Exploring Parameter-Efficient Fine-Tuning to Enable Foundation Models in Federated Learning

1st Guangyu Sun
Center for Research in Computer Vision
University of Central Florida
Orlando, FL, USA
guangyu.sun@ucf.edu

2nd Umar Khalid
Center for Research in Computer Vision
University of Central Florida
Orlando, FL, USA
umar.khalid@ucf.edu

3rd Matias Mendieta
Center for Research in Computer Vision
University of Central Florida
Orlando, FL, USA
matias.mendieta@ucf.edu

4th Pu Wang
Department of Computer Science
University of North Carolina at Charlotte
Charlotte, NC, USA
pu.wang@uncc.edu

5th Chen Chen
Center for Research in Computer Vision
University of Central Florida
Orlando, FL, USA
chen.chen@crcv.ucf.edu

***Abstract*—Federated learning (FL) has emerged as a promising paradigm for enabling the collaborative training of models without centralized access to the raw data on local devices. In the typical FL paradigm (*e.g.*, FedAvg), model weights are sent to and from the server each round to participating clients. Recently, the use of small pre-trained models has been shown to be effective in federated learning optimization and improving convergence. However, recent state-of-the-art pre-trained models are getting more capable but also have more parameters, known as the "Foundation Models." In conventional FL, sharing the enormous model weights can quickly put a massive communication burden on the system, especially if more capable models are employed. Can we find a solution to enable those strong and readily available pre-trained models in FL to achieve excellent performance while simultaneously reducing the communication burden? To this end, we investigate the use of parameter-efficient fine-tuning in federated learning and thus introduce a new framework: FedPEFT. Specifically, we systemically evaluate the performance of FedPEFT across a variety of client stability, data distribution, and differential privacy settings. By only locally tuning and globally sharing a small portion of the model weights, significant reductions in the total communication overhead can be achieved while maintaining competitive or even better performance in a wide range of federated learning scenarios, providing insight into a new paradigm for practical and effective federated systems.**

***Index Terms*—federated learning, parameter-efficient fine-tuning, vision transformers, image classification, action recognition**

I. Introduction

Federated learning (FL) [1] has become increasingly prevalent in the research community, having the goal of enabling collaborative training with a network of clients without needing to share any private data. One key challenge for this training paradigm is overcoming data heterogeneity. The participating devices in a federated system are often deployed across a variety of users and environments, resulting in a non-IID data distribution. As the level of heterogeneity intensifies, optimization becomes increasingly difficult. Various techniques have been proposed for alleviating this issue. These primarily consist of modifications to the local or global objectives through proximal terms, regularization, and improved aggregation operations [2, 3, 4, 5, 6]. More recently, some works have investigated the role of model initialization in mitigating such effects [7, 8]. Inspired by the common usage of pre-trained models for facilitating strong transfer learning in centralized training, researchers employed widely available pre-trained weights for initialization in FL and were able to close much of the gap between federated and centralized performance.

Still, while pre-trained initializations are effective for alleviating heterogeneity effects in FL, another key challenge is left unaddressed; that is, communication constraints. This is often the primary bottleneck for real-world federated systems [9]. In the standard FL framework [10], updates for all model parameters are sent back and forth between the server and participating clients each round. This can quickly put a massive communication burden on the system, especially if more capable models beyond very small MLPs are used.

When employing strong pre-trained models, the number of parameters can be large, such as for current state-of-the-art transformers. For example, ViT-Base (ViT-B) [11] has 84 million parameters, let alone the current significant progress in large foundation models (*e.g.*, GPT-4 [12] has more than 1 trillion parameters). Those large models would simply exacerbate the communication overhead to insurmountable levels. As a compromise, most existing FL work focuses on the performance of smaller Convolutional Neural Networks (*e.g.*, ResNet [13]) on smaller datasets (*e.g.*, CIFAR-10 [14], EMINIST [15]). Considering the thriving progress in large pre-trained Foundation Models [16], *an efficient framework enabling these large pre-trained models will be significant for the FL community.*

Based on the previous study on centralized training [17, 18, 19, 20], we note that pre-trained models have strong representations, and updating all the weights during fine-tuning

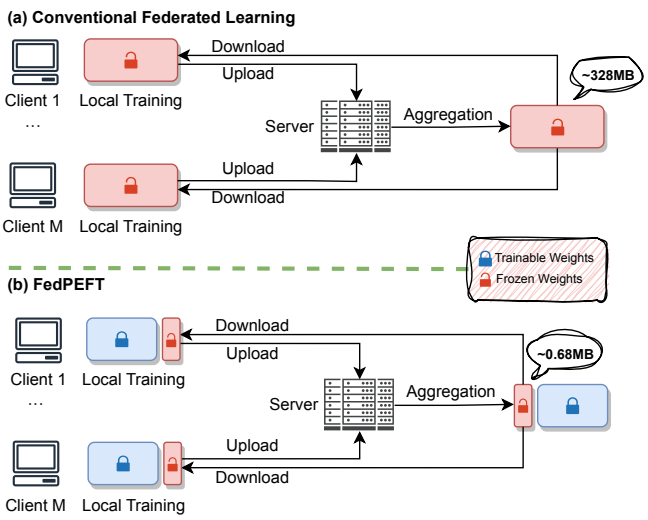


Fig. 1: **Process in a federated learning communication round with M participating clients.** We use ViT-Base as an instance to analyze the communication costs. (a) Conventional federated learning framework, where the entire model will be sent during the communication. (b) FedPEFT, which is our proposed parameter-efficient framework for federated learning.

is often not necessary. Various parameter-efficient fine-tuning methods (*e.g.*, fine-tuning only a subset of the parameters or the bias terms) for centralized training have been proposed in the literature and show that successful and efficient adaptation is possible, even under domain shift [18, 17, 19]. We reason that such insight is applicable to FL, where each client can be thought of as a shifted domain on which we are fine-tuning. By leveraging pre-trained weights, it may be possible to simply update a small portion of the weights for each client. This will significantly reduce the communication burden on the system, as the updates communicated with the server will consist of just a fraction of the total model parameters.

Can we reap these potential communication benefits while still achieving strong performance in FL? Unfortunately, operating conditions in FL are difficult, requiring successful convergence under varying data heterogeneity levels, random client availability, and differential privacy procedures. Therefore, we are unable to properly assess this possibility of benefit based on existing literature, as *diverse parameter-efficient fine-tuning methods have not been systematically explored in such situations in FL.* To fill this gap, we explore the viability of a Federated Parameter-Efficient Fine-Tuning (FedPEFT) framework with a systemic empirical study on a comprehensive set of FL scenarios including *communication analysis* about communication cost for each method to enable pre-trained models, *capability analysis* of each method with unlimited communication budget, and *robustness analysis* of each method when additional constraints (*i.e.*, differential privacy or data scarcity) applied. The framework is illustrated in Fig. 1. We deploy parameter-efficient fine-tuning methods to adapt pre-trained models and enable massive reductions in communication overheads.

