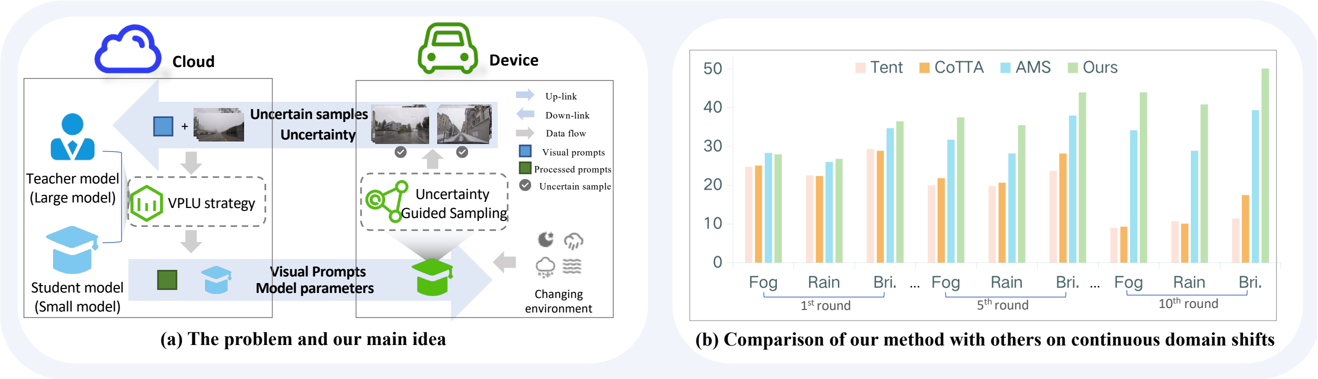
Cloud-Device Collaborative Adaptation to Continual Changing Environments in the Real-world

Yulu Gan[1]*, Mingjie Pan[1*], Rongyu Zhang[2], Zijian Ling[3],
Lingran Zhao[1], Jiaming Liu[1], Shanghang Zhang[1†]
[1]National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University,
[2]The Chinese University of Hong Kong, Shenzhen, [3]Imperial College London



(a) The problem and our main idea

(b) Comparison of our method with others on continuous domain shifts

Figure 1. (a) Models deployed on devices are preferably lightweight. However, device models will suffer from severe performance degradation when facing continual distribution shift data. Our main idea is to improve the continual domain adaptation capability of the device model by performing our proposed Cloud-Device Collaborative Adaptation paradigm. (b) We compare our method with previous works [16, 34, 35]. Our method surpasses the state-of-the-art approach and exhibits a solid ability when facing continual distribution shifts.

Abstract

When facing changing environments in the real world, the lightweight model on client devices suffers from severe performance drops under distribution shifts. The main limitations of the existing device model lie in (1) unable to update due to the computation limit of the device, (2) the limited generalization ability of the lightweight model. Meanwhile, recent large models have shown strong generalization capability on the cloud while they can not be deployed on client devices due to poor computation constraints. To enable the device model to deal with changing environments, we propose a new learning paradigm of Cloud-Device Collaborative Continual Adaptation, which encourages collaboration between cloud and device and improves the generalization of the device model. Based on this paradigm, we further propose an Uncertainty-based Visual Prompt Adapted (U-VPA) teacher-student model to transfer the generalization capability of the large model on the cloud to the device model. Specifically, we first design the Uncertainty Guided Sampling (UGS) to screen out challenging data continuously and transmit the most out-of-distribution samples from the device to the cloud. Then we propose a Visual Prompt Learning Strategy with Uncertainty guided updating (VPLU) to specifically deal with the selected samples with more distribution shifts. We transmit the visual prompts to the device and concatenate them with the incoming data to pull the device testing distribution closer to the cloud training distribution. We conduct extensive experiments on two object detection datasets with continually changing environments. Our proposed U-VPA teacher-student framework outperforms previous state-of-the-art test time adaptation and device-cloud collaboration methods. The code and datasets will be released.

1. Introduction

Real-world usually contains various environmental changes along with continual distribution shifts [35]. People usually deploy economically lightweight models on devices to boost the scalability and practicability. The lightweight model can suffer severe performance degrada-

[*]Equal contribution
[†]Corresponding author

tion under continual distribution shifts [25, 29, 33, 35]. The main challenges are: (1)The poor computational ability of devices. Due to the properties of device infrastructure, the deployed model can not be updated on time, thus lagging its performance for the real world under distribution shift. (2) The limited generalization ability of the lightweight model. Since lightweight models are of relatively small capacity, they can not handle continually changing environments. In contrast, recent large models that are trained on the cloud server show significant generalization ability [2, 39]. While in industry, these large models can not be directly applied due to the limited infrastructure.

Therefore, we enable the device model to tackle real-world environmental changing by proposing a Cloud-Device Collaborative Continual Adaptation paradigm, as shown in Fig .1 (a). Previous Cloud-Device Collaborative methods only focus on improving model representation on variance in video frames but neglect the model generalization ability for continually changing data distribution. In our new paradigm, we fully exploit the sufficient knowledge of the large cloud model and transfer the continual generalization ability to the device lightweight model.

In particular, we design an Uncertainty-based Visual Prompt Adapted (U-VPA) teacher-student model, which consists of an Uncertainty Guided Sampling (UGS) strategy and a Visual Prompt Learning Strategy with Uncertainty Guided Updating (VPLU). Due to the communication bandwidth constraint [11, 13, 19], different with some work targeting filtering the parameters [22, 26], we design the UGS to screen out the most environment-specific samples and decrease the required bandwidth compared with transmitting the whole sequence. To leverage the strong generalization ability of large models, we introduce the VPLU to align source-target domain distribution and transfer the representation of the large teacher model to the light student model. The light student model and visual prompts are then delivered to the device, thus alleviating the continuously changing scenarios in the real world.

