Hybrid-TTA: Continual Test-time Adaptation via Dynamic Domain Shift Detection

Hyewon Park Hyejin Park Jueun Ko Dongbo Min†
Ewha Womans University, Seoul, South Korea
{hwpark, clrara, jueun.ko, dbmin}@ewha.ac.kr

Abstract

Continual Test Time Adaptation (CTTA) has emerged as a critical approach to bridge the domain gap between controlled training environments and real-world scenarios. Since it is important to balance the trade-off between adaptation and stabilization, many studies have tried to accomplish it by either introducing a regulation to fully trainable models or updating a limited portion of the models. This paper proposes ***Hybrid-TTA****, a holistic approach that dynamically selects the instance-wise tuning method for optimal adaptation. Our approach introduces Dynamic Domain Shift Detection (DDSD), which identifies domain shifts by leveraging temporal correlations in input sequences, and dynamically switches between Full or Efficient Tuning for effective adaptation toward varying domain shifts. To maintain model stability, Masked Image Modeling Adaptation (MIMA) leverages auxiliary reconstruction task for enhanced generalization and robustness with minimal computational overhead. Hybrid-TTA achieves* $0.6\%p$ *gain on the Cityscapes-to-ACDC benchmark dataset for semantic segmentation, surpassing previous state-of-the-art methods. It also delivers about 20-fold increase in FPS compared to the recently proposed fastest methods, offering a robust solution for real-world continual adaptation challenges.*

1. Introduction

Deep learning has gathered significant attention in computer vision thanks to its powerful representation capability and versatility. Despite achieving high performance on curated datasets, deep learning models face a major challenge: the gap between laboratory-controlled environments and ever-changing real world. These models often struggle to maintain accuracy when faced with diverse and unpredictable realistic scenarios. In this regard, studies including

This work was supported by IITP grant funded by MSIT (No. RS-2022-00155966: AI Convergence Innovation Human Resources Development (Ewha Womans University) and No. RS-2021-II212068: AI Innovation Hub). † Corresponding author

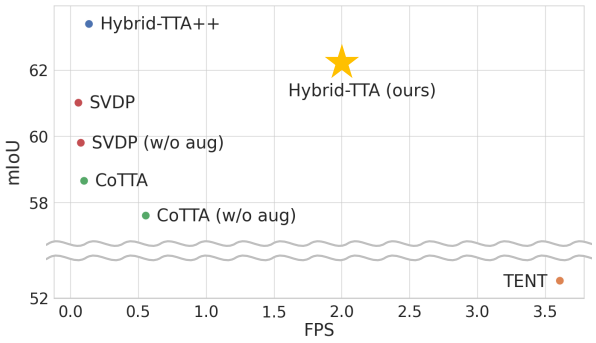


Figure 1. **Performance vs. FPS on Cityscapes-to-ACDC benchmark.** Hybrid-TTA achieves state-of-the-art 62.2% mIoU while being significantly faster than other methods with comparable mIoU performance. Hybrid-TTA++ utilizes Test-time Augmentation strategy as well as SVDP and CoTTA, resulting in 63.4%. See Tab. 4.

Domain Generalization (DG) [4, 8, 22, 23, 36, 47] and Domain Adaptation (DA) [2, 9, 15–17, 48] focus on enhancing robustness and adaptability of the model to ensure reliable performance in new, unseen domains. However, both of them have their own drawbacks: DG, which aims to generalize well to unseen domains, exhibits inherent limitations in adapting to continually changing target environments, as it cannot utilize information from target datasets. DA often struggles with flexibility because it typically assumes a stationary target dataset, limiting its effectiveness in constantly changing target domains.

Given the dynamic nature of target domains during testing, Continual Test-Time Adaptation (CTTA) has emerged as a solution, enabling models to adapt in real-time. Since CTTA operates in an online fashion, it commonly relies on unsupervised learning methods, such as pseudo-label generation with Mean Teacher [37, 41]. Unfortunately, unsupervised pseudo labels inevitably contain noise and inaccuracies, leading to error accumulation in the model [1]. Furthermore, as the model adapts to new, unfamiliar target domains, it gradually loses knowledge of the source domain, ultimately leading to catastrophic forgetting and overall per-

formance degradation.

These challenges have spurred the development of recent state-of-the-art methods [10, 25, 45, 46], attempting to separate domain-agnostic and domain-specific components of a model and focus on training the latter, inspired by Parameter-Efficient Fine-Tuning methods [3, 18]. On the other hand, other approaches [21, 26, 27, 41] continue to update all parameters, while mitigating these issues by either partially restoring source pre-trained weights or adding regularization term to stabilize the adaptation.

Full Tuning (FT) [21, 25–27, 29, 41, 45], which updates the entire model parameters, is well known for its adaptability, but comes with notable drawbacks. First, *unstable training and error accumulation*, where the reliance on self-generated pseudo labels causes error accumulation and performance degradation. Second, *low efficiency*, as updating the entire model for each target image requires high computational costs. In contrast, by updating only a subset of the model parameters, Efficient Tuning (ET) [10, 13, 40, 46] can better preserve the knowledge of the source domain while improving training efficiency. However, the *limited adaptability* of ET can confine the model to a rudimentary grasp of target domains, leading to suboptimal convergence. Since the pros and cons of both strategies are in a trade-off, we found that applying the appropriate updating strategy at the right time yields the desired performance, balancing plasticity and stability. This leads to the important question: *When is the right time to use Full or Efficient Tuning?*

To respond to this perplexing question, we propose **Hybrid-TTA**, a holistic approach that dynamically alternates between FT and ET, selecting the optimal training strategy for each target instance. Hybrid-TTA consists of two synergistic strategies; Dynamic Domain Shift Detection **(DDSD)** and Masked Image Modeling Adaptation **(MIMA)**.

Our approach is based on a teacher-student framework, similar to recent CTTA methods [25, 41], where the teacher model is updated via an exponential moving average (EMA) of the student model to generate pseudo-labels for the target image. The DDSD module detects domain shifts by capturing a significant drop in temporal correlation among adjacent input images. Since images from a new domain with different visual features exhibit lower temporal correlation with previous images, they increase the prediction discrepancy between the pseudo-label of the teacher model and the prediction map of the student model. This discrepancy upheaval is captured by DDSD based on a *dynamic loss thresholding*, enabling flexible domain shift detection across varying target domains. If a domain shift is detected, the entire model parameters are updated via FT to maximize the model adaptability toward the domain change. Otherwise, only a small proportion of the parameters are updated via ET to maintain the model stability. Note that in this paper, we focus on Adapter Tuning (AT) [3] as an exemplification of ET, as depicted in Fig. 2.

Furthermore, we propose **Masked Image Modeling Adaptation (MIMA)** in juxtaposition with DDSD, leveraging the ability of Masked Image Modeling (MIM) for continual model adaptation. MIM [14, 43] is well-known for its robustness and generalizability as a self-supervised learning method and has been applied for model adaptation [11, 17, 26]. Prior work [17] proposed Masked Image Consistency (MIC), an Unsupervised Domain Adaptation (UDA) method that uses masked images as input for the student model, to enhance the learning of contextual relationships and better utilize contextual cues in the target domain. However, the fundamental characteristics of UDA — accessible source dataset, mini-batch training, and repeated training on target dataset — favor MIC, making its direct application to CTTA challenging and necessitating an optimized approach.

Our simple yet effective CTTA-tailored MIM method, MIMA, integrates image reconstruction as an auxiliary task alongside the primary task of semantic segmentation. While providing the model with randomly masked input images as MIC, MIMA extends its capability by training on both tasks simultaneously, encouraging both the primary segmentation loss and the reconstruction loss to contribute to adaptation. This dual objective approach leverages the self-supervising nature of MIM for stable pseudo-label generation, and maximizes the use of image-driven information from the original target image. It is noteworthy that while recent CTTA methods [25, 41, 45] rely on Test-time Augmentation for a stable adaptation at the cost of low frame per second (FPS) (See Tab. 4), our method does not employ any additional pseudo-label enhancement techniques while delivering the state-of-the-art performance.

We summarize our main contributions as follows:

- We introduce a novel CTTA framework that alternates ET and AT for instance-wise model adaptation, balancing the trade-off between model adaptability and stability for dynamic target domains.
- To determine the training strategy for each target instance, we propose Dynamic Domain Shift Detection (DDSD) based on dynamic loss thresholding, which selects the optimal tuning method for each target instance.
- We introduce Masked Image Modeling Adaptation (MIMA), which integrates self-supervised image reconstruction as an auxiliary task to maximize learning from target images, leading to improved model stability and robustness.
- Our approach outperforms state-of-the-art methods on multiple CTTA benchmark datasets for semantic segmentation, while significantly reducing inference time.

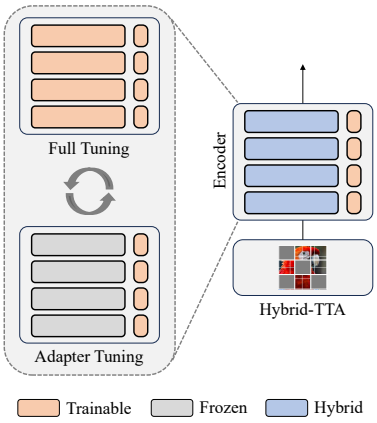


Figure 2. **Model structure of Hybrid-TTA.** Full Tuning (FT) and Adapter Tuning (AT) are used in hybrid, dynamically alternated fashion by Dynamic Domain Shift Detection (DDSD). When AT is activated, only adapter parameters, implemented via AdaptMLP [3], are trained.