The contribution of this paper is summarized as follows:

- We explored several PEFT methods in FL as the FedPEFT framework to simultaneously addresses data heterogeneity and communication challenges. FedPEFT allows for the utilization of powerful pre-trained models in federated learning while keeping communication costs extremely low.
- We present a systematic study of the FedPEFT framework with various fine-tuning methods under heterogeneous data distributions, client availability ratios, and increasing degrees of domain gap relative to the pre-trained representations on both *image* and *video* domains, showing the capability of FedPEFT. (Sections IV-B and IV-C)
- To ensure FedPEFT is practical for the complex environments of FL, we further analyze the robustness of FedPEFT among low-data regimes and differential privacy operations. (Sections IV-D)

II. Related Work

Federated Learning. FL is a decentralized training paradigm composed of two procedures: local training and global aggregation. Therefore, most existing work focuses on either local training [4, 21, 2] or global aggregation [22, 23] to learn a better global model. Another line of work cuts into this problem by applying different initializations to help both procedures. [8] shows that initializing the model with pre-trained weights can make the global aggregation of FedAvg more stable, even when pre-trained with synthetic data. Furthermore, [7] presents the effectiveness of pre-training with different local and global operations. However, these works focus purely on the effect of initialization in a standard FedAvg framework and do not consider the communication constraints of the system. Our work pushes the envelope further by leveraging strong pre-trained models (even large, capable transformers) in FL while effectively handling the communication issue via parameter-efficient fine-tuning.

Communication in Federated Learning. Communication constraints are a primary bottleneck in federated learning. To reduce the communication cost, several previous work leverage model compression techniques [24, 25]. Such works do not change the training paradigm but rather post-process the local model to reduce communication costs. For instance, [24] proposes approaches that parameterize the model with fewer variables and compress the model in an encoding-decoding fashion. However, the minimal requirement to maintain all the information is still high when facing today's large models. Meanwhile, another line of work changes the training paradigm by learning federated ensembles based on several pre-trained base models [26]. In this way, only the mixing weights of the base models will be communicated in each round. This approach aims to reduce the burden of downloading and uploading the entire model in each round. However, the base models are not directly trained, and the final performance is highly related to the base models. Meanwhile, model ensembles will take more time and space, which is often limited on the client side. Our framework follows the strategy of this line of work that does not transmit the entire model, but we use only one pre-trained model instead of several base

models and only transmit a subset of the parameters instead of the model ensembles. Therefore, no additional time or space is required.

Parameter-Efficient Fine-tuning. Fine-tuning is a prevalent topic in centralized transfer learning, especially in this era of the "Foundation Model" [16]. A significant line of work is to reduce the trainable parameter number, *i.e.*, parameter-efficient fine-tuning (PEFT) [20, 27, 28, 29, 30, 31, 17, 32]. PEFT has emerged as a pivotal area of research in the field of natural language processing (NLP) [19, 33, 34, 35] and further adapted into more fields such as computer vision (CV) [18, 36, 37, 38, 39, 40, 41]. With different focuses, PEFT methods can be divided into three categories: 1) Input Adaptation methods such as prompt-tuning [18, 30] focusing on adding learnable context to the input data, 2) Backbone Adaptation methods such as adapter-tuning [38, 19, 33], and 3) Specification methods such as bias-tuning [17, 31]. PEFT enables easier access and usage of pre-trained models by reducing the memory cost needed to conduct fine-tuning due to fewer computed gradients. In federated learning, PEFT has an additional benefit that is not salient in centralized training: reducing communication costs. By introducing PEFT to federated learning, our work can take advantage of a strong (and even large) pre-trained model while significantly reducing communication costs. Several works study the PEFT methods under FL settings in NLP [42, 43, 44, 45]. Complementarily, *our work provides a comprehensive study of PEFT in various FL settings in computer vision under both image and video tasks and insights on various scenarios, including more privacy requirements or under limited data.*

III. Federated Parameter-Efficient Fine-Tuning

A. Problem Formulation

In this section, we formally describe the federated learning objective and federated parameter-efficient fine-tuning. Using a classification task as an example, K samples in a dataset $\mathbb{D} = \{(\boldsymbol{x}_k, y_k)_{k=1}^{K}\} = \cup_{n=1}^{N}\mathbb{D}_n$, where $\boldsymbol{x}$ is the input and $y \in \{0, 1, \ldots, C-1\}$ is the label, are distributed among N clients. Each client has a local model $\{\phi_i\}_{i=1}^{N}$ parameterized by $\{\boldsymbol{\theta}_i \cup \delta_i\}_{i=1}^{N}$, where $\boldsymbol{\theta}$ is the pre-trained weights and δ is the trainable parameters. The goal of federated learning is *to learn a global model* ϕ parameterized by $\boldsymbol{\theta} \cup \delta$ on the server from M sampled client models in T communication rounds by minimizing the global objective F as

$$\min_{\delta} F(\phi) = \frac{1}{|\mathbb{D}_{test}|} \sum_{i=1}^{|\mathbb{D}_{test}|} \ell(y_k, \phi_i^{(t)}(\boldsymbol{x}_k)) \tag{1}$$

on a hold-off test set $\mathbb{D}_{test}$ with a loss function ℓ. *Compared with traditional FL updating the entire ϕ, only δ is updated in FedPEFT.*

At the beginning of training, $\boldsymbol{\theta}^{(0)}$in the global model $\phi^{(0)}$ is initialized with pre-trained weights, and $\delta^{(0)}$ is randomly initialized, where the superscript t indicates the model at round t. In each round t, M clients will be selected for

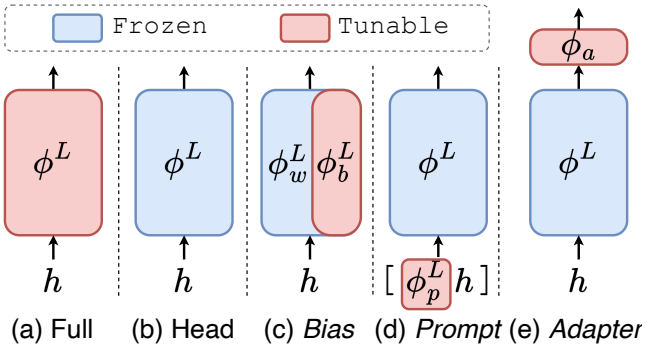


Fig. 2: Methods to fine-tune each layer in a pre-trained backbone, where h means the input, ϕ means the pre-trained layer, and ϕ_w, ϕ_b mean its weight and bias parameters, respectively.

communication, and $\{\phi_i^{(t)}\}_{i=1}^{M}$ will be initialized by $\phi^{(t)}$ and updated by

$$\min_{\delta_i} F_i(\phi_i^{(t)}) = \frac{1}{|\mathbb{D}_i|} \sum_{k=1}^{|\mathbb{D}_i|} \ell(y_k, \phi_i^{(t)}(\boldsymbol{x}_k)) \tag{2}$$

for E epochs, where F_i is the local objective. After the local updates, the server will receive and aggregate the trainable parameters $\{\delta_i^{(t)}\}_{i=1}^{M}$ with the FedAvg algorithm to a new global model

$$\phi^{(t+1)} = \sum_{m=1}^{M} \frac{|\mathbb{D}_m|}{\sum_{i=1}^{M} |\mathbb{D}_i|} \phi_i^{(t)}. \tag{3}$$

