2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 1-5, 2023. Detroit, USA

CEFHRI: A Communication Efficient Federated Learning Framework for Recognizing Industrial Human-Robot Interaction

Umar Khalid[1], Hasan Iqbal[2], Saeed Vahidian[3], Jing Hua[2], Chen Chen[1]

***Abstract*— Human-robot interaction (HRI) is a rapidly growing field that encompasses social and industrial applications. Machine learning plays a vital role in industrial HRI by enhancing the adaptability and autonomy of robots in complex environments. However, data privacy is a crucial concern in the interaction between humans and robots, as companies need to protect sensitive data while machine learning algorithms require access to large datasets. Federated Learning (FL) offers a solution by enabling the distributed training of models without sharing raw data. Despite extensive research on Federated learning (FL) for tasks such as natural language processing (NLP) and image classification, the question of how to use FL for HRI remains an open research problem. The traditional FL approach involves transmitting large neural network parameter matrices between the server and clients, which can lead to high communication costs and often becomes a bottleneck in FL. This paper proposes a communication-efficient FL framework for human-robot interaction (CEFHRI) to address the challenges of data heterogeneity and communication costs. The framework leverages pre-trained models and introduces a trainable spatiotemporal adapter for video understanding tasks in HRI. Experimental results on three human-robot interaction benchmark datasets: HRI30, InHARD, and COIN demonstrate the superiority of CEFHRI over full fine-tuning in terms of communication costs. The proposed methodology provides a secure and efficient approach to HRI federated learning, particularly in industrial environments with data privacy concerns and limited communication bandwidth. Our code is available at https://github.com/umarkhalidAI/CEFHRI-Efficient-Federated-Learning.**

I. Introduction

Human-robot interaction (HRI) is a rapidly growing field encompassing social [1] and industrial applications [2], [3]. In social settings, HRI involves interactions for entertainment, education, therapy, and personal assistance [4]. Conversely, Industrial HRI focuses on collaboration between humans and robots in industrial environments. Comprehending Industrial HRI is vital for the design and utilization of secure and efficient robotic systems in industrial environments, ultimately leading to increased productivity and optimized interaction between humans and robots. In the context of Industrial HRI, machine learning plays a crucial role in facilitating the adaptability of robots to ever-changing and intricate industrial environments [5]. This enhances their capability to execute tasks precisely and autonomously, while also reducing the potential for accidents and harm to human workers.

Regarding the interaction between humans and robots in industrial settings, data privacy is an essential concern as companies need to protect sensitive data while machine learning algorithms often require access to large datasets to make accurate predictions [6]. Federated Learning (FL) [7] offers a solution by enabling distributed training without sharing raw data. FL ensures privacy while improving human-robot interaction and productivity. Despite its potential, FL faces challenges in remote locations with limited bandwidth, hindering communication and data transfer, impacting its widespread deployment and effective utilization [8]. The substantial communication costs in FL due to parameter/data transmission between clients and servers also present a bottleneck.

Our Approach. To address the FL communication cost challenge, we propose a communication-efficient Federated Learning (FL) framework for Human-Robot Interaction (HRI) action recognition, named CEFHRI. We leverage pre-trained video models and fine-tune a carefully designed adapter specifically for video understanding on HRI datasets. The CEFHRI framework addresses challenges related to data heterogeneity [9] and large communication costs in HRI. The study is the first to investigate communication efficiency in FL for video understanding in the context of HRI. The key contributions of this study can be summarized as follows:

- This paper offers a pioneering study that systematically explores the effects of pre-training in the context of human-robot interaction (HRI) through FL.
- The study introduces a parameter-efficient fine-tuning framework, CEFHRI, within the Vision Transformer architecture. CEFHRI addresses challenges of data heterogeneity and large communication costs by using a light-weight trainable spatiotemporal adapter.
- Furthermore, the proposed CEFHRI framework is evaluated for preserving model privacy on the server while achieving efficient transfer learning.
- The proposed methodology is suitable for industrial environments prioritizing data privacy and facing communication and bandwidth constraints, providing a communication-efficient alternative for HRI federated learning. It can also be extended to other FL scenarios

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

[1]Center For Research in Computer Vision, University of Central Florida, Orlando, FL, USA umarkhalid@knights.ucf.edu and chen.chen@crcv.ucf.edu

[2]Dept. of Computer Science, Wayne State University, Detroit, MI, USA hasan.iqbal.cs@wayne.edu and jinghua@wayne.edu

[3]Dept. of Electrical and Computer Engineering, Duke University, Durham, NC, USA saeed.vahidian@duke.edu.

2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-9190-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IROS55552.2023.10341467

