CD-TTA: Compound Domain Test-time Adaptation for Semantic Segmentation

Junha Song  Kwanyong Park  InKyu Shin  Sanghyun Woo  InSo Kweon
KAIST

Abstract

Test-time adaptation (TTA) has attracted significant attention due to its practical properties which enable the adaptation of a pre-trained model to a new domain with only target dataset during the inference stage. Prior works on TTA assume that the target dataset comes from the same distribution and thus constitutes a single homogeneous domain. In practice, however, the target domain can contain multiple homogeneous domains which are sufficiently distinctive from each other and those multiple domains might occur cyclically. Our preliminary investigation shows that domain-specific TTA outperforms vanilla TTA treating compound domain (CD) as a single one. However, domain labels are not available for CD, which makes domain-specific TTA not practicable. To this end, we propose an online clustering algorithm for finding pseudo-domain labels to obtain similar benefits as domain-specific configuration and accumulating knowledge of cyclic domains effectively. Moreover, we observe that there is a significant discrepancy in terms of prediction quality among samples, especially in the CD context. This further motivates us to boost its performance with gradient denoising by considering the image-wise similarity with the source distribution. Overall, the key contribution of our work lies in proposing a highly significant new task compound domain test-time adaptation (CD-TTA) on semantic segmentation as well as providing a strong baseline to facilitate future works to benchmark.

1. Introduction

As demonstrated in a wide range of works [13, 15, 67], deep neural networks (DNNs) achieve competitive performance when training (source domain) and testing (target domain) data come from the same distribution. However, when the test data has different characteristics from one of the training dataset, the model often suffers a severe accuracy drop due to the domain shift [25, 54].

The domain shift is common in real-world problems and should be overcome especially for safety-critical applications such as self-driving cars and medical care systems [56, 45, 38]. A common technique for mitigating this domain gap is unsupervised domain adaptation which requires access to the target dataset during the model training stage to perform joint training on the source and target domain [57, 59]. In practice, however, the target domain might change and thus can not be known beforehand. Moreover, the training data might also be lost after the training is done, which results in a practical scenario: the practitioner has and only has a model pre-trained on a certain source domain, but needs to deploy it in a domain that is different from the source one. How to reduce the domain gap during the inference stage is a non-trivial challenge, for which test-time adaptation (TTA) has come as a timely solution and attracted significant attention.

Prior works on the task of TTA [60, 28] implicitly assume that the target samples come from the same distribution, for which their performance is often evaluated by treating the target samples as a single homogeneous domain. In practice, however, the target domain might consist of multiple homogeneous domains that are sufficiently different from each other [30, 39, 46], such as different weather conditions in autonomous vehicles or races in medical care sys-
Table 1. Comparison of various settings to mitigate the domain shift. We propose to study compound domain test-time adaptation. On the table, domain labels indicate whether the information on which domain each sample belongs exists, offline means the state of having all data for adaptation in advance, and conversely, online means the state in which samples can be accessed one by one on inference.

| Setting | Source / Target          | Target domain | Target State | Domain labels | Target scenario |
|---------|--------------------------|---------------|--------------|---------------|-----------------|
| Unsupervised domain adaptation [9] | labeled / Target | single | offline | ✓ | - |
| Compound domain adaptation [30] | labeled / unlabeled | multiple | offline | ✓ | - |
| Source-free domain adaptation [29] | unknown / unlabeled | single | offline | ✓ | - |
| Test-time adaptation (TTA) [60] | unknown / unlabeled | single | online | ✓ | - |
| Continual test-time adaptation (CoTTA) [61] | unknown / unlabeled | multiple | online | × | evolving domains |
| Compound domain test-time adaptation (CD-TTA) | unknown / unlabeled | multiple | online | × | cyclical domains |

tems. Besides, those multiple domains are likely to occur cyclically as seen in Fig. 1. Unfortunately, existing approaches [60, 28, 63, 41] in this scenario have a problem to overfit only the last domain viewed without leveraging the knowledge learned in the past. Therefore, in this paper, we propose to study compound domain test-time adaptation (CD-TTA) and present a strong baseline that can accumulate the learned knowledge and generalize the model in seen domains during test-time adaptation.

First, we performed a preliminary investigation and found that domain-specific TTA achieves better performance since it is not influenced by the domain discrepancy between the target sub-domains. However, domain labels are unknown in the real-world compound domain, and thus such a naive domain-specific TTA is not available even though it yields superior performance. To obtain a similar benefit without the need for domain labels, we propose an online domain clustering method to predict pseudo domain labels, which are leveraged to optimize domain-specific model parameters.

Moreover, we observed that there is a large discrepancy in terms of prediction quality among different target datasets in the context of CD-TTA. For example, the night images tend to have poorer predictions than the daytime counterpart. Since the performance drop is in general interpreted as domain shift, we assume that the prediction performance is highly correlated by the image-wise similarity with the source distribution. Motivated by this finding, we attempt a distribution similarity-based gradient denoising. Specifically, we evaluate the weight to multiply by un-supervised loss using the similarity between target distribution and source distribution.

