Backpropagation-Free Multi-modal On-Device Model Adaptation via Cloud-Device Collaboration

WEI JI*, National University of Singapore, Singapore

LI LI*, National University of Singapore, Singapore

ZHEQI LV*, Zhejiang University, China

WENQIAO ZHANG, Zhejiang University, China

MENGZE LI, Zhejiang University, China

ZHEN WAN, Fudan University, China

WENQIANG LEI, Sichuan University, China

ROGER ZIMMERMANN, National University of Singapore, Singapore

In our increasingly interconnected world, where intelligent devices continually amass copious personalized multi-modal data, a pressing need arises to deliver high-quality, personalized device-aware services. However, this endeavor presents a multifaceted challenge to prevailing artificial intelligence (AI) systems primarily rooted in the cloud. As these systems grapple with shifting data distributions between the cloud and devices, the traditional approach of fine-tuning-based adaptation (FTA) exists the following issues: the costly and time-consuming data annotation required by FTA and the looming risk of model overfitting. To surmount these challenges, we introduce a Universal On-Device Multi-modal Model Adaptation Framework, revolutionizing on-device model adaptation by striking a balance between efficiency and effectiveness. The framework features the Fast Domain Adaptor (FDA) hosted in the cloud, providing tailored parameters for the Lightweight Multi-modal Model on devices. To enhance adaptability across multi-modal tasks, the AnchorFrame Distribution Reasoner (ADR) minimizes communication costs. Our contributions, encapsulated in the Cloud-Device Collaboration Multi-modal Parameter Generation (CDC-MMPG) framework, represent a pioneering solution for on-Device Multi-modal Model Adaptation (DMMA). Extensive experiments validate the efficiency and effectiveness of our method, particularly in video question answering and retrieval tasks, driving forward the integration of intelligent devices into our daily lives.

CCS Concepts: • **Information systems → Mobile information processing systems**; **Personalization**; • **Human-centered computing → Mobile computing**.

Additional Key Words and Phrases: Cloud-device collaboration, model adaptation, multi-modal

---

*These authors contributed equally to this research.

Authors' addresses: Wei Ji, National University of Singapore, Singapore, jiwei@nus.edu.sg; Li Li, National University of Singapore, Singapore, lili02@u.nus.edu; Zheqi Lv, Zhejiang University, China, zheqilv@zju.edu.cn; Wenqiao Zhang, Zhejiang University, China, wenqiaozhang@zju.edu.cn; Mengze Li, Zhejiang University, China, mengzeli@zju.edu.cn; Zhen Wan, Fudan University, China, wz2311602492@gmail.com; Wenqiang Lei, Sichuan University, China, wenqianglei@gmail.com; Roger Zimmermann, National University of Singapore, Singapore, dcsrz@nus.edu.sg.
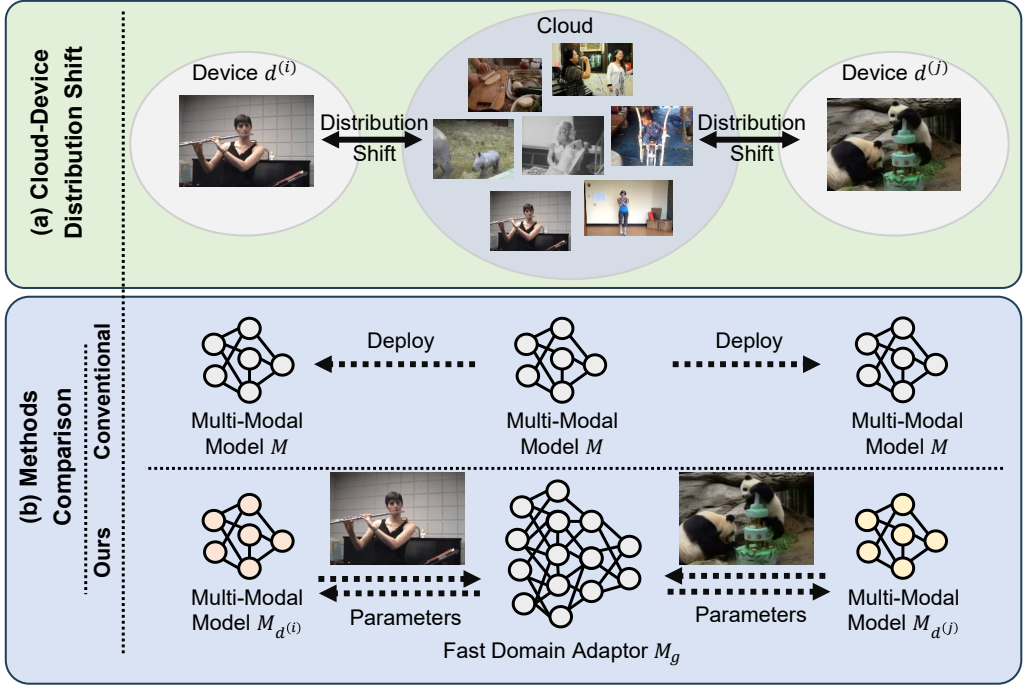

111

Fig. 1. (a) Multi-modal data on cloud and different devices exist in different distributions due to the personalized preference of users. (b) Compared with conventional methods of deploying models on different devices, we propose an FDA that can achieve a balance of efficiency and effectiveness.

1 INTRODUCTION

In today's interconnected world, the proliferation of intelligent devices, ranging from ubiquitous smartphones to the ever-expanding Internet of Things (IoT) ecosystem, has become an integral part of our daily lives. These devices serve as data collection powerhouses, continuously amassing vast repositories of personalized multi-modal data, which can include a wide array of input modalities such as text, images and videos. The potential locked within this trove of multi-modal data arriving continuously is immense, promising to unlock high-quality and tailored device-aware services for individual users. Despite promising, the personalized device service involves analyzing the dynamic nature of the multi-modal data that underscore users' intentions. The prevailing artificial intelligence (AI) systems, primarily trained and deployed in cloud-based environments, face a profound challenge in adapting to the dynamic device data when using a static cloud model for all individual users, mainly due to the distribution shift of the cloud and device data, as shown in Figure 1. In other words, high-quality personalized service requires AI systems to undergo continual refinement and adaptation to accommodate the evolving landscape of personalized multi-modal data.

Intuitively, one of the straightforward adaptation strategies is to fine-tune the cloud model based on the device's multi-modal data, which can kindly alleviate the cloud-device data distribution shift to model users' intentions. Nevertheless, we contend that the fine-tuning-adaptation (FTA) paradigm may not satisfactorily resolve device model personalization, which can be summarized as two key aspects: (1) **Undesirable Annotation.** FTA often necessitates manually annotating data to guide

model adaptation, which typically hinges on expensive and labor-intensive device data labeling. Additionally, this retraining process can result in substantial delays, hindering the AI system's ability to deliver real-time, context-aware responsiveness. This situation is further exacerbated by the inherent complexity of multi-modal data understanding, *i.e.*, with data streams that encompass textual, visual, and auditory information, the intricacies of labeling and retraining are more time-consuming, resulting in a higher time delay and thus diminishing its practicality for users. (2) **Overfitting Risk.** In real-world applications, the majority of devices could be characterized by sparse and specialized multi-modal data where model fine-tuning may inadvertently lead to overfitting issues, even leading to device model performance degradation. In other words, FTA imposes significant demands for on-device data quantity, hampering its ability to generalize effectively across diverse device ecosystems. Based on the aforementioned insights, a meaningful optimization goal of device model personalization is to appropriately tap onto the personalized multi-modal data, and thus strike the delicate balance between the effectiveness and efficiency of personalized adaptation.

