, Lili Ju, and Song Wang. Dannet: A one-stage domain adaptation network for unsupervised nighttime semantic segmentation. In *CVPR*, 2021. 7

[69] Xinyi Wu, Zhenyao Wu, Lili Ju, and Song Wang. A one-stage domain adaptation network with image alignment for unsupervised nighttime semantic segmentation. *TPAMI*, 2021. 7

[70] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *NeurIPS*, 2021. 1, 5

[71] Qi Xu, Yinan Ma, Jing Wu, Chengnian Long, and Xiaolin Huang. Cdada: A curriculum domain adaptation for nighttime semantic segmentation. In *ICCV*, 2021. 7

[72] Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Ultrasr: Spatial encoding is a missing key for implicit image function-based arbitrary-scale super-resolution. *arXiv preprint arXiv:2103.12716*, 2021. 3, 4

[73] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *CVPR*, 2020. 1

[74] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 3

[75] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022. 3

[76] Pan Zhang, Bo Zhang, Ting Zhang, Dong Chen, Yong Wang, and Fang Wen. Prototypical pseudo label denoising and target structure learning for domain adaptive semantic segmentation. In *CVPR*, 2021. 1, 2, 3, 6

[77] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. In *ICCV*, 2017. 1

[78] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, 2017. 1

[79] Zhedong Zheng and Yi Yang. Rectifying pseudo label learning via uncertainty estimation for domain adaptive semantic segmentation. *IJCV*, 129(4):1106–1120, 2021. 1, 2, 3, 5, 6, 7, 8

[80] Shuaifeng Zhi, Tristan Laidlow, Stefan Leutenegger, and Andrew J Davison. In-place scene labelling and understanding with implicit scene representation. In *ICCV*, 2021. 3

[81] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain adaptive ensemble learning. *TIP*, 30:8008–8018, 2021. 3

[82] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *ECCV*, 2018. 1, 2, 6

[83] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *ICCV*, 2019. 1, 2