Communication-Efficient Federated Video Human Activity Recognition via Discriminative Subspace Discovery

Allassan Tchangmena A Nken[1] , **Susan McKeever**[3] , **Peter Corcoran**[2] and **Ihsan Ullah**[1,4]
[1]School of Computer Science, Visual Intelligence Lab, University of Galway, Galway, Ireland
[2]School of Electrical Engineering, University of Galway, Galway, Ireland
[3]School of Computer Science, Technological University Dublin, Dublin, Ireland
[4]Data Science Institute, University of Galway, Galway, Ireland
{a.tchangmenaanken1, ihsan.ullah, peter.corcoran}@universityofgalway.ie,
susan.mckeever@TUDublin.ie

Abstract

Federated learning (FL) enables privacy-preserving video human activity recognition without centralizing data. Despite its potential, FL remains fundamentally constrained by the high communication overhead incurred during training. Existing methods, such as model quantization, gradient sparsification and low-rank approximation, aim to mitigate this overhead but often at the cost of model expressivity, leading to reduced classification accuracy. In this work, we take a representation-centric perspective and show that for video-base activity recognition using pretrained embeddings, class-discriminative information lies in a very low-dimensional subspace, that preserves linear separability between activity classes. Motivated by this observation, we propose **Federated discriminative subspace discovery**, a framework that collaboratively identify and operate within such a subspace. Instead of communicating full classifier updates, each client first projects its local video embeddings using a shared random projection matrix. It then computes the covariance of the projected embeddings and transmits this statistic to the server for aggregation. The server reconstructs a global covariance matrix, performs singular value decomposition, and selects the top-k eigenvectors that define the global class-discriminative subspace. Subsequent communication occurs only in this compact subspace. Extensive experiments on UCF101, HMDB51, and Toyota-SmartHome demonstrate that, learning and sharing the classifier in the discovered subspace, reduces communication costs by up to 61%, with a moderate drop in activity recognition accuracy compared to the full model. Our code is publicly available at .

1 Introduction

Video-based human activity recognition (HAR) is a core task in computer vision, with applications ranging from surveillance to healthcare [Azar *et al.*, 2019; Benfold and Reid, 2011; Ijaz *et al.*, 2022]. While deep learning models achieve high accuracy, training them requires large-scale video datasets [Kay *et al.*, 2017; Hara *et al.*, 2018], often collected from distributed users or devices. Federated learning (FL) [McMahan *et al.*, 2017] has emerged as a promising paradigm for training models in a privacy-preserving manner, allowing clients to collaboratively learn a global model, without collecting or sharing data and by communicating only model updates with a server. Recent works have extended FL to video-based HAR for domains-specific activity recognition [Doshi and Yilmaz, 2022; Gautam *et al.*, 2024], as well as video self-supervised learning, few-shot learning, and representation learning [Rehman *et al.*, 2022; Tu *et al.*, 2024; Yang *et al.*, 2022].

However, the high dimensionality of video models, especially modern transformer-based backbones such as multiscale vision transformers (MViT) [Fan *et al.*, 2021], and 3D convolutional neural networks such as ResNet3D-18 [Hara *et al.*, 2018] results in substantial communication overhead during FL, limiting scalability and practicality. Prior work attempts to reduce communication or computation cost via model quantization [Molchanov *et al.*, 2019; Horvóth *et al.*, 2022; Amiri *et al.*, 2020], gradient sparsification [Alistarh *et al.*, 2018], pruning [Han *et al.*, 2015], or low-rank approximations [Grativol *et al.*, 2024; Guo *et al.*, 2024]. While effective, these approaches often trade off model capacity, causing a noticeable drop in recognition accuracy. Moreover, they operate directly on full model embeddings, ignoring structural properties of embedding's representations that could be exploited to reduce communication more efficiently. A model embedding is the feature vector produced by a pretrained network (e.g. MViT), that captures high-level semantic information relevant for downstream tasks such as classification. Working with embeddings is common in FL [He *et al.*, 2020; Tu *et al.*, 2024; Rehman *et al.*, 2022] as they are lightweight alternatives to the original high-dimensional inputs (e.g. video frames). Furthermore, linear classifiers trained on such embeddings achieve strong performance ([Fanì *et al.*, 2024; Xu *et al.*, 2023]), making them a natural choice for communication-efficient FL. Therefore, in this work we operate directly on **video embeddings** extracted from pretrained models.

