Supplementary Material for Cloud-Device Collaborative Adaptation to Continual Changing Environments in the Real-world

A. Overview

The following aspects are included in this supplementary material.

- Supplementary experimental analysis
  - Performance with single-stage detector.
  - Additional ablation study.
    - Effect of transferring different number of parameters in the downlink phase.
    - Effect of visual prompts' size and location.
- Additional visualization results.
  - Heat-map visualization of student and teacher models before and after joint optimization.
  - Visualization of the visual prompts.
- Additional related work:
  - Continual domain adaptation.
- Demo video

B. Supplementary Experimental Analysis

B.1. Performance with single-stage detector.

To validate that our proposed method works for different networks, we replace the student model with another prevalent detector – YOLOV3 [6], and adopt the lightweight network MobilenetV2 [2] as the backbone. Experiments show that our method is effective on both single-stage and two-stage detectors. Besides, the single-stage detector is of less parameters and faster computing speed compared with the two-stage detectors, thus is suitable for client device deployment.

We compare the model's performance with several baselines. All experiments are conducted with a total of ten rounds. Each round contains five different environments: fog, motion blur, rain, snow, and brightness. Our proposed method is higher than the other methods from the first round of Motion, and the performance gap is widening continually. In conclusion, the total average performance of the model over ten rounds is 15.7% higher than that of the source-only method, even surpassing the state-of-the-art methods.

B.2. Additional Ablation Study

- **Effect of transferring different number of parameters in the downlink.** The number of parameters of visual prompt is tiny, accounting for only 0.4% of the student model parameters. Therefore, visual prompt can improve the performance of device model with negligible transmission overhead. Regarding AMS [3], we only update and transmit subsets of the device model and compare the performance with VPA used or not. As shown in Table 1, the reduction of model parameters will significantly reduce model's performance, but the degradation will be alleviated after VPA is used. Model using VPA can achieve better performance with only 40% parameters updated, which is similar to the performance of the integrated model.

Table 1. **Additional Ablation study.** Effect analysis of the number of parameters transmitted on the downlink. Experiments show that our proposed VPA can consistently improve the model's performance for different parametric quantities delivered from the cloud to the device.

	5%	10%	30%	40%	100%
w/o VPA	22.6	23.5	24.2	24.4	27.1
VPA	24.0	25.3	26.2	26.8	31.0
Gain	+1.4	+2.7	+2.7	+2.4	+3.9

Table 2. **Additional Ablation study.** Effect analysis of the Prompt location.

	50x50	100x100	200x200	300x300
Top-left	29.1	29.3	31.0	25.8
Top-right	29.1	29.0	29.9	24.4
Lower-left	29.2	29.1	28.4	24.6
Lower-right	28.9	29.2	28.6	24.8
Middle	27.9	26.7	25.3	19.4
Random	28.1	27.8	28.4	24.0

- **Effect of visual prompts' size and location.** We investigate the effect of the visual prompts' size and location on the model's performance. (1) Location: As shown in Table 2, the model shows relative poor performance when the visual prompt is placed in the middle compared with other locations. While it achieves best performance when the visual prompt is placed in the upper left corner. The visual prompts will reformulate the data to pull the distribution of the test data closer to the training data, thus improving the model's performance. However, for the target detection task, too much object occlusion is detrimental to the model learning, so placing the visual prompt in the back-

Table 3. **Continual generalization experiment with one-stage detector.** Object detection results (mAP@0.5 in %) on the Cityscapes-to-Cityscapes-C online continual test-time adaptation task. Gain(%) means the improvement of our method compared with Source-only. We evaluate five test conditions continually for ten times to verify the long-term adaptation performance. All results are evaluated on the YOLOV3 architecture with the largest corruption severity level 5. Our approach surpasses the SOTA method and exhibits significant continual generalization and anti-forgetting abilities.

Time	$t \longrightarrow$																
Round	1					5					10					All	
Condition	Fog	Motion	Rain	Snow	Brightness	Fog	Motion	Rain	Snow	Brightness	Fog	Motion	Rain	Snow	Brightness	Mean	Gain
Source-only [6]	15.0	5.2	19.0	0.4	16.7	15.0	5.2	19.0	0.4	16.7	15.0	5.2	19.0	0.4	16.7	11.3	/
TENT-continual [8]	15.2	5.0	19.2	0.5	16.9	11.4	3.4	15.8	0.3	14.0	8.6	2.1	7.7	0.1	8.3	8.7	-2.6
CoTTA [9]	15.6	5.4	19.3	0.5	17.3	13.4	5.5	18.6	0.4	16.9	14.3	5.3	16.1	0.5	17.1	11.7	+0.4
Pseudo-Label [4]	**20.2**	12.4	25.5	0.9	32.3	33.6	17.5	30.1	1.6	40.2	35.9	18.0	31.4	1.7	40.8	22.1	+10.8
Ours	16.7	**13.0**	**25.7**	**2.9**	**32.4**	**37.5**	**19.3**	**35.1**	**4.1**	**43.0**	**41.5**	**21.8**	**37.1**	**3.9**	**48.7**	**27.0**	**+15.7**

ground area is better.