2. Related Work

Continual Test-Time Adaptation CTTA addresses the challenge of adapting models in dynamic environments where target domains change over time, while traditional Test-Time Adaptation (TTA) methods assume a static target domain [19, 24, 44]. Recent CTTA works have employed techniques such as self-training and entropy minimization to adapt the model online [40, 41]. [41] mitigates error accumulation and catastrophic forgetting using Test-time Augmentation and stochastic weight reset, fully updating the source model for continual target domains.

Several parameter-efficient CTTA methods aim to mitigate catastrophic forgetting while reducing computational overhead. For instance, [35] employs a meta-network for stability, while [10, 12] update a small subset of parameters named visual domain prompts to improve efficiency. [25] leverages multi-rank adapters to capture both domain-agnostic and -specific features. Meanwhile, [26] employs HOG-feature reconstruction for robust adaptation, though its reliance on a handcrafted module may limit overall performance. [40] minimizes prediction entropy by adapting the batch normalization parameters, but struggles due to its limited adaptability.

Masked Image Modeling (MIM) MIM has gained prominence in self-supervised learning, enabling models to learn robust visual representations by reconstructing masked portions of images. Notable examples include [14] and [43] which mask significant parts of an image and train the model to reconstruct them, learning high-quality feature representations that benefit various downstream tasks.

Building on these principles, [17] uses MIM in unsupervised domain adaptation (UDA) for semantic segmentation by minimizing the consistency loss between teacher's pseudo-label from the original image and student's prediction map from the masked image. Recent works also incorporate MIM into Test-Time Training (TTT), such as [11] using MIM for both source training and test-time feature refinement in TTT, improving its adaptability to continual domain shifts. Also, [28] was proposed which leverages the model's ability of reconstruction to generate pseudo-labels for the main task.

Domain Shift Detection for CTTA Recent studies including [34] attempt to quantify distributional changes in input dataset via uncertainty estimation and statistical divergence metrics. In contrast, other techniques [6, 29] detect domain shifts by analyzing model responses through feature distance or confidence measurements. Our work, building on these ideas, determines domain shifts by measuring the difference in model responses between the student and temporally correlated teacher networks.

3. Preliminary

Continual Test-time Adaptation (CTTA) CTTA aims to adapt a model, which is initially trained on source data $\mathcal{D}_s = (\mathcal{X}_s, \mathcal{Y}_s)$, to multiple unlabeled target data $\mathcal{D}_T = \{\mathcal{X}_{T_1}, \mathcal{X}_{T_2}, \ldots, \mathcal{X}_{T_n}\}$ during deployment, where n represents the number of unseen domains. The entire adaptation process is constrained by two key challenges: (1) the model cannot access any source data during test-time adaptation, and (2) the model can only access each instance of target domain once, necessitating efficient and effective use of respective target instance. When the target domain data $\mathcal{X}_{T_i}$ consists of N_i target samples for $i = 1, \ldots, n$, for a simplicity of notations, we denote x_t by the t^{th} target instance for $t = 0, \ldots, |\mathcal{D}_T| - 1$, where $|\mathcal{D}_T| = \sum_{i=1}^{n} N_i$. Similarly, in the following sections of this paper, we omit the domain notation $T_1, ..., T_n$ as all target domains are considered to be integrated into a single sequence.

To tackle the challenges of CTTA as an unsupervised online learning, a common strategy is to use a weight-averaged teacher model to generate pseudo-labels for the target data [37, 41]. In source training, the student encoder f_θ and task decoder h_ϕ is trained on the source dataset $\mathcal{D}_s$, establishing the source-trained parameters θ_s, ϕ_s. During test-time, the teacher model f'_θ, h'_ϕ is firstly initialized with these parameters θ_s, ϕ_s. As the model encounters new target data over time t, the teacher model generates pseudo-labels $\hat{y}'_t = h_{\phi'_t}(f_{\theta'_t}(x_t))$ for each input x_t. The student model is then updated by minimizing the task loss $\mathcal{L}^{seg}_{\theta_t, \phi_t}$ between its own prediction $\hat{y}_t$ and the teacher's pseudo-label $\hat{y}'_t$.

After each update of the student model, the weights of the teacher model θ', ϕ' are adjusted using an exponential moving average (EMA) of the student's updated weights:

$$\theta'_{t+1} = \alpha \cdot \theta'_t + (1 - \alpha) \cdot \theta_{t+1} \tag{1}$$

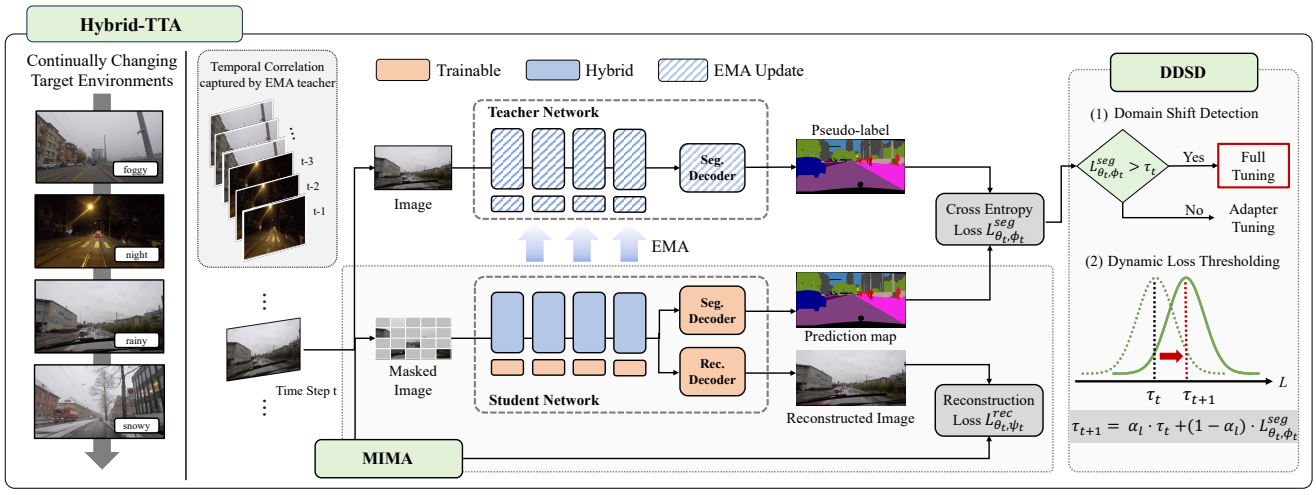


Figure 3. **Overview of the Hybrid-TTA framework.** Continual Test-time Adaptation with **Masked Image Modeling Adaptation (MIMA)** under continually changing target environments (e.g., foggy, night, rainy, snowy conditions). Here, **Dynamic Domain Shift Detection (DDSD)** is used along with MIMA, which detects the domain shift and switches between Full Tuning and Adapter Tuning via Dynamic Loss Thresholding to maintain optimal adaptation performance.

where α is a smoothing factor that governs the influence of the student's weights on the teacher model's updates. (1) applies for both θ' and ϕ'.

Masked Image Modeling (MIM) MIM is broadly utilized for effective pretraining [14, 43]. Input images are partially occluded with random patch masks and passed through the Transformer encoder, and the original intensities of the masked patches are reconstructed in a decoder. The training objective is typically set as:

$$\min_{\theta,\psi} \mathop{\mathbb{E}}_{x\sim\mathcal{D}} \mathcal{L}_{rec}(x \odot M, g_\psi(z) \odot M), \quad z = f_\theta(\tilde{x}), \\ \tilde{x} = x \odot (1-M) + e \odot M \tag{2}$$

where $\odot$ represents an element-wise multiplication, and $\mathcal{D}$ is the train data distribution. M denotes a patch mask, whose element is 1 if masked, 0 otherwise[1]. $f(\cdot)$ and $g(\cdot)$ are encoder and reconstruction decoder respectively, with parameters θ and ψ, and z is the learned representation. The masked input $\tilde{x}$ is generated by replacing the masked patches in x with trainable mask tokens e. The reconstruction loss $\mathcal{L}_{rec}$ is computed only for masked regions.

4. Proposed Method

We introduce **Hybrid-TTA**, a specially designed CTTA framework, illustrated in Fig. 3. To effectively manage the wide range of domain gaps presented by continually changing domains, a CTTA model must achieve the dual objectives of adaptability and stability, two distinctive qualities in a trade-off. Since FT offers flexible adaptation but struggles with instability, while AT is more reliable and efficient but has limited adaptability, Hybrid-TTA aims to accomplish both goals by dynamically alternating these adaptation methods through DDSD, as well as fully exploiting useful image features through complementary application of segmentation and reconstruction tasks via MIMA.

4.1. Dynamic Domain Shift Detection (DDSD)

As previously discussed, correctly determining when to use FT or AT is crucial for maximizing the benefits of Hybrid-TTA. Given that images from new, unseen domains tend to require adaptability and those from familiar domains prioritize stability, *domain shifts* can serve as a key indicator for alternating FT and AT. However, detecting domain shift is a challenging task due to the vast number of possibilities in real-world environments. Dynamic Domain Shift Detection (DDSD) aims to solve the problem using two distinctive approaches, temporal correlation and dynamic loss thresholding.