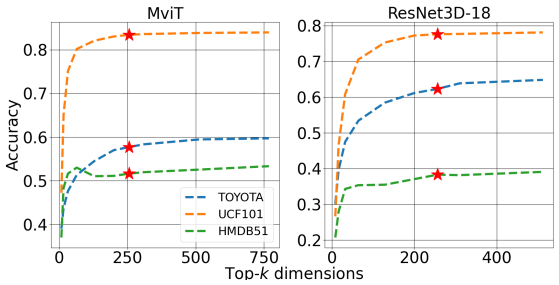

Figure 1: For both MViT and ResNet-3D-18, increasing the subspace dimensionality leads to an increase in accuracy, which then begins to plateau at approximately 256 dimensions.

To better understand the structure of video embeddings, we conducted an empirical analysis on three commonly used video HAR datasets: UCF101 [Soomro *et al.*, 2012], HMDB51 [Kuehne *et al.*, 2011], and Toyota-Smarthome [Das *et al.*, 2019]. For each dataset, we extracted video embeddings from the fully connected layer of a pretrained MViT backbone, resulting in a 768-dimensional (768-D) feature vector per video. We then calculated the covariance matrix of all embeddings and performed singular value decomposition (SVD) to obtain the eigenvectors corresponding to the top-k largest eigenvalues. Each embedding was projected into this subspace of eigenvectors, and a linear classifier was trained on progressively larger subspaces ($k \in \{30, 60, 120, ..., 768\}$ dimensions) to measure recognition accuracy. As illustrated in Figure 1, the accuracy increases sharply with dimensionality and saturates around 256-D (marked by a ★), indicating that most **class-discriminative** information resides in a compact **subspace** of the original 768-D embedding space, that preserves linear-separability of activity classes. We call a subspace *discriminative* if projecting data onto it preserves linear separability (and, ideally, classification margins) for the downstream labels under a classifier. We repeated the same procedure on 512-D ResNet3D-18 embeddings and observed a similar saturation trend, confirming the robustness of this observation across architectures. This observation is consistent with prior work demonstrating the low-rank structure and redundancy of deep visual representations [Sanyal *et al.*, 2018]. Our analysis suggest that, a simple classifier does not require the full dimensional embedding space; training within a 256-D class-discriminative subspace is sufficient.

Motivated by this insight, we propose **Federated discriminative subspace discovery**; a framework, enabling clients to collaboratively identify and train within this compact subspace for **communication-efficient FL**. Instead of exchanging full classifier updates, clients contribute only the information needed to estimate this shared subspace and learn a lighter classifier on top of it. Concretely, each client first projects its local video embeddings using a shared random projection matrix. It then computes the covariance of the projected embeddings and transmits a quantized version of this statistic to the server. The server aggregates the received covariance matrices, performs SVD, and extracts the top-k eigenvectors to form a global discriminative subspace.

Contribution. (1) We show that for pretrained video embeddings, linear classification accuracy saturates in a low-dimensional leading-eigenspace, suggesting that decision-relevant information concentrates in a compact subspace. (2) We propose a federated learning framework that recovers the shared discriminative subspace from, quantized covariance statistics of projected client embeddings, and uses this subspace as the exclusive training/communication interface. (3) Experiments on UCF101, HMDB51, and Toyota-Smarthome demonstrate that our approach reduces communication costs by up to 61%, with a moderate drop in activity recognition accuracy.

2 Background and Related Work

2.1 Federated learning (FL)

FL enables multiple clients to collaboratively train a global model without sharing raw data. In each training round, all clients receive the global model from a server, locally optimize it on their own data, and send updated model parameters back to the server, which aggregates them to form a new global model. A standard aggregation strategy is the Federated Averaging algorithm (FedAvg) [McMahan *et al.*, 2017], which updates the global model as a weighted average of the clients' local models:

$$\theta_i^{(t+1)} = \sum_{i=1}^{M} \frac{n_i}{N} \theta_i^{(t)}. \tag{1}$$

Here, $\theta_i^{(t)}$ denotes the model parameters trained by client i at round t, n_i is the client's dataset size, and N is the total number of samples across all clients. Client datasets may be independent and identically distributed (IID), where all clients sample their data from the same underlying distribution, or non-IID, where each client's data follows a different distribution: for example, due to varying activity categories, visual styles, or recording conditions. Such statistical heterogeneity causes local updates to diverge, slowing global convergence.