Experimental results show that our method outperforms the state-of-the-art methods on synthetic and real-world distribution shift datasets, as shown in Fig .1 (b). Besides, we can achieve the same performance as the entire data with fewer reflowed data (42% of total data). As another benefit, fewer reflowed data reduce the bandwidth pressure of the uplink. As for the downlink, we can deliver the visual prompts (0.43% of the model's parameters) with almost negligible bandwidth to the device and apply the visual prompt to the input data to improve the performance of the device model by 3.9% in mAP.

Our contributions can be summarized as follows:

- We make the first attempt to deal with continually changing scenarios by proposing a Cloud-Device Collaborative Continual Adaptation paradigm, which aims to transfer the generalization ability from the large cloud model to the lightweight device model. Our method is a general paradigm that can apply to real-world systems.
- We design an Uncertainty-based Visual Prompt Adapted (U-VPA) teacher-student model, which consists of UGS and VPLU. We introduce UGS to screen out the most environment-specific samples and decrease the required bandwidth compared with transmitting the whole sequence. We propose a VPLU to align source-target data distribution and transfer the representation of the large teacher model to the lightweight student model.
- Experiments show that our proposed framework and method surpass other state-of-the-art methods and can effectively improve the continuous domain adaptation capability of the device model.

2. Related work

Test time adaptation. Test-time adaptation(TTA) aims to get better model generalization performance on target data without access to source data. Feature modulation is used in TENT [34] to optimize BN layers of the input pre-trained model by entropy minimization. SHOT [20] takes entropy minimization and a diversity regularizer to achieve information maximization. [18, 24] both take a productive way to enhance model performance on the target domain.

We differ from TTA in that we consider continuously changing domains. Besides, our model is deployed on the device. There will be restrictions on computing resources and uplink and downlink transmission bandwidth. Considering TTA can effectively use the test data to make updates, we leverage TTA to enable the cloud to improve generalization performance continuously.

Prompt learning. Prompt is originally from the natural language process (NLP) and aims to adapt the large-scale pre-trained language model to downstream tasks. Recently, several methods [1, 9, 21, 40, 41] are proposed to prove the feasibility of using prompts to adapt the large pre-trained model to downstream tasks in computer vision or vision-language tasks. [1, 21, 37] take prompt as additional input containing a small number of learnable parameters and explore the design of prompts.

Prompt learning aims at mining the pre-trained model's information. We introduce visual prompt tuning to improve the performance of the device model on the current domain, with a negligible computational overhead in our Cloud-Device Collaboration (DCC) framework.

Cloud-device collaboration. Cloud-device collaboration aims to bring benefits to both the cloud and the device jointly. Several methods [4, 6, 7, 15] try to unload part or