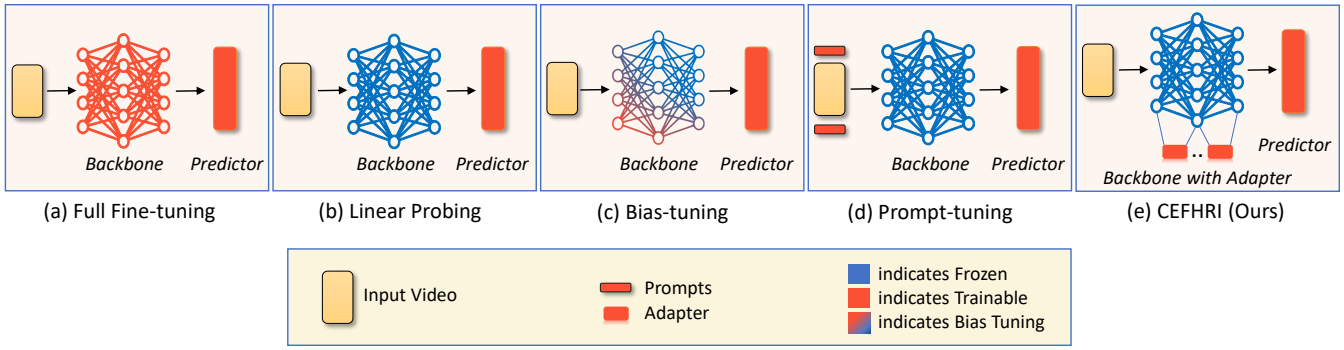

Fig. 1: Methods to fine-tune the pre-trained model (a) Full fine-tuning trains all the parameters of the pre-trained model. (b) Linear probing only fine-tunes the penultimate linear layer (c) Fine-tunes the bias term along with the penultimate layer. (d) Prompt tokens are concatenated with the input tokens, where prompt tokens are fine-tuned with the linear classifier layer. (e) Adapters are added to each layer of the model, where only the adapters and penultimate linear layers are fine-tuned.

involving video understanding tasks.

II. Related Work

a) Industrial Human-Robot Interaction: In recent years, industrial human-robot interaction (HRI) has emerged as a prominent research field, attracting considerable interest [10], [11]. However, there are a limited number of studies that have explored the computer vision domain specifically action recognition in industrial HRI settings [12]. One study by Huang et al. [13] proposes a model for gesture recognition that uses Convolutional Neural Networks (CNNs) to recognize hand gestures performed by human workers in industrial environments. Rizzi et al. [14] explores the use of deep learning to improve the performance of robotic grasping in industrial environments, by training a model to predict the optimal grasp point for objects based on visual cues. [15] applied deep learning using a combination of convolutional and recurrent neural networks to improve the perception of robots in an industrial setting, enabling them to recognize and locate objects on a conveyor belt.

b) Efficient Fine-Tuning: [16] proposes a novel structured pruning method for parameter-efficient fine-tuning that preserves important network structures while discarding unimportant connections. [17] realizes transfer learning in NLP tasks with adapter modules, which adds a few trainable parameters per task while keeping the backbone frozen. Some recent works [18]–[20] extend the adapter design to use the image foundation models for video understanding. [21], [22] propose prompt tuning for adapting language models. In the visual domain, [23] introduces visual prompt tuning as a highly efficient and effective approach for large-scale Transformer models in vision. Another efficient fine-tuning alternative is bias-tuning [24], [25] which is a sparse fine-tuning method that only fine-tunes a subset of the bias terms of the model during training.

c) Federated Learning: Federated learning is an innovative machine learning technique enabling model training across numerous decentralized devices or servers, all while avoiding the need to transfer data to a central server [26], [27]. Until now, FedAvg [26] has been widely considered the primary benchmark for federated learning. In FedAvg, the weight *aggregation* is performed by averaging the model weights obtained from different devices. FedAvg works well when the data is homogenous. However, in heterogeneous environments, the global model that is suitable for all clients faces convergence challenges. FedProx[28], FedAdapt [29], [30], FedNova [31] and SCAFFOLD [32] are some enhancements proposed to FedAvg which have attempted to develop modified versions of the algorithm that can handle non-IID data. In our evaluation, we establish that FedAvg has the tendency to achieve satisfactory performance in the heterogenous environment given a pre-trained foundation model.

Although there are some recent works on FL for HRI [33], [34], no study has yet explored HRI for action recognition. Our research constitutes the initial investigation into the video understanding of HRI within Federated Learning (FL) settings. Additionally, this study serves as a fundamental basis for the utilization of parametric-efficient fine-tuning in conjunction with pre-trained models for the purpose of video comprehension within FL.

III. Efficient Federated Learning for Human Robot Interaction Recognition

A. Federated Learning in HRI Action Recognition

Industrial HRI action recognition is a multi-class classification task where, given T training samples of a video dataset $X^T = \{(x_t, y_t)_{t=1}^T\}$, with $y \in \{0, 1, .., C-1\}$ for C classes, the goal is to achieve accurate classification on a set of unseen videos $X^u = \{x_1, ..., x_M\}$, where M is the number of videos in the test set. To set up the HRI recognition task in the FL setting, we assume a system of N clients that can coordinate with the centralized server without sharing their

local data. Further, let $\mathcal{X}$ be a subset of $\mathbb{R}^p$ representing the instance space, $\mathcal{Z}$ a subset of $\mathbb{R}^d$ denoting the latent feature space and $\mathcal{Y}$ a subset of $\mathbb{R}$ representing the output space. The server model F, parameterized by $\Theta := [\boldsymbol{\theta}^f; \boldsymbol{\theta}^p]$, comprises two components: a feature extractor $f : \mathcal{X} \rightarrow \mathcal{Z}$ parameterized by $\boldsymbol{\theta}^f$, and a predictor $p : \mathcal{Z} \rightarrow \Delta^{\mathcal{Y}}$ parameterized by $\boldsymbol{\theta}^p$, where $\Delta^{\mathcal{Y}}$ is the simplex over $\mathcal{Y}$.

In each communication round, the server randomly selects a subset of available clients $\mathbb{S}_r \in [\mathbb{N}]$ with $|\mathbb{S}_r| = n$ and broadcasts the model to all clients. The clients, upon receiving the model from the server, perform several steps of stochastic gradient descent (SGD) updates on their local training data $T_k \subset X^T$, obtain the updated model, and send their model parameters $\Theta := [\boldsymbol{\theta}_k^f; \boldsymbol{\theta}_k^p]_{k=1}^n$ back to the server.