Overall, we identify a blind spot in existing works on TTA for failing to take compound and cyclical domain (CD) into account and conduct an investigation to mitigate the CD-induced concerns for more practical TTA. Our key contributions lie in proposing a significant yet unexplored task CD-TTA as well as providing a strong baseline with two technical novelties: online domain clustering and distribution similarity-based gradient denoising.

2. Related Works

Unsupervised Domain Adaptation (UDA) aims to transfer the knowledge from labeled source dataset to unlabeled target domain. Recent UDA methods can be largely categorized into Adversarial learning based DA [57, 59, 32, 37, 50] and Self-training based DA [71, 72, 52, 34]. Despite the effectiveness of traditional UDA, their experimental settings show two major practical limitations: Traditional UDA (1) inevitably requires concurrent access to both source and target data and (2) mainly focuses to adapt a single target domain.

Open-Compound Domain Adaptation Traditional UDA methods consider one target domain and cannot be well generalized to more complex scenarios for which the multiple target domains are given. To address this problem, open compound domain adaptation (OCDA) [30] investigates an adaptation of a mixture of multiple domains. Follow-up works [39, 11] for OCDA aims to learn domain-invariant representations that are robust on multiple target domains, i.e. compound domain. In addition, they expect their trained models to generalize on unseen target domains, i.e. open domain. Note that, for adaptation, they utilize labeled source dataset and multiple target datasets with multiple losses and long training.

Source Free Domain Adaptation In some major areas like autonomous driving, the source datasets may be commercial and/or private, restricting the use of the labeled source data during the adaptation process. With this practical constraint, source-free domain adaptation (SFDA) [29, 27, 24, 69] investigates a new scenario where only a well-trained source model and unlabeled target data are available. Specifically, the methods for SFDA optimize the model in an offline-manner without considering any resource limitations.

Test-time Adaptation Unlike aforementioned settings, test time adaptation (TTA) requires adapting the model in an online-manner, and thus emphasizes the efficiency of adaptation. After its practical setting is proposed [19, 55, 60], recent works improve the performance of TTA with delicate designs of self-supervised loss [41, 55, 68, 51], calibration for normalization parameters [64, 23, 14], and prototypical optimization [18]. Compared to these works which attempt to adapt to a single distribution, CoTTA [61] considers continuously evolving target distributions. This adaptation is usually made not to forget source knowledge learned from train dataset. In contrast, our methods aim not to lose the knowledge learned during TTA, i.e. we expect that our model become generalized in encountered test domains.
3. Problem statement

3.1. Background: Test-time adaptation

For the task of TTA, we are given only unlabeled test dataset $X_{T}$ which is drawn from a different distribution from the source one. The samples from the target domain can be represented as $X_{T} = \{x_{n}^{T}\}$, where $N_{T}$ is the number of target samples. The goal of TTA is to adapt the deployed model $f_{0}$ with parameters $\theta$ to $X_{T}$ in an online manner and to mitigate the performance drop caused by domain shift practically while inference and adaptation are conducted simultaneously. Several works [60, 40, 36] update a small part of the model parameters by using a carefully designed self-supervisory loss since efficiency and stability are the key concerns for the task of TTA. Specifically, a pioneering work, TENT [60], optimize only affine parameters in batch normalization layers by minimizing the entropy of model predictions: $H(\hat{y}) = -\sum_{p} p(\hat{y}) \log p(\hat{y})$, where $\hat{y} = f_{0}(x_{i})$.

**Preliminary background on BN.** Batch normalization (BN) [16] has become a de facto standard component in modern DNNs to boost convergence. As shown in Eq 1, it consists of normalization statistics and affine parameters. With $j$-th features $F$ as the BN input, normalization statistics are denoted as $\mu^{j} = \mathbb{E}[F^{j}]$ and $\sigma^{j} = \mathbb{E}[(\mu^{j} - F^{j})^{2}]$. Affine parameters are denoted as $\gamma_{j}, \beta_{j}$.

$$BN(F^{j}) = \gamma^{j}\left(\frac{F^{j} - \mu^{j}}{\sigma^{j}}\right) + \beta^{j} \tag{1}$$

Calibrated normalization parameters. Recent works for TTA [64, 14, 23] have shown that performance degradation can occur when source normalization parameters are directly substituted with target normalization statistics. The reason is known for the mismatch between target statistics and the source-trained model parameters. To solve this problem, in contrast to only using the target domain BN statistics (called as $t$-BN), they propose to mix up the source and target normalization statistics (termed as $\alpha$-BN):

$$\mu^{j} = \alpha\mu^{j}_{s} + (1 - \alpha)\mu^{j}_{t}, \tag{2}$$

$$\sigma^{j} = \alpha\sigma^{j}_{s} + (1 - \alpha)\sigma^{j}_{t}, \tag{3}$$

where $\mu_{s}, \sigma_{s}$ denote the running statistics tracked during source pre-training and $\mu_{t}, \sigma_{t}$ denote the normalization statistics calculated on features $F^{j}$ of a batch of $x_{n}^{T}$ in the target domain. We follow the same strategy for our method.