This procedure is repeated from $t = 0$ to $t = T-1$. During the client-server communication, we only take the communication cost for the model into consideration, assuming the remaining communication costs are fixed. Therefore, the communication cost C is proportional to the transmission parameters number, thus can be formulated as

$$C \propto |\delta| \cdot M, \tag{4}$$

We take the one-way communication cost (*i.e.*, upload or download) as the metric. *The final goal of this problem is to minimize the C while maintaining server accuracy.*

B. FedPEFT

In conventional federated learning, updates for the entire model need to be repeatedly sent to and from the server, resulting in significant communication costs, especially when larger, more capable modern neural network architectures are employed. To reduce this heavy burden, we deploy parameter-efficient fine-tuning methods to adapt pre-trained models to the local clients rather than fully fine-tuning all parameters, which is described in Algorithm 1. In the FedPEFT framework, illustrated in Fig. 1, only a small amount of parameters in the local model will be downloaded, trained, and uploaded in each communication round. For instance, FedPEFT reduces the size of communication each round from **328MB** (85.88M parameters)/Client to **0.68MB** (0.17M parameters)/Client when using a pre-trained ViT-Base as the backbone in our default setting introduced in Section IV-A.

Algorithm 1: Algorithm of FedPEFT framework

Input: N clients indexed by i, participating-client number M, communication rounds T, trainable parameters δ of the model where $|\delta| << |\phi|$, pre-trained model weights $\boldsymbol{\theta}$, random initialized $\delta^{(0)}$, and local epoch number E.

Server executes:
initialize $\phi^{(0)}$ and $\{\phi_i^{(0)}\}_{i=1}^N$ with $\boldsymbol{\theta}$ and $\delta^{(0)}$
for each round $t = 0, 2, ..., T-1$ **do**
  $\mathbb{S}_t \leftarrow$ (random set of M clients)
  for each client $i \in \mathbb{S}_t$ **in parallel do**
    $\delta_i^{(t)} \leftarrow \delta^{(t)}$
    $\delta_i^{(t+1)} \leftarrow$ **ClientUpdate**$(\delta_i^{(t)}, i)$
  $\delta^{(t+1)} = \sum_{i=1}^{M} \frac{|\mathbb{D}_i|}{\sum_{i=1}^{M}|\mathbb{D}_i|}\delta_i^{(t+1)}$
  $\phi^{(t+1)} \leftarrow \{\boldsymbol{\theta}, \delta^{(t+1)}\}$
return $\phi^{(T)}$

ClientUpdate (δ, i):
$\delta \leftarrow$ perform local training on δ with $\mathbb{D}_i$ for E epochs
return δ

To implement FedPEFT, we provide a canonical baseline approach (Head-tuning) and three prototypes leveraging different representative parameter-efficient fine-tuning methods (Bias, Adapter, and Prompt), which are detailed in the following.

To reduce the number of trainable parameters, one intuitive method, Head-tuning, is to freeze the backbone ϕ and only train the head c. This method is historically the most common fine-tuning procedure, and therefore we use it as a baseline for FedPEFT. However, the adaptation ability of this method is limited, as no adjustment is made to the network representation prior to the final output head. This can be problematic in the presence of a more intense domain shift. Therefore, we consider the following approaches as primary prototypes for FedPEFT:

FedPEFT-Bias. Bias-tuning [17] aims to adapt the pre-trained backbone by only fine-tuning a specific group of parameters, the bias term. In this way, the backbone can be trained with moderate adjustments to prevent damaging the upstream representation. We show the output of each layer ϕ^l given hidden states h as $\boldsymbol{h} := \phi_w^L \boldsymbol{h} + \phi_b^L$, where ϕ_w^L and ϕ_b^L are weight and bias of ϕ_L, blue parameters are frozen, while red parameters are trainable.

FedPEFT-Adapter. Instead of directly tuning existing parameters in the backbone like Bias-tuning, Adapter-tuning [19] adds a few parameters called adapters inside the backbone ϕ instead. Usually, adapters will be deployed in each layer of the backbone to perform transformations on different levels of the pre-trained feature while the backbone stays frozen. We show the output of each layer ϕ^l given hidden states h as $\boldsymbol{h} := \phi_a(\phi_w^L(\boldsymbol{h}))$,where ϕ_a is the adapter.

FedPEFT-Prompt. Prompt-tuning [18] takes a slightly different approach from the other fine-tuning methods. Specifically, it concatenates trainable parameters, called prompt embeddings, to the input embedding and hidden states in each layer. We show the output of each layer ϕ^l given hidden states h as $\boldsymbol{h} := \phi_w^L([\phi_p^L, \boldsymbol{h}])$, where ϕ_p^L is the prompt for layer L and $[\cdot, \cdot]$ is the concat operation.

We illustrate the differences between all baseline and prototype methods in Fig. 2. Besides the above prototypes, our framework is compatible with other PEFT methods such as LoRA [29].

C. Convergence Guarantee

Based on the convergence of FedAvg in [1, 3, 46], in this section, we will comment on the convergence of FedPEFT. For ease of notation, we consider, at each round, S to be the number of clients sampled. We require the following assumptions.

Assumption 1: (Global minimum) For the global objective F, there exists $\phi^\star$ such that, $F(\phi^\star) = F^\star \leq F(\phi)$, for all $\phi \in \mathbb{R}^d$.

Assumption 2: (β-Smoothness) The loss function $f_i : \mathbb{R}^d \to \mathbb{R}$ at each node is β-smooth, i.e. $f_i(y) \leq f_i(x) + \nabla f_i(x)^\top (y - x) + \frac{\beta}{2}\|y - x\|^2$ for all $x, y \in \mathbb{R}^d$.

Remark 1: (PEFT-FT gap) The above assumption implies that F is β-smooth. Therefore, the gap between PEFT and Full Fine-tuning can be proved as $|F(\boldsymbol{\theta}^{(T)} \cup \delta^{(0)}) - F(\boldsymbol{\theta}^{(T)} \cup \delta^{(0)}) = \mathcal{O}(\frac{\beta}{2}(\|\boldsymbol{\theta}^{(T)} - \boldsymbol{\theta}^{(0)}\|^2 + \|\delta^{(T)} - \delta^{(0)}\|^2))|$.

Assumption 3: There exist constants $G \geq 0, B \geq 1$, such that for all $x \in \mathbb{R}^d$, the stochastic noise, $\xi_{i,t}$ follows

$$\frac{1}{N}\sum_{i=1}^{N} \|\nabla f_i(x)\|^2 \leq G^2 + B^2 \|F(x)\|^2.$$

Assumption 4: (Bounded variance) Let $g_i(\phi) := \nabla f_i(\phi, z_{i(k)})$ be the unbiased stochastic gradient of f_i with bounded variance. That is, there exists, $\sigma \geq 0$ such that, $\mathbb{E}_{z_{i(k)}}\left[\|g_i(\phi) - \nabla f_i(\phi)\|^2\right] \leq \sigma^2$, for all ϕ, i, where $z_{i(k)}$ is the k^{th} sample data at the i^{th} client.