Assuming the global model is initialized by $F(\Theta)$, the clients minimize their loss in each round r using SGD training as follows:

$$\min_{\Theta} F_k^{(r)}(\Theta) = \frac{1}{|T_k|} \sum_{t=1}^{|T_k|} \ell_k(\Theta, x_t), \tag{1}$$

where ℓ is the loss function, r is the communication round, and $|T_k|$ represents the number of the training samples of the kth client. The server collects all the parameter updates from clients and conducts model averaging as [26]

$$\Theta^{(r+1)} = \sum_{k=1}^{n} \frac{|T_k|}{\sum_{i=1}^{n} |T_i|} \Theta_k^{(r)}. \tag{2}$$

This parameter exchange between the clients and server goes on from $r = 0$ to $r = R - 1$ until the convergence of the global model.

B. Problem definition

As stated in Section III-A, the global model convergence in FL is accomplished after several rounds of parameter exchange between the local nodes and the global server. The procedure is repeated for r rounds. In this paper, we let $\mathbb{C}$ represent the communication cost associated with the FedAvg baseline. This cost is directly related to the number of parameters shared by the clients as $\mathbb{C} \propto |\boldsymbol{\theta}| \cdot |\mathbb{S}_r|$, where $\boldsymbol{\theta} \subseteq \Theta$ the parameters that need to be transmitted. We aim to minimize $\mathbb{C}$ without an accuracy drop for the HRI recognition task. As the communication cost is directly related to the number of trainable parameters, one intuitive method is to freeze the backbone $\boldsymbol{\theta}^f$ and only train the penultimate linear layer $\boldsymbol{\theta}^p$. However, the drawbacks of linear probing in terms of performance, which will be discussed in Section IV-B, have prompted us to introduce our federated learning framework for human-robot interaction recognition **(CEFHRI)** which keeps tunable parameters, $\boldsymbol{\theta} << \Theta$ using Vision Tansformers [35].

C. Proposed Framework

In this section, we provide a succinct overview of the Vision Transformer (ViT) and its application in the video understanding task of FL followed by an introduction to spatial-temporal adaptation. We further discuss the utility of such adaptation to preserve server privacy.

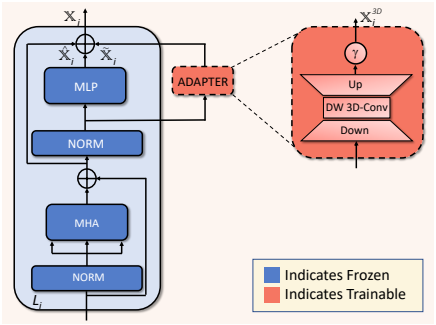

Fig. 2: The adapter design of the proposed CEFHRI framework. The adapter is inserted in the i^{th} layer of the backbone model while keeping other blocks frozen.

a) Overview: The transformer architecture [35] has been extensively utilized in vision applications such as video surveillance and action recognition. In this work, we investigate the role of FL in recognizing human-robot interaction which is another video understanding task. To this end, we introduce CEFHRI, an FL framework that employs the vision transformer for efficient decentralized HRI learning. Within the CEFHRI framework, the local and global models are customized by inserting an adapter module. Taking inspiration from the adapter design in [19], [36], we propose a variant of spatio-temporal adapter [19] architecture, called the ST-Adapter specifically designed for a video pre-trained model.

b) Adapter Design: The designed adapter architecture allows for the preservation of both the spatial and temporal characteristics of videos, while simultaneously facilitating efficient fine-tuning. The present discourse will concentrate on creating adapters for deep transformer backbones. In particular, we have introduced a depth-wise 3D convolu-

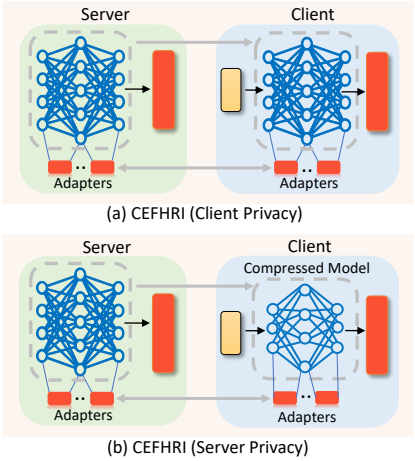

Fig. 3: (a) CEFHRI with client data privacy. (b) CEFHRI with server model privacy

tion layer in a standard configuration, placed between the bottlenecks of the vanilla adapter [36] and parallel to the MLP block. The visual representation of this arrangement is depicted in the accompanying Fig. 2.

Since the ST-Adapter branch is inserted alongside the MLP block, we term the adapted MLP block as the ST-adapted MLP and its output as $\mathbb{X}^{3D}$. The 3D branch will transform the features in the subsequent manner:

$$\tilde{\mathbb{X}}_i = \gamma \cdot ((DWConv3D(NL(\hat{\mathbb{X}}_i) \cdot W_{DOWN})) \cdot W_{UP}) \quad (3)$$

where $DWconv3D$ represents the depth-wise 3D-convolution. Here, γ is a tuneable parameter to scale the adapter output as in [36]. Before applying $DWConv3D$, the features are reshaped from $\mathbb{X}_i^{'} \in \mathbb{R}^{N \times \hat{d}}$ to $\mathbb{X}_i^{''} \in \mathbb{R}^{D \times H \times W \times \hat{d}}$, where $\mathbb{X}^{'} = NL(\hat{\mathbb{X}}_i)$. $\tilde{\mathbb{X}}^i$ features are then fused with the MLP branch output features to generate $\mathbb{X}^{3D}$,

$$\mathbb{X}_i^{3D} = \mathbf{MLP}(NL(\hat{\mathbb{X}}_i)) + \tilde{\mathbb{X}}_i \quad (4)$$

Further, $\hat{\mathbb{X}}$ features are fused with $\mathbb{X}^{3D}$ using a residual connection as following,

$$\mathbb{X}_i = \mathbb{X}_i^{3D} + \hat{\mathbb{X}}_i \quad (5)$$

We provide a detailed comparison of the proposed adapter design with other parametric-efficient finetuning techniques in Section IV-B under various FL settings. We further evaluate the existing adapter designs from the literature in Section V.