To address these multifaceted challenges and pave the way for intelligent device-driven AI systems that harmonize adaptability and computational efficiency, we present a universal device multi-modal model adaptation framework, a groundbreaking approach that redefines the landscape of on-device AI adaptation. This framework is designed to revolutionize the way AI systems harness multi-modal data, simultaneously unlocking efficiency and effectiveness in the adaptation process. In our approach, we introduced a Fast Domain Adaptor (FDA) hosted on the cloud, which takes device-captured images as input and produces customized parameters for the Lightweight Multi-modal Model on the device. These customized parameters are tailored to different data distributions present on the device. Moreover, the FDA adapts to various multi-modal tasks by requiring distinct data inputs. However, certain multi-modal tasks, like Video QA, necessitate transmitting extensive data such as multiple frames or the entire video to FDA, incurring substantial communication costs and bandwidth requirements. To mitigate this challenge and enhance FDA's adaptability across different multi-modal tasks, we developed the AnchorFrame Distribution Reasoner (ADR). ADR standardizes the input for the FDA across various multi-modal tasks. For instance, in the case of Video QA, ADR selects the first frame of each video as the AnchorFrame. Using a combination of Variation AutoEncoder (VAE) and KeyFrame, ADR maps the AnchorFrame to a standard distribution through training. Once both FDA and ADR are trained, they are deployed on the cloud to deliver personalized model parameter services for lightweight multi-modal models on the device. The device's model can then upload the AnchorFrame to obtain personalized model parameters tailored to the data distribution of the current image or video input. An important highlight of our framework is the absence of backpropagation during the domain adaptation process for the device model. This unique approach enables our framework to achieve rapid domain adaptation of the device model, resulting in exceptionally low latency.

Summing up, our contributions are summarized below:

(1) We propose a **C**loud-**D**evice **C**ollaboration **M**ulti-**M**odal **P**arameter **G**eneration (CDC-MMPG) framework to accomplish efficient on-**D**evice **M**ulti-modal **M**odel **A**daptation (DMMA). Its core is Fast Domain Adaptor (FDA) which can generate multi-modal equipment model parameters based on the data distribution of multi-modal data. To the best of our knowledge, we are the first ones to solve the DMMA. Our proposed CDC-MMPG framework is a universal one that can be adaptive to other modalities.

(2) We propose the AnchorFrame Distribution Reasoner (ADR) to further reduce the communication cost of the CDC-MMPG. ADR successfully addresses the issue of high bandwidth dependence while maintaining nearly consistent performance.

(3) Extensive experiments on video question answering and video retrieval tasks have verified the efficiency and effectiveness of our proposed method.

2 RELATED WORKS

Cloud-Device Collaboration. In the rapidly evolving sphere of cloud-device collaboration, deep learning has become a crucial component, strategically combining the features of cloud-based and on-device machine learning. Although Federated Learning (FL), notably represented by FedAVG [5], has traditionally been a significant player, its perceived simplicity has limited its applicability in various practical situations. This dynamic field has seen the rise of innovative methods and techniques designed to optimize the interaction between cloud resources and mobile devices. For instance, MPDA [41] skillfully utilizes cloud-based samples to enhance the performance of on-device models. Another interesting approach involves deploying multiple models with similar functionalities, coordinated by a Meta Controller for optimal task allocation, thus expanding the possibilities of collaborative model adaptation [3, 49]. Moreover, DUET [23], inspired by the HyperNetwork concept [26, 38, 47, 50], simplifies the device model adaptation process, eliminating the need for on-device training [55]. IDEAL [22] expands parameter generation-based models to the domain of recommender systems, particularly DC-CDR [52]. This enhances the generalization capabilities of device recommendation models, though it requires consideration of factors such as request frequency and communication revenue. However, the above works are tailored for recommender system, which can not be directly migrated to the multi-modal domain field, mainly due to that they ignore the inherent complexity of multi-modal data. In this paper, we present CDC-MMPG, tailored for multi-modal on-device model adaptation via cloud-device collaboration, which simultaneously obtained effectiveness and efficiency.

Domain Adaptation. Domain Adaptation (DA) plays a crucial role in transferring a network that was initially trained on a labeled source domain to effectively perform on a target domain. Various techniques have been developed to align the feature distributions between the source and target domains, such as maximum mean discrepancy and correlation alignment. These methods aim to reduce the distribution discrepancy and enable the network to generalize well across different domains, improving its performance in the target domain. Lots of methods are proposed to facilitate the development of the domain adaptation, such as multi-stage methods [4, 12, 14, 15, 53], gradual transfer strategies [2, 16, 31, 33], and so on. Recent advancements in domain adaptation embrace curriculum-based strategies inspired by curriculum learning to enhance the effectiveness of the adaptation process [13, 27, 29], Source-free domain adaptation (SFDA) has emerged as a response to privacy and copyright concerns, where access to the source domain data is restricted or unavailable [2, 4, 8, 10, 18, 20, 27, 29, 30, 35, 44, 53]. SFDA presents a more challenging variant known as Test-Time Adaptation (TTA), which necessitates online adaptation during the inference phase. In addition, many researchers introduce the domain adaptation setting into different tasks to enhance model generalization [1, 6, 19, 24, 25, 28, 36, 37, 42, 43, 48, 51, 54]. In summary, domain adaptation (DA) methods focus on aligning feature distributions between the source and target domains. Recent advancements have highlighted the importance of curriculum-based strategies to improve the adaptation process. [2, 4, 27, 29, 46, 53]. In this paper, we propose a model generation framework to solve the personalized adaption problem.

3 METHODOLOGY

In this section, we provide a detailed description of our proposed **C**loud-**D**evice **C**ollaboration **M**ulti-**M**odal **P**arameter **G**eneration (CDC-MMPG) framework, which includes **F**ast **D**omain **A**daptor (FDA) and **A**nchorFrame **D**istribution **R**easoner (ADR).
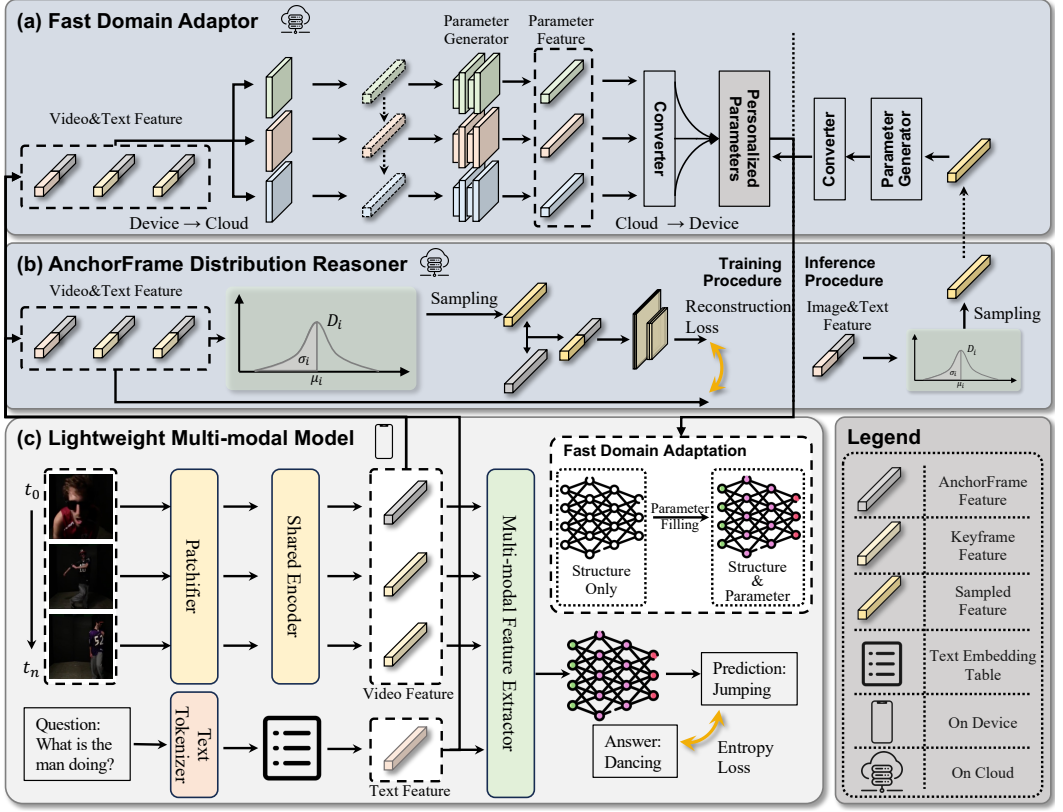
Fig. 2. Illustration of the overall pipeline of our method, CDC-MMPG. (a) and (b) represent the Cloud model, which reconstructs the video features uploaded from the device and reasons out the personal parameters of the device model based on the reconstructed video features. (c) represents the lightweight multi-modal device-side model, which extracts the multi-modal features, and uploads the video features to the cloud model for the personal device-model parameter prediction. After being updated with the personal parameters, the lightweight multi-modal device-side model will further analyze the multi-modal features and make the final prediction.