3.2. Compound domain Test-time Adaptation

Realistic target data usually consists of multiple homogeneous domains, for example, the domains of images collected from autonomous vehicles is various due to variations in sensor sensitivity, road conditions, weather, and season [6, 4]. In order to find a strong baseline for this setting, we established and verified the following conjecture.

**Conjecture 1.** We conjecture that domain-specific TTA outperforms its counterpart that does not consider different characteristics inside each domain. The rationale behind this conjecture is that each homogeneous domain has its domain-specific parameters that are likely to be the most compatible with the corresponding domain.

**Verifying conjecture 1.** To verify the above conjecture, we consider a compound target domain that has three distinctive homogeneous domains: cloudy domain, rainy domain and snowy domain in C-driving dataset [30]. Table 2 reports the results of domain-specific TTA. For differentiation, we term the domain where the TTA is optimized as $\text{adapted-domain}$ (i.e. columns in table) and the domain where the adapted model is evaluated on $\text{eval-domain}$ (i.e. rows in table). In conclusion, we observe that the performance is the highest when $\text{adapted-domain}$ is set to the same as $\text{eval-domain}$. For example, on the rainy eval-domain Table 2 (b), the mIoU after TTA with the adapted-domain set to rainy one is 28.98 which is higher than 27.23 (cloudy adapted-domain) and 27.15 (snowy adapted-domain). Alternatively, we treat the whole target domain as a compound one (i.e. compound BN). Its performance is worse than domain-specific TTA, which confirms our above conjecture as well. Our results align with findings in [1, 48, 17, 47] that utilize domain-specific BN beyond the task of TTA.

**CD-TTA.** The above findings motivate a new TTA setting for considering target domain as multiple distinctive homogeneous domains. Besides, the fact that those multiple domains are likely to occur recurrently in the real world inspires us to accumulate the learned knowledge and generalize the model in seen domains while test-time adaptation.

| (a) Oracle domain label | cloudy | rainy | snowy |
|------------------------|--------|-------|-------|
| Source only            | 32.79  | 26.96 | 27.48 |
| compound BN            | 33.63  | 27.59 | 28.12 |
| cloudy BN              | 34.61  | 27.43 | 28.20 |
| rainy BN               | 33.52  | 28.49 | 28.51 |
| snowy BN               | 33.61  | 27.20 | 29.21 |

| (b) Pseudo domain label by offline clustering [33] |
|------------------------|--------|-------|-------|
| Model                  | cloudy | rainy | snowy |
| Source only            | 32.79  | 26.96 | 27.48 |
| compound BN            | 33.63  | 27.59 | 28.12 |
| cloudy≈BN              | 34.71  | 27.23 | 28.08 |
| rainy≈BN               | 33.24  | 28.98 | 28.50 |
| snowy≈BN               | 33.41  | 27.15 | 29.52 |
Figure 2. Overview of our approach. [Step2]: we extract the domain-discriminative feature of the incoming image and predict the pseudo domain label with online domain clustering. A domain-specific BN is selected for inference and adaptation. [Step3]: we compute unsupervised loss leveraging the segmentation result and refine it with the weight for gradient denoising in order to create a reliable backward signal.

We name this task as compound domain test-time adaptation (CD-TTA). To our best knowledge, our work is the first to propose this new task as well as to address its challenge.

4. Approach

An overview of our method is shown in Fig. 2. [Step1] We first construct TTA framework by building multiple branches of batch normalization layers. [Step2] Pseudo domain label to each sample is predicted by clustering samples in an online manner. According to predicted pseudo domain labels, we choose a branch and use it to produce the segmentation results. [Step3] Finally, we attempt to calculate reliable unsupervised loss by rectifying with instance-wise gradient denoising weight and optimize the selected BN affine parameters.

4.1. Domain Label Prediction

As shown in Sec. 3.2, a straightforward approach for CD-TTA is to perform domain-specific adaptation which requires information about which domains each target sample belongs to (i.e. domain label). However, it is not always available because several TTA applications may not be able to access domain labels [66]. Thus, the first challenge to solve is the lack of ground-truth domain labels. For obtaining pseudo domain labels, [33] propose offline domain clustering method. They assign pseudo domain labels from domain-discriminative features using a standard clustering algorithm, k-means [12], where they divide whole target datasets into $K$ clusters before multi-target domain generalization. Additionally, the pseudo domain labels are used for reporting the results of domain-specific TTA as shown in Table 2 (b).