Algorithm 1 The CEFHRI Framework

Server: initialize the foundation model F parametrized by Θ, and the compressed model F^c with Θ^c
Clients: fetch the compressed model weights Θ^c, and initialize the tunable adapter weights, $\boldsymbol{\theta}$
Require: $\mathbb{N}$ clients, sampling rate $\mathbb{R} \in (0, 1]$, communication round budget R, shared weight $\boldsymbol{\theta} \subseteq \Theta$.

for each round $r = 0, 1, 2, ..., R-1$ **do**
  $\mathbb{S}_r \leftarrow$ Server samples $|\mathbb{S}|$ clients from N clients

  Client update $(k, \Theta^c, \boldsymbol{\theta}^{(r)})$**:**
    for each client $k \in \mathbb{S}_r$ **do**
      $F_k \leftarrow \{\Theta^c, \boldsymbol{\theta}_k^{(r)}\}$
      training F_k by SGD on $\mathbb{D}_k$ for E epochs
      $\boldsymbol{\theta}_k^{(r+1)} \leftarrow \boldsymbol{\theta}_k^{(r)} - \eta \nabla F_k$
      return $\boldsymbol{\theta}_k^{(r+1)}$ to server

  Server executes:
    for each client $k \in \mathbb{S}_r$ **do**
      $\boldsymbol{\theta}_k^{(r+1)} \leftarrow$ ClientUpdate$(k, \Theta^c, \boldsymbol{\theta}^{(r)})$
      $\theta^{(r+1)} \leftarrow$ aggregate updated parameters $\boldsymbol{\theta}_k^{(r+1)}$ by clients as in Eq. 2

c) Server-Side Privacy: We have additionally expanded the applicability of the proposed adapter design to a situation wherein the foundation model owner, i.e., the server is precluded from disclosing their model to the data owner [37]. To overcome this challenge, we devise a strategy to fine-tune the proposed adapter on the data owner's dataset without accessing the complete pre-trained model weights. In order to safeguard model ownership while simultaneously enhancing efficiency, we have implemented a technique known as lossy compression on the frozen backbone model as shown in Fig. 3. We leverage the findings of [38] which reveal that the discriminative information crucial for accurate classification is predominantly captured within the class tokens of the final few blocks. Specifically, we have selectively dropped a few layers from the model to produce a compressed variant, an adapter counselor, F^c with parameters Θ^c such that $\Theta^c < \Theta$. The purpose of utilizing F^c is to furnish approximate gradient directions to update the adapters, while simultaneously maintaining similarity to the original frozen component weights, Θ. Nonetheless, it is imperative that the F^c precision be restrained, as a higher degree of accuracy could potentially divulge information regarding the original model. Furthermore, a smaller F^c size facilitates a more efficient fine-tuning process for downstream users. We report our results for various compression ratios in Section V. The outline of the proposed framework with client data privacy and server model privacy has been described in Algorithm 1.

IV. Experiments

We evaluate CEFHRI for downstream human-robot interaction task across a wide range of FL settings. The experimental setup is described in Section IV-A, and the effectiveness of CEFHRI is demonstrated in Section IV-B.

A. Experiment Setup

a) Architecture and Datasets: In our experiments, we use Kinetics-400 [39] dataset for pre-training. For the downstream FL task, we select three datasets with varying degrees of domain gap as compared to Kinetics-400: (i) InHARD [40], (ii) HRI30 [12] and (iii) COIN Dataset [41]. We adopted experimental settings from [36], [42]. We use plain ViT-B/16 [35] with supervised pre-train weights from VideoMAE [42] official repository, where pre-trained checkpoints on Kinetics-400 [39] dataset are publicly available. An extra BatchNorm layer [43] without affine transformation is inserted before the penultimate layer as in [36]. For all datasets, we use 8 frames with a temporal stride of 4. The tubelet size is set to 2 as in VideoMAE's default settings [42]. The VideoMAE codebase includes a downsampling layer that converts original frames based on the tubelet size ratio. Hence, the final number of tokens for the transformer block is $4 \times 14 \times 14$, where 4 is the downsampled number of frames while the original input video has 8 frames.

b) Evaluation Metrics: We compare the proposed CEFHRI with four commonly used fine-tuning baselines: (1) Full Fine-tuning, (2) Linear Probing (fine-tuning classification head only), (3) Bias-Tuning, and (4) Prompt-Tuning. Here full fine-tuning indicates that all the parameters of the foundation model are fine-tuned on the downstream dataset. Linear probing indicates that

TABLE I: Human-robot interaction action recognition performance in terms of % accuracy under FL settings where maximum communication rounds, $R = 40$. All results represent the HRI recognition accuracy for different FL settings across three datasets. Here, N and $|\mathbb{S}_r|$ indicate total clients and sampled clients respectively.