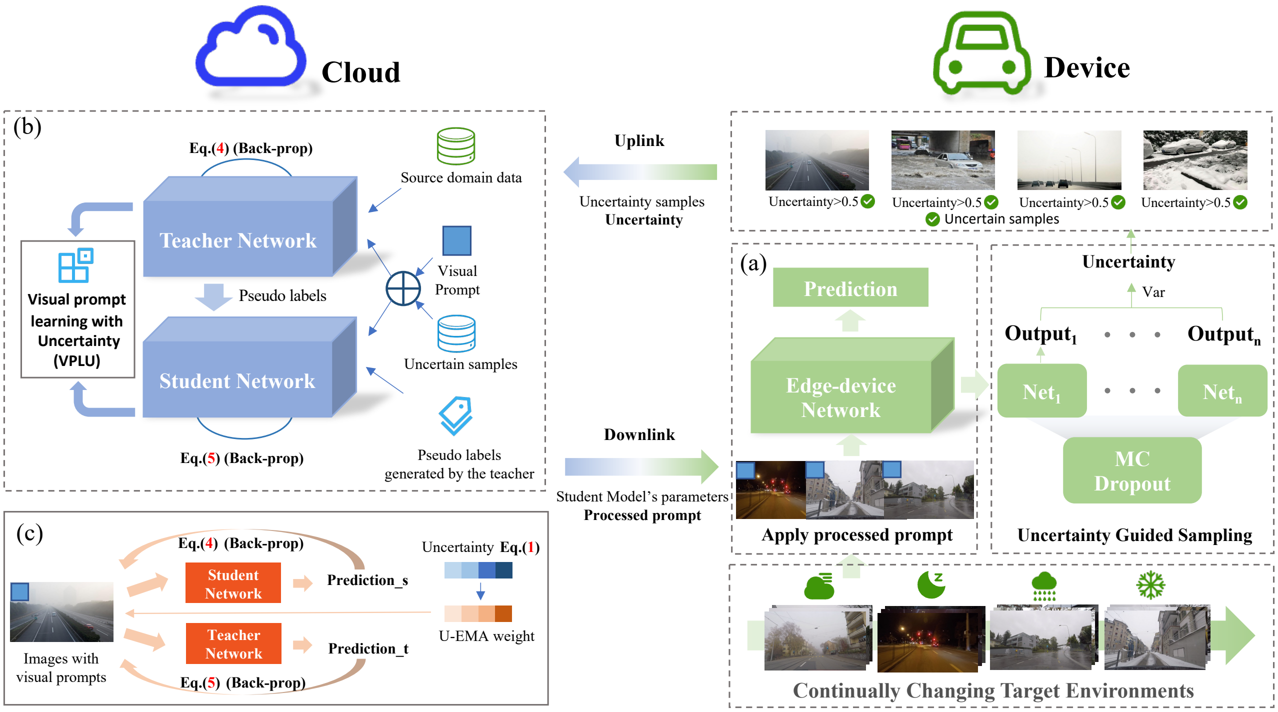


Figure 2. **The whole framework. (a): Testing on the device**. On the client device, we select data that need to be sent to cloud by Uncertainty Guided Sampling (UGS). These uncertainty samples are then used in the cloud to update the teacher-student models. The visual prompt trained on the cloud will be delivered to the device through the downlink and applied to the input data x. Then the reformulated input data will be put into the device model for testing. **(b) Training on the cloud.** We add our proposed visual prompt to the uncertain samples at the pixel level to get the reformulated images. We then put the reformulated images into the teacher-student model. The teacher-student model and visual prompts are optimized by align loss and supervised loss. **(c) Visual prompt learning strategy with uncertainty guided sampling (VPLU).** We update the visual prompts by uncertainty-based EMA update. The update weight of the visual prompt will be larger when the uncertainty is higher. The student model is updated using the uncertainty samples and the pseudo labels.

all computation to the cloud to alleviate the computation deficiency on the device. But they do not leverage the collaboration between the cloud and the device. AMS [16] and DCCL [38] have considered the collaboration but they do not take the real-world continual distribution shifts into account. Compared with methods above, our proposed Cloud-Device Collaborative Continual Adaptation paradigm can deal with the real-world environment changes.

In our work, we use Uncertainty Guided Sampling (UGS) and visual prompt for more efficient bandwidth usage. Besides, our model can adapt to different domains during test time. Our teacher model on the cloud and our student model on the device can be improved simultaneously.

3. Method

3.1. Preliminary

Given a lightweight model θ_{device} that is deployed on the device, our goal is to perform well on the continually changing target domains $D_{t1}, D_{t2}, ..., D_{tT}$ with limited communication bandwidth, where $D_{ti} = \{(x_i^T)\}_{i=1}^{N_t}$, and N_t represents the scale of the target domain. The distributions of the target domains are arbitrary since they can change or reoccur over time. We thus introduce a large teacher model θ_{tea} and lightweight student model θ_{stu} cloned from θ_{device}, to boost the generalization ability of θ_{device}.

3.2. Overall framework

In our Cloud-Device Collaborative Continual Adaptation paradigm, the key idea is to improve the device model's performance continuously by receiving adapted parameters from the cloud server. So we propose Visual Prompt Adapted (U-VPA) teacher-student model. As shown in Fig. 2, the U-VPA teacher-student model contains Uncertainty Guided Sampling (UGS) and Visual Prompt Learning Strategy with Uncertainty Guided Updating (VPLU).

When perceiving scene changing or distribution shift, poor computational device intends to get help from a cloud server with powerful computing ability. However, due to the communication bandwidth constraint, we must consider the required bandwidth when concurring the distribution shifts. Therefore, we design a Uncertainty Guided Sampling (UGS) to screen out the most environment-specific samples and decrease the required bandwidth compared with transmitting the whole sequence. After receiving the learned parameters from the cloud to deal with these samples of relatively large distribution shifts, it can adapt to the new scene when inference.

On the cloud server, due to its strengths in computing, we intend to leverage the strong generalization ability of large models to tackle the received environment-specific and challenging samples. We introduce Visual Prompt Learning Strategy with Uncertainty guided Updating (VPLU) to align source-target domain distribution and transfer the representation of the large teacher model to the light student model. The light student model and visual prompts is then delivered to the device, thus alleviating the continuously changing scenarios in the real-world.

We will discuss the details of UGS and VPLU in Sec. 3.3.1 and Sec. 3.3.2.

3.3. U-VPA teacher-student model

3.3.1 Uncertainty Guided Sampling(UGS)

Due to bandwidth limitations, it is not practical for the device to return all data. Therefore, we select images that need to be transmitted based on uncertainty estimation. Inspired by the Dropout methods [8, 12], since we are doing an object detection task, we can obtain n probabilities $p_i(y|x)$ of the model by dropout method, and we calculate the variance of these n prediction $p_i(y|x)$ as uncertainty.

$$V_{unc} = (\frac{1}{n}\sum_{i}^{n}(p_i(y|x) - \mu)^2)^{\frac{1}{2}} \tag{1}$$

where $p(y|x)$ represents the predictions of the input image x and μ is the mean. V_{unc} is the variance of n probabilities of the model, which represents the uncertainty of the model for the data. The device model filters the input images x whose uncertainty is larger than a threshold and sends back to the cloud server.

3.3.2 Visual Prompt Learning Strategy with Uncertainty Guided Updating (VPLU).

We optimize model on the cloud server by using the data transmitted from the device. It then delivers the learned parameters and processed visual prompt to the device model for inference. We thus introduce VPLU with a large teacher model and light student model, to mitigate variant domain distribution and improve the generalization ability of light model. Note that, teacher model is initialized by the source model pre-trained on source domain. It takes source and target domain as input, while student model only trains on the target domain.

Cross-domain alignment. In order to mitigate distribution shift, we align the features in the source and target domains. Inspired by [10], we apply feature alignment on teacher model to improve its performance on target domain. We use a domain discriminator to classify data in feature space and try to distinguish which domain the data comes from. Besides, we use a gradient reversal layer to achieve a uniform training process. Source domain data is used to supervise teacher model, avoiding performance degradation.