3.1 Preliminary

Problem Formulation. For the on-Device Multi-modal Model Adaptation (DMMA) in the device-cloud collaboration system, we have access to a set of devices $\mathcal{D} = \{d^{(i)}\}_{i=1}^{\mathcal{N}_d}$, each device with its personal i.i.d multi-modal history samples $\mathcal{S}_{H^{(i)}}$ and multi-modal real-time samples $\mathcal{S}_{R^{(i)}}$ in current session, where $\mathcal{N}_d$ represents the number of devices. The goal of on-Device Multi-modal Model Adaptation is to generalize a trained cloud model $\mathcal{M}_g(\cdot; \Theta_g)$ learned from $\{\mathcal{S}_{H^{(i)}}\}_{j=1}^{\mathcal{N}_d}$ to each specific local device model $\mathcal{M}_{d^{(i)}}(\cdot; \Theta_{d^{(i)}})$ conditioned on real-time samples $\mathcal{S}_{R^{(i)}}$, where Θ_g and $\Theta_{d^{(i)}}$ respectively denote the learned parameters for the cloud model and the i-th device model:

$$\underbrace{\text{CDC-MMPG}}_{\text{DMMA Model}} : \underbrace{\mathcal{M}_g(\{\mathcal{S}_{H^{(i)}}\}_{i=1}^{\mathcal{N}_d}; \Theta_g)}_{\text{Cloud Model}} \rightarrow \underbrace{\mathcal{M}_{d^{(i)}}(\mathcal{S}_{R^{(i)}}; \Theta_{d^{(i)}})}_{\text{Device Model}}. \tag{1}$$

Figure 2 illustrates the overview of our CDC-MMPG framework which consists of two modules to improve the generalization ability of the trained models on the device: (a) *Fast Domain Adaptor*

(FDA) aims to learn a global benchmark model based on the history samples of all distributions and generate the network parameters for the distribution-specific device model based on the real-time device samples (in Sec. 3.2); (b) *AnchorFrame Distribution Reasoner* (ADR) standardizes the input for the FDA across various multi-model samples. (in Sec. 3.3).

Model Pipeline. As shown in Algorithm 1, there are three steps in the pipeline of the CDC-MMPG model: (1) Training the cloud model $\mathcal{M}_g(.)$, including the FDA module and the ADR module, with the history samples $\mathcal{S}_H$. (2) Uploading the real-time samples $\mathcal{S}_{R(i)}$ from the device side to the cloud side. Then, the cloud side model $\mathcal{M}_g(.)$ generates the personalized parameters for the device model $\mathcal{M}_{d(i)}(.)$. (3) With the model parameters passed from the cloud side, the device model $\mathcal{M}_{d(i)}(.)$ is updated and makes the final prediction based on the input read-time samples.

Multi-modal Feature Extraction. We extract the multi-modal representation needed for the subsequent processes from the input examples. Specifically, given a real-time input video V_r and a corresponding language query Q_r, we employ the DeiT [32] as the visual feature extractor $f_v(.)$ and the BERT-base language model [7] as the text encoder $f_t(.)$, to extract the features of the input sample.

$$F_v = f_v(V_r), \tag{2}$$
$$F_t = f_t(Q_r), \tag{3}$$

where $F_v = \{F_v^i\}_{i=1}^{N_f}$ represents the features of all video frames and N_f is the video frame number. F_t is the language query feature.

We adopt the spatial-temporal positional embedding and modality type embedding following [34] to the extracted features F_v and F_t and get the corresponding embedded features $E_v = \{E_v^i\}_{i=1}^{N_f}$ and E_t. Then, the cross-modal fusion module $g(.)$ consisted of t transformer layers is employed to fuse the visual features E_v and the language feature E_t:

$$F_m = g(E_v, E_t), \tag{4}$$

where $F_m = \{F_m^i\}_{i=1}^{N_f}$ is the generated multi-modal features after the feature fusion.

3.2 Fast Domain Adaptor

When adapting the cloud model to the personalized device real-time samples of specific distribution, one straightforward strategy is fine-tuning the cloud model on the device. However, it may be unreachable due to a lack of sufficient training samples and corresponding annotations on the device. To tackle the aforementioned challenge, we propose a novel solution called the Fast Domain Adaptor (FDA), which is implemented as a cloud-based service. The FDA is designed to process device-captured images as input and generate customized parameters specifically tailored to the Lightweight Multi-modal Model deployed on the device. These customized parameters are carefully crafted to adapt to the unique data distributions encountered on each individual device.

Specifically, we further sample $D(D > 1)$ frames from F_m randomly, upload the sampled multi-modal features from the device side to the cloud side, and average the included D frames of multi-modal features into the global representation F_g. Then, we project the feature F_g before it is further analyzed:

$$E_g = f_p(F_g), \tag{5}$$

where $f_p(.)$ represents the multi-layer perceptron consisting of two linear layers and a normalization layer. The output feature E_g is treated as the real-time sample embedding for hyper-network [9, 11] input:

$$\Theta_d = f_h(E_g), \tag{6}$$

where $f_h(.)$ is the multi-layer perceptron. The generated Θ_d is the parameters for a single linear layer including both the linear weights and bias, which will be passed to the device model.

The generated model parameters Θ_d are passed from the cloud side to the device side. The device model $\mathcal{M}_d$ is updated with the model parameters Θ_d and makes the prediction P_n for the input multi-model real-time sample:

$$P_n = \mathcal{M}_d(F_m; \Theta_d), \tag{7}$$

where F_m is the extracted multi-modal feature.

3.3 AnchorFrame Distribution Reasoner

The FDA is designed to adapt to diverse multi-modal tasks by utilizing specific data inputs. However, certain tasks, such as Video QA, require transmitting significant amounts of data, such as multiple frames or the entire video, to the FDA. This results in considerable communication costs and bandwidth requirements. To address this challenge and improve the adaptability of the FDA across various multi-modal tasks, we introduce the AnchorFrame Distribution Reasoner (ADR). ADR plays a crucial role in standardizing the input format for the FDA, ensuring compatibility across different multi-modal tasks.