Clients	Method	COIN			InHARD			HRI30		
		$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$	$\alpha = 0.1$	$\alpha = 0.5$	$\alpha = 1.0$
$N = 16, \lvert\mathbb{S}_r\rvert = 16$	Full Fine-Tuning	46.1±0.1	46.9±0.1	47.3±0.2	81.3±0.2	82.4±0.1	83.2±0.1	76.2±0.2	81.5±0.1	87.2±0.1
	Linear Probing	33.4±0.1	34.1±0.1	34.4±0.2	44.7±0.2	51.2±0.4	55.6±0.1	35.1±0.1	36.6±0.1	42.6±0.2
	Bias-Tuning	33.7±0.1	34.6±0.1	34.8±0.1	47.5±0.2	54.2±0.1	60.6±0.1	45.7±0.1	48.8±0.1	50.1±0.2
	Prompt-Tuning	36.1±0.1	36.9±0.1	37.1±0.2	69.8±0.1	70.6±0.3	71.1±0.1	67.6±0.1	72.4±0.1	75.8±0.2
	CEFHRI	**44.6**±0.1	**45.1**±0.1	**45.8**±0.3	**80.6**±0.1	**81.3**±0.1	**82.1**±0.3	**73.6**±0.1	**81.2**±0.1	**85.8**±0.2
$N = 16, \lvert\mathbb{S}_r\rvert = 4$	Full Fine-Tuning	38.8±0.3	43.9±0.1	45.2±0.3	73.56±0.2	74.4±0.1	77.2±0.1	73.8±0.1	82.6±0.1	85.1±0.2
	Linear Probing	29.4±0.3	32.6±0.2	32.9±0.3	33.3 ±0.2	38.8±0.1	42.6 ±0.1	39.6±0.1	35.6±0.1	34.8±0.2
	Bias-Tuning	30.5±0.2	32.9±0.2	33.4±0.3	39.5±0.2	42.2±0.1	48.1±0.1	49.3±0.1	47.6±0.1	45.0±0.2
	Prompt-Tuning	32.4±0.2	35.2±0.2	36.5±0.2	63.1±0.1	65.4±0.1	68.8±0.4	64.6±0.1	72.6±0.1	74.7±0.2
	CEFHRI	**38.6**±0.1	**43.1**±0.1	**44.2**±0.2	**72.9**±0.2	**73.3**±0.3	**76.5**±0.1	**73.2**±0.1	**80.6**±0.1	**81.5**±0.2

only the penultimate linear layer is fine-tuned. Bias-tuning [24], [44] aims to fine-tune only the bias parameters for the downstream task while prompt-tuning [21], [23] concatenates prompt tokens to the input embedding, where each prompt token is a learnable d-dimensional vector. The baselines are compared with CEFHRI with respect to two evaluation criteria (i) **Communication Efficiency,** (ii) **Human-Robot Interaction recognition**.

c) Training Settings: We distribute the training data between clients based on the Dirichlet distribution (with $\alpha \in \{0.1, 0.5, 1.0\}$) to achieve heterogeneous data partitioning across clients. Lower α indicates a higher degree of data heterogeneity [45]. In all experiments, we assume $N = 16$ clients are available, and we set the sampling rate to 25%, and 100% which means either $|\mathbb{S}_r| = 4$ or $|\mathbb{S}_r| = 16$ in each round. Each client performs $E = 8$ local epochs with a batch size of 8 before global aggregation is performed for $R = 40$ communication rounds. We keep the model weights frozen during the fine-tuning of CEFHRI techniques. We set $\gamma = 2.5$, while the bottleneck dimension, $\hat{d} = 64$ in our default settings for the adapter. We followed the training settings of [20] except the learning rate which we selected as 0.001. For prompt-tuning, the number of introduced tokens is set to 8 based on optimal performance as mentioned in [36]. Any modification to the default settings will be clearly stated.

B. Main Results

The results reported in this section ensure the client's data privacy with the assumption that the client has an access to the non-compressed version of the pre-trained model.

a) Human-Robot Interaction Recognition: Table I displays the interaction recognition results of the CEFHRI framework. Our assessment shows that the CEFHRI framework surpasses all other approaches, except for full fine-tuning, across all datasets and FL configurations. As reported in Table I, the CEFHRI framework achieves satisfactory adaptation performance despite having significantly fewer trainable parameters than full fine-tuning. Moreover, the results presented in Table I indicate that irrespective of the value of α, there is no substantial difference in the performance gap for CEFHRI. This highlights the crucial role of the pre-trained model in mitigating the impact of heterogeneous data. These findings are in agreement with recent studies conducted on image classification tasks [46], [47].

b) Communication Cost: In this paper, we only consider the uploading (i.e., from clients to server) communication cost, $\mathbb{C}$, for FedAvg [26] baseline defined as $\mathbb{C} = R \times |\boldsymbol{\theta}| \times |\mathbb{S}_r| \times$ 4B, where R stands for the total number of communication rounds; $|\mathbb{S}_r|$ denotes the number of participating clients, and $|\boldsymbol{\theta}| \subseteq |\Theta|$ represents the subset of the total number of model parameters[1] that are exchanged between each client and server in each round, with each parameter occupying 4 bytes (4B) of storage. With this in mind, we report the communication efficiency results against the pre-specified target accuracy for each dataset in Table II. Upon comparison to full fine-tuning, it becomes evident that the CEFHRI framework can achieve the target accuracy with a significantly reduced communication cost of at least $\approx 35\times$ lower. Despite the linear probing approach having the fewest tunable parameters, it falls short of attaining the target accuracy for any of the datasets being examined. Fig. 4 further illustrates the performance vs cost tradeoff between the baselines and CEFHRI. This suggests that the proposed parameter-efficient fine-tuning prototype of the CEFHRI framework is efficacious in mitigating communication overhead, without compromising recognition performance.

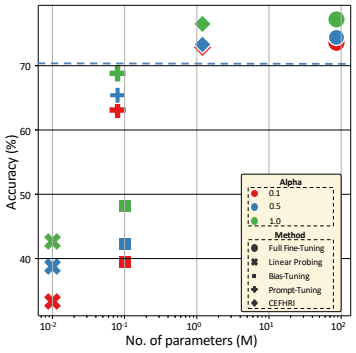

Fig. 4: Federated training accuracy vs # of parameters plot for all fine-tuning methods under consideration. The figure illustrates that only CEFHRI can achieve the target accuracy of 70% on InHard dataset given $R = 40$, $N = 16$, and $|\mathbb{S}_r| = 4$ for numerous levels of data heterogeneity.