Cross-model transfer. After improving teacher model's domain-invariant representation ability, we transfer its ability against distribution shifts and generalization to student model via visual prompts and knowledge transferring.

Visual Prompt Adaptation(VPA) learning. Specifically, inspired by the prompt learning in NLP [23], which can effectively explores the knowledge of source domain, we propose visual prompts as hints to continually transfer teacher model's generalized representation to the student model.

In particular, visual prompts ϕ are learnable parameters. During the forward process of the model, we can obtain the reformulated image x^* by adding visual prompts on the input images x at the pixel level.

$$x^* = x + \phi, \tag{2}$$

In our framework, whether in training on server or inference on device, we all use the reformulated image x^* as model's input, instead of origin images x.

Uncertainty-aware Prompt Updation(U-EMA). Visual prompt acts on the input data and participates in the forward process of both teacher and student models, so it can also benefit from their optimization during training. Moreover, to make the prompt update more stable and consider the current model's uncertainty to data, we update the visual prompt using by uncertainty-based exponential moving average(U-EMA). The weight of the exponential moving average is based on the uncertainty estimation as follows:

$$\phi_t \leftarrow \beta\phi_{t-1} + (1-\beta)\phi_t, \tag{3}$$

Note that ϕ_t represents the parameters of the visual prompt that need to be updated, and ϕ_{t-1} indicates the visual prompt parameters of the last training step t. We follow [30] to set $\alpha = 0.999$. And $\beta = \alpha - V_{unc}$, where V_{unc} is defined in Eq. (1) as the uncertainty of current data. Higher uncertainty means a larger distribution shift in the current data. To save more domain-related information, the prompt update weight will be larger at this time. That is, the larger the uncertainty, the greater the weight of the prompt update.

As another essential role of the visual prompt, the processed visual prompt will be sent to the device to improve the adaptability of the device model by applying the visual prompt on input images.

Knowlegde transfer. Along with visual prompt adapted tuning, we align teacher student feature and adopt teacher model to generate pseudo label for student model. It is penalized by Knowledge transfer loss $\mathcal{L}_k$, to promote the consistency between teacher student model representation. Specifically, The teacher model will generate predictions for the reformulated images x^* as pseudo labels. Teacher model has stronger generalization ability, so the generated pseudo labels are of high quality. To further improve the pseudo labels quality, we screen those predictions with confidence higher than the threshold as the input to the student model. Overall, we transfer the knowledge from the teacher to the student by feeding selected images x^* and corresponding pseudo labels into the student model for training.

3.4. Training objectives

The teacher and student models are jointly optimized on the cloud server, using uncertain samples $x_t^i \in D_t$ returned from device. The source model is trained on source pairs of $\{x_s^i, y_s^i\}_{i=1}^{n_s} \in D_s$ where $x_s^i \in X_s$ and $y_s^i \in Y_s$ are images and labels, respectively.

For the training process of the teacher model, given the reformulated taregt images x_t^*, as the Eq. (4) shown, we use adversal alignment loss [31] and supervision loss to optimize the teacher model (including visual prompt ϕ).

$$\begin{aligned} L_{tea} = -\lambda L_d(G_d(F(x_s^*;\theta_{tea}); F(x_t^*;\theta_{tea})), y_d) \\ +L_{Det}(X(x_s^*, \theta_{tea}), y_s), \end{aligned} \tag{4}$$

Note that the first part is adversal alignment loss [31], and another one is the loss function of Faster RCNN [27] (using labeled data in source domain D_s). X represents the model which predict detection outputs. L_d represent domain classifier. F indicates the feature extractor and G_d is the domain classifier. λ is the weight of alignment loss.

For the optimization of the student model, we use pseudo labels y_t^p generated by teacher model as supervision:

$$L_{stu} = L_{Det}(X(x_t^*, \theta_{stu}), y_t^p), \tag{5}$$

The visual prompt benefited from both L_{tea} and L_{stu}. And we update the visual prompt by uncertainty-aware exponential moving average as Eq. (3) shown.

4. Experiments

In Sec.4.1, we give the details of the dataset and experiment setting. we evaluate the performance of CCA on continual changing scenarios in Sec.4.2 We give discussion on our motivations and ablation study in Sec.4.3, Sec.4.4.

4.1. Experimental setup

4.1.1 Datasets and settings

We evaluate our proposed method on two test time adaptation benchmark tasks for object detection: Cityscapes-to-Cityscape-C and Cityscapes-to-ACDC-Detection dataset.

Cityscape-C is originally created to benchmark robustness tasks [14]. We select the five most relevant corruptions to the autonomous driving scenario (including brightness, motion_blur, rain, fog and snow). Each corruption dataset contains five levels of severity. The corruptions are applied to images from the validation set of the clean Cityscapes dataset [5]. There are 2,500 images in five corruption datasets.

For Cityscapes-to-Cityscapes-C, we follow [3] to train the pre-trained model on the source domain (Cityscapes). Specifically, we use the SGD optimizer with a learning rate of 1e-2 as the official implementation to train the source model.

ACDC-Detection is created from the Adverse Conditions Dataset (ACDC) [28]. The ACDC dataset shares the same classes with Cityscapes and is collected in various adverse visual conditions: Fog, Night, Rain and Snow. We convert the segmented labels into labels for object detection. We follow CoTTA [35] to set our task. We use 400 unlabeled images from each adverse condition for the adaptation. To mimic the scenario in real life where a similar environment might be revisited and to evaluate the continuous generalization capability of our methods, we repeat the same sequence group (of the three conditions) 10 times (i.e., in total 30: Fog → Rain → Snow → Fog ...). This also provides an evaluation of the adaptation performance in the long term.