Based on the vanilla FedAvg framework, we can give our main convergence result as

Theorem 1: Let F satisfies Assumptions 1-4. Then

$$\mathbb{E}\left[\|\nabla F(\phi^{(T)})\|^2\right] \leq \mathcal{O}\left(\frac{\beta\sqrt{(F(\phi^{(0)}) - F^\star)}}{\sqrt{TEM}} + P\right),$$

where $P = \frac{\beta}{2}(\|\boldsymbol{\theta}^{(T)} - \boldsymbol{\theta}^{(0)}\|^2 + \|\delta^{(T)} - \delta^{(0)}\|^2)$.

D. Privacy Discussion

Federated learning inherently ensures data privacy, as it keeps the training data localized. Consequently, our proposed FedPEFT framework, by design, does not introduce any additional risk of privacy leakage beyond what is intrinsic to FL itself. However, it is crucial to acknowledge the vulnerability of FL systems to gradient inversion attacks [47, 48], where an adversary could potentially reconstruct original training data from shared gradients. This type of attack typically requires smaller batch sizes to be effective, as the precision of the information contained within the gradients diminishes with larger batch sizes, significantly reducing the feasibility of such attacks [49]. In light of this, FedPEFT inherently encourages the use of larger batch sizes compared to conventional full fine-tuning methods, as FedPEFT requires gradients for much

fewer parameters and, therefore, very little memory cost during training. Enabling larger batch sizes not only optimizes the training efficiency but also fortifies the privacy-preserving nature of the federated learning framework, further mitigating the risks associated with gradient inversion attacks.

IV. Experiments

The capability of full fine-tuning in terms of accuracy has been illustrated in recent work [7, 8]. Thus we regard it as a competitive baseline for FedPEFT. To verify the performance of FedPEFT comprehensively, we evaluate the server accuracy with each method from three perspectives and aim to answer the following questions:

Communication Analysis: When faced with a limited communication budget, there are several solutions to reduce costs, *e.g.*, sampling fewer clients each round or using a lightweight model. Can FedPEFT outperform other solutions in terms of *communication cost and accuracy*? **(RQ1)**

Capability Analysis: When the communication budget is amply sufficient for all approaches, we want to evaluate the trade-off of training fewer parameters with FedPEFT. Can FedPEFT outperform full fine-tuning and training from scratch within *various federated learning settings* and *increasing levels of downstream domain gap*? **(RQ2)**

Robustness Analysis: In a lot of application scenarios, there will be additional challenges for FL, such as privacy-preserving requirements (*i.e.*, differential privacy) and data scarcity (*i.e.*, very small amount of data on each client). We want to evaluate the robustness of each method under such scenarios. Can FedPEFT outperform full fine-tuning in terms of *robustness*? **(RQ3)**

A. Experiments details

Dataset. For our study, we focus on computer vision (CV) applications as our testbed. Specifically, we investigate the performance of each method on the *Image* and *Video* domains with image classification and action recognition tasks. For the image classification, we employ **ImageNet-21K** [52] as the pre-training dataset. Then we select three datasets for the downstream tasks that have *increasing degrees of domain gap* compared to ImageNet-21k, and we visualize and quantify the domain gap in Section IV-C: Resisc45 [53], CIFAR-100 [14] and PCam [54]. For video domain analysis, we take video action recognition task for evaluation. we employ **Kinetics-400** [55] as the pre-training dataset and select three datasets with varying degrees of domain gap as compared to Kinetics-400: UCF101 [56], HMDB51 [57] and UCF-CRIME [58].

Experimental Setting. Our default experimental setting is to split the dataset across $N = 64$ clients and sample $M = 8$ clients each round. The global aggregation will be performed after $E = 10$ local epochs. A total of $T = 50$ rounds of communication will be performed. To simulate heterogeneous data, we partition samples in each class to all clients following a Dirichlet distribution, as common in the literature [4, 5, 21], with $\alpha = 0.1$ for CIFAR-100 and Resisc45 and $\alpha = 0.5$ for PCam based on the class number. Any modifications to this setting in subsequent experiments will be clearly noted.

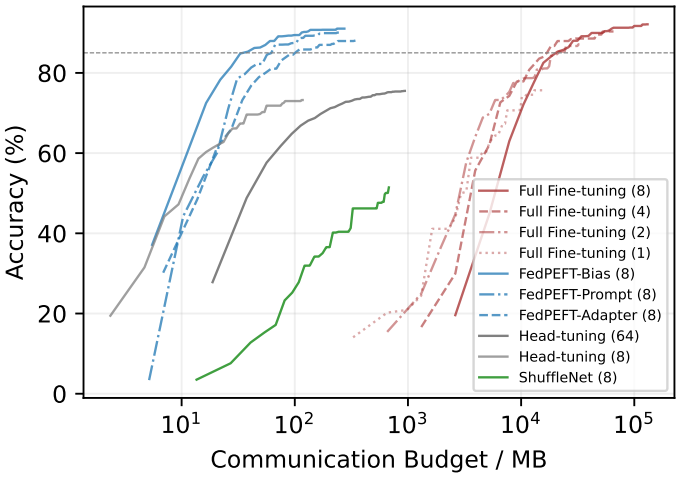


Fig. 3: **Server accuracy given the total communication budget.** The communication cost is computed with 4B/parameter, and the max number of communication rounds is 50. The number in the bracket next to the method indicates the number of participating clients m. The transparency of the line indicates the ratio between m and total client number $N = 64$. The horizontal dashed line shows a target accuracy of 85%.

For each action recognition experiment, the data is split across $N = 32$ clients and sample $M = 4$ clients each round with constant $\alpha = 0.1$ across all three video datasets.

Implementation Detail. We choose ViT-B [11] with image size 224 and patch size 16 as our backbone. For the image domain, the backbone is pre-trained on ImageNet-21K [52], as available in the timm library [59]. The images for the downstream datasets are resized to 224×224. Images from CIFAR-100 are augmented by random cropping with a padding of 4 and random horizontal flipping, and Resisc45 and PCam are augmented only with random horizontal flipping. For the video domain, we choose vanilla ViT-Base (ViT-B/16) [11] with joint space-time attention as our backbone model [60] using VideoMAE [61] pre-trained weights on Kinetics-400 [55]. We perform the experiments on 8 Nvidia RTX A5000 GPUs with a batch size of 64. All reported main results are run under 3 random seeds and averaged.

B. RQ1: Communication Analysis

To verify the effectiveness of FedPEFT and answer the first research question (**RQ1**, Section IV), we compare it with three baselines while monitoring the communication budget: a) Full fine-tuning of our default model (ViT-B). We vary the number of participating clients to show different levels of communication requirements. b) Head-tuning. The communication cost of head-tuning is naturally lower than other methods, so we increase the participating clients to make it a stronger baseline. c) Fully fine-tune a light-weighted model (ShuffleNet V2 0.5× [50] for images and X3D-S [51] for videos) with a similar communication overhead.