A key limitation of FedAvg is communication overhead: each client must upload and download full model parameters at every round. The **Total communication cost** (TCC) per round for M clients is:

$$\text{TCC} = 2MQ\,|\theta|\,, \tag{2}$$

where Q is the number of bits used to represent each parameter and $|\theta|$ is the total number of parameters in the model. For large models with millions of parameters, this cost becomes substantial. Moreover, under non-IID conditions, global convergence typically requires more communication rounds, further amplifying the communication bottleneck. These challenges motivate the need for communication-efficient FL techniques.

2.2 Communication Efficient FL

To reduce the communication cost in Equation 2, one can either decrease the number of transmitted parameters $|\theta|$, or (and) reduce the number of bits Q used to represent each parameter. We therefore review prior work that targets one or both of these directions.

Reducing the number of bits Q. This strategy, commonly referred to as *quantization*, reduces the precision of model parameters during transmission. Neural network parameters are typically stored as 32-bit floating-point (FP32) format; quantization compresses them to 16-bit, 8-bit, or even sub-8-bit formats to lower communication cost.

A large body of work explores quantization for communication-efficient FL. For example, Amiri and Gunduz [Amiri *et al.*, 2020] propose a lossy FL (LFL) algorithm in which both global models and clients updates are quantized prior to transmission in order to reduce bandwidth usage. However, LFL assumes that the same set of clients participates in every communication round, an assumption that is impractical in real-world scenarios. To relax the requirement of full client participation, Reisizadeh et al. [Reisizadeh *et al.*, 2020] introduce FedPAQ, which supports partial participation by combining periodic model averaging with quantized communication. Specifically, clients perform several local updates before sending a quantized version of their local model to the server only at a periodic intervals, which significantly reduces communication overhead. From an optimization perspective, Alistarh et al. [Alistarh *et al.*, 2017] introduced quantized stochastic gradient descent (QSGD), which compresses gradients in a lossy manner to reduce communication and speed up distributed training. In general, quantization schemes introduce bias into the gradient estimates, which can lead to degraded model accuracy, especially under non-IID data or aggressive compression levels.

Reducing the number of parameters $|\theta|$. A complementary line of work reduces communication by decreasing the number of parameters transmitted at each round. This is typically achieved through pruning, low-rank approximation (LoRA), or gradient-sparsification, which shrink the effective model size before transmission.

LoRA methods reduce the number of communicated parameters by representing client updates through low-rank factors or structured matrices. Prior work has shown that LoRA is highly effective for communication-efficient FL. For example, Grativol et al. [Grativol *et al.*, 2024] introduce FLoCoRA, which sends low-rank decompositions of client updates to the server for aggregation, following a federated averaging style protocol. Recent techniques further explore LoRA-style adapters for efficient model update transmission in FL [Yang *et al.*, 2025; Guo *et al.*, 2024; Wu *et al.*, 2024]. While these approaches substantially reduce communication, the imposed low-rank structure restricts the allowable update space, which can limit performance for high-capacity video models such as MViT or ResNet-3D-18.

A related class of methods reduces communication by enforcing sparsity. Gradient sparsification communicates only the most significant gradient entries, i.e., those whose magnitudes change the most. Alistarh et al. [Alistarh *et al.*, 2018] propose Top-k sparsification, where each client transmits only its k largest-magnitude gradient coordinates, while the remaining entries are stored in a residual buffer for future updates. Model sparsification, or pruning, applies sparsity directly at the parameter level: redundant weights or channels are removed, yielding a sparse model with fewer parameters to transmit. Federated pruning methods such as PruneFL [Jiang *et al.*, 2022] prune client models during training to maintain a shared global sparse structure across clients. However, sparsification methods, whether applied to gradients or model pruning, often require careful synchronization of sparsity patterns across clients which is challenging to ensure.

2.3 Subspace Learning in FL

A growing body of work explores learning shared low-dimensional subspaces in FL and distributed learning[Grammenos *et al.*, 2020; Narayanamurthy *et al.*, 2022; Kannan *et al.*, 2014]. The most closely related approach is Federated Principal Component Analysis (FedPCA) [Grammenos *et al.*, 2020], which incrementally estimates local PCA subspaces on each client and merges them into a global subspace. While effective for compact representations, these methods are typically *task-agnostic*, and aim only at data compression.