For the Cityscapes-to-ACDC-Detection task, we use the same way to train the pre-trained model on the source domain (Cityscapes) as the Cityscapes-to-Cityscapes-C task.

4.1.2 Implementation Details

In this paper, all experiments are conducted with PyTorch. For the Cityscapes-to-Cityscapes-C task and Cityscapes-to-ACDC-Detection task, the batch size is set to 8. We follow [3] to train the pre-trained model on the source domain (Cityscapes). The number of samples used for uncertainty estimation is set to 10 in the experiment.

We use FasterRCNN with the backbone of ResNet-101 as the large teacher model on the cloud. And we use FasterRCNN with the backbone of ResNet-18 as the lightweight student model (same as the device model). The above large and light models are initialized with corresponding pre-trained models on the source domain (Cityscapes dataset). For the UGS module, we set the uncertainty threshold to 0.008, which filters out 42% samples in the Cityscape-C dataset. For the VPA module, We set the visual prompts with size 3x200x200, which can be added to the left-up space of the image. The student model's parameters and visual prompts (about 0.4% of the student model's parameters) will be transmitted to the device.

We use mAP@0.5(%) as the evaluation metric. The

Table 1. **Continual generalization capability on Cityscapes-to-Cityscapes-C.** 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 the four test conditions continually for ten times to evaluate the long-term adaptation performance. All results are evaluated on the FasterRCNN 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 [27]	24.7	9.5	22.4	1.3	29.0	24.7	9.5	22.4	1.3	29.0	24.7	9.5	22.4	1.3	29.0	17.4	/
TENT-continual [34]	24.8	9.2	22.5	1.4	29.4	20.0	6.4	19.8	0.8	23.7	9.0	3.8	10.7	0.5	11.4	13.0	-4.4
CoTTA [35]	25.1	10.1	22.4	1.3	28.9	21.8	9.5	20.7	4.2	28.2	9.3	3.9	10.1	3.3	17.5	14.7	-2.7
Pseudo-Label [17]	27.3	17.2	24.7	6.9	33.7	30.1	20.1	26.9	8.0	36.5	31.0	20.9	27.7	7.2	38.2	24.1	+6.7
AMS [16]	**28.4**	17.7	26.0	7.6	34.7	31.8	20.4	28.2	7.7	38.0	34.2	21.4	28.9	7.0	39.4	25.1	+7.7
Ours(proposed)	28.0	**17.8**	**26.8**	**10.0**	**36.5**	**37.5**	**24.2**	**35.5**	**14.2**	**44.0**	**44.0**	**27.3**	**40.9**	**15.1**	**50.1**	**31.0**	**+13.6**

Table 2. **Continuous generalization capability on Cityscapes-to-ACDC-Detection.** Object detection results (mAP@0.5 in %) on the Cityscapes-to-ACDC-Detection online continual test-time adaptation task. Gain(%) means the improvement of our method compared with Source-only. We evaluate the four test conditions continually for ten times to evaluate the long-term adaptation performance. All results are evaluated on the FasterRCNN architecture. Our results outperform the SOTA method, thus verifying the effectiveness of our method on the real distribution shift.

Time	$t \longrightarrow$																	
Round	1				4				7				10				All	
Condition	Fog	Night	Rain	Snow	Fog	Night	Rain	Snow	Fog	Night	Rain	Snow	Fog	Night	Rain	Snow	Mean	Gain
Source-only [27]	20.5	7.4	14.3	14.2	20.5	7.4	14.3	14.2	20.5	7.4	14.3	14.2	20.5	7.4	14.3	14.2	14.1	/
TENT-continual [34]	20.5	7.5	14.3	13.9	20.0	7.2	13.4	13.0	18.6	6.6	12.3	12.1	16.6	6.0	11.1	10.9	12.8	-2.3
CoTTA [35]	20.7	7.5	14.7	14.3	20.9	7.4	14.7	14.4	20.4	7.3	13.8	14.4	16.3	6.4	11.7	12.2	13.8	-0.3
Pseudo-Label [17]	20.9	8.7	15.5	17.6	23.4	10.3	16.9	18.9	24.3	10.9	17.8	19.5	25.1	11.5	17.5	20.0	17.6	+3.5
AMS [16]	**21.6**	9.0	**16.0**	**17.2**	23.2	10.7	17.2	19.1	24.7	11.4	17.2	19.9	26.1	11.7	17.8	20.5	17.8	+3.7
Ours(proposed)	21.1	**9.3**	15.7	**17.2**	**24.7**	**11.8**	**17.8**	**20.2**	**26.3**	**12.0**	**18.5**	**20.6**	**26.1**	**12.3**	**18.0**	**20.8**	**18.5**	**+4.4**

overall process of test-time adaptation strictly follows CoTTA [35]. The process first tests the coming batch of data then train with it. We do batch-level updates on the cloud model, and the performance on the device model is taken as the average of the results of all the data.

Baselines. Our method is compared with multiple baselines, including those for test time adaptation (TTA), continual test time adaptation (CTTA) and cloud-end collaboration methods as follows: *Source-only [27]* . Test the model performance on the different target domians using the pre-trained source model (training on the source domain). *Pseudo-label [17].* Use pseudo label generated by the teacher model to supervise the student model. *Tent-continual [34].* Tent is an effective method for the test-time adaptation (TTA). Tent fine-tune the batch normalization layer during training to adapt the model to the target domain. *CoTTA [35].* CoTTA is an attractive approach for the continual test-time adaptation (CTTA). CoTTA mitigate error accumulation and catastrophic forgetting with weighted averaging, data augmented averaging, and random recovery neurons. *AMS [16].* AMS is an cloud engagement approach. To solve the resource limitation of the device model, AMS do distillation in the cloud, and then transmit the model's parameters to the device in the cloud, the teacher network will not update.