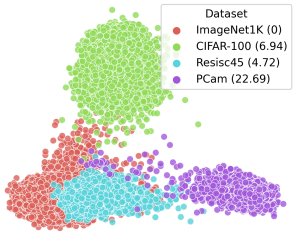
As demonstrated in Table I, all FedPEFT methods achieve better results in many cases compared with other approaches, even with significantly fewer communicated parameters. We

TABLE I: **Communication analysis on the image (upper) and video (lower) domains.** The communication cost is computed with 4B/parameter. The averaged final accuracy, *i.e.*, $t = T = 50$, and the standard deviation of three different random seeds are reported for each data set. The number of tuned parameters is computed based on CIFAR-100, but it may be slightly different for each dataset. The first section shows the change in accuracy when decreasing the participating client number. The gray numbers indicate the baseline performance with no decrease in the participating client number. The second section shows the change in accuracy when we reward the low communication cost of head-tuning by increasing the number of participating clients. The third section shows the accuracy when we fully fine-tune a lightweight model, ShuffleNet V2 $0.5\times$ [50] for image domain or X3D-S [51] for video domain. The fourth section shows the performance of each prototype of FedPEFT.

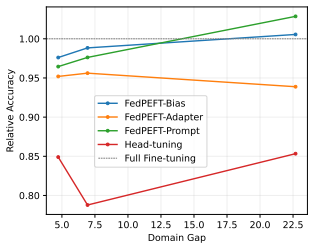
Model	Method	# Tuned Params × Clients	Comm. Cost	Resisc45	CIFAR-100	PCam
ViT-B	Full Fine-tuning	85.88M × 8	2.56GB	91.49±0.82	91.73±0.43	85.41±2.41
ViT-B	Full Fine-tuning	85.88M × 4	1.28GB	**92.13±0.87**	89.69±0.30	81.93±3.54
ViT-B	Full Fine-tuning	85.88M × 2	656MB	87.68±1.32	87.03±0.18	82.20±1.22
ViT-B	Full Fine-tuning	85.88M × 1	328MB	73.38±1.95	74.79±0.77	80.18±1.83
ViT-B	Head-tuning	0.08M × 8	2.44MB	77.30±1.03	72.45±0.08	74.82±2.40
ViT-B	Head-tuning	0.08M × 64	19.53MB	83.58±0.45	75.45±0.16	77.82±0.37
ShuffleNet	Full Fine-tuning	0.44M × 8	13.43MB	63.52±0.50	49.81±1.94	76.52±3.35
ViT-B	FedPEFT-Bias	0.18M × 8	5.49MB	**89.04±0.80**	**90.79±0.25**	85.51±0.66
ViT-B	FedPEFT-Adapter	0.23M× 8	7.02MB	87.20±0.78	87.74±0.55	78.67±1.85
ViT-B	FedPEFT-Prompt	0.17M× 8	5.19MB	83.35±0.76	89.78±0.84	**86.50±0.85**
Model	**Method**	**# Tuned Params × Clients**	**Comm. Cost**	**UCF101**	**HMDB51**	**UCF-CRIME**
ViT-B	Full Fine-tuning	86.30M × 4	1.29GB	94.22±0.23	70.34±0.28	34.37±0.76
ViT-B	Full Fine-tuning	86.30M × 2	656MB	93.85±0.33	69.06±0.48	32.03±0.26
ViT-B	Full Fine-tuning	86.30M × 1	328MB	92.61±0.16	59.88±0.27	24.21±0.26
ViT-B	Head-tuning	0.08M × 4	1.22MB	88.57±0.30	64.98±0.32	32.03±0.76
ViT-B	Head-tuning	0.08M × 32	9.76MB	89.74±0.45	63.87±0.66	32.81±0.76
X3D-S	Full Fine-tuning	3.07M × 4	47.96MB	36.68±1.67	27.74±0.24	21.42±0.76
ViT-B	FedPEFT-Bias	0.18M × 4	2.75MB	92.34±0.30	69.37±0.32	34.38±0.52
ViT-B	FedPEFT-Adapter	0.23M× 4	3.51MB	92.82±0.44	69.96±0.24	33.76±0.26
ViT-B	FedPEFT-Prompt	0.17M× 4	2.60MB	**93.82±0.66**	**70.87±0.16**	**34.38±0.26**

find that full fine-tuning needs several orders of magnitude of communication to achieve a comparable result with FedPEFT. For instance, it needs at least **187**$\times$ and **477**$\times$ more parameters to reach and outperform FedPEFT on CIFAR-100. Interestingly, full fine-tuning performs well on Resisc45 where the domain gap is smaller, even when the participating-client number is low. However, when the domain gap increases, more participating clients will be needed to outperform FedPEFT, and finally it fails to outperform FedPEFT even without reducing the participating-client number on PCam where a large domain gap exists. Meanwhile, head-tuning lags behind most other approaches, but the performance is stable with different levels of domain gap, while the ShuffleNet model only achieves 71%, 55%, and 89% of accuracy on Resisc45, CIFAR-100, and PCam with **2.4**$\times$ the communication cost compared with FedPEFT-Bias. Besides, the standard deviation of FedPEFT is lower than most other solutions, especially when the domain gap is large, showing the stability of FedPEFT. For the video domain, the conclusion is consistent.

In Fig. 3, we also report the server accuracy that can be achieved for each method given the communication budgets using CIFAR-100 as an example. The communication cost per communication round for full fine-tuning is even higher than the total communication cost for FedPEFT to converge to similar final server accuracy. Meanwhile, all FedPEFT prototypes only require megabytes level communication, while full fine-tuning requires gigabytes level communication to reach a given target accuracy (*e.g.*, 85% in Fig. 3), showing the efficiency of FedPEFT. For the inter-prototype comparison,



(a) Extracted feature. The number in the bracket is the domain gap to the pre-training dataset.



(b) Relative accuracy of each method compared with full fine-tuning with different domain gaps.

Fig. 4: **Visualization and analysis of domain gap.**

FedPEFT-Bias stands out for its highest efficiency. We provide further discussions on the performance of each prototype in Section IV-C.

C. RQ2: Capability Analysis

To study and understand our second research question (**RQ2**, Section IV), we analyze the impact of the domain gap between the model pre-training dataset and the dataset for FL (Section IV-C) and systemically perform experiments on CIFAR-100 across different federated learning scenarios by varying client status and data distribution (Section IV-C).

Capability with Domain Gap. Domain gap is a realistic concern when deploying pre-trained models for downstream tasks. As shown in Tab. I, the performance on different datasets varies a lot. Besides the difference in the difficulty between all datasets, the domain gap to the pre-trained dataset is also a key concern here since Resisc45 is a remote sensing dataset, and PCam is a medical dataset. To further discuss

TABLE II: **Capability analysis for different federated learning settings on CIFAR-100 and UCF-101.** Training from scratch in more complicated settings will lead to a lower result than in the first setting, which is omitted here. Bold-style shows the best performance among all methods or among prototypes in FedFEFT.