In contrast, we use the recovered subspace as a purely discriminative representation: clients project embeddings into a shared subspace, and all learning and communication occur exclusively within it. We evaluate performance by task accuracy rather than reconstruction error. This changes the objective from ***compress the data*** to ***compress the decision-relevant geometry***, enabling aggressive communication reduction while preserving class separability.

3 Methodology

We now present our federated discriminative subspace discovery framework (Figure 2). The method consists of four main components: (1) extraction of video embeddings, (2) random projection to obtain compressed local statistics, (3) computation and quantization of projected covariance matrices, and (4) server-side aggregation and subspace derivation via singular value decomposition (SVD).

Video Embedding Extraction. Each client i holds a local dataset $D_i = \{v_1, \ldots, v_{n_i}\}$ of n_i video samples. We assume that each client is equipped with a frozen pretrained model $G_\theta(\cdot)$, such as MViT or ResNet3D-18. For every video $v_j \in D_i$, the client extracts a feature embedding from the model's final fully connected layer:

$$x_j = G_\theta(v_j) \in \mathbb{R}^d.$$

Here, d denotes the embedding dimensionality (e.g., $d = 768$ for MViT and $d = 512$ for ResNet3D-18). Collecting all embeddings gives the matrix:

$$X_i = \begin{bmatrix} x_1^\top \\ \vdots \\ x_{n_i}^\top \end{bmatrix} \in \mathbb{R}^{n_i \times d},$$

where each row corresponds to the embedding of a local video. These embeddings serve as the input to our discriminative subspace discovery pipeline.

Random Projection. Computing the full covariance matrix of X_i yields a $d \times d$ symmetric matrix[1], which is

[1]A symmetric matrix is a square matrix whose elements are mirrored across the main diagonal.

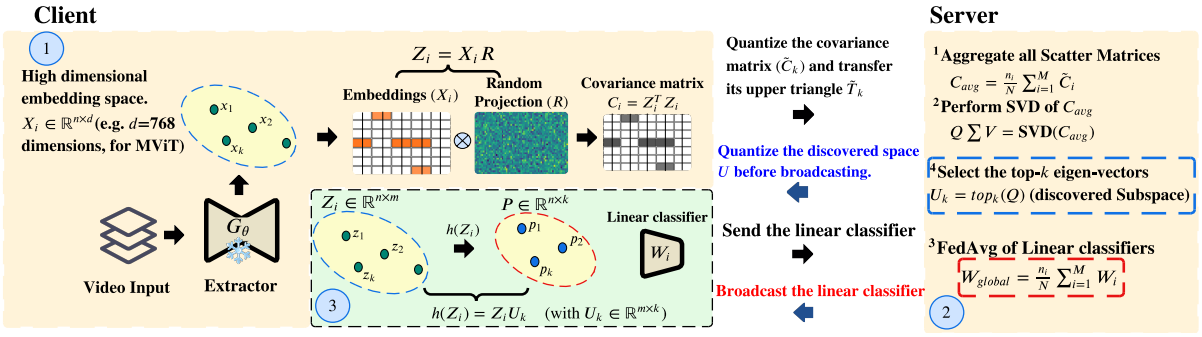

Figure 2: For a given client i, we proceed as follows. (1) We extract video embeddings $X_i \in \mathbb{R}^{n\times d}$ using a frozen feature extractor G_θ (e.g., MViT). These embeddings are then projected using a random projection matrix $\mathcal{R}$ to obtain $Z_i \in \mathbb{R}^{n\times m}$, from which the client computes the covariance matrix C_i. (2) A quantized version of the covariance matrix is transmitted to the server for aggregation. The server estimates a shared discriminative subspace U_i, which is broadcast back to the clients.(3) Each client then projects Z_i onto this subspace to obtain $P_i \in \mathbb{R}^{n\times k}$ and trains a linear classifier. Since $k \ll m \ll d$, the overall communication cost during FL is substantially reduced.