**Online domain clustering.** A practical CD-TTA, however, requires adaptation to perform in an online mode, in other words, we can not get access to the whole dataset in advance. Therefore, we propose a online clustering algorithm to acquire pseudo domain labels. First, we need to extract domain-distinguishable features, which are composed of convolutional feature statistics $s_{l(n)}$ (i.e. mean and standard deviations) obtained from $l$-th lower layer of the feature extractor $f_l$:

$$
s_{l(n)} = \text{statistics} \left\{ f_l \left( x^{(n)} \right) \right\} = \left\{ (\mu_1^n, \sigma_1^n), (\mu_2^n, \sigma_2^n), \ldots, (\mu_c^n, \sigma_c^n) \right\}^{(n)}
$$

(4)

where $c$ denotes the number of channels in the layer. Then, we prepare a stack of feature statistics from a few lower layers: $S = \{s^l\}$, which is leveraged to determine how close each sample is to. The distance between two samples, $S^{(n_1)}$ and $S^{(n_2)}$ is calculated by Bhattacharya distance function
We verify the effectiveness of above distance function as well as the domain-distinguishable features in Sec. 5.4.

Our algorithm find the index of domain-centroids \( \{c_k\}_{k=1}^{K} \) closest to the current domain feature \( S^{(n)} \). Specifically, domain-centroids are first initialized by the features obtained from the first \( K \) samples and updated by exponential moving average as Eq. \( (6) \) whenever the incoming sample is assigned a pseudo domain label sequentially.

\[
c_{d} \leftarrow \eta \cdot c_{d} + (1 - \eta) \cdot S^{(n)}
\]

where \( \eta \) denotes momentum. After assigning a target sample among \( K \) clusters, we utilize these cluster assignments as pseudo domain labels \( d \). As an additional note, our clustering algorithm is perfectly suited for a TTA setting since it only needs to store \( K \) domain-centroids (i.e. memory efficiency) and only execute \( k \) distance calculations whenever a batch of images are coming (i.e. time efficiency).

Note that the estimated domain labels facilitate domain-specific TTA that ensure that the knowledge of each domain is accumulated on only one compatible parameter. Moreover, in the cyclic domain scenario, when a previously seen domain comes again, our clustering approach immediately predicts the domain and help to use the parameters previously accumulated knowledge. Even if a new domain is encountered, our method is likely to make a more appropriate prediction using the nearest domain knowledge. It would be better, for example, if we can utilize the knowledge we learned in the night domain when our vehicle enters a tunnel.

4.2. Gradient denoising

In the task of TTA, there is often a discrepancy in terms of prediction quality among different target samples. This phenomenon tends to be much more significant in the task of CD-TTA. For example, an image in the rainy domain tends to have poorer predictions than that from the cloudy domain. Intuitively, the contribution of each sample to CD-TTA should be subject to such prediction quality. A low-quality prediction is likely to result in unreliable unsupervised signals, resulting in false adaptation and performance degradation. Thus, it is expected that CD-TTA might be improved by weighting its contribution to the model adaptation based on its image quality. However, it is non-trivial to quantify the prediction quality since the GT prediction labels are not available. To this end, we formulated the following conjecture and verified it.

Conjecture 2. We conjecture that the prediction performance is highly correlated to the image-wise similarity between the source distribution and the one of a current test sample. The rationale lies in a common interpretation of the performance drop as domain shift. In other words, if the distribution of the target image is more different from that of the source domain, the prediction performance is expected to be worse.

Verifying conjecture 2. To verify the above conjecture, we analyze Pearson correlation coefficient [8] between the mIoU (i.e. mean intersection over union [31]) score of the segmentation results and the similarity between feature statistics of a target sample and source statistics saved in batch normalization layers. As shown in Fig. 3(a), their correlation value of 0.724, suggesting the two values have a positive correlation. In order words, a higher similarity is likely to yield a more correct segmentation prediction. Alternatively, we also investigate the correlation with probability \( \mathbb{E}[p(\hat{y})] \) and entropy \( \mathbb{E}[-\sum_c p(\hat{y}) \log p(\hat{y})] \), which are shown in Fig. 3(b) and Fig. 3(c) respectively. The results demonstrate that entropy and probability can also quantify the prediction quality, but their ability is inferior considering their (absolute) correlation coefficient is smaller than that with distribution similarity.

Distribution similarity-based gradient denoising. The above findings motivate a distribution similarity-based gradient denoising method. Specifically, we consider a measure of cosine similarity between feature statistics \( s_l \) of a target sample and source running statistics \( SRS^l \) in BN layers as follows:

\[
w(x) = \mathbb{E}_{l \sim L} \left[ \text{CosineSim}(SRS^l, s_l) \right]
\]

where \( L = \{\text{res0}, \text{res1}, \text{res2}, \text{res3}, \text{res4}\} \).