4.1.1. Domain Shift Detection via Temporal Correlation

In continually changing environment, a domain shift refers to a change in visual signals over time, such as an abrupt change in weather. These shifts cause distributional changes in the input image sequence, consequently altering the model's predictions over time. This can be understood through the concept of *temporal correlation*, which captures the consistency in model responses across consecutive time steps. In short, a domain shift causes the model to respond differently, reducing the temporal correlation.

To leverage this concept for domain shift detection, it

[1] $x \odot (1-M)$ represents "visible patches" and vice versa.

is necessary to maintain the history of previously seen images to compare with current input. Previous studies often utilize a replay buffer [5] to retain historical information by sampling and storing a subset of prior inputs, which requires additional memory. In contrast, we argue that the Mean Teacher model [37] can be viewed as an alternative mechanism for maintaining past domain characteristics, thanks to its evolving nature via EMA. Unlike the student model, which undergoes rapid updates, the teacher model evolves gradually, allowing it to retain temporal correlations in its predictions. So, if a domain shift occurs, the teacher model's predictions would differ from those of the student model, lowering the temporal correlation.

DDSD captures this discrepancy by comparing the student's predicted segmentation map and the teacher's pseudo-label using the cross-entropy loss. It effectively quantifies the divergence between dense segmentation maps while avoiding additional computational overhead, making DDSD exceptionally efficient for detecting domain shifts. Using this information, DDSD dynamically selects the optimal tuning strategy for each instance: FT is applied when a significant loss upheaval (*i.e.*, low correlation) indicates a major domain shift; AT is used when the loss remains relatively unchanged (*i.e.*, high correlation), implying the domain is stable.

4.1.2. Dynamic Loss Thresholding

Detecting domain shifts through temporal correlation often relies on a fixed threshold. Yet, this approach lacks flexibility, especially in CTTA scenarios where the model is continually evolving and the degree of domain shift varies across target instances.

To overcome this limitation, we introduce *Dynamic Loss Thresholding*, where the loss threshold evolves over time in addition to model adaptation. Assume $\mathcal{L}^{seg}_{\theta_t,\phi_t}$ is a segmentation loss, calculated between the teacher's pseudo label and the student's prediction map at time step t. An initial loss threshold τ_0 is set to 0, under the assumption that all upcoming target instances are under domain shift. At each time step t, this threshold is updated using an EMA of the newly calculated loss:

$$\tau_{t+1} = \alpha_l \cdot \tau_t + (1 - \alpha_l) \cdot \mathcal{L}^{seg}_{\theta_t,\phi_t} \tag{3}$$

where α_l is a smoothing factor for the EMA of loss threshold with default value 0.999 [37].

Hence, if a loss value at time t exceeds the threshold, *i.e.* $\mathcal{L}^{seg}_{\theta_t,\phi_t} > \lambda_d \cdot \tau_t$, DDSD identifies a domain shift, assuming that the temporal discrepancy between the student and teacher model is significant. Note that λ_d is sensitivity hyperparameter (See supplementary material).

4.2. Masked Image Modeling Adaptation (MIMA)

As previously mentioned, the critical challenge of CTTA is that the target images arrive without ground truth labels. This makes adaptation particularly difficult, as continually changing target domains often cause unstable adaptation. To mitigate error accumulation caused by pseudo-labels and fully exploit the visual information provided by target images, we incorporate a self-supervised learning method, Masked Image Modeling (MIM), to our CTTA framework.

As illustrated in Fig. 3, the target input image is masked and fed into the feature extraction encoder of the student model to generate a masked image representation. This representation is subsequently passed through two distinctive decoders for segmentation and image reconstruction, producing a prediction map and a reconstructed image, respectively. The segmentation loss is computed between the teacher's pseudo-labels and the student's predictions as following:

$$\mathcal{L}^{seg}_{\theta_t,\phi_t} = CE(\hat{y}'_t, h_{\phi_t}(f_{\theta_t}(\tilde{x}_t))), \tag{4}$$

where $\hat{y}'_t = h_{\phi'_t}(f_{\theta'_t}(x_t))$ represents the pseudo-label generated by the teacher model. For image reconstruction, the model aims to reconstruct the original image from the masked input $\tilde{x}$ based on (2). At time step t, the reconstruction loss between the original and reconstructed images is calculated using $L1$ loss, as below:

$$\mathcal{L}^{rec}_{\theta,\psi} = \frac{1}{|M|} \parallel (x - g_\psi(f_\theta(\tilde{x}))) \odot M \parallel_1, \tag{5}$$

A final loss is computed as in (6) and updated at each time t, as the model continuously adapts during the TTA process.

$$\mathcal{L}_{total} = \mathcal{L}^{seg}_{\theta,\phi} + \lambda_r \cdot \mathcal{L}^{rec}_{\theta,\psi} \tag{6}$$

Note that the reconstruction decoder is randomly initialized, not requiring pre-training or warm-up stage. Further experiments with respect to the reconstruction loss weight λ_r are provided in the supplementary material.

It is noteworthy that Masked Image Consistency (MIC) was proposed in [17] as a UDA plug-in that trains the student network to predict the complete segmentation map from randomly masked images to enhance contextual understanding of the target domain. However, key characteristics of UDA — access to the source dataset, mini-batch training, and repeated training on target dataset — enable MIC to maintain model stability while fully leveraging its potential. In contrast, the absence of these characteristics in CTTA makes models vulnerable to error accumulation, leading to unstable pseudo-labels. To fully exploit the useful features of the target image in a challenging CTTA setup where adaptation is conducted once and only with a single target image, we introduce MIMA, a MIM-based strategy that integrates masked image reconstruction and masked image segmentation, enhancing generalization and robustness as validated in experiments of supplementary material.

Table 1. Comparison of CTTA segmentation performance on **Cityscapes-to-ACDC benchmark** over 10 rounds. Mean is the average score of mIoU. The best results are highlighted in **bold**. Only 1, 4, 7 and 10th round scores are written, while the entire scores are provided in suppl. material. Scores of additional methods including VDP [10] and C-MAE [26] can also be found in supplementary material.

	t →																
Test	1				4				7				10				All↑
Condition	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	Mean
(a) Source	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	56.7
(b) BN Stats Adapt [33]	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	62.3	38.0	54.6	53.0	52.0
(c) TENT-continual [40]	69.0	40.2	60.1	57.3	66.5	36.3	58.7	55.0	64.2	32.8	55.3	50.9	61.8	29.8	51.9	47.8	52.3
(d) CoTTA [41]	70.9	41.2	62.4	59.7	70.9	41.0	62.7	59.7	70.9	41.0	62.8	59.7	70.9	41.0	62.8	59.7	58.6
(e) EcoTTA [35]	68.5	35.8	62.1	57.4	68.1	35.3	62.3	57.3	67.2	34.2	62.0	56.9	66.4	33.2	61.3	56.3	55.2
(f) SVDP [45]	71.0	43.2	64.8	62.0	**71.8**	43.9	66.6	62.3	71.9	43.6	66.5	62.3	71.6	43.5	**66.6**	62.0	61.0
(g) ViDA [25]	**71.6**	43.2	**66.0**	**63.4**	70.9	44.0	66.0	**63.2**	**72.3**	44.8	66.5	62.9	**72.2**	45.2	66.5	62.9	61.6
(h) Hybrid-TTA	70.3	**44.5**	65.1	63.2	70.1	**49.3**	**66.9**	63.1	69.6	**49.4**	**66.7**	**63.0**	69.9	**49.5**	66.5	**63.0**	**62.2**

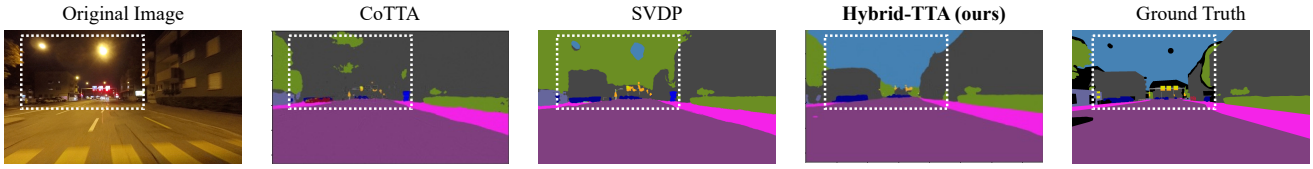


Figure 4. **Qualitative comparison on the Night domain of Cityscapes-to-ACDC benchmark.**

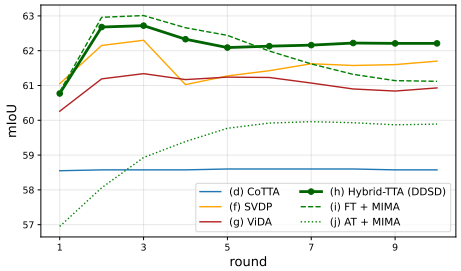


Figure 5. **Performance of various CTTA methods on Cityscapes-to-ACDC benchmark.**

5. Experiments

5.1. Experimental Setups

Dataset and Task Setup Experiments are carried out on the Cityscapes-to-ACDC [7, 32] benchmark, which assesses test-time performance of CTTA methods under four weather conditions (fog, night, rain, and snow) repeated cyclically for 10 times to evaluate long-term adaptation. We down-sampled Cityscapes to 1024×1024 and ACDC to 960×544, as the image sizes need to be multiple of the mask patch size 32. We also follow the experimental setup of [29] for assessment on the OnDA benchmark, which applies synthetic rain [38] of various rain intensities (25mm-200mm) to the Cityscapes [7] dataset. Note that the five domains in OnDA benchmark are cyclically repeated 5 times. We adopt two types of benchmarks to show the model performance in different environments, considering that OnDA benchmark contains a smaller domain gap with the source dataset (Cityscapes) compared to ACDC.