4.2. Results and Analysis

Synthetic continual distribution shift. We evaluate our proposed method on the synthetic continual distribution shift dataset (Cityscapes → Cityscapes-C task). As shown in Table 1, TENT-continual [34] has a decreasing performance when facing continuously changing domains. Because TENT does not consider continuous distribution shifts, it uses self-entropy to update the bn layer, which is prone to catastrophic forgetting when facing constant distribution shifts. CoTTA [35] is a method designed specifically for continuous test time adaptation. CoTTA applies weighted averaging, data-augmented averaging, and random recovery neurons to overcome catastrophic forgetting and error accumulation. However, when doing the object detection task on the continuously changing domains, the model's performance still decreases, indicating that solutions used by CoTTA are insufficient against catastrophic forgetting. We believe this is caused by the poor generalization abilities of the lightweight device model. Therefore, it is reasonable to introduce the cloud for the Cloud-Device collaborative continual adaptation paradigm.

Both Pseudo-Label and AMS [16] use the Cloud-Device collaboration framework, which transfers the teacher's knowledge to the student on the cloud. The generalization of the large model provides continuous guidance for the student model, which avoids the problem of catastrophic for-

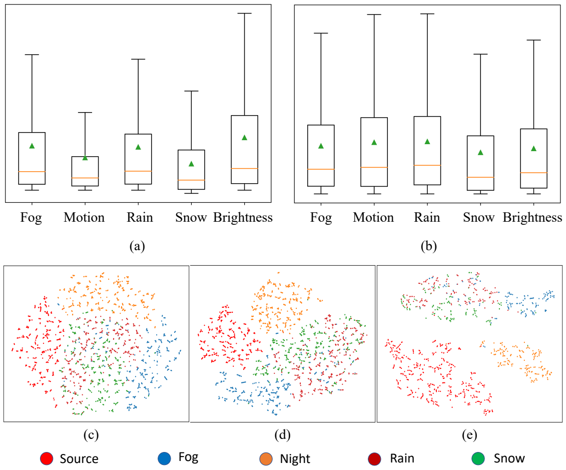


Figure 3. **(a)-(b): Statistical distribution of confidence and uncertainty.** (a): distribution of confidence. (b): distribution of uncertainty. Comparing (a) and (b), the uncertainty value is more stable when facing continuous distribution shift, while the confidence value fluctuates significantly with continuous distribution shift. **(c)-(e): T-SNE [32] visualization of samples in different domains.** (c): all samples. (d): samples filtered by confidence. (e): samples filtered by UGS. Samples filtered by UGS are more environment-specific. They are more worthy of being transmitted to the cloud server for optimizing the teacher-student models.

getting and error accumulation. (Pseudo-Label and AMS are 6.7% and 7.7% higher than Source-only).

However, pseudo label and AMS do not consider updating the teacher model. Our U-VPA framework jointly optimizes the teacher and student model, which makes our method surpass Source-only by 13.6%, outperforming all the above methods. This validates that our Cloud-Device collaboration paradigm can effectively improve the continuous domain adaptation ability of the device.

Real-world continual distribution shift. We also evaluate our proposed method on the real-world continual distribution shift dataset (Cityscapes → ACDC-Detection task). As shown in Table 2, our method improves by 4.4% over the source-only method. By comparing the performance change of the same domain in 10 rounds, we can find that our method does not have the problem of catastrophic forgetting. For example, the TENT-continual method faces the problem of catastrophic forgetting. The mAP(%) drops in the fourth, seventh, and tenth rounds compared to the first round when facing the night domain, specifically -0.2%, -0.5%, -4.1%. In contrast, our method leads to a continuous performance improvement (e.g., our method's mAP(%) increases in the fourth, seventh, and tenth rounds, respectively, compared to the first round when facing the Night domain, specifically, +2.5%, +0.2%, +0.3%).

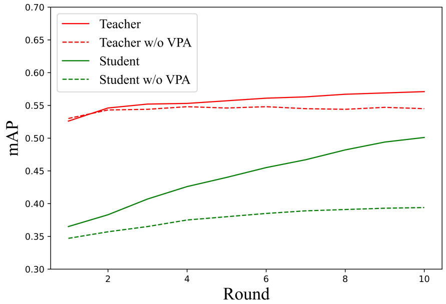


Figure 4. **Detection results (mAP@0.5) of both teacher and student model.** Solid lines are results of models using VPA under the brightness domain. Teacher and student models have jointly improved in continuous domain adaptation. Dotted lines are results without using VPA. The teacher quickly reaches the bottleneck, and student's improvement is slow. The analysis above illustrates that the proposed U-VPA framework enables the teacher-student model jointly improve the continuous domain adaption capability.

4.3. Discussion

Teacher-student joint optimization with VPA. Previous approaches (e.g., AMS) focus on improving the student model. However, the student model's performance is correlated with the teacher model. Our method enables the teacher and student models to improve performance together by using the visual prompt as a bridge and using the temporal domain information brought by the visual prompt.

The optimization of the teacher model comes from two modules: (1) feature alignment, (2) visual prompt adaptation(VPA). Our experiments have proved that although both of the above can bring about gains, visual prompt adaptation is the main reason for the success of our framework.