Client	Method	Image			Video		
		# Tuned Params	Homogeneous	Heterogeneous	# Tuned Params	Homogeneous	Heterogeneous
$N = 16$ $M = 16$	Scratch	85.88M	38.44	35.72	86.30M	27.34	22.57
	Full Fine-tuning	85.88M	**93.70**	**93.50**	86.30M	**95.32**	**95.17**
	Head-tuning	0.08M	78.11	77.59	0.08M	89.90	85.44
	FedPEFT-Bias	0.18M	91.89	90.25	0.18M	93.14	92.88
	FedPEFT-Adapter	0.23M	90.21	88.77	0.23M	93.73	93.48
	FedPEFT-Prompt	0.17M	**92.09**	**90.37**	0.17M	**94.23**	**94.09**
$N = 16$ $M = 2$	Full Fine-tuning	85.88M	**93.32**	**87.01**	86.30M	**95.12**	**94.78**
	Head-tuning	0.08M	76.65	62.80	0.08M	89.68	84.32
	FedPEFT-Bias	0.18M	91.35	86.18	0.18M	93.05	92.64
	FedPEFT-Adapter	0.23M	89.48	80.08	0.23M	93.53	93.27
	FedPEFT-Prompt	0.17M	**91.60**	**85.54**	0.17M	**94.11**	**93.89**
$N = 64$ $M = 64$	Full Fine-tuning	85.88M	**93.66**	**92.81**	86.30M	**79.21**	**78.78**
	Head-tuning	0.08M	78.45	75.51	0.08M	59.56	52.74
	FedPEFT-Bias	0.18M	**92.71**	**91.71**	0.18M	78.21	75.33
	FedPEFT-Adapter	0.23M	90.50	89.26	0.23M	78.41	75.88
	FedPEFT-Prompt	0.17M	91.87	90.96	0.17M	**78.69**	**76.11**
$N = 64$ $M = 8$	Full Fine-tuning	85.88M	**93.50**	**92.09**	86.30M	**78.78**	**76.66**
	Head-tuning	0.08M	77.59	72.55	0.08M	56.54	46.62
	FedPEFT-Bias	0.18M	**92.49**	**91.02**	0.18M	77.19	74.42
	FedPEFT-Adapter	0.23M	90.39	88.05	0.23M	77.69	75.18
	FedPEFT-Prompt	0.17M	92.00	89.90	0.17M	**78.56**	**75.38**

the impact of the domain gap for each method, we use our default setting shown in Section IV-A on the image domain as an instance to visualize and quantify the domain gap between each downstream dataset and the pre-training dataset. Specifically, we adapt Linear Discriminant Analysis (LDA) for all extracted features for the test samples in each dataset from the pre-trained backbone to reduce the dimension. We compute the center of each dataset and then compute the **distance to the center** of the pre-training dataset as the quantifying result of the domain gap, as shown in Fig. 4a.

In Fig. 4b, we present the performance of all approaches with an increasing degree of domain gap compared to the ImageNet-21k pre-training dataset. Interestingly, full fine-tuning falls further behind as the data domain gap widens in the PCam scenario, largely unable to keep up with FedPEFT despite requiring a massive communication budget. This phenomenon when a pre-trained model meets out-of-domain data has been studied under centralized settings [62]. It was found that the pre-trained upstream representations are still meaningful even with a domain gap. Therefore, fully fine-tuning the backbone with out-of-domain data can damage the high-level semantics inside these upstream representations due to overfitting, especially when the data size is small. This is particularly relevant in FL, where overfitting and subsequent client drift [63, 4, 3] are prone to occur.

On the opposite end of the spectrum, by not fine-tuning the backbone at all, head-tuning maintains similar accuracy despite the domain gap. This shows the robustness of the pre-trained high-level semantics across domains, supporting the conclusion that **there is meaningful high-level semantics inside of the upstream representations**.

Still, the tight restriction on head-tuning is perhaps a bit too far, as the accuracy on all datasets is still low overall. Between the two extremes of head-tuning and full fine-tuning, FedPEFT approaches may be able to suitably adapt the upstream representations without excessively damaging them. Specifically, FedPEFT-Bias operates with parameter-level control for each parameter pair containing weight and bias terms. The representation can then preserve the high-level semantics by freezing the weight term (maintaining the direction in the feature space) and still adapting via the bias term (shifting in the feature space). FedPEFT-Adapter and FedPEFT-Prompt have slightly different mechanisms, controlling the backbone by transforming the intermediate hidden representations via adapters and prompts. Specifically, FedPEFT-Prompt adds additional hidden states before the original hidden states without changing the original representation, while FedPEFT-Adapter transforms hidden states into a new space. Consequently, FedPEFT-Prompt shows stronger robustness in handling larger domain gaps than FedPEFT-Adapter. Of these approaches, FedPEFT-Prompt is the most stable under the domain gap, surpassing full fine-tuning by 1.1% on PCam. Overall, we hypothesize that **more fine-tuning freedom will be better when the domain gap is minor, but moderate fine-tuning is needed to maintain, as well as control, the high-level semantics when the domain gap is large**.

Capability with Different FL Settings. In application scenarios, the setting of federated learning can vary substantially. It is important to show the capability to maintain high performance in diverse settings. We present results for all approaches with different client availability ratios and data distributions in Table II and draw the following conclusions from the experiments:

First, we see that **fine-tuning the pre-trained model shows a significant improvement over training from scratch**, especially in heterogeneous scenarios. This finding is in agree-

TABLE III: **Robustness analysis for privacy-preserving.** The red number indicates the performance difference.

Method	Image		Video	
	w/o DP	w/ DP	w/o DP	w/ DP
Full Fine-tuning	92.09	77.61 (-14.48)	94.22	86.34 (-7.88)
Head-tuning	72.55	62.20 (-10.35)	88.57	83.09 (-5.48)
FedPEFT-Bias	91.02	**84.98** (-6.04)	92.34	86.61 (-5.73)
FedPEFT-Adapter	88.05	79.05 (-9.00)	92.82	85.57 (-7.25
FedPEFT-Prompt	89.90	78.35 (-11.55)	93.82	**87.64** (-6.18)

ment with other very recent works [8, 7], which note the stabilization effect of pre-trained initialization in federated optimization. When only fine-tuning the head, the performance is still much better than training the entire model from scratch but remains low in comparison to other methods across all settings. We again find that head-tuning simply lacks adaptation ability, holding too closely to the upstream representation.

On the other hand, we find that **FedPEFT achieves comparable results ($\geq 95\%$) to full fine-tuning with less than 0.3% of the trainable parameters.** This ability to maintain accuracy performance in various scenarios is crucial for FL, as oftentimes, the exact setting and distributions are not known ahead of time. Meanwhile, for inter-prototype comparison, we find that **FedPEFT-Bias outperforms other prototypes in almost all settings in the image domain, while FedPEFT-Prompt shows leading performance among all prototypes in the video domain.** This provides guidance in application to choose the prototype.