Specifically, during the inference process, we randomly select one frame from the whole video on the device side and upload the multi-modal feature F_m^i corresponding to the selected i-th frame to the cloud side. Then, the **V**ariational **A**uto-**E**ncoder (VAE) including an encoder Ψ_E and a decoder Ψ_D are applied to further analyze. Firstly, we obtain the latent distribution by analyzing the uploaded feature F_m^i with the VAE encoder Ψ_E:

$$\Psi_E(F_m^i) \sim \mathcal{N}(\mu^{real}, \sigma^{real}), \tag{8}$$

where $\mathcal{N}(\mu^{real}, \sigma^{real})$ represents the Gaussian distribution with mean μ^{real} and standard deviation σ^{real}. Then we sample a new representation R_m^i from the obtained specific distribution:

$$R_m^i \sim \mathcal{N}(\mu^{real}, \sigma^{real}). \tag{9}$$

The R_m^i is then reasoned on by the VAE decoder Ψ_D:

$$F_m^{i\,\prime} = \Psi_D(R_m^i). \tag{10}$$

The $F_m^{i\,\prime}$ and F_m^i are utilized to adaptively generate the new feature F_g' through the adaptive generator $\Phi_a(.)$. The generated new features will be forwarded to the hyper-network with the goal of leading it to generalize to the target domain. Thus, we can update the Eq. 5 to:

$$F_g' = \Phi_a(F_m^{i\,\prime}, F_m^i). \tag{11}$$

The feature F_g' is applied to replace the global representation of the input sample F_g defined in Section 3.2 for further prediction of the personalized model parameters.

The framework uses limited communication bandwidth: one-frame multi-modal feature upload and hyper-network parameters (a few linear layers) download, promising the generalization ability and extremely low communication delay of the device model.

During the training process, all history samples are directly saved on the cloud side and participated in the training process, following the setting of previous methods [45]. Given the sampled

---

Algorithm 1: CDC-MMPG Framework

---

1 Initialize the multi-modal feature extraction model: $\mathcal{M}_{vl}$; the hyper-network model in FDA: $h(\cdot)$; the encoder and the decoder in ADR: Ψ_E, Ψ_D; the adaptive generator: Φ_a.

2 **Phase I:** Input: Historical data $\mathcal{S}_H$.

3   1) Train the ADR on the cloud using $\mathcal{S}_H$ and the training objectives in Eq. 18.

4 Output the trained models: $\mathcal{M}'_{vl}, h'(\cdot), \Psi'_E, \Psi'_D, \Phi'_a$.

5 **Phase II:** Input: Real-time data $\mathcal{S}_R$.

6   1) Extract multi-modal features and process the one-frame feature to the cloud $F^i_{cloud}, F_g \leftarrow \mathcal{M}'_{vl}(\mathcal{S}_R)$.

7   2) Obtain latent distribution of F_{cloud}: $\mathcal{N}(\mu, \sigma)$, where $\sigma, \mu \leftarrow \Psi'_E(F^i_{cloud})$.

8   3) Randomly sample from this distribution $F_{rand} \leftarrow \mathcal{N}(\mu, \sigma)$.

9   4) Decode the randomly sampled distribution $F'_{rand} \leftarrow \Psi'_D(F_{rand})$.

10   5) Get adaptive layer embedding $E_m \leftarrow \Phi_a(, F'_{rand}, F^i_{cloud})$.

11   6) Generate adaptive parameters using the layer embedding $\Theta_a \leftarrow h'(E_m)$.

12 Output the parameters: Θ_a.

13 **Phase III:** Input: Real-time data $\mathcal{S}_R$.

14   1) Download the adaptive parameters to the device $\Theta^d_a \leftarrow \Theta_a$.

15   2) Share feature backbone between the cloud and device $\mathcal{M}_d \leftarrow \mathcal{M}_{vl}$.

16   3) Make the prediction with Θ^d_a: $P_n \leftarrow \mathcal{M}_d(\mathcal{S}_R; \Theta^d_a)$

17 Output the P_n.

---

multi-modal features F_g and the randomly selected feature F^i_m, the variational auto-encoder is applied to further analyze the same as the inference process. Firstly, the encoder Ψ_E project F_m into the latent space and generate a new feature F_h:

$$F_h = \Psi_E(F_g). \tag{12}$$

Then, to reconstruct the input feature F_g, the decoder Ψ_d decodes the feature F_h to the feature F'_g:

$$F'_g = \Psi_D(F_h). \tag{13}$$

In order to train the variational auto-encoder, we use the **K**ullback-**L**eibler (KL) loss L_{kl} and reconstruction loss L_{Rec} as follows:

$$L_{kl} = \mathbb{D}_{kl}(\mathcal{N}(\mu', \sigma')||\mathcal{N}(0, I)), \tag{14}$$

$$L_{Rec} = \text{MSE}(F_g, F'_g), \tag{15}$$

where $\mathcal{N}(0, I)$ represents the normal distribution, $\mathbb{D}_{kl}(.)$ represents the Kullback-Leibler divergence calculation and $\text{MSE}(\cdot)$ denotes the mean square loss.

Inspired by [21], in order to incorporate the various distribution shifts into the feature of the single video frame, the adaptive generator generates new features F_a by re-normalizing the F'_g to have the same channel-wise mean and standard deviation as the F^i_m. This process can be formulated as follows:

$$\Phi_a(F'_g, F^i_m) = \delta(F'_g)\left(\frac{F^i_m - \delta(F^i_m)}{\gamma(F^i_m)}\right) + \gamma(F'_g), \tag{16}$$

where $\delta(\cdot)$ and $\gamma(\cdot)$ denotes channel-wise mean and standard deviation operations, respectively.

Following previous works on variational auto-encoder generalization, we additionally compute the loss values between the generated new features and F_g, F_m^i, respectively, to train the whole module:

$$L_{AG} = \lambda MSE(F_a, F_g) + MSE(F_a, F_m^i), \tag{17}$$

where λ denotes a hyper-parameter, and F_a denotes the feature generated by the adaptive generator. Thus, the overall training loss L for the whole framework is:

$$L = L_{AG} + L_{Rec} + L_{KL}. \tag{18}$$

Table 1. Results of our proposed method in Open-ended Video QA task on MSRVTT-QA and MSVD-QA. We adopt accuracy as the evaluation metric. The number of learnable parameters on the device, the number of learnable parameters on the cloud, and the time delay are additionally measured to show the efficiency of our proposed method. "F-linear" denotes only fine-tuning the classifiers after the multi-modal feature extractor. "F-hyper" denotes only fine-tuning the classifiers and a simple hyper-network without an adaptive generator.

Methods	VSRVTT-QA				MVSD-QA			
	Acc.	D-Param.	C-Param.	Time Delay	Acc.	D-Param.	C-Param.	Time Delay
F-linear	13.6	11.6M	/	60000ms	17.3	11.5M	/	60000ms
Fine-tuning	36.7	11.6M	/	60000ms	34.3	11.5M	/	60000ms
F-hyper	11.1	11.6M	28.1M	5.55ms	6.34	11.5M	18.7M	3.71ms
Ours	**37.1**	11.6M	55.9M	5.55ms	**35.4**	11.5M	46.5M	3.71ms

Table 2. Results of our proposed method in Open-ended Video QA task on TGIF. We adopt accuracy as the evaluation metric. The number of learnable parameters on the device, the number of learnable parameters on the cloud, and the time delay are additionally measured to show the efficiency of our proposed method. "F-linear" denotes only fine-tuning the classifiers after the multi-modal feature extractor. "F-hyper" denotes only fine-tuning the classifiers and a simple hyper-network without an adaptive generator.