As shown in Fig. 4, if only feature alignment is used, the performance of the teacher model reaches the bottleneck after the second round, and the student's optimization will be very slow (dotted lines in the figure).

While our method benefits from VPA, the teacher and student models are jointly improved, demonstrating that our proposed U-VPA teacher-student model can optimize the student-teacher model together.

Uncertainty Guided Sampling (UGS) strategy. We demonstrate in Table 5 that using only a small amount of reflowed data can outperform the results using total data by the UGS strategy. Our model's performance using the UGS strategy to select part of the data to transmit into the cloud for training is 31%. Compared to the model performance of 28.3% obtained with the total data, there is a 2.7% improvement. The amount of data using the random picking strategy is the same as that with the UGS strategy. And the UGS strategy has a better performance than random picking. The above shows the effectiveness of the UGS strategy.

We further reveal the advantages of uncertainty when the

Table 3. **Ablation study.** We conduct ablation studies for proposed Pseudo-Label, VPA, UGS, and U-EMA. All the experiments are done on Cityscapes-to-Cityscapes-C, and the evaluation metric is mAP (%). The experimental result shows that all these proposed modules positively impact the performance improvement of the devices model when facing continuous domains.

	Pseudo-Label	VPA	UGS	U-EMA	Fog	Motion	Rain	Snow	Brightness	Mean	Gain
1					24.7	9.5	22.4	1.3	29.0	17.4	/
2	✓				31.0	20.9	27.7	7.2	38.2	24.1	+6.7
3	✓	✓			37.1	22.2	32.7	11.2	43.1	27.4	+10.0
4	✓	✓	✓		38.3	24.8	36.5	12.3	46.7	28.3	+10.9
5	✓	✓	✓	✓	44.0	27.3	40.9	15.1	50.1	31.0	+13.6

Table 4. **Comparing the difference between selecting samples with confidence and uncertainty.** Experiments show that better performance is obtained by choosing the reflowed data model through uncertainty estimation.

	Fog	Motion	Rain	Snow	Brightness	Mean
Confidence	39.6	24.9	37.2	14.3	45.1	27.4
Uncertainty	44.0	27.3	40.9	15.1	50.1	31.0

Table 5. **Effect of uplink transmission of full and partial data on model's performance.** Our method returns partial data (4̃2%) for training but achieves better performance than using full data. This indicates that our UGS strategy reduces the data transmission communication cost and improves the model performance.

	25%	50%	75%	100%
Random	23.5	26.1	27.9	28.3
UGS	28.2	31.0	31.2	/
Gain	+4.7	+4.9	+3.3	/

distribution shifts exist. We find uncertainty is more reliable compared with the confidence score. Regarding Fig.3 (a)-(b), in the presence of distribution shifts, confidence distribution varies significantly, while uncertainty distribution is balanced. Therefore, the samples selected using confidence will face the problem of long-tail distribution, which is not conducive to cloud model training [36]. Uncertainty can more accurately reflect the difficulty of identifying samples by device without using the training of cloud model and discover the samples that are more worthy of attention.

Besides, we explore the properties of the data filtered by different filter strategies. In Fig.3 (c)-(e), each point represents a sample filtered by the corresponding strategy. We find that samples selected by uncertainty are environmental-specific, so using UGS can filter out more valuable data.

4.4. Ablation study

Effect of the Uncertainty Guided Sampling (UGS). Our proposed UGS strategy not only reduces the communication cost of the uplink from the device to the cloud but also improves performance by 0.9%, as shown in Table 3. Furthermore, as shown in Table 5, the model using UGS outperforms baseline with only 42% data for train, which significantly improves the compression ratio of the video.

Effect of the Visual Prompt Adapt (VPA). Our proposed VPA teacher-student model uses visual prompts to make the student-teacher model jointly improve performance. As shown in Table 3, compared to the pure pseudo label method, the usage of VPA can increase performance by 3.3%. Furthermore, we discussed in Sec. 4.3 that VPA could significantly increase the upper limit of the teacher-student framework from challenging continuously changing distribution shift scenarios. The combination of VPA and other modules can further improve performance steadily.

Effect of the Uncertainty-aware prompt EMA (U-EMA) update mechanism. The U-EMA update mechanism we proposed in training the cloud model makes the cloud model pay more attention to the images worthy of attention by dynamically adjusting the prompt weights with uncertainty. As shown in Table 3, U-EMA can bring a 2.7% performance improvement.

5. Conclusion & Limitations

We propose a Cloud-Device Collaborative Continual Adaptation paradigm to deal with continual changing environments on devices. We design an Uncertainty Guided Sampling (UGS) strategy to transmit the most out-of-distribution samples from device to cloud. Besides, we design a Visual Prompt Learning Strategy with Uncertanty Guided Updating (VPLU) to transfer the generalization capability of large model on the cloud to the device model. Experiments of our method on a wide range of continually changing environments show an improvement of 4.4–13.6% in mAP@0.5 on the object detection task. Our proposed method offers flexibility in its application to most existing cloud-device systems. However, it is worth noting that our experiments were conducted using simulated datasets rather than real systems. Moving forward, an interesting direction for future research would be to test our method on a real system with specific bandwidth limitations, a specific cloud, and a specific device. By taking into account the hardware characteristics of the system, there should have rooms to improve its performance further.

Acknowledgement. This research was partly supported by the foundation of the National Key R&D Program of China (2022ZD0116305). We thank Xiaoqi Li for the insightful discussions and for helping with writing.