Implementation Details We follow the setup of the previous work [41], using SegFormer MiT-B5 [42] trained on down-sampled Cityscapes (1024×1024). The same pre-trained weights are used for initialization by all participating methods for a fair comparison. To implement Adapter Tuning, we incorporate AdaptMLP [3] in each transformer layer, which adds 0.17M extra parameters to the model, occupying 0.1% out of total model parameters. The adapter parameters are initialized according to [3] without any warm-up training. All experiments are conducted using NVIDIA A6000-ada and RTX3090. Further details are in the supplementary material.

5.2. Main Results

Cityscapes-to-ACDC benchmark Tab. 1 provides a comparative study with various CTTA methods on the Cityscapes-to-ACDC benchmark. Here, 'Mean' represents an average mIoU score over 10 rounds of experiments. Fig. 5 visualizes the 10-round scores of some methods. Note that (c) TENT-continual is a variant of TENT [40] originally proposed for test-time adaptation, with a slight modification to continuously update batch normalization parameters using entropy loss to make it suitable for CTTA. A source-trained model (a) SegFormer MiT-B5 works as a baseline with no adaptation. The performance of (b) BN Stats Adapt [33] and (c) TENT-continual [40] gradually declines due to catastrophic forgetting and limited adaptability. (e) EcoTTA [35], a CNN based ET method, shows a similar performance degradation. (d) CoTTA [41] maintains stable performance by preventing error accumulation with test-time augmentation, but fails to effectively adapt to the target domain, resulting in only moderate gains.

Current state-of-the-art methods, (f) SVDP [45] and (g) ViDA [25], achieve a significant performance gain compared to previous methods (a-e) by more than 2.4%p. Our

Table 2. Comparison of CTTA segmentation performance on **OnDA benchmark** over 5 rounds. Mean is the average score of mIoU. The best results are highlighted in **bold**. Only 1, 3 and 5th round scores are written, while the entire scores are provided in suppl. material.

t →	1						3						5						All↑
Test / **Intensity (mm)**	**25**	**50**	**75**	**100**	**200**	**mean**	**25**	**50**	**75**	**100**	**200**	**mean**	**25**	**50**	**75**	**100**	**200**	**mean**	Mean
(a) Source	67.7	65.6	62.9	58.1	45.3	59.9	67.7	65.6	62.9	58.1	45.3	59.9	67.7	65.6	62.9	58.1	45.3	59.9	62.4
(b) TENT-continual [40]	66.8	64.8	62.3	58.1	45.6	59.5	64.2	61.3	58.6	54.6	43.1	56.4	61.1	58.1	55.4	51.8	41.2	53.5	56.5
(c) CoTTA [41]	68.9	67.6	66.8	65.6	59.2	65.6	**68.7**	67.5	66.6	65.6	60.1	65.7	**68.6**	**67.6**	66.7	65.7	60.7	65.9	65.8
(d) SVDP [45]	**69.2**	**68.1**	66.9	65.5	61.7	66.3	67.5	66.8	66.1	65.3	62.6	65.7	67.4	66.6	66.1	65.4	62.1	65.5	65.7
(e) Hybrid-TTA	68.2	67.7	**67.4**	**66.8**	**62.9**	66.6	68.1	**67.7**	**67.1**	**66.5**	**64.5**	66.8	67.4	67.1	**66.7**	**66.0**	**64.3**	66.3	**66.7**

approach, (h) Hybrid-TTA, shows outstanding results compared to (f) and (g). In particular, as Fig. 5 shows, our method achieves a large performance leap in the first couple of rounds, as the DDSD module in Hybrid-TTA prefers to choose FT in the early stages of adaptation to overcome the initial domain gap between the source and target domains. We notice that the performance remains fairly stable over the remaining 8 rounds, where DDSD mostly selects AT to ensure stable adaptation and computational efficiency. This phenomenon is discussed in detail in Sec. 5.4.

In addition, Fig. 5 validates the effectiveness of DDSD of Hybrid-TTA by comparing it with FT and AT. In this graph, (h) Hybrid-TTA (DDSD) represents the result of Tab. 1. (i) always applies FT throughout all rounds, while (j) uses only AT. Note that MIMA is used in all cases for a fair comparison. The graph shows that (i), despite achieving good performance in the first three rounds, rapidly declines due to error accumulation and catastrophic forgetting caused by excessive updates, resulting in average performance of 61.9%. On the other hand, (i) exhibits suboptimal performance due to its limited adaptability, resulting in 59.27%. In contrast, (h), where DDSD orchestrates the adaptation, effectively leverages the strengths of both strategies, leading to outstanding results of 62.2%.

Furthermore, our approach shows a significant improvement in the Night domain (44.5%→49.5%) of Tab. 1, where (f) and (g) shows relatively moderate gains (43.2%→43.5%, 43.2%→45.2%). This is impressive because the Night domain is the most challenging with the largest domain gap from the source dataset due to the poor lighting condition [25]. Fig. 4 confirms better visual results of our method in the Night domain. Further qualitative comparisons are provided in the supplementary material.

OnDA benchmark Tab. 2 presents comparative results on the OnDA benchmark. (a) Source and (b) TENT-continual show a significant performance drop from easy (25mm) to hard (200mm) domains (*e.g.,* 67.7%→45.3%), due to limited adaptation capability. Moreover, the average performance of (b) per round severely declines due to catastrophic forgetting. (c) CoTTA shows fine mean mIoU (65.8%) thanks to its increased learning capacity and robust pseudo-labels. (c) also maintains steady performance, even slightly improving in the hard domain of 200mm (59.2%→60.7%), but still experiences severe drop from easy to hard domains ($-9.7\%p$) in the 1^{st} round. (d) SVDP reduces the impact of the domain shift, as the score gap from easy to hard domains in the 1^{st} round is reduced ((c) $-9.7\%p$→(d) $-7.5\%p$), but still shows moderate performance in the hard domain (61.7%→62.1%).

Table 3. **Ablation Studies in Cityscapes-to-ACDC benchmark.** DDSD represents the experiment where DDSD participated in tuning method selection. Gain denotes the performance gain compared with (a).

	FT	DDSD	MIMA	mIoU	Gain	FPS
(a)	✓			59.3±0.11		1.976
(b)	✓		✓	61.9±0.147	$+2.6\%p$	1.608
(c)		✓		59.8±0.19	$+0.5\%p$	2.546
(d) - ours		✓	✓	**62.2**±0.148	$+2.9\%p$	2.004

In contrast, (e) Hybrid-TTA achieves a total mean of 66.7%, outperforming all methods by approximately $1\%p$. It also shows significant improvement in 200mm domain (62.9%→64.3%) due to its improved adaptability. Moreover, (e)'s mean mIoU per round remains stable (66.6%→66.3%), with reduced performance drop from easy to hard domains ($-5.3\%p$) in the 1^{st} round, indicating a stable adaptation throughout the experiment.

5.3. Ablation Studies

In Tab. 3, we conducted ablation studies to assess the impact of DDSD and MIMA on the Cityscapes-to-ACDC benchmark. Note that 'FT' refers to experiments where only FT was used, and the results are averaged over 5 runs with standard deviations being reported. The results in (a) and (b) show that MIMA's self-supervised learning strategy significantly improves segmentation performance, yielding a $+2.6\%p$ gain. In (c), DDSD boosts segmentation performance by $+0.5\%p$ and also increases frames per second (FPS) by 130% (1.976→2.546). In (d), combining DDSD and MIMA results in a $+2.9\%p$ mIoU gain and a slight improvement in FPS (1.976→2.004). This result aligns with the performance visualization in Fig. 5, where (h) Hybrid-TTA (DDSD) mitigates the performance drop of (i) FT+MIMA during the last five rounds. Further analysis on DDSD is discussed in Sec. 5.4.

Table 4. **Performance and FPS comparison on various methods.** 10-round experiment on Cityscapes-to-ACDC benchmark (as in Tab. 1) is used to measure the performance and Frames Per Second (FPS). (a) Hybrid-TTA++ indicates that test-time augmentation strategy [41] is additionally utilized for performance improvement, and (w/o aug) of (d), (f) means it is removed for FPS gain.

Method	Tuning type	Performance (mIoU)	FPS
(a) Hybrid-TTA++	DDSD	63.4	0.136
(b) Hybrid-TTA (ours)	DDSD	62.2	2.004
(c) SVDP [45]	FT	61.0	0.058
(d) SVDP (w/o aug)	FT	59.8	0.076
(e) CoTTA [41]	FT	58.7	0.100
(f) CoTTA (w/o aug)	FT	57.6	0.555
(g) Source	no adaptation	56.9	7.984
(h) TENT-continual [40]	ET	52.5	3.609

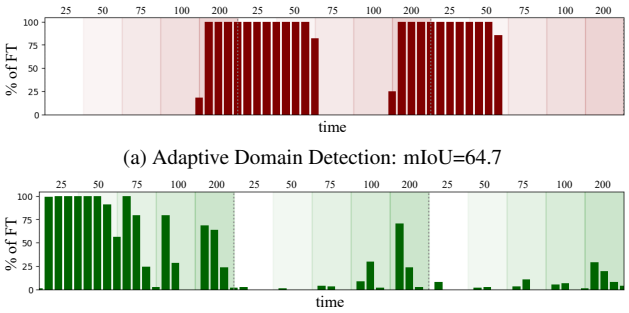
5.4. Analysis

Adaptation Time and Performance Tab. 4 compares the performance and FPS of various methods on the Cityscapes-to-ACDC benchmark, which is also displayed in Fig. 1. 10-round experiment, as in Tab. 1, measured performance and average FPS. All experiments were carried out in a fair environment using NVIDIA RTX 6000 Ada.