Methods	TGIF			
	Accuracy	D-Param.	C-Param.	Time Delay
F-linear	27.3	11.6M	/	60000ms
Fine-tuning	54.7	11.6M	/	60000ms
F-hyper	19.6	11.6M	18.7M	5.70ms
Ours	**55.2**	11.6M	46.5M	5.70ms

4 EXPERIMENT

In this section, we evaluate our proposed method on three popular multi-modal tasks and four relevant datasets. Additionally, we perform ablation studies to investigate the impact of each module in our method, the influence of the number of frames sampled from the whole video, and the time delay between the cloud and the device with various internet conditions.

4.1 Datasets

We conducted our experiments on three multi-modal tasks: multiple-choice Video QA, open-ended Video QA, and video-text retrieval. For the open-ended Video QA task, we use the MSRVTT-QA dataset [39], MSVD-QA dataset [39] and the TGIF dataset [17]. For the multiple-choice Video QA task, we use the MSRVTT-QA dataset [39]. For the video-text retrieval task, we experiment on the MSRVTT dataset [40].

(1) **MSRVTT.** The MSRVTT is a large-scale dataset for the open-domain video captioning, which consists of 10K video clips and each video clip is annotated with 20 English sentences. The standard splits use 6,513 clips for training, 497 clips for validation, and 2,990 clips for testing.

(2) **MSRVTT-QA.** The MSRVTT-QA dataset contains 10K video clips and 243k question answer pairs. Specifically, we used MSRVTT-QA dataset [39] for open-ended VQA and multiple-choice VQA. Following [39], We use the traditional data split which is 65% for the training set, 5% for the validation set and 30% for test set.

(3) **MSVD-QA.** The MSVD-QA dataset has a total number of 1,970 video clips and 50,505 question answer pairs. Following [39], we split the dataset based on videos that training set takes 61%, validation set takes 13%, and test set takes 26% of the total number of videos.

(4) **TGIF.** The Tumblr GIF (TGIF) dataset contains 100K animated GIFs and 120K sentences describing visual content of the animated GIFs. One sentence is provided for every animated GIF for the training and validation splits, and three sentences per GIF for the test split. Specifically, there are 80K training samples, 11K validation samples and 11k test samples.

4.2 Evaluation Metrics

Following previous works, we adopt accuracy for the open-ended Video QA task, and VR@K, TR@K for video-text retrieval task. Specifically, VR@K denotes the recall of video to text retrieval and TR@K denotes the recall of text to video retrieval. For both of these two tasks, K is set to 1,2,5 respectively. Additionally, we calculate the number of learnable parameters in each model. Meanwhile, for the practical scenario of cloud-device collaboration, we also measure the time delay for the cloud-device communication process.

4.3 Tasks and Implementation Details

We evaluate our method on three tasks:

(1) **Video-text Retrieval:** We evaluate two sub-tasks including video-to-text retrieval and text-to-video retrieval. The text-to-video retrieval task requires retrieving the target video using

Table 3. Results of our proposed method in Text-video Retrieval task. The number of learnable parameters on the device, the number of learnable parameters on the cloud, and the time delay are additionally measured to show the efficiency of our proposed method.

Methods	MSRVTT					
	VR@1	VR@5	VR@10	TR@1	TR@5	TR@10
F-linear	2.0	7.7	14.1	3.1	10.1	16.9
Fine-tuning	4.8	17.6	27.9	**6.2**	20.7	31.8
F-hyper	2.6	8.6	14.7	2.7	11.4	17.6
Ours	**6.6**	**21.3**	**33.8**	5.6	**21.2**	**33.3**

Table 4. Ablation studies of different numbers of sampled frames. We make the evaluation on three datasets in Open-ended Video-question Answering task. "Time/Epoch" denotes the time of fine-tuning for each epoch.

Frames	MSRVTT-QA		MSVD-QA		TGIF	
	Accuracy	Time/Epoch	Accuracy	Time/Epoch	Accuracy	Time/Epoch
2	36.1	740s	34.1	204s	54.3	245s
3	36.7	897s	**34.3**	228s	54.7	310s
4	37.0	1309s	**34.3**	318s	54.7	448s
5	**37.1**	2092s	34.0	377s	**55.2**	702s

language queries. The video-to-text retrieval task requires retrieving the target text using videos.

(2) **Open-ended Video-question Answering:** It requires answering questions according to the context of the video. The answers are originally in free-form natural language, but it is a common practice to convert the task to a classification task by representing the answer with a class label.

(3) **Multiple-choice Video-question Answering:** Given a video with a query and 5 candidate captions, the task is to find the one that fits the query out of 5 possible candidates. The correct answer is the ground-truth (GT) caption, and four other negatives are chosen from other captions that have different activity-phrase labels from the correct answer.

Implementation Details. We employ the All-in-One-Ti [34] as the baseline model, which includes the DeiT [32] as its visual backbone and BERT-base [7] as its semantic encoder. Specifically, we only use the embedding layers of BERT for text embedding. The All-in-one-Ti is adopted as the multi-modal encoder both for the device model and the cloud model. The value of D is set to 3. During the training process of ADR, we employ the adamw optimizer with a polynomial decay scheduler. The learning rate is set to 2e-5 and the training epoch is set to 10. During the training process of the rest of the modules, we froze the parameters in the adaptive generator, and the same adamw optimizer is used with the 1e-4 learning rate used. The total training epoch for the rest modules is set to 40. For the hyper-parameters and other settings during the training process, λ is set to 0.1, and the hidden layer of the hyper-network is set to 96 (for Open-ended VQA and Multiple-choice VQA) and 256 (for Video-text Retrieval). During the inference process, all the parameters on the device model and the cloud model are frozen except the last few dynamic linear layers of the device model. The cloud model and the device model share the same parameters of the multi-modal encoder. Then the cloud generates dynamic parameters using hyper-network for the device model for better generalization.

4.4 Performance Comparison

In this section, we introduce how we design the baseline method to showcase the efficiency of our proposed method, and we present the experimental results of our method on various tasks and datasets.

4.4.1 Baseline Methods.

For we are the first to explore this field, we design the other three baseline methods to show the superiority of our proposed method. As shown in Tab. 1 and Tab. 2, we adopt accuracy as the evaluation metric. The number of learnable parameters on the device, the number of learnable

parameters on the cloud, and the time delay are additionally measured to show the efficiency of our proposed method. "F-linear" denotes only fine-tuning the classifiers after the multi-modal feature extractor. "F-hyper" denotes only fine-tuning the classifiers and a simple hyper-network without an adaptive generator. Besides our proposed method, we also design the fine-tuning approach, fine-tuning only the linear classifiers approach (F-linear), and fine-tuning both the linear classifiers and solely hyper-network approach. For the fine-tuning approach, we simply add layers of MLPs after the multi-modal feature extractor and do fine-tuning on the whole model. For fine-tuning only the linear classifiers approach, we froze the parameters in the multi-modal feature extractor and fine-tuned the layers of MLPs after the feature extractor. For the fine-tuning of both the linear classifiers and the solely hyper-network approach, we employ a simple hyper-network without the proposed adaptive generator and do fine-tuning for both linear classifiers and the simple hyper-network.