\[
\min_{\gamma, \delta} \mathbb{E}_{x \sim \mathcal{X}_1} \left[ w(x)^\delta \mathcal{L}(f_\theta(x)) \right]
\]
Table 3. Experimental results. (a) GTA5 \(\rightarrow\) C-driving: We evaluate the semantic segmentation results, reporting the mIoU accuracy of 19-classes. (b) SYNTHIA \(\rightarrow\) C-driving: We show quantitative results with the mIoU accuracy of 13-classes. C-driving score denote the average value of all domains.

| (a) GTA5 \[42\] \(\rightarrow\) C-driving (30) | Model # | Un-sup. loss | Time of day | Weather | C-driving (30) |
|-----------------------------------------------|---------|-------------|-------------|---------|----------------|
| Source only                                   | -       | 29.61 11.16 26.07 22.28 | 32.79 26.96 27.48 29.08 | 25.68 |
| \(\tau\)-BN \[60\]                          | Entropy [49] | 28.81 11.41 22.82 21.01 | 30.34 24.95 25.47 26.92 | 23.97 |
| \(\alpha\)-BN \[64\]                         | 31.68 15.43 25.29 24.13 | 33.63 27.59 28.14 29.79 | 26.96 |
| Ours                                          | 32.92 15.91 27.49 25.44 | 35.25 27.74 28.97 30.65 | 28.05 |
| \(\tau\)-BN \[60\]                          | MaxSquare [3] | 28.91 11.69 23.02 21.21 | 30.55 25.24 25.66 27.15 | 24.18 |
| \(\alpha\)-BN \[64\]                         | 32.08 15.76 26.19 24.68 | 34.21 27.90 28.87 30.33 | 27.50 |
| Ours                                          | 33.02 16.17 27.51 25.57 | 35.50 27.92 29.15 30.85 | 28.21 |
| \(\tau\)-BN \[60\]                          | Pseudo label [26] | 32.84 11.47 22.85 21.05 | 30.32 25.05 25.53 26.97 | 24.01 |
| \(\alpha\)-BN \[64\]                         | 31.15 14.26 24.27 23.23 | 33.14 27.13 27.72 29.33 | 26.28 |
| Ours                                          | 32.71 15.07 27.56 25.11 | 35.31 27.86 29.07 30.75 | 27.93 |

| (b) SYNTHIA \[43\] \(\rightarrow\) C-driving (30) | Model # | Un-sup. loss | Time of day | Weather | C-driving (30) |
|-----------------------------------------------|---------|-------------|-------------|---------|----------------|
| Source only                                   | -       | 25.36 8.12 19.90 17.80 | 24.59 22.52 24.71 23.94 | 20.87 |
| \(\tau\)-BN \[60\]                          | Entropy [49] | 22.21 10.23 17.92 16.79 | 21.94 20.56 21.53 21.34 | 19.06 |
| \(\alpha\)-BN \[64\]                         | 25.30 10.68 20.59 18.86 | 25.45 23.02 24.51 24.32 | 21.59 |
| Ours                                          | 25.84 11.21 21.61 19.55 | 26.70 23.34 24.86 24.97 | 22.26 |
| \(\tau\)-BN \[60\]                          | MaxSquare [3] | 22.20 10.21 17.91 16.77 | 21.93 20.54 21.51 21.32 | 19.05 |
| \(\alpha\)-BN \[64\]                         | 25.64 11.10 20.94 19.23 | 25.65 23.17 24.66 24.49 | 21.86 |
| Ours                                          | 26.11 11.46 21.55 19.70 | 26.85 23.31 24.78 24.98 | 22.34 |
| \(\tau\)-BN \[60\]                          | Pseudo label [26] | 22.26 10.29 17.97 16.84 | 21.99 20.64 21.60 21.41 | 19.12 |
| \(\alpha\)-BN \[64\]                         | 24.38 10.18 19.70 18.09 | 24.84 22.72 23.92 23.83 | 20.96 |
| Ours                                          | 25.20 10.56 21.51 19.09 | 26.21 23.05 24.88 24.71 | 21.90 |

5. Experiments

5.1. Datasets and Setups

We evaluate our proposed methods for segmentation adaptation on scenarios where compound domain test-time adaptation should be taken into account. The model that would be deployed to user applications is pre-trained on GTA5 \[42\] or SYNTHIA \[43\], which contain 24,966 and 9,400 synthetic labeled images respectively. After deployed, the applications encounter real samples in C-Driving (from BDD) \[30, 65\] which has attributes including ‘weather’ set (cloudy, rainy, snowy) or ‘time of day’ set (daytime, twilight, night). In this case, the model needs to handle naturalistic images from compound domain using test-time adaptation since the synthetic source data has the lack of photorealism compared to real target data.

5.2. Implementation Details

Pre-training segmentation model. For the segmentation models, we use DeepLabV3 \[2\] with a ResNet50 \[13\] backbone and initialize models with ImageNet \[7\] weights. We train on the source domain for 20 epochs (GTA5 and SYNTHIA) using an SGD optimizer. Since the model may be distributed to applications in the best possible condition in practice, we need to use the best pre-trained model. Specifically, we store weights every 0.5 epoch, and we use the weight with the highest source-only score in the validation split of C-driving domains. The detailed training settings are the followings: learning rate of \(1 \times 10^{-4}\) for GTA5 and \(5 \times 10^{-5}\) for SYNTHIA, weight decay of \(2 \times 10^{-4}\) and momentum of 0.9, batch size of 8 and resize images to 960x540.