(b) Hybrid-TTA achieves the best performance of 62.2%, except for (a) Hybrid-TTA++, a variant of our method in which test-time augmentation [25, 41, 45] was applied. (a) shows the highest performance of 63.4%, a 1.2%p improvement due to enhanced pseudo-labeling via multiple augmentations. (b) reaches 2.004 FPS while delivering outstanding performance (62.2%). In contrast, (h) TENT-continual shows poor performance due to its limited adaptation capacity. Despite being significantly faster, both (h) TENT-continual and (g) Source yield low results of 52.5% and 56.9%, respectively. (c) SVDP and (e) CoTTA achieve good segmentation performance (58.7% and 61.0%) but are impractical for online adaptation, requiring about 17 seconds and 10 seconds to process a single image. In (d) SVDP (w/o aug) and (f) CoTTA (w/o aug), where the test-time adaptation strategy is removed, FPS improves slightly at the cost of reduced performance, but their FPS remains significantly lower than ours.

Domain Shift Detection Fig. 6 compares the behaviors of two domain shift detectors, DDSD and Adaptive Domain Detection (ADD) proposed in HAMLET [6], to examine their initial adaptation to new domains and reaction afterwards. We conducted cyclic adaptation using 5 rain intensities (25mm-200mm) 3 times on the OnDA benchmark. This benchmark, with more subtle domain shifts, is used to evaluate the sensitivity of domain shift detectors. The changing background colors indicate domain changes, while the y-axis shows the ratio of FT activation every 125 timesteps, *e.g.*, if FT is activated 100 times out of 125 timesteps and AT 25 times, the ratio is 80%.

Fig. 6a illustrates the behavior of ADD, which detects domain shifts by comparing the feature distance between



(a) Adaptive Domain Detection: mIoU=64.7

(b) DDSD: mIoU=66.73

Figure 6. **Performance comparison of Domain Shift Detection methods.** (a) is Adaptive Domain Detection [6] and (b) is ours.

the student model and ImageNet pretrained model. When a domain shift is detected, ADD activates FT for a fixed time window (*e.g.*, 1000 timesteps) for adaptation. We observe that ADD sticks to AT for the first 4 domains in the first round without detecting any shifts, and then detects the first domain shift at the start of 200mm domain, triggering FT for a certain timesteps. This is because ADD uses a fixed threshold to detect domain shifts, ignoring shifts with gaps less than 200mm. This static thresholding limits the the model's adaptability, leading to suboptimal performance with an mIoU of 64.7%. Moreover, we noticed that the FT was being activated longer than expected, until 25mm and 50mm domains in the second round. This overflow happens due to the fixed FT activation window of ADD [6], highlighting the need for instance-wise domain shift detection.

In contrast, Fig. 6b shows that DDSD activates FT at the beginning of the experiment, as it assumes that the model faces a large domain shift between source and target domains. After the first round, DDSD begins to prefer AT over FT, resulting in a reduced FT ratio. In the 200mm domain of the 2^{nd} round, the FT ratio dramatically increases, as 200mm is farther from source domain requiring additional adaptation. This happens again in the third round, but with significantly reduced FT ratio. This indicates that DDSD evolves over time with the changing target domains, leading to a superior mIoU of 66.7%.

6. Conclusion

In this paper, we have proposed Hybrid-TTA that dynamically selects the optimal tuning method for each input instance. This framework promotes the interaction between the model and ever-changing input sequences on target domain, resulting in improved performance. By integrating an auxiliary reconstruction task within segmentation CTTA, our model extracts generalized features and maintains robustness across domains. Experiments demonstrate that our method achieves outstanding performance in semantic segmentation with significantly reduced computational overhead. It is expected that our hybrid framework provides new insights on developing reliable techniques for CTTA.

References

[1] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2020. 1

[2] Mu Chen, Zhedong Zheng, Yi Yang, and Tat-Seng Chua. Pipa: Pixel-and patch-wise self-supervised learning for domain adaptative semantic segmentation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 1905–1914, 2023. 1

[3] Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678, 2022. 2, 3, 6, 11

[4] Junhyeong Cho, Gilhyun Nam, Sungyeon Kim, Hunmin Yang, and Suha Kwak. Promptstyler: Prompt-driven style generation for source-free domain generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15702–15712, 2023. 1

[5] Wonguk Cho, Jinha Park, and Taesup Kim. Complementary domain adaptation and generalization for unsupervised continual domain shift learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11442–11452, 2023. 5

[6] Marc Botet Colomer, Pier Luigi Dovesi, Theodoros Panagiotakopoulos, Joao Frederico Carvalho, Linus Härenstam-Nielsen, Hossein Azizpour, Hedvig Kjellström, Daniel Cremers, and Matteo Poggi. To adapt or not to adapt? real-time adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16548–16559, 2023. 3, 8

[7] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016. 6

[8] Yu Ding, Lei Wang, Bin Liang, Shuming Liang, Yang Wang, and Fang Chen. Domain generalization by learning and removing domain-specific features. *Advances in Neural Information Processing Systems*, 35:24226–24239, 2022. 1

[9] Zhekai Du, Xinyao Li, Fengling Li, Ke Lu, Lei Zhu, and Jingjing Li. Domain-agnostic mutual prompting for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23375–23384, 2024. 1

[10] Yulu Gan, Yan Bai, Yihang Lou, Xianzheng Ma, Renrui Zhang, Nian Shi, and Lin Luo. Decorate the newcomers: Visual domain prompt for continual test time adaptation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7595–7603, 2023. 2, 3, 6, 13

[11] Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 35:29374–29385, 2022. 2, 3

[12] Yunhe Gao, Xingjian Shi, Yi Zhu, Hao Wang, Zhiqiang Tang, Xiong Zhou, Mu Li, and Dimitris N Metaxas. Visual prompt tuning for test-time domain adaptation. *arXiv preprint arXiv:2210.04831*, 2022. 3

[13] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022. 2

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022. 2, 3, 4

[15] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Daformer: Improving network architectures and training strategies for domain-adaptive semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9924–9935, 2022. 1

[16] Lukas Hoyer, Dengxin Dai, and Luc Van Gool. Hrda: Context-aware high-resolution domain-adaptive semantic segmentation. In *European conference on computer vision*, pages 372–391. Springer, 2022.

[17] Lukas Hoyer, Dengxin Dai, Haoran Wang, and Luc Van Gool. Mic: Masked image consistency for context-enhanced domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11721–11732, 2023. 1, 2, 3, 5, 12

[18] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *European conference on computer vision*, pages 709–727. Springer, 2022. 2

[19] Jogendra Nath Kundu, Naveen Venkat, R Venkatesh Babu, et al. Universal source-free domain adaptation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4544–4553, 2020. 3

[20] Daeun Lee, Jaehong Yoon, and Sung Ju Hwang. Becotta: Input-dependent online blending of experts for continual test-time adaptation. *arXiv preprint arXiv:2402.08712*, 2024. 13

[21] Jae-Hong Lee and Joon-Hyuk Chang. Continual momentum filtering on parameter space for online test-time adaptation. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[22] Haoliang Li, Sinno Jialin Pan, Shiqi Wang, and Alex C Kot. Domain generalization with adversarial feature learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5400–5409, 2018. 1

[23] Ziyue Li, Kan Ren, Xinyang Jiang, Yifei Shen, Haipeng Zhang, and Dongsheng Li. Simple: Specialized model-sample matching for domain generalization. In *The Eleventh International Conference on Learning Representations*, 2022. 1

[24] Hyesu Lim, Byeonggeun Kim, Jaegul Choo, and Sungha Choi. Ttn: A domain-shift aware batch normalization in test-time adaptation. *arXiv preprint arXiv:2302.05155*, 2023. 3

[25] Jiaming Liu, Senqiao Yang, Peidong Jia, Renrui Zhang, Ming Lu, Yandong Guo, Wei Xue, and Shanghang Zhang.