Open-ended Video-question Answering. For Open-ended Video-question Answering, the responses are typically expressed in unrestricted natural language. However, a prevalent approach involves transforming this task into a classification problem by encoding the answers as class labels. To achieve this, we incorporate layers of MLPs with a hidden layer dimension of 96 following the extraction of multi-modal features. The dimension of the MLP's output layer varies according to the specific label size associated with the datasets, for instance, 1501 labels for the MSRVTT-QA dataset.

Multiple-choice Video-question Answering. For Multiple-choice Video-question Answering, where both the questions and candidate answers are presented as sentences, our approach involves concatenating the question and the answer candidates. To distinguish between them, we employ the special token [SEP]. Subsequently, we determine the prediction by selecting the candidate with the highest output logit, signifying its likelihood of being the correct answer.

Video-text Retrieval. For Video-text Retrieval, the retrieval process encompasses two directions: text-to-video retrieval and video-to-text retrieval. In each direction, the respective modality is first extracted, and subsequently, a comparative analysis is conducted. The predictions are ultimately generated through the application of MLPs.

4.4.2 Main Results.

In this section, we present the experimental results of open-ended Video QA in Tab. 1 and Tab. 2, multiple-choice Video QA in Tab. 5, and Video-text Retrieval in Tab. 3 on four popular datasets.

Effectiveness. Tab. 1,Tab. 2, Tab. 5, and Tab. 3 provide a comprehensive overview of the superiority of our method compared to other baseline approaches across various datasets and evaluation criteria. Notably, our method consistently outperforms alternative baseline models in nearly all aspects across these datasets. For example, consider the task of Open-ended Video-question Answering on the MSRVTT-QA dataset. Our proposed method excels in terms of both accuracy and time efficiency when compared to all other baseline models. The conventional fine-tuning method demands a significant amount of time to complete the fine-tuning process. As demonstrated in Tab. 4, when selecting three frames from the entire video, the time required for each fine-tuning epoch on the MSRVTT dataset is approximately 897 seconds. In contrast, our method achieves nearly real-time communication, effectively reducing the time delay between the cloud and the device to as little as 5.55ms, contingent on internet conditions. Furthermore, our proposed method even surpasses the fine-tuning approach in terms of performance, indicating its superior fast generalization capabilities on personalized samples. Tab. 1 and Tab. 2, which denote the results on the MSVD-QA and TGIF datasets, it is evident that our method significantly outperforms other baseline models, thus affirming the effectiveness of our proposed approach.

Table 5. Results of our proposed CDC-MMPG framework in the multiple-choice Video QA task. The number of learnable parameters on the device, the number of learnable parameters on the cloud, and the time delay are measured to show the efficiency of our proposed method.

Methods	MSRVTT-QA			Time Delay
	Acc.	D-Param.	C-Param.	
F-linear	3.58	11.4M	/	≥ 60000ms
Fine-tuning	75.6	11.4M	/	≥ 60000ms
F-hyper	46.0	11.4M	37.2K	≥ 5.70ms
Ours	**77.0**	11.4M	20.3M	≥ 5.70ms

Table 6. Time delay of our framework in different circumstances (various internet speeds) on different datasets. Our proposed method largely reduces the time delay between the cloud and the device, realizing almost real-time communication. " ↑ " denotes the upload process from the device to cloud. " ↓ " denotes the parameters downloaded from the cloud to device.

Datasets	Size	4G: 5MB/s	4G: 15MB/s	5G: 50MB/s	5G: 100MB/s
MSRVTT	↑:0.75KB ↓:568.5KB	↑:0.15ms ↓:111ms	↑:0.05ms ↓:37.0ms	↑:0.01ms ↓:11.1ms	↑:0.007ms ↓:5.55ms
MSVD	↑:0.75KB ↓:379.4KB	↑:0.15ms ↓:74ms	↑:0.05ms ↓:24.7ms	↑:0.01ms ↓:7.41ms	↑:0.007ms ↓:3.71ms
TGIF	↑:0.75KB ↓:583.8KB	↑:0.15ms ↓:114ms	↑:0.05ms ↓:38.0ms	↑:0.01ms ↓:11.4ms	↑:0.007ms ↓:5.70ms

Extensibility. Our proposed approach exhibits the capacity to enhance accuracy while concurrently reducing time delays across all datasets associated with the three specific tasks. As delineated in Tab. 3, our methodology yields a significant performance improvement, particularly in the domain of text-video retrieval. Our assessment encompasses a two-directional approach, addressing both text-to-video retrieval and video-to-text retrieval. Remarkably, our method surpasses the majority of baseline methods, not only in terms of accuracy but also with respect to time delays. In the context of multiple-choice Video QA, our approach consistently asserts its superiority across all evaluation metrics. This is evident from the data presented in Tab. 5.

4.5 Ablation Studies

Number of sampled frames. In Tab. 4, we investigate the influence of the number of sampled frames D from the entire video on the final results. Simultaneously, we measure the time of fine-tuning for each epoch. To be specific, we employ the standard fine-tuning approach to evaluate this factor across three distinct datasets, all in the Open-ended Video-question Answering task. The batch size for all the datasets is set to 256, and the hardware is a single A100 GPU. It becomes evident that as the number of sampled frames increases, the time of fine-tuning for each epoch experiences a rapid escalation, which is in line with intuition. For instance, when only two frames are sampled, the fine-tuning process demands approximately 740 seconds on MSRVTT-QA dataset. However, if five frames are sampled from each video, the time requirement extends to around 2092

seconds, which is almost three times longer than the former scenario. Interestingly, the model's performance does not exhibit a consistent upward trend with the increasing number of sampled frames. Besides, the model's performance reaches its peak when the number of sampled frames is set at three or four. To make a trade-off between performance and training cost, we set the number of sampled frames as three in our model.

Different modules in our proposed CDC-MMPG framework. In accordance with the results in Tab. 1 and Tab. 2, we have devised three additional baseline methods to assess the efficacy of our comprehensive framework. To be more specific, our assessment involves a comparative analysis of the performance outcomes between the fine-tuning model and our proposed model, serving to emphasize the latter's superiority. Furthermore, we have undertaken an evaluation of the "F-linear" model to facilitate an ablation study of the classifiers subsequent to the multi-modal feature extractor. In addition, the "F-hyper" model has been examined in order to conduct an ablation study focused on the hyper-network and the adaptive generator. It is noteworthy that the omission of our proposed adaptive generator renders a simple hyper-network prone to encountering sub-optimal performance, particularly evident in certain datasets, including MSRVTT-QA, MSVD-QA, and TGIF. This diminished performance is primarily attributed to the issue of over-fitting, leading to the emergence of spurious correlations between visual cues and predictions. Our adaptive generator, as proposed, substantially improves the performance of the hyper-network through the implementation of anchor-frame distribution reasoning.