D. RQ3: Robustness Analysis

In this section, we further investigate our third research question **(RQ3)** in two critical FL scenarios evaluated by CIFAR-100 and UCF-101.

Differential Privacy. A fundamental property of federated learning is privacy protection. However, various works [47, 48] have demonstrated how the client data can be reconstructed from the raw gradient updates received by the server in some scenarios. To protect client data privacy from such attacks, differential privacy (DP) [9, 64, 65, 66] has become standard practice. Therefore, we first study FedPEFT and other baselines under DP.

To integrate DP, we apply a Gaussian mechanism within the local optimization of each iteration [66] with $\varepsilon = 5$ and $\delta = 0.001$. We maintain the remaining FL settings as described in Section IV-A, and show the results in Table III. Interestingly, when comparing all methods, full fine-tuning experiences the sharpest drop with DP. This causes its accuracy to fall lower than all the FedPEFT prototypes. To understand this effect, we note that DP applies noise to all trainable parameter gradients. Full fine-tuning, therefore, requires such noise on all model parameters, resulting in a more pronounced negative effect on final performance. On the other hand, the other fine-tuning methods maintain some part of the backbone frozen and have significantly fewer trainable parameters on which adding noise is necessary, limiting the performance drop. Overall, FedPEFT allows for stronger accuracy in DP-enabled federated systems than even full fine-tuning while still maintaining extremely low communication needs.

TABLE IV: **Robustness analysis for data scarcity.** K indicates the total sample number of all clients.

Method	$K = 1000$		$K = 1500$		$K = 2000$	
	Image	Video	Image	Video	Image	Video
Full Fine-tuning	66.52	87.50	67.47	88.34	77.67	90.54
Head-tuning	52.13	83.46	56.52	85.56	60.15	86.76
FedPEFT-Bias	**76.40**	85.34	**81.14**	87.36	**83.83**	88.85
FedPEFT-Adapter	71.34	86.17	76.91	88.12	79.22	90.28
FedPEFT-Prompt	63.77	**87.46**	71.94	**88.22**	76.89	**90.45**

Data Scarcity. We explore another common yet challenging robustness condition in FL; that is when very little data is available on individual clients. Such data scarcity scenarios are even a tricky problem in centralized training. Fewer training data will incur damage to the pre-trained representation due to overfitting. In our evaluation for FL, we reduce the total sample number K to 1000, 1500, and 2000. As shown in Table IV, we find that FedPEFT outperforms full fine-tuning and head-tuning under such low-data scenarios, further revealing its capability to appropriately adapt pre-trained representations to the FL task at hand.

For the inter-prototype comparison, FedPEFT-Bias and FedPEFT-Prompt remain leading the performance in image and video domains, consistent with the conclusion in common scenarios, showing their robustness.

E. Insights and Takeaways

Our research findings contribute significant insights to leverage parameter-efficient fine-tuning in federated learning, aiming at reducing communication costs.

- FedPEFT stands out under stringent communication budgets by offering significant advantages over traditional approaches, such as reducing the number of participating clients, utilizing smaller models, or solely training the classification head. Remarkably, the total communication overhead for 50 rounds in the FedPEFT framework is less than that of a single round in a conventional FL setting.
- In scenarios without communication limitations, FedPEFT can achieve server accuracies exceeding 95% while only requiring less than 0.3% of the parameters to be trainable.
- The continuum of fine-tuning freedom, ranging from Full fine-tuning through FedPEFT-Bias, FedPEFT-Adapter, FedPEFT-Prompt, to Head-tuning, varies across different scenarios. Typically, FedPEFT-Bias and FedPEFT-Prompt emerge as the top performers in image and video processing tasks. Notably, FedPEFT-Prompt demonstrates superior adaptability when bridging larger domain gaps between pre-trained models and downstream tasks.
- The effectiveness of FedPEFT is further illustrated in real-world applications characterized by stringent privacy requirements or data scarcity. Even under such conditions, FedPEFT-Bias and FedPEFT-Prompt maintain exceptional performance across image and video domains.

V. Conclusion

In this paper, we introduce FedPEFT, a new federated learning framework leveraging strong pre-trained models and massively reducing communication costs. We integrate three

effective prototypes within the FedPEFT framework: Bias, Adapter, and Prompt. With a thorough empirical study, we then evaluate FedFEFT and other baselines in three key areas: communication, capability, and robustness. We find FedPEFT to be a promising approach for practical FL systems, capable of handling many of the harsh conditions in FL while alleviating the critical communication bottleneck. As a general framework, FedPEFT can also be leveraged with other PEFT methods and in application domains other than computer vision. We hope this work can inspire new perspectives in federated learning through the combined innovation of strong pre-trained models and parameter-efficient fine-tuning methodologies.

VI. Acknowledgement

This work is partially supported by the NSF/Intel Partnership on MLWiNS under Grant No. 2003198 and the NSF Grant No. 2008447.

References

[1] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[2] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated Optimization in Heterogeneous Networks, April 2020. arXiv:1812.06127.

[3] Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR, 2020.

[4] Matias Mendieta, Taojiannan Yang, Pu Wang, Minwoo Lee, Zhengming Ding, and Chen Chen. Local Learning Matters: Rethinking Data Heterogeneity in Federated Learning. *arXiv:2111.14213 [cs]*, March 2022. arXiv: 2111.14213.

[5] Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. *arXiv preprint arXiv:2111.04263*, 2021.

[6] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H Vincent Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33:7611–7623, 2020.

[7] John Nguyen, Kshitiz Malik, Maziar Sanjabi, and Michael Rabbat. Where to Begin? Exploring the Impact of Pre-Training and Initialization in Federated Learning, June 2022. arXiv:2206.15387 [cs].

[8] Hong-You Chen, Cheng-Hao Tu, Ziwei Li, Han-Wei Shen, and Wei-Lun Chao. On Pre-Training for Federated Learning, June 2022. arXiv:2206.11488 [cs].

[9] Peter Kairouz et al. Advances and Open Problems in Federated Learning. *arXiv:1912.04977*, March 2021.

[10] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data, February 2017. arXiv:1602.05629 [cs].

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, June 2021. arXiv:2010.11929 [cs].

[12] OpenAI. Gpt-4 technical report, 2023.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Alex Krizhevsky. Learning Multiple Layers of Features from Tiny Images. *University of Toronto*, 2012.

[15] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pages 2921–2926. IEEE, 2017.

[16] Rishi Bommasani et al. On the Opportunities and Risks of Foundation Models, July 2022. arXiv:2108.07258 [cs].

[17] Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. TinyTL: Reduce Memory, Not Parameters for Efficient On-Device Learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 11285–11297. Curran Associates, Inc., 2020.

[18] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual Prompt Tuning, July 2022. arXiv:2203.12119 [cs].

[19] Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. AdapterHub: A Framework for Adapting Transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online, 2020. Association for Computational Linguistics.