References

[1] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring visual prompts for adapting large-scale models. 2022. 2

[2] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie Chen, Kathleen Creel, Jared Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Ahmad Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf H. Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, and William Yang Wang. On the opportunities and risks of foundation models. *arXiv: Learning*, 2021. 2

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5

[4] Sandeep Chinchali, Apoorva Sharma, James Harrison, Amine Elhafsi, Daniel Kang, Evgenya Pergament, Eyal Cidon, Sachin Katti, and Marco Pavone. Network offloading policies for cloud robotics: a learning-based approach. *Autonomous Robots*, 2019. 2

[5] 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 *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[6] Daniel Crankshaw, Xin Wang, Giulio Zhou, Michael J. Franklin, Joseph E. Gonzalez, and Ion Stoica. Clipper: A low-latency online prediction serving system. *networked systems design and implementation*, 2016. 2

[7] Sadjad Fouladi, John Emmons, Emre Orbay, Catherine Wu, Riad S. Wahby, and Keith Winstein. Salsify: Low-latency network video through tighter integration between a video codec and a transport protocol. *networked systems design and implementation*, 2018. 2

[8] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *international conference on machine learning*, 2015. 4

[9] 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, 2023. 2

[10] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. *international conference on machine learning*, 2015. 4

[11] Amir Ghodrati, Babak Ehteshami Bejnordi, and Amirhossein Habibian. Frameexit: Conditional early exiting for efficient video recognition. *arXiv: Computer Vision and Pattern Recognition*, 2021. 2

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. *international conference on machine learning*, 2017. 4

[13] Amirhossein Habibian, Davide Abati, Taco S. Cohen, and Babak Ehteshami Bejnordi. Skip-convolutions for efficient video processing. 2021. 2

[14] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019. 5

[15] Yiping Kang, Johann Hauswald, Cao Gao, Austin Rovinski, Trevor Mudge, Jason Mars, and Lingjia Tang. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. *architectural support for programming languages and operating systems*, 2017. 2

[16] 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, 3, 6

[17] Dong-Hyun Lee. Pseudo-label : The simple and efficient semi-supervised learning method for deep neural networks. 2022. 6

[18] Rui Li, Qianfen Jiao, Wenming Cao, Hau-San Wong, and Si Wu. Model adaptation: Unsupervised domain adaptation without source data. *computer vision and pattern recognition*, 2020. 2

[19] Xiaoqi Li, Jiaming Liu, Shizun Wang, Cheng Lyu, Ming Lu, Yurong Chen, Anbang Yao, Yandong Guo, and Shanghang Zhang. Efficient meta-tuning for content-aware neural video delivery, 2022. 2

[20] Jian Liang, Dapeng Hu, and Jiashi Feng. Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. *international conference on machine learning*, 2020. 2

[21] Ser-Nam Lim, Bharath Hariharan, Luming Tang, Menglin Jia, Bor-Chun Chen, Claire Cardie, and Serge Belongie. Visual prompt tuning. 2022. 2

[22] Jiaming Liu, Ming Lu, Kaixin Chen, Xiaoqi Li, Shizun Wang, Zhaoqing Wang, Enhua Wu, Yurong Chen, Chuang Zhang, and Ming Wu. Overfitting the data: Compact neural video delivery via content-aware feature modulation, 2021. 2

[23] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv: Computation and Language*, 2021. 4

[24] Yuang Liu, Wei Zhang, and Jun Wang. Source-free domain adaptation for semantic segmentation. *computer vision and pattern recognition*, 2021.

[25] Robert A. Marsden, Felix Wiewel, Mario Döbler, Yang Yang, and Bin Yang. Continual unsupervised domain adaptation for semantic segmentation using a class-specific transfer. 2022. 2

[26] Zhongnan Qu, Cong Liu, and Lothar Thiele. Deep partial updating: Towards communication efficient updating for on-device inference. 2020. 2

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015. 5, 6

[28] 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 (ICCV)*, October 2021. 5

[29] Christian Simon, Masoud Faraki, Yi-Hsuan Tsai, Xiang Yu, Samuel Schulter, Yumin Suh, Mehrtash Harandi, and Manmohan Chandraker. On generalizing beyond domains in cross-domain continual learning. 2022. 2

[30] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Learning*, 2017. 4

[31] Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. *computer vision and pattern recognition*, 2017. 5

[32] Laurens Van Der Maaten. Barnes-hut-sne. *arXiv preprint arXiv:1301.3342*, 2013. 7

[33] 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

[34] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *Learning*, 2021. 1, 2, 6

[35] 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. 1, 2, 5, 6

[36] Tong Wang, Yousong Zhu, Chaoyang Zhao, Wei Zeng, Jinqiao Wang, and Ming Tang. Adaptive class suppression loss for long-tail object detection. *computer vision and pattern recognition*, 2021. 8

[37] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. 2022. 2

[38] Jiangchao Yao, Feng Wang, Kunyang Jia, Bo Han, Jingren Zhou, and Hongxia Yang. Device-cloud collaborative learning for recommendation. *knowledge discovery and data mining*, 2021. 3

[39] Chongzhi Zhang, Mingyuan Zhang, Shanghang Zhang, Daisheng Jin, Qiang Zhou, Zhongang Cai, Haiyu Zhao, Shuai Yi, Xianglong Liu, and Ziwei Liu. Delving deep into the generalization of vision transformers under distribution shifts. *arXiv: Computer Vision and Pattern Recognition*, 2021. 2

[40] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Conditional prompt learning for vision-language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[41] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision (IJCV)*, 2022. 2