[1]Each dataset has different # of classes, so the # of model parameters are different owing to varying linear classification head parameters.

TABLE II: The **communication cost** is computed with 4Bytes/parameter with $N = \mathbb{S} = 16$, $\alpha = 0.5$, and $R = 40$. The number in the bracket indicates the target action recognition accuracy %. — indicates that the target accuracy is not achieved under that particular setting. **Parameters** are the trainable parameters of each client. ↓ indicates smaller values are better.

Method	COIN (45)			InHARD (80)			HRI30 (80)		
	Parameters (M)↓	Rounds↓	Comm. cost↓	Parameters (M) ↓	Rounds ↓	Comm. cost ↓	Parameters (M)↓	Rounds ↓	Comm. cost ↓
Full Fine-Tuning	86.36	20	5.39× 20 GB	86.23	14	5.38×14 GB	86.25	21	5.39×21 GB
Linear Probing	0.13	-	-	0.01	-	-	0.01	-	-
Bias-Tuning	0.23	-	-	0.10	-	-	0.10	-	-
Prompt-Tuning	0.20	-	-	0.08	-	-	0.09	-	-
CEFHRI	1.33	35	85.12× 35 MB	1.20	38	76.80×38 MB	1.21	37	77.44×37 MB

c) CEFHRI vs Parametric Efficient Model: One straightforward solution to reduce the communication cost is to use tiny video networks such as the X3D-S mode [48]. Nevertheless, the architectures with low capacity ar incapable of achieving satisfactory performance even witl pre-training, as is evidenced in Table III. In particular, X3D-S [48] could only achieve 30.8% HRI recognition performanc on HRI30 [12], while CEFHRI yields 84.6% accuracy fo $\alpha = 0.5$. Our comprehension is that models with lov capacity are not effective in producing desirable perfor mance under data heterogeneity scenarios. Consequently, th CEFHRI approach offers a practical remedy for mitigatin the decrease in performance while still keeping the commu nication cost as minimal as that of smaller models, such a X3D-S, for the purpose of HRI recognition.

V. Ablation Studies

a) Impact of Model Initialization: To demonstrate the applicability of the pre-trained model in the CEFHRI framework, we undertake a more in-depth examination of two fundamental questions: **(1)** How does the model initialization impact the industrial HRI action recognition performance in FL settings? **(2)** To what extent can pre-training alleviate the accuracy drop caused by the heterogeneity of data from clients in the context of industrial HRI action recognition?

Our findings show that using a pre-trained model can drastically improve the performance of industrial HRI action recognition in FL. Fig. 5 demonstrates the significant advantage of using a pre-trained model over training from scratch for the InHARD [40], and HRI30 [12] datasets. It is evident that the server achieves notable performance within a fewer number of rounds. This is in contrast to random initialization, which is unable to narrow the performance gap even after 100 rounds for heterogeneous FL settings. Our results indicate that using pre-trained models not only outperforms random initialization but also effectively mitigates the data heterogeneity effect in the industrial HRI action recognition

TABLE III: Performance comparison with parameter-efficient model **X3D-S**. Here, $R = 40$, $N = 16$, and $|\mathbb{S}_r| = 4$. All results represent the averaged percentage accuracy of HRI recognition over several runs for $\alpha = 0.5$. ↑ indicates larger values are better and ↓ indicates smaller values are better. # of parameters indicates the tunable parameters.

Dataset	Architecture	# of Parameters ↓	Accuracy(%) ↑
COIN	X3D-S	3.34M	15.9±0.3
	CEFHRI	1.33M	43.1±0.1
InHARD	X3D-S	3.00M	26.9±0.1
	CEFHRI	1.20M	73.3±0.3
HRI30	X3D-S	3.03M	30.8±0.1
	CEFHRI	1.21M	80.6±0.1

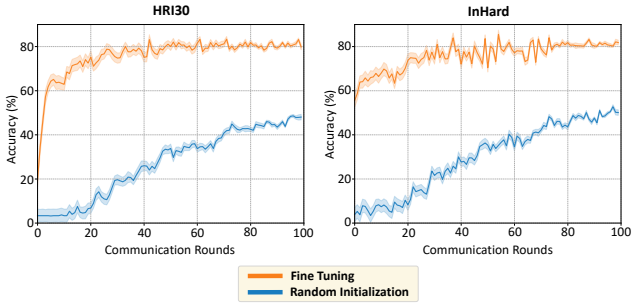

Fig. 5: FL results for the industrial HRI action recognition task reveal the advantages of utilizing the pre-trained model. The figure illustrates that employing a pre-trained model in FL saves communication costs while achieving superior performance. Here, $R = 100$, $N = 16$, and $|\mathbb{S}| = 16$.

b) Performance Gain of the Proposed Adapter Design: CEFHRI adapter is different in design from [20] and distinct in application from [19]. Firstly, the ST-adapter proposed by [19] is specifically designed for frozen CLIP models, and its effectiveness in FL has not been explored. Secondly, the sequential implementation doesn't maintain the original features achieved by the frozen backbone. We argue that the parallel design of the adapter maintains the original features as the adapter branch is added parallel to the original MLP block and only scaled feature aggregation is performed. As shown in Table IV, our analysis indicates that the parallel adapter structure outperforms the sequential design. As a result, we use the parallel ST-adapter design by default for the CEFHRI prototype, which distinguishes it structurally from that of [36].

TABLE IV: Performance comparison with different adapter designs. Here, $R = 40$, $N = 16$, and $|\mathbb{S}_r| = 4$. All results represent the averaged percentage accuracy of HRI action recognition over several runs for $\alpha = 0.5$.