Vida: Homeostatic visual domain adapter for continual test time adaptation. *arXiv preprint arXiv:2306.04344*, 2023. 2, 3, 6, 7, 8, 11, 15

[26] Jiaming Liu, Ran Xu, Senqiao Yang, Renrui Zhang, Qizhe Zhang, Zehui Chen, Yandong Guo, and Shanghang Zhang. Continual-mae: Adaptive distribution masked autoencoders for continual test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 28653–28663, 2024. 2, 3, 6, 13

[27] Jing Ma. Improved self-training for test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23701–23710, 2024. 2

[28] Youssef Mansour, Xuyang Zhong, Serdar Caglar, and Reinhard Heckel. Ttt-mim: Test-time training with masked image modeling for denoising distribution shifts. In *European Conference on Computer Vision*, pages 341–357. Springer, 2024. 3, 13

[29] Theodoros Panagiotakopoulos, Pier Luigi Dovesi, Linus Härenstam-Nielsen, and Matteo Poggi. Online domain adaptation for semantic segmentation in ever-changing conditions. In *European Conference on Computer Vision*, pages 128–146. Springer, 2022. 2, 3, 6, 11

[30] Stephan R Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In *European conference on computer vision*, pages 102–118. Springer, 2016. 13

[31] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3234–3243, 2016. 13

[32] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Acdc: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10765–10775, 2021. 6, 13

[33] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in neural information processing systems*, 33:11539–11551, 2020. 6

[34] Ziqi Shi, Fan Lyu, Ye Liu, Fanhua Shang, Fuyuan Hu, Wei Feng, Zhang Zhang, and Liang Wang. Controllable continual test-time adaptation. *arXiv preprint arXiv:2405.14602*, 2024. 3

[35] Junha Song, Jungsoo Lee, In So Kweon, and Sungha Choi. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11920–11929, 2023. 3, 6

[36] Zhaorui Tan, Xi Yang, and Kaizhu Huang. Rethinking multi-domain generalization with a general learning objective. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23512–23522, 2024. 1

[37] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 1, 3, 5, 11

[38] Maxime Tremblay, Shirsendu Sukanta Halder, Raoul De Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision*, 129:341–360, 2021. 6

[39] Riccardo Volpi, Pau De Jorge, Diane Larlus, and Gabriela Csurka. On the road to online adaptation for semantic image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19184–19195, 2022. 13

[40] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020. 2, 3, 6, 7, 8, 13, 15

[41] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022. 1, 2, 3, 6, 7, 8, 11, 13, 14, 15

[42] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 6

[43] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9653–9663, 2022. 2, 3, 4, 11

[44] Shiqi Yang, Yaxing Wang, Joost Van De Weijer, Luis Herranz, and Shangling Jui. Generalized source-free domain adaptation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8978–8987, 2021. 3

[45] Senqiao Yang, Jiarui Wu, Jiaming Liu, Xiaoqi Li, Qizhe Zhang, Mingjie Pan, Yulu Gan, Zehui Chen, and Shanghang Zhang. Exploring sparse visual prompt for domain adaptive dense prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16334–16342, 2024. 2, 6, 7, 8, 11, 13, 14, 15

[46] Xu Yang, Xuan Chen, Moqi Li, Kun Wei, and Cheng Deng. A versatile framework for continual test-time domain adaptation: Balancing discriminability and generalizability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23731–23740, 2024. 2

[47] Huaxiu Yao, Xinyu Yang, Xinyi Pan, Shengchao Liu, Pang Wei Koh, and Chelsea Finn. Improving domain generalization with domain relations. *arXiv preprint arXiv:2302.02609*, 2023. 1

[48] Wenyu Zhang, Qingmu Liu, Felix Ong Wei Cong, Mohamed Ragab, and Chuan-Sheng Foo. Universal semi-supervised domain adaptation by mitigating common-class bias. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23912–23921, 2024. 1

Hybrid-TTA: Continual Test-time Adaptation via Dynamic Domain Shift Detection

Supplementary Material

1. Implementation Details

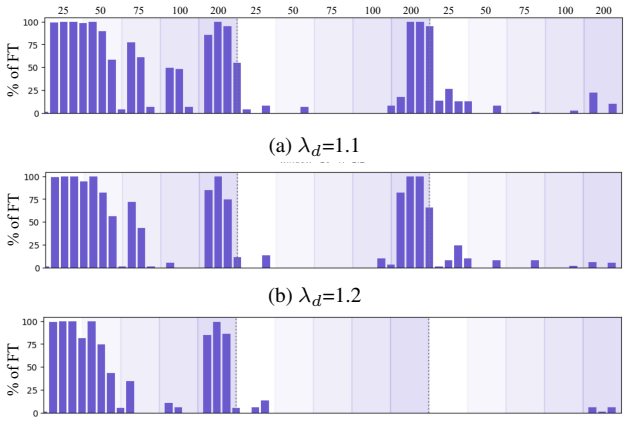
We report the implementation details used to train in the described method, including network architectures, and hyperparameters.

Model Architecture For semantic segmentation, we use SegFormer MiT-B5 for feature extractor f_θ and segmentation decoder h_ϕ. For reconstruction, SimMIM [43] is used for the implementation of reconstruction decoder g_ψ. To implement Adapter Tuning, we utilize AdaptMLP from [3] into each transformer layers to implement efficient tuning, where Full Tuning updates all parameters including the adapter, and Adapter Tuning updates only the adapter parameters (which occupies 0.1% of the entire parameters).

Hyperparameters For test-time adaptation, we set the batch size to 1, and used Adam optimizer with learning rate of $6 \times 10^{-5}/8$, in reference of [25, 41]. For DDSD implementation, we set the initial DDSD threshold τ to be 0, and default α_l to be 0.999, following [37]. For MIMA implementation, we masked the images with masking ratio 0.6 and mask patch size 32, as in [43]. Unlike recent CTTA studies [25, 41, 45] that used multi-scale input with flipping as the test-time Augmentation, we did not use any augmentation strategy for our main results. λ_d=1.1 and λ_r=0.3 were heuristically selected based on a small subset of the first target domain and kept fixed across all experiments to avoid test-time overfitting. The same hyperparameters are used for both benchmarks.

2. Ablation Study

Hyperparameter λ_d We conducted ablation experiments to investigate the role the hyperparameter λ_d in DDSD works within the system, using OnDA benchmark [29]. Only the segmentation loss L_{seg} has engaged in this ablation study to exclude the influence of MIMA. We performed cyclic adaptation over 5 rain intensities (25mm-200mm) 3 times, to understand how DDSD adapts to a completely new domains and how it reacts after some initial adaptations. Since 25mm is closer to the source domain and 200mm farther, we consider 200mm to be harder to adapt to, requiring more adaptation than 25mm. The domain changes are visualized by changing background colors, according to the changing rain intensities written at the top. The height of each bar represents the ratio of Full Tuning (FT) being activated in every 125 timesteps, *e.g.,* if FT is activated 100 times out of 125, the ratio is 80%.



(a) λ_d=1.1

(b) λ_d=1.2

(c) λ_d=1.3

Figure 7. **Ablation on hyperparameter λ_d.** Percentage of FT usage in OnDA benchmark. x-axis represents time and y-axis represents FT usage ratio. 5 domains are cyclically repeated 3 times.

Fig 7 illustrates how DDSD evolves over time. After aggressive adaptation in the first round, triggered by the initial encounter with the target domain, DDSD begins to favour AT over FT, leading to a reduced FT ratio. This change occurs because the target domain becomes relatively familiar to both the student and teacher models. However, in 200mm domain during the first round, the FT ratio increases, as 200mm is significantly more distant from any other domains and thus exhibits lower temporal correlation. This phenomenon consistently holds across all experiments, regardless of the size of λ_d value.

The effect of hyperparameter λ_d is particularly evident in 100mm domain during the first round. In Fig. 7a, the FT ratio is nearly 50%, while in Fig. 7b and Fig. 7c, the ratio drops sharply to below 10%. Similarly, in 200mm domain during the second round, FT is not activated in Fig. 7c, unlike the other two graphs.

From these observations, we can conclude that as λ_d increases, DDSD becomes less sensitive to domain shifts. This is due to the role of λ_d a sensitivity hyperparameter (described in Sec. 4.1), which acts as a scaling factor for the dynamic threshold τ_t. As the dynamic threshold increases, a larger loss is required to activate FT, and vice versa.

Hyperparameter λ_r In Tab. 6, we conducted ablation studies with respect to λ_r in (6), which determines the in-

Table 5. Ablation Study in Cityscapes-to-ACDC benchmark for **20 rounds.** Mean is the average score of mIoU.

Test Condition	FT	DDSD	MIMA	1 F	1 N	1 R	1 S	10 F	10 N	10 R	10 S	20 F	20 N	20 R	20 S	Mean ↑ 10R	Mean ↑ 20R
(a)	✓			70.6	42.7	62.7	61.7	69.3	42.7	61.2	59.7	65.5	39.3	55.9	58.5	59.3	55.9
(b)	✓		✓	70.3	44.5	65.1	63.3	68.9	49.2	64.9	61.5	66.2	48.2	62.0	57.5	61.9	59.6
(c)		✓		70.9	43.2	64.0	60.7	70.5	43.0	64.3	60.5	69.6	43.6	62.5	59.1	59.8	58.9
(d) - ours		✓	✓	70.3	44.5	65.1	63.2	69.9	49.5	66.5	63.0	68.8	50.1	63.9	59.9	62.2	61.5

Table 6. **Ablation on hyperparameter** λ_r Performance in Cityscapes-to-ACDC benchmark.

	λ_r	**Performance (mIoU)**
(a)	0.0	61.3
(b)	**0.3**	**61.9**
(c)	0.5	61.3
(d)	1.0	60.6

fluence of the reconstruction loss $\mathcal{L}^{rec}_{\theta,\psi}$. To exclude the influence of DDSD, we only utilized Full Tuning.

In (a), λ_r is set to 0, meaning that the reconstruction loss does not participate in test-time Adaptation process. This setup is similar to MIC [17], where the model is adapted solely through the consistency loss between student and teacher prediction maps. Since some important visual information is masked out from the target image, (a) yields a suboptimal performance of 61.3%. In (b), λ_r is set to 0.3, achieving the best performance. However, as λ_r increases in (c) and (d), performance declines. This degradation occurs because the influence of the segmentation loss diminishes as reconstruction loss dominates the adaptation process with big λ_r. Consequently, adaptation towards segmentation, which is our primary task, becomes weaker.