Time delay between cloud and device in various internet conditions. Tab. 6 illustrates the time delay associated with data transmission between cloud services and devices under various internet connectivity conditions. Specifically, we have conducted this experiment in four distinct scenarios, each reflecting a different internet condition: 4G (5MB/s), 4G (15MB/s), 5G (50MB/s), and 5G (100MB/s). These scenarios correspond to real-world situations involving older mobile devices with limited 4G connectivity, older mobile devices with optimal 4G connectivity, modern mobile devices with 5G connectivity, and modern mobile devices with advanced 5G internet connectivity. Our proposed method demonstrates remarkable improvements in terms of time delay. For example, during the inference process, a mere 0.75KB of features is required to be uploaded to the cloud, a task that consumes only about 0.007ms in a 5G (100MB/s) environment. Even in the presence of constrained 4G (5MB/s) internet, the time required for uploading remains negligible. The duration of parameter downloading, however, fluctuates according to the dataset in use, largely due to variations in parameter size. For instance, the parameter size is 583.8KB for the TGIF dataset, the largest among the datasets, whereas the parameter size for the MSVD-QA dataset, the smallest, amounts to 379.4KB. Notably, even when dealing with the demanding TGIF dataset, our proposed approach demonstrates the capability to achieve real-time predictions. For instance, in a 5G (100MB/s) environment, the time required for parameter downloading is merely 5.70ms. Even in scenarios featuring limited 4G (5MB/s) connectivity, the time delay remains nearly imperceptible in practical terms.

5 CONCLUSION

In this paper, we focus on the challenge of enabling efficient and effective adaptation of AI systems to personalized multi-modal data generated by intelligent devices. The shifting data distribution between the cloud and devices necessitates novel approaches to ensure high-quality personalized services. We introduced a universal on-device Multi-modal Model Adaptation Framework, featuring the Fast Domain Adaptor (FDA) and the AnchorFrame Distribution Reasoner (ADR). FDA, hosted in the cloud, tailors model parameters for on-device Lightweight Multi-modal Models, optimizing adaptability across diverse data distributions. ADR further standardizes input, reducing communication costs. Our contributions, consolidated within the Cloud-Device Collaboration Multi-modal Parameter Generation (CDC-MMPG) framework, constitute a pioneering solution for

on-Device Multi-modal Model Adaptation (DMMA), demonstrated through extensive experiments. Looking forward, promising directions include expanding the framework to accommodate various modalities, refining personalized data handling techniques, and further reducing communication costs in multi-modal tasks.

REFERENCES

[1] Sabbir Ahmed, Abdullah Al Arafat, Mamshad Nayeem Rizve, Rahim Hossain, Zhishan Guo, and Adnan Siraj Rakin. 2023. SSDA: Secure Source-Free Domain Adaptation. In *ICCV*.

[2] Chaoqi Chen, Weiping Xie, Wenbing Huang, Yu Rong, Xinghao Ding, Yue Huang, Tingyang Xu, and Junzhou Huang. 2019. Progressive feature alignment for unsupervised domain adaptation. In *CVPR*.

[3] Hong-You Chen, Jike Zhong, Mingda Zhang, Xuhui Jia, Hang Qi, Boqing Gong, Wei-Lun Chao, and Li Zhang. 2023. Federated Learning of Shareable Bases for Personalization-Friendly Image Classification. arXiv:2304.07882 [cs.CV] https://arxiv.org/abs/2304.07882

[4] Pengfei Chen, Leida Li, Jinjian Wu, Weisheng Dong, and Guangming Shi. 2021. Unsupervised curriculum domain adaptation for no-reference video quality assessment. In *ICCV*.

[5] Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. 2022. Fedavg with fine tuning: Local updates lead to representation learning. *NIPS* (2022).

[6] Victor G. Turrisi da Costa, Giacomo Zara, Paolo Rota, Thiago Oliveira-Santos, Nicu Sebe, Vittorio Murino, and Elisa Ricci. 2022. Dual-Head Contrastive Domain Adaptation for Video Action Recognition. In *WACV*.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.

[8] Ning Ding, Yixing Xu, Yehui Tang, Chao Xu, Yunhe Wang, and Dacheng Tao. 2022. Source-Free Domain Adaptation via Distribution Estimation. In *CVPR*.

[9] Tan M. Dinh, Anh Tuan Tran, Rang Nguyen, and Binh-Son Hua. 2022. HyperInverter: Improving StyleGAN Inversion via Hypernetwork. In *CVPR*.

[10] Rinon Gal, Or Patashnik, Haggai Maron, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. 2022. StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. *ACM Trans. Graph.* (2022).

[11] David Ha, Andrew Dai, and Quoc V. Le. 2017. HyperNetworks. In *ICLR*.

[12] Wei Ji, Li Li, Hao Fei, Xiangyan Liu, Xun Yang, Juncheng Li, and Roger Zimmermann. 2024. Towards Complex-query Referring Image Segmentation: A Novel Benchmark. *ACM Trans. Multimedia Comput. Commun. Appl.* (Nov. 2024). https://doi.org/10.1145/3701733

[13] Li Li, Wei Ji, Yiming Wu, Mengze Li, You Qin, Lina Wei, and Roger Zimmermann. 2024. Panoptic Scene Graph Generation with Semantics-Prototype Learning. *AAAI* 38, 4 (Mar. 2024), 3145–3153. https://doi.org/10.1609/aaai.v38i4.28098

[14] Li Li, You Qin, Wei Ji, Yuxiao Zhou, and Roger Zimmermann. 2024. Domain-Wise Invariant Learning for Panoptic Scene Graph Generation. In *ICASSP*. 3165–3169. https://doi.org/10.1109/ICASSP48485.2024.10447193

[15] Li Li, Chenwei Wang, You Qin, Wei Ji, and Renjie Liang. 2023. Biased-Predicate Annotation Identification via Unbiased Visual Predicate Representation. In *ACM MM (MM '23)*. 4410–4420. https://doi.org/10.1145/3581783.3611847

[16] Ming Li, Jike Zhong, Chenxin Li, Liuzhuozheng Li, Nie Lin, and Masashi Sugiyama. 2024. Vision-Language Model Fine-Tuning via Simple Parameter-Efficient Modification. arXiv:2409.16718 [cs.CV] https://arxiv.org/abs/2409.16718

[17] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. 2016. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *CVPR*.

[18] Mattia Litrico, Alessio Del Bue, and Pietro Morerio. 2023. Guiding Pseudo-Labels With Uncertainty Estimation for Source-Free Unsupervised Domain Adaptation. In *CVPR*.

[19] Nannan Lu, Hanhan Xiao, Zhanguo Ma, Tong Yan, and Min Han. 2022. Domain Adaptation With Self-Supervised Learning and Feature Clustering for Intelligent Fault Diagnosis. *IEEE Transactions on Neural Networks and Learning Systems* (2022), 1–14. https://doi.org/10.1109/TNNLS.2022.3219896

[20] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. 2020. Adversarial Style Mining for One-Shot Unsupervised Domain Adaptation. In *NIPS*.

[21] Yawei Luo, Ping Liu, Tao Guan, Junqing Yu, and Yi Yang. 2020. Adversarial Style Mining for One-Shot Unsupervised Domain Adaptation. In *NIPS*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.).

[22] Zheqi Lv, Zhengyu Chen, Shengyu Zhang, Kun Kuang, Wenqiao Zhang, Mengze Li, Beng Chin Ooi, and Fei Wu. 2023. Ideal: Toward high-efficiency device-cloud collaborative and dynamic recommendation system. *arXiv preprint arXiv:2302.07335* (2023).

[23] Zheqi Lv, Wenqiao Zhang, Shengyu Zhang, Kun Kuang, Feng Wang, Yongwei Wang, Zhengyu Chen, Tao Shen, Hongxia Yang, Beng Chin Ooi, et al. 2023. DUET: A Tuning-Free Device-Cloud Collaborative Parameters Generation Framework for Efficient Device Model Generalization. In *WWW*. 3077–3085.

[24] Xianzheng Ma, Zhixiang Wang, Yacheng Zhan, Yinqiang Zheng, Zheng Wang, Dengxin Dai, and Chia-Wen Lin. 2022. Both Style and Fog Matter: Cumulative Domain Adaptation for Semantic Foggy Scene Understanding. In *CVPR*.