Dataset	Adapter Design	Accuracy(%)
InHARD	Adaptformer [36]	55.9 ±0.3
	CLIP-Adapter [19]	52.1±0.1
	CEFHRI-Adapter	73.3±0.3
HRI30	Adaptformer [36]	60.7 ±0.3
	CLIP-Adapter [19]	57.1±0.1
	CEFHRI-Adapter	80.6±0.1

c) Performance while Preserving Server Privacy: In order to protect the privacy of the foundation model with l layers, we implement a model compression strategy by

TABLE V: Performance comparison with different # of layers dropped. Here, $R = 40$, $N = 16$, and $|\mathbb{S}| = 4$. All results represent the averaged percentage accuracy of HRI action recognition over several runs for $\alpha = 0.5$.

Dataset	Layers Dropped	Accuracy(%)
InHARD	1	72.3±0.3
	2	71.8±0.1
	3	70.1±0.1
	4	67.3±0.1
HRI30	1	79.8 ±0.2
	2	79.1±0.1
	3	77.5±0.1
	4	72.1±0.1

removing a specified number of layers, n from the ViT-B/16 [35] model. The resulting compressed model is fine-tuned using knowledge distillation, with mean square error, under the supervision of the original model using Kinetics400 [39] dataset for 20 epochs. The fine-tuned compressed model is then distributed to the clients as a point of reference. The clients then use this reference model to train the $l - n$ adapters, which are later inserted into the server model's last $l - n$ layers. The results of our experiments indicate that knowledge distillation was essential in attaining the best performance. We also find that dropping the last n layers produced superior results compared to dropping the first n layers of the transformer model, which is consistent with [38]. Therefore, the reported results in Table V are generated using the strategy of dropping the last n layers for model compression. Here, we can observe that as the # of removed layers is more than 3, the performance drop is significant.

VI. Conclusion

In this research, we present a new FL framework called CEFHRI, which aims to improve the performance of human-robot interaction recognition tasks in industrial settings through the utilization of pre-trained video models. To mitigate the communication overhead that is commonly encountered in FL systems, the CEFHRI framework proposes a parameter-efficient fine-tuning prototype. We conduct a comprehensive evaluation of the CEFHRI framework by comparing its performance against other baselines with regard to both recognition for human-robot interaction and communication cost. Our findings indicate that the CEFHRI framework not only effectively addresses the communication bottleneck issue but also outperforms other FL fine-tuning techniques, such as linear probing and bias-tuning. Additionally, our results show that the proposed CEFHRI adapter performs satisfactorily in scenarios where the downstream dataset has a major domain shift compared to the pre-trained dataset. In conclusion, our work sheds new light on the potential for improving communication efficiency in FL for video understanding tasks and lays the groundwork for future advancements in this area.

References

[1] C. Breazeal, Designing sociable robots. MIT Press, 2004.

[2] H. He, S. Li, and X. Chen, "Industrial robots: A survey on the recent progress and future directions," IEEE/CAA Journal of Automatica Sinica, vol. 6, no. 4, pp. 847–863, 2019.

[3] A. Kumar, J. Kini, A. Mian, and M. Shah, "Self supervised learning for multiple object tracking in 3d point clouds," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 3754–3761.

[4] H. Mahdi, S. A. Akgun, S. Saleh, and K. Dautenhahn, "A survey on the design and evolution of social robots—past, present and future," Robotics and Autonomous Systems, p. 104193, 2022.

[5] R. Choudhury, G. Swamy, D. Hadfield-Menell, and A. D. Dragan, "On the utility of model learning in hri," in 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2019, pp. 317–325.

[6] N. Karim, U. Khalid, A. Esmaeili, and N. Rahnavard, "Cnll: A semi-supervised approach for continual noisy label learning," in 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, pp. 3877–3887.

[7] M. Mendieta, T. Yang, P. Wang, M. Lee, Z. Ding, and C. Chen, "Local learning matters: Rethinking data heterogeneity in federated learning," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8397–8406, June 2022.

[8] S. Vahidian, M. Morafah, and B. Lin, "Personalized federated learning by structured and unstructured pruning under data heterogeneity," IEEE ICDCS, 2021.

[9] X. Li, Y. Zhang, L. Wang, and X. Zhang, "Pre-trained federated learning: A comprehensive survey," IEEE Communications Surveys & Tutorials, vol. 24, no. 1, pp. 702–729, 2022.

[10] A. Roitberg, A. Perzylo, N. Somani, M. Giuliani, M. Rickert, and A. Knoll, "Human activity recognition in the context of industrial human-robot interaction," in Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2014 Asia-Pacific. IEEE, 2014, pp. 1–10.

[11] A. Hentout, M. Aouache, A. Maoudj, and I. Akli, "Human–robot interaction in industrial collaborative robotics: a literature review of the decade 2008–2017," Advanced Robotics, vol. 33, no. 15-16, pp. 764–799, 2019.

[12] F. Iodice, E. De Momi, and A. Ajoudani, "Hri30: An action recognition dataset for industrial human-robot interaction," in 2022 26th International Conference on Pattern Recognition (ICPR). IEEE, 2022, pp. 4941–4947.

[13] J. Huang, S. Guo, J. Chen, and Y. Yang, "Convolutional neural networks for hand gesture recognition in industrial human-robot interaction," Journal of Intelligent Manufacturing, vol. 32, no. 5, pp. 1175–1187, 2021.

[14] A. Rizzi, E. Battaglia, A. Marino, A. Nava, M. Matteucci, and V. Caglioti, "Learning grasping affordances from visual cues for industrial robotics," Robotics and Autonomous Systems, vol. 112, pp. 14–24, 2019.