Long-term Adaptation Although 0.5%p improvement of DDSD in Tab. 3 may appear minor, we believe its core strength–stability–has been underestimated within the scope of the original 10-round experiment. Fig. 5 clearly illustrates that DDSD maintains stable performance, whereas Full Tuning (FT) exhibits a sharp performance decline after the 5$^{\text{th}}$ round, indicating that DDSD's advantage will grow over time. To further support this observation, we extend the ablation study reported in Tab. 3 from 10 rounds to 20 rounds. As shown in Tab. 5, (c) DDSD consistently outperforms FT at both the 10$^{\text{th}}$ and 20$^{\text{th}}$ rounds, with an widening performance gap (0.5→**3.0** between (a) FT and (c) DDSD, 0.3→**1.9** between (b) FT+MIMA and (d) DDSD+MIMA). These results strongly confirm the superior long-term stability of DDSD.

3. Analysis

Source Domain Forgetting Tab. 7 presents the performance on Cityscapes dataset after each round of adaptation on OnDA benchmark. Since Cityscapes serves as the source dataset for adaptation, we aim to assess the severity of catastrophic forgetting after each round of adaptation. 'Mean' represents the mIoU performance on Cityscapes, while 'Target Mean' refers to the performance on synthetic rain dataset, provided for comparison.

Table 7. Performance comparison of variants of proposed method on **Source Dataset** (CityScapes). *ours* refers to Hybrid-TTA, our main result, *Source* refers to SegFormer MiT-B5 with no adaptation.

Test	1	2	3	4	5	Mean	Target Mean
(a) Source	78.0	78.0	78.0	78.0	78.0	78.0	62.4
(b) AT	77.7	77.4	77.2	77.3	77.0	**77.3**	60.5
(c) DDSD	76.7	76.2	75.6	75.4	75.2	75.8	64.2
(d) DDSD+MIMA	77.0	76.0	75.3	75.0	74.7	75.5	**66.7**
(e) FT	76.5	75.6	75.2	74.9	74.7	75.4	65.0
(f) AT+MIMA	77.3	75.9	74.8	74.0	73.1	75.0	65.3
(g) FT+MIMA	76.8	75.7	74.8	74.2	73.7	75.0	66.3

(a) Source, SegFormer MiT-B5 without any adaptation, serves as the upper bound, achieving 78% on the source dataset, with moderate target performance of 62.4%. (b) AT, which updates only the adapter parameters, achieves the best source performance of 77.3%, but its target performance (60.5%) is lower than Source. This is because AT effectively preserves source knowledge and prevents catastrophic forgetting but has limited adaptability. However, this limitation is significantly mitigated by incorporating MIMA into AT, as (f) AT+MIMA shows 65.3% of target performance at the cost of slight reduction in source performance.

Here, we can assume that MIMA encourages the model to lose the source knowledge and collect target knowledge, as a similar pattern is observed with (e) FT and (g) FT+MIMA, where (g) exhibits lower source performance but higher target performance compared to FT. As seen from these findings, losing source knowledge is not necessarily disadvantageous, because the model learns as much as it forgets.

Nevertheless, excessive loss of source knowledge can degrade target performance. Notably, (d) DDSD+MIMA (ours) achieves an impressive 66.7% mIoU on the target dataset, with a gain of $+4.3\%p$ compared to Source, while substantially preserving source performance ($-2.5\%p$).

Table 8. Performance comparison in **OASIS benchmark**. GTA [30] as the source dataset, and ACDC [32] as the test dataset.

Test	1				2				3				
Condition	F	N	R	S	F	N	R	S	F	N	R	S	Mean ↑
Source	43.4	19.6	41.3	38.1	43.4	19.6	41.3	38.1	43.4	19.6	41.3	38.1	35.6
TENT [40]	43.8	19.8	42.1	38.7	43.9	18.8	39.9	35.9	43.0	17.1	37.4	33.5	34.5
SVDP [45]	**46.9**	**25.2**	45.6	41.8	48.0	25.0	43.8	39.6	45.1	22.5	42.6	40.1	38.8
C-MAE [26]	46.1	20.0	43.2	38.9	45.8	18.9	42.9	39.1	46.2	20.5	43.2	38.1	36.9
Ours	45.1	23.0	**46.6**	**42.5**	**48.9**	**26.9**	**48.2**	**43.3**	**49.2**	**28.8**	**48.0**	**43.3**	**41.2**

Table 9. Performance Comparison on **Cityscapes-to-ACDC benchmark** over 3 rounds.

Test	1				2				3				
Condition	F	N	R	S	F	N	R	S	F	N	R	S	Mean↑
CoTTA [41]	70.9	41.2	62.4	59.7	70.9	41.1	62.6	59.7	70.9	41.0	62.7	59.7	58.6
VDP [10]	70.5	41.1	62.1	59.5	70.4	41.1	62.2	59.4	70.4	41.0	62.2	59.4	58.2
BeCoTTA [20]	**72.3**	42.0	63.5	60.1	**72.4**	41.9	63.5	60.2	**72.3**	41.9	63.6	60.2	59.5
C-MAE [26]	71.9	**44.6**	**67.4**	**63.2**	71.7	44.9	66.5	63.1	**72.3**	45.4	**67.1**	63.1	61.8
Ours	70.3	44.5	65.1	**63.2**	71.8	**48.2**	**67.1**	**63.7**	71.2	**49.3**	**67.1**	**63.3**	**62.2**

GTA-to-ACDC benchmark We now dive deeper in Hybrid-TTA performance on the OASIS [39] benchmark protocol, as detailed in Tab. 8. The OASIS protocol involves training models on a synthetic source dataset (GTA [30]), tuning hyperparameters on a synthetic validation dataset (SYNTHIA [31]), and finally evaluating model performance on a real-world test dataset (ACDC [32]). It is widely acknowledged that the OASIS benchmark presents greater challenge than the Cityscapes-to-ACDC benchmark due to a significantly larger domain gap between source and target datasets.

Our base model was trained on GTA following the training protocol described in [39]. We select the best hyperparameters in SYNTHIA dataset as follows: λ_d=1.2, λ_r=0.2. Finally, we present CTTA results on ACDC dataset over 3 rounds, where Ours outperforms SoTA [26, 45], demonstrating our robustness under various environments.

Although using FT under large domain shifts may seem counter-intuitive, severe shifts require greater adaptability than what efficient tuning can offer (*e.g.*, TENT in Tab. 8), at the cost of stability and with the forgetting risks associated with FT. To manage this trade-off and achieve balance, DDSD selectively triggers FT under significant distribution shifts, while avoiding unnecessary updates in stable regimes.

Performance Comparison with Parameter-efficient Fine-tuning Methods Tab. 9 provides a detailed comparative study of various CTTA strategies on the Cityscapes-to-ACDC benchmark, extending the results discussed in 5.2, but evaluated over 3 adaptation rounds to highlight early-stage adaptation behaviors and performances.

VDP [10], a pioneering work that introduces a prompt-based Parameter-efficient Fine-tuning (PEFT) strategy for CTTA, achieves a relatively disappointing result of 58.2, notably underperforming even the baseline CoTTA. This suggests that prompt-based PEFT adaptation strategies might struggle in scenarios involving significant domain shifts, such as from CityScapes to ACDC.

BeCoTTA [20], which employs a Mixture-of-Experts (MoE) based Parameter-efficient Fine-Tuning CTTA strategy, demonstrates slightly better average mIoU (59.5) compared to VDP (59.4).

Performance Comparison with Continual-MAE Tab. 9 also provides performance of Continual-MAE [26], another pioneering strategy based on MIM, which attains a considerably stronger performance of 61.8. While our method employing MIMA outperforms C-MAE, our method also differs significantly in methodology. Specifically, C-MAE reconstructs HOG features to emphasize geometric changes (*e.g.*, shape), promoting the learning of domain-invariant features. On the other hand, MIMA reconstructs RGB values, not only enhancing robust feature extraction, but also capturing domain-specific low-level cues (*e.g.*, color, texture, brightness). This enables the model to better capture cross-domain feature discrepancies, allowing DDSD to detect domain shifts more effectively.

While MIM has been previously explored in TTA as a domain-agnostic regularizer [26] or pseudo-label generator [28], our work departs from these conventional usages in both design and intent. The integration between MIMA and DDSD is not merely synergistic but functionally complementary: MIMA enhances sensitivity to domain discrepancies while improving robustness on familiar data, and DDSD leverages this sensitivity to selectively detect domain shifts. We believe this task-driven coupling of MIM and domain shift detection in CTTA is novel, offering a new perspective on how reconstruction-based signals can be actively utilized beyond pretraining.

4. Performance Comparison (Full scores)

We provide the entire performance results of segmentation CTTA experiments in Tab. 10 and Tab. 11. Our proposed method, Hybrid-TTA, achieves 0.6%p mIoU improvement over the previous state-of-the-art method on Cityscapes-to-ACDC benchmark. Moreover, it outperforms other CTTA methods on OnDA benchmark by 0.9%p. Notably, Hybrid-TTA also achieves more than 20 times higher FPS than any other CTTA method with comparable mIoU performance, including CoTTA and SVDP (See Fig. 1 and Tab. 4), offering a robust solution for real-world online continual adaptation challenges.

5. Qualitative Results

Fig. 8 is qualitative comparison of segmentation results on Fog, Night, Rain and Snow domains for Cityscapes-to-ACDC benchmark. Hybrid-TTA (column 4) is showing remarkable segmentation results compared to other methods including CoTTA [41] and SVDP [45], notably SVDP being currently the state-of-the-art method in semantic segmentation CTTA.