Test-time adaptation on compound domain. TTA requires simple and rapid adaptation, so TENT \[60\] only optimize BN parameters and reports performance after 1 epoch of test-time optimization. We also utilize the same setting for all the baselines and our methods. Specifically, as the details of Fig. 4, we configure each domain with 2000 images in ‘weather’ or ‘time of day’ sets from C-Driving train-set and assume that three domains in the set

![Figure 4. Visualization of evaluation setting. Details in Sec. 5.2.](image-url)
then daytime and twilight ones. In addition, with pseudo difference, ‘time of day’ sets have a more distinct characteristic difference compared to ‘weather’ sets. It seems because $\alpha$ have a higher performance difference from $\beta$

days. In particular, we find that our methods consider the domain discrepancy between the target domains, we utilize the same segmentation model, and report their best results with hyper-parameters to be selected by cross-validation. For clear clustering. For $l$ th block, $DDR_l$ is computed as:

$$
DDR_l = \frac{\mathbb{E}_{m \sim N_c, k \sim N_j} \sum_{i \neq j} dist \left(s_i^{(m)}, s_j^{(k)}\right)}{\mathbb{E}_{m \sim N_c, k \sim N_j} \sum_{i = j} dist \left(s_i^{(m)}, s_j^{(k)}\right)}
$$

We summarize the result of quantitative analysis in Fig. 5(a), using C-driving dataset [30]. We find that the Bhattacharyya distance of Eq. (5) is the most principled choice as it has the best domain distinction rate in most cases, and the low-level features are more distinguishable to cluster the samples into each latent domain which aligns with previous works [10, 70].

5.4. Ablation Study and Analysis

Component-wise ablation. We summarize the ablations in Table 4, where we observe that our proposed methods bring favorable performance improvements. Specifically, if we do not apply online domain clustering, the model incurs sub-optimization due to the discrepancy between domains $(28.05 \rightarrow 27.35 \text{ mIoU})$. If we drop the importance-weighting stage, the model misses a chance to receive refined self-supervisory signals $(28.05 \rightarrow 27.64 \text{ mIoU})$.

Analysis on design choices. We conduct experiments to examine the effect of other design choices about: exponent value $\delta$ in Eq. (8) and alternative weighting strategies. Specifically, we find that (a) our distribution similarity-based weighting method outperform other weighting methods, probability and entropy aforementioned in Sec. 4.2 and (b) the large $\delta$ is likely to prevent sufficient entropy reduction to mitigate performance drop.

Discriminative features and distance function. As highlighted in Sec. 4.1, the key elements of the clustering algorithm are domain-distinguishable features and distance function, so we need to carefully design them. We consider different layer blocks in network-backbone and four distance metrics to compare two multivariate gaussian distributions $(p, q)$: euclidean, statistics-divergency [62], wasserstein [44], and bhattacharyya [21]. For quantitative comparisons, we calculate domain distinction rate (in short, $DDR$) which represent average expected value of the distance between images with different domains over one with the same domain. Theoretically, if the ratio is high, it means that the corresponding layer and distance metric are helpful for clear clustering. For $l$ th block, $DDR_l$ is computed as:

$$
DDR_l = \frac{\mathbb{E}_{m \sim N_c, k \sim N_j} \sum_{i \neq j} dist \left(s_i^{(m)}, s_j^{(k)}\right)}{\mathbb{E}_{m \sim N_c, k \sim N_j} \sum_{i = j} dist \left(s_i^{(m)}, s_j^{(k)}\right)}
$$

We summarize the result of quantitative analysis in Fig. 5(a), using C-driving dataset [30]. We find that the Bhattacharyya distance of Eq. (5) is the most principled choice as it has the best domain distinction rate in most cases, and the low-level features are more distinguishable to cluster the samples into each latent domain which aligns with previous works [10, 70].

Sensitivity to learning rate. Since TTA only sees the test data during adaptation, the stability can be largely affected by hyperparameters like learning rates. In Fig. 5(b), we run $\alpha$-BN [64] and our CD-TTA using entropy minimization with various learning rates, and can show that our methods perform robustly and show good stability during
adaptation with the result of higher mean and lower standard deviation. It seems because our method tries to avoid sub-optimization due to discrepancy between domains and optimizes the model using only refined unsupervised loss.

**Varying the number of latent domains.** The number of clusters $K$ is a crucial parameter in the clustering algorithm. As shown in Fig. 5(a), the higher the $K$, the fewer images are used to adapt a specific BN and more the gaps inside the compound domain are taken into account. Fig. 5(b) shows the effect of $K$ on performance, we observe that setting $K$ to 3 provides the best result.