[15] G. Spina, P. Rocco, S. Aldegheri, D. Cattin, M. Bicego, and E. Menegatti, "Deep learning for visual perception of robotic manipulators in industrial environments," Robotics and Computer-Integrated Manufacturing, vol. 64, p. 101927, 2020.

[16] Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang, "Structured pruning for efficient neural network inference," in International Conference on Learning Representations, 2019.

[17] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for nlp," in International Conference on Machine Learning. PMLR, 2019, pp. 2790–2799.

[18] T. Yang, Y. Zhu, Y. Xie, A. Zhang, C. Chen, and M. Li, "Aim: Adapting image models for efficient video action recognition," arXiv preprint arXiv:2302.03024, 2023.

[19] J. Pan, Z. Lin, X. Zhu, J. Shao, and H. Li, "Parameter-Efficient Image-to-Video Transfer Learning," June 2022, arXiv:2206.13559 [cs]. [Online]. Available: http://arxiv.org/abs/2206.13559

[20] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "Adaptformer: Adapting vision transformers for scalable visual recognition," arXiv preprint arXiv:2205.13535, 2022.

[21] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. Association for Computational Linguistics, 2021, pp. 3045–3059.

[22] J. Zhang, S. Vahidian, M. Kuo, C. Li, R. Zhang, G. Wang, and Y. Chen, "Towards building the federated GPT: federated instruction tuning," CoRR, vol. abs/2305.05644, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.05644

[23] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual Prompt Tuning," July 2022, arXiv:2203.12119 [cs]. [Online]. Available: http://arxiv.org/abs/2203.12119

[24] H. Cai, C. Gan, L. Zhu, and S. Han, "TinyTL: Reduce Memory, Not Parameters for Efficient On-Device Learning," in Advances in Neural Information Processing Systems, vol. 33. Curran Associates, Inc., 2020, pp. 11 285–11 297. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/81f7acabd411274fcf65ce2070ed568a-Abstract.html

[25] E. B. Zaken, S. Ravfogel, and Y. Goldberg, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," arXiv preprint arXiv:2106.10199, 2021.

[26] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Artificial intelligence and statistics. PMLR, 2017, pp. 1273–1282.

[27] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," arXiv preprint arXiv:1806.00582, 2018.

[28] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," Proceedings of Machine learning and systems, vol. 2, pp. 429–450, 2020.

[29] S. Li, T. Wang, K. Zhou, Y. Niu, J. Yang, and S. Hu, "Fedadapt: Overcoming model degradation for federated learning with non-iid data," in Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2020, pp. 377–392.

[30] Y. Zhao, M. Li, L. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-iid data," arXiv preprint arXiv:1806.00582, 2018.

[31] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in Advances in Neural Information Processing Systems, vol. 33. Curran Associates, Inc., 2020, pp. 7611–7623.

[32] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "Scaffold: Stochastic controlled averaging for federated learning," in International Conference on Machine Learning. PMLR, 2020, pp. 5132–5143.

[33] J. J. Gamboa-Montero, F. Alonso-Martin, S. Marques-Villarroya, J. Sequeira, and M. A. Salichs, "Asynchronous federated learning system for human–robot touch interaction," Expert Systems with Applications, vol. 211, p. 118510, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417422015901

[34] X. Su, F. Yuan, R. Zhang, J. Liu, M. Boltz, and X. Zhao, "Deploying a human robot interaction model for dementia care in federated learning," in 2022 IEEE/ACM Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2022, pp. 184–185.

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," June 2021, arXiv:2010.11929 [cs]. [Online]. Available: http://arxiv.org/abs/2010.11929

[36] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition," May 2022, arXiv:2205.13535 [cs]. [Online]. Available: http://arxiv.org/abs/2205.13535

[37] G. Xiao, J. Lin, and S. Han, "Offsite-tuning: Transfer learning without full model," arXiv preprint arXiv:2302.04870, 2023.

[38] M. M. Naseer, K. Ranasinghe, S. H. Khan, M. Hayat, F. Shahbaz Khan, and M.-H. Yang, "Intriguing properties of vision transformers," Advances in Neural Information Processing Systems, vol. 34, pp. 23 296–23 308, 2021.

[39] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al., "The kinetics human action video dataset," arXiv preprint arXiv:1705.06950, 2017.

[40] M. Dallel, V. Havard, D. Baudry, and X. Savatier, "Inhard-industrial human action recognition dataset in the context of industrial collaborative robotics," in 2020 IEEE International Conference on Human-Machine Systems (ICHMS). IEEE, 2020, pp. 1–6.

[41] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J. Zhou, "Coin: A large-scale dataset for comprehensive instructional video analysis," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1207–1216.

[42] Z. Tong, Y. Song, J. Wang, and L. Wang, "Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training," arXiv preprint arXiv:2203.12602, 2022.

[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in International conference on machine learning. PMLR, 2015, pp. 448–456.

[44] E. B. Zaken, Y. Goldberg, and S. Ravfogel, "Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models," in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022. Association for Computational Linguistics, 2022, pp. 1–9.

[45] S. Vahidian, S. Kadaveru, W. Baek, W. Wang, V. Kungurtsev, C. Chen, M. Shah, and B. Lin, "When do curricula work in federated learning?" CoRR, vol. abs/2212.12712, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2212.12712

[46] H.-Y. Chen, C.-H. Tu, Z. Li, H.-W. Shen, and W.-L. Chao, "On Pre-Training for Federated Learning," June 2022, arXiv:2206.11488 [cs]. [Online]. Available: http://arxiv.org/abs/2206.11488

[47] J. Nguyen, K. Malik, M. Sanjabi, and M. Rabbat, "Where to Begin? Exploring the Impact of Pre-Training and Initialization in Federated Learning," June 2022, arXiv:2206.15387 [cs]. [Online]. Available: http://arxiv.org/abs/2206.15387

[48] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 203–213.