[25] Ahmadreza Mosallanezhad, Mansooreh Karami, Kai Shu, Michelle V Mancenido, and Huan Liu. 2022. Domain adaptive fake news detection via reinforcement learning. In *WWW*. 3632–3640.

[26] Hod Lipson Oscar Chang, Lampros Flokas. 2020. Principled Weight Initialization for Hypernetworks. In *ICLR*.

[27] Subhankar Roy, Evgeny Krivosheev, Zhun Zhong, Nicu Sebe, and Elisa Ricci. 2021. Curriculum graph co-teaching for multi-target domain adaptation. In *CVPR*.

[28] Cristiano Saltori, Fabio Galasso, Giuseppe Fiameni, Nicu Sebe, Fabio Poiesi, and Elisa Ricci. 2023. Compositional Semantic Mix for Domain Adaptation in Point Cloud Segmentation. *TPAMI* (2023), 1–14. https://doi.org/10.1109/TPAMI.2023.3310261

[29] Yang Shu, Zhangjie Cao, Mingsheng Long, and Jianmin Wang. 2019. Transferable curriculum for weakly-supervised domain adaptation. In *AAAI*.

[30] Tao Sun, Cheng Lu, Tianshuo Zhang, and Haibin Ling. 2022. Safe Self-Refinement for Transformer-Based Domain Adaptation. In *CVPR*.

[31] Zhensu Sun, Li Li, Yan Liu, Xiaoning Du, and Li Li. 2022. On the importance of building high-quality training datasets for neural code search. In *ICSE*. 1609–1620. https://doi.org/10.1145/3510003.3510160

[32] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. 2021. Training data-efficient image transformers amp & distillation through attention. PMLR.

[33] Cheng-Hao Tu, Hong-You Chen, Zheda Mai, Jike Zhong, Vardaan Pahuja, Tanya Berger-Wolf, Song Gao, Charles Stewart, Yu Su, and Wei-Lun Chao. 2023. Holistic Transfer: Towards Non-Disruptive Fine-Tuning with Partial Target Data. arXiv:2311.01420 [cs.LG] https://arxiv.org/abs/2311.01420

[34] Alex Jinpeng Wang, Yixiao Ge, Rui Yan, Ge Yuying, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. 2023. All in One: Exploring Unified Video-Language Pre-training. *CVPR* (2023).

[35] Fan Wang, Zhongyi Han, Yongshun Gong, and Yilong Yin. 2022. Exploring Domain-Invariant Parameters for Source Free Domain Adaptation. In *CVPR*.

[36] Haixin Wang, Jinan Sun, Xiang Wei, Shikun Zhang, Chong Chen, Xian-Sheng Hua, and Xiao Luo. 2023. DANCE: Learning A Domain Adaptive Framework for Deep Hashing. In *WWW*. 3319–3330.

[37] Rui Wang, Zuxuan Wu, Zejia Weng, Jingjing Chen, Guo-Jun Qi, and Yu-Gang Jiang. 2023. Cross-Domain Contrastive Learning for Unsupervised Domain Adaptation. *TMM* 25 (2023), 1665–1673.

[38] Zhou Xian, Shamit Lal, Hsiao-Yu Tung, Emmanouil Antonios Platanios, and Katerina Fragkiadaki. 2021. HyperDynamics: Meta-Learning Object and Agent Dynamics with Hypernetworks. In *ICLR*.

[39] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. 2017. Video Question Answering via Gradually Refined Attention over Appearance and Motion. ACM MM.

[40] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. MSR-VTT: A Large Video Description Dataset for Bridging Video and Language. CVPR.

[41] Yikai Yan, Chaoyue Niu, Renjie Gu, Fan Wu, Shaojie Tang, Lifeng Hua, Chengfei Lyu, and Guihai Chen. 2022. On-Device Learning for Model Personalization with Large-Scale Cloud-Coordinated Domain Adaption. In *KDD*. 2180–2190.

[42] Ceyuan Yang, Yujun Shen, Zhiyi Zhang, Yinghao Xu, Jiapeng Zhu, Zhirong Wu, and Bolei Zhou. 2023. One-Shot Generative Domain Adaptation. In *ICCV*.

[43] Jihan Yang, Shaoshuai Shi, Zhe Wang, Hongsheng Li, and Xiaojuan Qi. 2023. ST3D++: Denoised Self-Training for Unsupervised Domain Adaptation on 3D Object Detection. *TPAMI* 45, 5 (2023), 6354–6371. https://doi.org/10.1109/TPAMI.2022.3216606

[44] Shiqi Yang, yaxing wang, kai wang, Shangling Jui, and Joost van de Weijer. 2022. Attracting and Dispersing: A Simple Approach for Source-free Domain Adaptation. In *NIPS*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.).

[45] Jiangchao Yao, Feng Wang, Kunyang Jia, Bo Han, Jingren Zhou, and Hongxia Yang. 2021. Device-Cloud Collaborative Learning for Recommendation. In *KDD*.

[46] Junjie Ye, Changhong Fu, Guangze Zheng, Danda Pani Paudel, and Guang Chen. 2022. Unsupervised Domain Adaptation for Nighttime Aerial Tracking. In *CVPR*.

[47] Liping Yi, Xiaorong Shi, Nan Wang, Ziyue Xu, Gang Wang, and Xiaoguang Liu. 2023. pFedLHNs: Personalized Federated Learning via Local Hypernetworks. In *ICANN 2023*.

[48] Ao Zhang, Hao Fei, Yuan Yao, Wei Ji, Li Li, Zhiyuan Liu, and Tat-Seng Chua. 2023. VPGTrans: Transfer Visual Prompt Generator across LLMs. In *NeurIPS*, Vol. 36. 20299–20319.

[49] Cheng Zhang, Tai-Yu Pan, Tianle Chen, Jike Zhong, Wenjin Fu, and Wei-Lun Chao. 2022. Learning with Free Object Segments for Long-Tailed Instance Segmentation. arXiv:2202.11124 [cs.CV] https://arxiv.org/abs/2202.11124

[50] Chris Zhang, Mengye Ren, and Raquel Urtasun. 2020. Graph HyperNetworks for Neural Architecture Search. In *ICLR*.

[51] Jingyi Zhang, Jiaxing Huang, Zichen Tian, and Shijian Lu. 2022. Spectral Unsupervised Domain Adaptation for Visual Recognition. In *CVPR*.
[52] Ruohan Zhang, Tianzi Zang, Yanmin Zhu, Chunyang Wang, Ke Wang, and Jiadi Yu. 2023. Disentangled Contrastive Learning for Cross-Domain Recommendation. In *International Conference on Database Systems for Advanced Applications*. Springer.
[53] Yang Zhang, Philip David, Hassan Foroosh, and Boqing Gong. 2019. A curriculum domain adaptation approach to the semantic segmentation of urban scenes. *TPAMI* 42, 8 (2019), 1823–1841.
[54] Yuyang Zhao, Zhun Zhong, Zhiming Luo, Gim Hee Lee, and Nicu Sebe. 2022. Source-Free Open Compound Domain Adaptation in Semantic Segmentation. *TCSVT* 32, 10 (2022), 7019–7032. https://doi.org/10.1109/TCSVT.2022.3179021
[55] Jike Zhong, Hong-You Chen, and Wei-Lun Chao. 2024. Making Batch Normalization Great in Federated Deep Learning. arXiv:2303.06530 [cs.LG] https://arxiv.org/abs/2303.06530