[20] Hao Chen, Ran Tao, Han Zhang, Yidong Wang, Wei Ye, Jindong Wang, Guosheng Hu, and Marios Savvides. Conv-Adapter: Exploring Parameter Efficient Transfer Learning for ConvNets, August 2022. arXiv:2208.07463 [cs].

[21] Qinbin Li, Bingsheng He, and Dawn Song. Model-Contrastive Federated Learning, March 2021. arXiv:2103.16257 [cs].

[22] Jianyu Wang, Qinghua Liu, Hao Liang, Gauri Joshi, and H. Vincent Poor. Tackling the Objective Inconsistency Problem in Heterogeneous Federated Optimization, July 2020. arXiv:2007.07481 [cs, stat].

[23] Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. Bayesian Nonparametric Federated Learning of Neural Networks, May 2019. arXiv:1905.12022 [cs, stat].

[24] Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated Learning: Strategies for Improving Communication Efficiency, October 2017. arXiv:1610.05492 [cs].

[25] Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and H. Brendan McMahan. Distributed Mean Estimation with Limited Communication, September 2017. arXiv:1611.00429.

[26] Jenny Hamer, Mehryar Mohri, and Ananda Theertha Suresh. FedBoost: A Communication-Efficient Algorithm for Federated Learning. In *Proceedings of the 37th International Conference on Machine Learning*, pages 3973–3983. PMLR, November 2020. ISSN: 2640-3498.

[27] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. Parameter-Efficient Image-to-Video Transfer Learning, June 2022. arXiv:2206.13559 [cs].

[28] Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. Few-Shot Parameter-Efficient Fine-Tuning is Better and Cheaper than In-Context Learning, August 2022. arXiv:2205.05638 [cs].

[29] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[30] Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Lam Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks, March 2022. arXiv:2110.07602 [cs].

[31] Elad Ben Zaken, Shauli Ravfogel, and Yoav Goldberg. Bit-

fit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*, 2021.

[32] Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*, 2021.

[33] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. AdapterFusion: Non-Destructive Task Composition for Transfer Learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 487–503, Online, 2021. Association for Computational Linguistics.

[34] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021.

[35] Hyojin Bahng, Ali Jahanian, Swami Sankaranarayanan, and Phillip Isola. Exploring Visual Prompts for Adapting Large-Scale Models, June 2022. arXiv:2203.17274 [cs].

[36] Taojiannan Yang, Yi Zhu, Yusheng Xie, Aston Zhang, Chen Chen, and Mu Li. Aim: Adapting image models for efficient video action recognition. *arXiv preprint arXiv:2302.03024*, 2023.

[37] Yuan Yao, Ao Zhang, Zhengyan Zhang, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. CPT: Colorful Prompt Tuning for Pre-trained Vision-Language Models, May 2022. arXiv:2109.11797.

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

[39] Shibo Jie and Zhi-Hong Deng. Convolutional bypasses are better vision transformer adapters. *arXiv preprint arXiv:2207.07039*, 2022.

[40] Junting Pan, Ziyi Lin, Xiatian Zhu, Jing Shao, and Hongsheng Li. St-adapter: Parameter-efficient image-to-video transfer learning. *Advances in Neural Information Processing Systems*, 35:26462–26477, 2022.

[41] Qiankun Gao, Chen Zhao, Yifan Sun, Teng Xi, Gang Zhang, Bernard Ghanem, and Jian Zhang. A unified continual learning framework with general parameter-efficient tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11483–11493, 2023.

[42] Zhuo Zhang, Yuanhang Yang, Yong Dai, Qifan Wang, Yue Yu, Lizhen Qu, and Zenglin Xu. Fedpetuning: When federated learning meets the parameter-efficient tuning methods of pre-trained language models. In *Annual Meeting of the Association of Computational Linguistics 2023*, pages 9963–9977. Association for Computational Linguistics (ACL), 2023.

[43] Haodong Zhao, Wei Du, Fangqi Li, Peixuan Li, and Gongshen Liu. Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

[44] Tuo Zhang, Lei Gao, Chaoyang He, Mi Zhang, Bhaskar Krishnamachari, and Salman Avestimehr. Federated Learning for Internet of Things: Applications, Challenges, and Opportunities. *arXiv:2111.07494 [cs]*, March 2022. arXiv: 2111.07494.

[45] Jinyu Chen, Wenchao Xu, Song Guo, Junxiao Wang, Jie Zhang, and Haozhao Wang. Fedtune: A deep dive into efficient federated fine-tuning with pre-trained transformers. *arXiv preprint arXiv:2211.08025*, 2022.

[46] Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Delta tuning: A comprehensive study of parameter efficient methods for pre-trained language models. *arXiv preprint 2203.06904*, 2022.

[47] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating Gradient Inversion Attacks and Defenses in Federated Learning, November 2021. arXiv:2112.00059.

[48] Ali Hatamizadeh, Hongxu Yin, Holger Roth, Wenqi Li, Jan Kautz, Daguang Xu, and Pavlo Molchanov. GradViT: Gradient Inversion of Vision Transformers, March 2022. arXiv:2203.11894 [cs].

[49] Dimitar I Dimitrov, Maximilian Baader, Mark Niklas Müller, and Martin Vechev. Spear: Exact gradient inversion of batches in federated learning. *arXiv preprint arXiv:2403.03945*, 2024.

[50] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.

[51] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition, 2020.

[52] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. ImageNet-21K Pretraining for the Masses, August 2021. arXiv:2104.10972 [cs].

[53] Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: Benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[54] Bastiaan S Veeling, Jasper Linmans, Jim Winkens, Taco Cohen, and Max Welling. Rotation Equivariant CNNs for Digital Pathology. June 2018. _eprint: 1806.03962.

[55] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017.

[56] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[57] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011.

[58] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6479–6488, 2018.

[59] Ross Wightman. PyTorch Image Models, 2019. Publication Title: GitHub repository.

[60] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021.

[61] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint 2203.12602*, 2022.

[62] Zhiqiang Shen, Zechun Liu, Jie Qin, Marios Savvides, and Kwang-Ting Cheng. Partial Is Better Than All: Revisiting Fine-tuning Strategy for Few-shot Learning, February 2021. arXiv:2102.03983 [cs].

[63] Farshid Varno, Marzie Saghayi, Laya Rafiee, Sharut Gupta, Stan Matwin, and Mohammad Havaei. Minimizing Client Drift in Federated Learning via Adaptive Bias Estimation. *arXiv:2204.13170 [cs]*, April 2022. arXiv: 2204.13170.

[64] M. A. P. Chamikara, P. Bertok, I. Khalil, D. Liu, S. Camtepe, and M. Atiquzzaman. Local Differential Privacy for Deep Learning. *IEEE Internet of Things Journal*, 7(7):5827–5842, July 2020. arXiv:1908.02997 [cs].

[65] Cynthia Dwork. Differential Privacy: A Survey of Results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, Lecture Notes in Computer Science, pages 1–19, Berlin, Heidelberg, 2008. Springer.

[66] Cynthia Dwork, Aaron Roth, and others. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014. Publisher: Now Publishers, Inc.