(2) Size: As shown in Table 2, in most cases, the performance is optimal when the prompt size is 200*200. When the prompt size is too large, the visual prompt will obscure more objects, which is unsuitable for the target detection task. When the size of the prompt is too small, visual prompt can not effectively bring the test data and training data distribution closer. Therefore, taking the size of 200*200 can get better performance.

B.3. Additional Visualization results

- **Heat-map visualization of student and teacher models before and after joint optimization.** As shown in Fig.2, due to the distribution shifts, the model may not be able to focus on the objects well. Especially for the client device model, the performance of the model declines sharply when facing the distribution shift due to the restriction of computing resources. Using our CCA paradigm to optimize the client device model can significantly improve its feature expression ability. The heat-map visualization results show that after training with U-VPA teacher-student framework, the device model can extract target features better. Besides, our framework jointly optimizes the teacher and student models, so that they can be promoted synchronously.

- **Visualization of the visual prompts.** We have done a visualization of the visual prompt. Fig 1 shows that the visual prompt is some additional learnable parameters that are directly added to the original image at the pixel level. The visual prompt does not overwrite the original object very much. The experimental results in the main paper show that adding visual prompts to the upcoming data can effectively improve the performance of model inference because visual prompts can close the distribution of target data and training data.

C. Additional Related Work

Continual domain adaptation. Continual domain adaptation aims to improve model performance on continually changing target domains. [7] takes a meta-learning approach for augmentation to address the catastrophic forgetting under continual domain adaptation setting. [1] assumes the continuity between each distribution shift and utilizes a replay mechanism at every adaptation stage to address the forgetting issue. GRCL [1] proposes a unified framework with gradient regularized and domain memory to learn better domain-invariant representation as well as preserving model performance on source domain. AdaGraph [5] builds a domain graph where edges represent domain relation strength and predict the property of a new domain using graph message propagation algorithm during test time.

Different from aforementioned methods, we address the continual domain adaptation during test time under restrictions on computational power device setting. This setting is a common scenario for perception system in real-world environment, especially in autonomous driving.

D. Demo Video

We provide a video demo (see the attached MP4 file), which contains the motivation of our proposed Cloud-Device Collaborative (CCA) paradigm (0-1'10), the flow of the whole framework (1'10-3'20), and the visualization of the test results (3'20-4'20) on the Cityscapes raw dataset (autonomous driving video).

References

[1] Andreea Bobu, Eric Tzeng, Judy Hoffman, and Trevor Darrell. Adapting to continuously shifting domains. *Learning*, 2018. 2

[2] Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, M. Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neu-

(a) (b)

Figure 1. **Visualization of the visual prompts.** The visual prompt is added to the corner of the image, which pulls the distributions of the upcoming data closer to the training data's distribution.

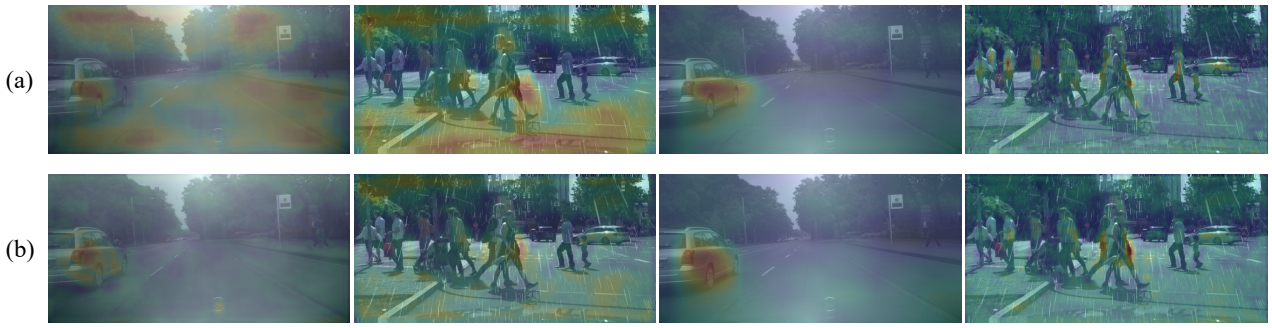

Figure 2. **Heat-map visualization of student and teacher models before and after joint optimization.** **(a)**Visualization of student model. **(b)**Visualization of teacher model. We compared the performance before and after using our CCA paradigm. After training with our U-VPA teacher-student framework, both student and teacher model can focus on the objects better, which means that our framework jointly optimizes the teacher and student models.

ral networks for mobile vision applications. *arXiv: Computer Vision and Pattern Recognition*, 2017. 1

[3] Mehrdad Khani, Pouya Hamadanian, Arash Nasr-Esfahany, and Mohammad Alizadeh. Real-time video inference on edge devices via adaptive model streaming. *arXiv: Learning*, 2020. 1

[4] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2022. 2

[5] Massimiliano Mancini, Samuel Rota Bulò, Barbara Caputo, and Elisa Ricci. Adagraph: Unifying predictive and continuous domain adaptation through graphs. *computer vision and pattern recognition*, 2019. 2

[6] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv: Computer Vision and Pattern Recognition*, 2018. 1, 2

[7] Riccardo Volpi, Diane Larlus, and Grégory Rogez. Continual adaptation of visual representations via domain randomization and meta-learning. *computer vision and pattern recognition*, 2020. 2

[8] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *Learning*, 2021. 2

[9] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, 2022. 2