**The effect of online clustering.** Fig. 5(c) is T-SNE visualization of flattened domain-distinguishable features $S^{(n)}$. The first row using oracle domain labels shows that our features are distinguishable enough to divide domains. The second row demonstrates that domain-centroids are well located for domain separation by our online clustering algorithm. We observe that $K$ domain-centroids are properly located. In a such state, we can expect that domain separation would be performed well by calculating the distance $D_B$ from the centroids.

**Weights for gradient denoising.** We predict the results with different domain samples of C-driving dataset and compute the weights for gradient denoising $w(x)^6$. We observe that if the segmentation results is more erroneous, the denoising-weight is likely to have a lower value. Therefore, we can anticipate that the backward signal produced by unsupervised loss from low quality prediction would be refined by this weight.

**5.5. Discussions**

Despite the impressive achievement of our method in CD-TTA, if we compare with other adaptation methods for UDA or SFDA losing constraints for practical adaptation as mentioned Sec. 2, most of TTA approaches show inferior performance improvement currently. Therefore, in order to exploit TTA techniques in real applications, we first need to study find a more sophisticated and reliable unsupervised loss [20, 5, 53], which would be better to be designed domain-specially for handling compound domain, since we can not rely on any supervisory signals in a TTA setup. Moreover, a long-term adaptation may lead to catastrophic failure [58, 22] due to training without ground-truth. Thus, exploration to solve this also seems to be a good research direction.

**6. Conclusion**

This work proposes a highly significant yet unexplored task termed compound domain test time adaptation (CD-TTA). In contrast to vanilla TTA, the new task CD-TTA considers the target domain to contain multiple homogeneous domains and these multiple domains might occur recurrently. Even though domain-specific TTA outperforms counterparts treating the compound domain as a single one, it is not feasible due to lack of ground-truth domain labels. To address this issue, we propose an online domain clustering method that helps predict pseudo domain labels. This algorithm is perfectly suitable for cyclical target domains because it enables us to leverage the domain parameter closest to the current target. Moreover, we introduce a distribution similarity-based gradient denoising method to alleviate the influence of unreliable self-supervision loss. Experiments demonstrate that our proposed methods reliably improve performance in compound domain.
References

[1] Woong-Gi Chang, Tackgeun You, Seonguk Seo, Suha Kwak, and Bohyung Han. Domain-specific batch normalization for unsupervised domain adaptation, 2019.

[2] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

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

[4] Sungha Choi, Sanghun Jung, Huiwon Yun, Joanne Kim, Seungryong Kim, and Jaegul Choo. Robustnet: Improving domain generalization in urban-scene segmentation via instance selective whitening, 2021.

[5] Sungha Choi, Seoungahn Yang, Seokeon Choi, and Sun grant Yoon. Improving test-time adaptation via shift-agnostic weight regularization and nearest source prototypes. arXiv preprint arXiv:2207.11707, 2022.

[6] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime, 2018.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–254, 2009.

[8] David Freedman, Robert Pisani, and Roger Purves. Statistics (international student edition). Pisani, R. Purves, 4th edn. WW Norton & Company, New York, 2007.

[9] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In ICML, 2015.

[10] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In CVPR, 2016.

[11] Rui Li, Qianfan Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9638–9647, 2020.

[12] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. JSTOR: Applied Statistics, 28(1):100–108, 1979.

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

[14] Xuefeng Hu, Gokhan Uzunbas, Sirius Chen, Rui Wang, Ashish Shah, Ram Nevatia, and Ser-Nam Lim. Mixnorm: Test-time adaptation through online normalization estimation, 2021.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In CVPR, 2017.

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICMIL, 2015.

[17] Takashi Isobe, Xu Jia, Shuaijun Chen, Jianzhong He, Yongjie Shi, Jianzhuan Liu, Huchuan Lu, and Shengjin Wang. Multi-target domain adaptation with collaborative consistency learning, 2021.

[18] Yusuke Iwasawa and Yutaka Matsuo. Test-time classifier adjustment module for model-agnostic domain generalization. In NeurIPS, 2021.

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

[20] Minguk Jang and Sae-Young Chung. Test-time adaptation via self-training with nearest neighbor information. arXiv preprint arXiv:2207.10792, 2022.

[21] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. IEEE transactions on communication technology, 15(1):52–60, 1967.

[22] Tommie Kerssies, Joaquín Vanschoren, and Mert Kilic kaya. Evaluating continual test-time adaptation for contextual and semantic domain shifts. arXiv preprint arXiv:2208.08767, 2022.

[23] Ansh Khurana, Sujoy Paul, Piyush Rai, Soma Biswas, and Gaurav Aggarwal. Sita: Single image test-time adaptation, 2021.

[24] Yunsheng Li, Lu Yuan, and Nuno Vasconcelos. Bidirectional learning for domain adaptation of semantic segmentation. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6929–6938, 2019.