In Night domain (row 3-4), Hybrid-TTA shows outstanding performance as it is demonstrated in the main paper. It is noticeable in Hybrid-TTA (row 3-4, column 4), particularly in distinguishing the sky from vegetation (green) and buildings (gray), where lighting conditions are poor. This is a significant achievement, given that other methods often misclassify these elements due to the altered appearance of objects at night.

For Rain domain (row 5-6), Hybrid-TTA excels in segmenting fine details such as fence (beige) and accurately identifying cars (deep blue) and sidewalk (pink) from road (purple), which other methods often confuse with vegetation (green) or terrain (light green). This highlights the model's ability to maintain clear boundaries and accurate object classification under adverse weather conditions.

In case of Snow domain (row 7-8), Hybrid-TTA effectively delineates sidewalk (pink) and other urban features, producing sharper segmentation maps compared to CoTTA and SVDP, which often blur these boundaries.

Overall, we can observe that Hybrid-TTA not only maintains robustness across diverse environmental conditions but also mitigate common segmentation issues, such as dirty segmentation maps observed in CoTTA (row 3, column 2) and SVDP (row 6, column 3). This robustness is particularly evident in the Night domain, which is typically considered the most challenging due to its low contrast and ambiguous object boundaries. These qualitative results underscore the effectiveness of Hybrid-TTA in real-world scenarios, where adaptability and precision are crucial.

Table 10. Performance comparison of Segmentation CTTA baselines on **Cityscapes-to-ACDC benchmark** over 10 cyclic rounds. Mean is the average score of mIoU.

Test	1				2				3				4				5				
Condition	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	Cont.
Source	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	-
CoTTA [41]	70.9	41.2	62.4	59.7	70.9	41.1	62.6	59.7	70.9	41.0	62.7	59.7	70.9	41.0	62.7	59.7	70.9	41.0	62.8	59.7	-
SVDP [45]	71.0	43.2	64.8	62.0	71.9	44.3	66.1	62.5	72.2	44.1	66.6	62.5	71.8	43.9	66.6	62.3	72.1	43.7	66.7	62.5	-
ViDA [25]	71.6	43.2	66.0	63.4	73.2	44.5	67.0	63.9	73.2	44.6	67.2	64.2	70.9	44.0	66.0	63.2	72.0	43.7	66.3	63.1	-
Hybrid-TTA	70.3	44.5	65.1	63.2	71.8	48.2	67.1	63.7	71.2	49.3	67.1	63.3	70.1	49.3	66.9	63.1	69.5	49.2	66.6	63.0	-

Test	6				7				8				9				10				All↑
Condition	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	**Fog**	**Night**	**Rain**	**Snow**	Mean
Source	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	69.1	40.3	59.7	57.8	56.9
CoTTA [41]	70.9	41.0	62.8	59.7	70.9	41.1	62.6	59.7	70.9	41.1	62.6	59.7	70.8	41.1	62.6	59.7	70.8	41.1	62.6	59.7	58.6
SVDP [45]	71.9	43.9	66.8	62.4	71.9	43.6	66.5	62.3	71.6	43.3	66.6	62.1	71.7	43.7	66.5	61.5	71.6	43.5	66.6	62.0	61.0
ViDA [25]	72.2	44.0	66.6	62.9	72.3	44.8	66.5	62.9	72.1	45.1	66.2	62.9	71.9	45.3	66.3	62.9	72.2	45.2	66.5	62.9	61.6
Hybrid-TTA	69.5	49.3	66.7	63.0	69.6	49.4	66.7	63.0	69.7	49.5	66.7	63.0	69.7	49.4	66.6	63.1	69.9	49.5	66.5	63.0	**62.2**

Table 11. Performance comparison of Segmentation CTTA baselines on **OnDA benchmark.** over 5 cyclic rounds. Mean is the average score of mIoU.

Test	1					2—					3					4					5					All↑
Intensity (mm)	**25**	**50**	**75**	**100**	**200**	**25**	**50**	**75**	**100**	**200**	**25**	**50**	**75**	**100**	**200**	**25**	**50**	**75**	**100**	**200**	**25**	**50**	**75**	**100**	**200**	Mean
Source	67.7	65.6	62.9	58.1	45.3	67.7	65.6	62.9	58.1	45.3	67.7	65.6	62.9	58.1	45.3	67.7	65.6	62.9	58.1	45.3	67.7	65.6	62.9	58.1	45.3	59.9
TENT-continual [40]	66.8	64.8	62.3	58.1	45.6	65.8	63.3	60.7	56.5	44.5	64.2	61.3	58.6	54.6	43.1	62.6	59.5	56.7	53.0	42.0	61.1	58.1	55.4	51.8	41.2	56.5
CoTTA [41]	68.9	67.6	66.8	65.6	59.2	69.2	68.1	67.2	66.0	60.3	68.7	67.5	66.6	65.6	60.1	68.6	67.5	66.6	65.8	60.3	68.6	67.6	66.7	65.7	60.7	65.8
SVDP [45]	69.2	68.1	66.9	65.5	61.7	67.7	67.2	66.6	64.7	61.2	67.5	66.8	66.1	65.3	62.6	67.7	67.1	66.2	65.4	62.2	67.4	66.6	66.1	65.4	62.1	65.7
Hybrid-TTA	68.2	67.7	67.4	66.8	62.9	68.7	68.2	67.5	66.8	64.4	68.1	67.7	67.1	66.5	64.5	67.6	67.3	66.9	66.2	64.4	67.4	67.1	66.7	66.0	64.3	**66.7**

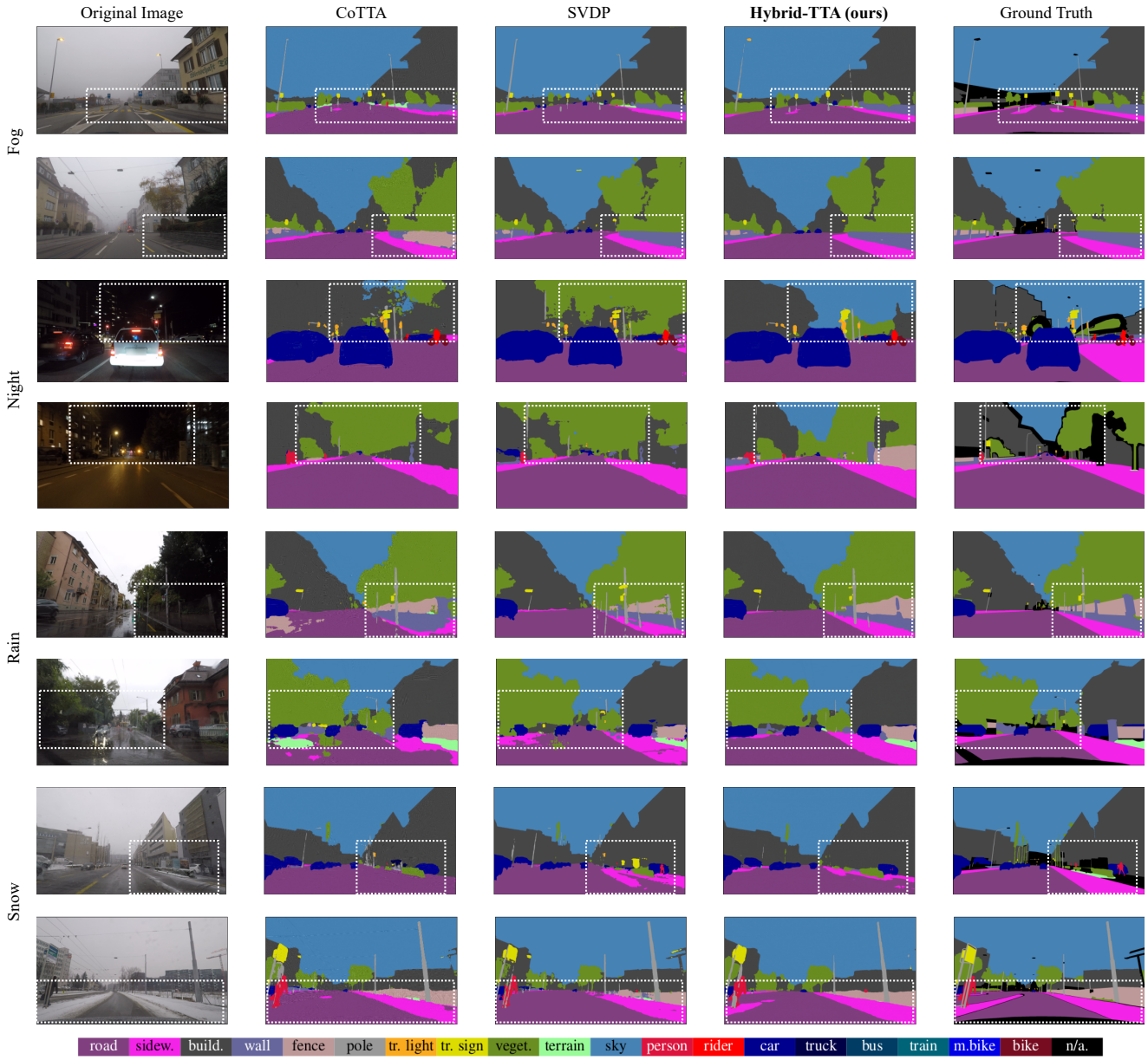


Figure 8. **Qualitative comparison of segmentation results on Fog, Night, Rain and Snow domains for Cityscapes-to-ACDC benchmark.**