[25] Jian Liang, D. Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. ICLR workshop, 2017.

[26] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttv++: When does self-supervised test-time training fail or thrive? Advances in Neural Information Processing Systems, 2021.

[27] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pages 248–254, 2009.

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

[29] Yuejiang Liu, Parth Kothari, Bastien van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttv++: When does self-supervised test-time training fail or thrive? Advances in Neural Information Processing Systems, 2021.

[30] Ziwei Liu, Zhongqi Miao, Xingang Pan, Xiaohang Zhan, Dahuaim Lin, Stella X Yu, and Boqing Gong. Open compound domain adaptation. In CVPR, 2020.

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

[32] 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, 2019.

[33] Toshihiko Matsuura and Tatsuya Harada. Domain generalization using a mixture of multiple latent domains. In AAAI, 2020.

[34] Ke Mei, Chuan Zhu, Jiaqi Zou, and Shanghang Zhang. Instance adaptive self-training for unsupervised domain adaptation. 2020.

[35] Chaithanya Kumar Mummadi, Robin Hutmacher, Kilian Rambach, Evgeny Levinkov, Thomas Brox, and Jan Hendrik
Inkyu Shin, Yi-Hsuan Tsai, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Sparsh Garg, In So Kweon, and Kuk-Jin Yoon. Multi-modal test-time adaptation for 3D semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16928–16937, 2022.

Inkyu Shin, Sanghyun Woo, Fei Pan, and InSo Kweon. Two-phase pseudo label densification for self-training based domain adaptation, 2020.

Yongyi Su, Xin Xu, and Kui Jia. Revisiting realistic test-time training: Sequential inference and adaptation by anchored clustering. arXiv preprint arXiv:2206.02721, 2022.

Baochen Sun, Jiashu Feng, and Kate Saenko. Correlation alignment for unsupervised domain adaptation. In Domain Adaptation in Computer Vision Applications. 2017.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei A. Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts, 2020.

Antonio Torralba and Alexei A Efros. Unbiased look at dataset bias. In CVPR 2011. IEEE, 2011.

Yi-Hsuan Tsai, Wei-Chih Hung, Samuel Schulter, Ki-hyk Sohn, Ming-Hsuan Yang, and Manmohan Chandraker. Learning to adapt structured output space for semantic segmentation, 2020.

Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games, 2016.

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 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 3234–3243, 2016.

Ludger Rüschendorf. The wasserstein distance and approximation theorems. Probability Theory and Related Fields, 70(1):117–129, 1985.

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. Adapting visual category models to new domains. In European conference on computer vision. Springer, 2010.

Antoine Saporta, Tuan-Hung Vu, Matthieu Cord, and Patrick Pérez. Multi-target adversarial frameworks for domain adaptation in semantic segmentation. In ICCV, 2021.

Mattia Segu, Alessio Tonioni, and Federico Tombari. Batch normalization embeddings for deep domain generalization, 2021.

Seonguk Seo, Yumin Suh, Dongwan Kim, Geeho Kim, Jongwoo Han, and Bohyung Han. Learning to optimize domain specific normalization for domain generalization, 2020.

C. E. Shannon. A mathematical theory of communication. The Bell System Technical Journal, 27(3):379–423, 1948.

Inkyu Shin, Kwanyong Park, Sanghyun Woo, and In So Kweon. Unsupervised domain adaptation for video semantic segmentation, 2021.

Inkyu Shin, Yi-Hsuan Tsai, Bingbing Zhuang, Samuel Schulter, Buyu Liu, Sparsh Garg, In So Kweon, and Kuk-Jin Yoon. Mm-tta: Multi-modal test-time adaptation for 3d semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16928–16937, 2022.
[67] Chaoming Zhang, Philipp Benz, Dawit Mureja Argaw, Seokju Lee, Junsik Kim, Francois Rameau, Jean-Charles Bazin, and In So Kweon. Resnet or densenet? introducing dense shortcuts to resnet. In WACV, 2021.

[68] Marvin Zhang, Sergey Levine, and Chelsea Finn. Memo: Test time robustness via adaptation and augmentation, 2022.

[69] Yuyang Zhao, Zhun Zhong, Zhiming Luo, Gim Hee Lee, and N. Sebe. Source-free open compound domain adaptation in semantic segmentation. ArXiv, abs/2106.03422, 2021.

[70] Kaiyang Zhou, Yongxin Yang, Yu Qiao, and Tao Xiang. Domain generalization with mixstyle. arXiv preprint arXiv:2104.02008, 2021.

[71] Yang Zou, Zhiding Yu, BVK Vijaya Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In Proceedings of the European Conference on Computer Vision (ECCV), pages 289–305, 2018.

[72] Yang Zou, Zhiding Yu, Xiaofeng Liu, B.V.K. Vijaya Kumar, and Jinsong Wang. Confidence regularized self-training. In The IEEE International Conference on Computer Vision (ICCV), October 2019.