expensive to transmit to the server. To reduce the transmission cost, each client first applies a shared random projection matrix $\mathcal{R} \in \mathbb{R}^{d\times m}$ (with $m \ll d$), on its data, before computing covariance statistics. A random projection is a dimensionality reduction technique that maps high-dimensional data into a lower-dimensional space while approximately preserving pairwise distances between points. Its effectiveness in dimensionality reduction and privacy preservation [Nken *et al.*, 2025; Gondara and Wang, 2020] makes it a suitable choice for our compression step. Our random projection matrix draws its entries from a normal distribution with mean $\mu = 0$ and variance $\sigma^2 = \frac{1}{m}$ (that is $\mathcal{R} \sim \mathcal{N}(0, \frac{1}{m})^{d\times m}$). The embedding matrix X_i is projected as:

$$Z_i = X_i\mathcal{R} \text{ thus, } Z_i \in \mathbb{R}^{n_i\times m}$$

Covariance Matrix Computation. After projection, each client computes the covariance matrix of its projected embeddings. Assuming the projected embeddings Z_i are mean-centered, the sample covariance is

$$C_i = \frac{1}{n_i - 1} Z_i^\top Z_i,$$

where $C_i \in \mathbb{R}^{m\times m}$. Since $m \ll d$, this covariance matrix is substantially smaller than the $d \times d$ covariance that would be formed from the original embeddings X_i.
However, transmitting C_i still incurs a non-negligible communication cost. With each matrix entry usually stored using Q_{32} bits ($Q_{32} = 32$), the communication cost (CC) of sending a single covariance matrix is

$$\text{CC} = Q_{32} m^2 \quad \text{bits.}$$

For example, with $m = 300$, transmitting C_i requires $300^2 \times 32$ bits $\approx$ 351 Kilobyte (KB) per client, which is already larger than communicating a 768×101 linear classifier[2] ($\approx$ 303 KB for UCF101). To reduce this overhead, we apply two strategies. First, we perform an **element-wise quantization** of the covariance matrix to $Q_4 = 4$ bits, prior to transmission. Second, since the quantized covariance matrix is symmetric, we only transmit its upper triangular component; the server reconstructs the full quantized matrix during aggregation. The resulting communication cost becomes

$$\text{CC} = \frac{Q_4\, m(m+1)}{2} \quad \text{bits.} \tag{3}$$

Following the previous example, the cost decreases from 351 KB to approximately 22.56 KB per client, a substantial reduction. Importantly, transmitting the quantized upper triangle of C_i can reduce the risk of leaking sensitive information, as it consists only of statistics derived from the projected embeddings rather than the embeddings or raw data themselves.

Algorithm 1 Federated Discriminative Subspace Discovery

Require: Pretrained feature extractor G_θ, random projection $\mathcal{R} \in \mathbb{R}^{d\times m}$, number of top eigenvectors k

for each communication round **do**
  for each client i **in parallel do**
    Extract embeddings: $X_i = G_\theta(v_i) \in \mathbb{R}^{n_i\times d}$
    Apply random projection: $Z_i = X_i\mathcal{R} \in \mathbb{R}^{n_i\times m}$
    Compute covariance: $C_i = \frac{1}{n_i-1} Z_i^\top Z_i$
    Quantize covariance: $\tilde{C}_i = Q(C_i)$
    Send upper triangle $\tilde{T}_i$ of $\tilde{C}_i$ to server
  end for
  Server reconstruct $\tilde{C}_i$ by mirroring $\tilde{T}_i$.
  Server aggregates covariances: $\tilde{C}_{\text{avg}} = \sum_{i=1}^{M} \frac{n_i}{N} \tilde{C}_i$
  Compute top-k eigenvectors: $SVD(\tilde{C}_{\text{avg}}) = U\Lambda U^\top$
  Broadcast $U_k \in \mathbb{R}^{m\times k}$ to clients (U_k=top$_k(U)$)
  for each client i **in parallel do**
    Project embeddings: $h_i = Z_i U_k$
    Train linear classifier W_i on (h_i, y_i)
    Send W_i to server
  end for
  Server aggregates classifiers: $W_{\text{global}} = \sum_{i=1}^{M} \frac{n_i}{N} W_i$
end for

Server-side Aggregation. Let $\tilde{T}_i$ denote the sent *upper-*

[2] A linear classifier taking 768-D embeddings as input and output 101 logits.

triangular portion of the quantized covariance matrix of client i. Upon receiving $\tilde{T}_i$, the server reconstructs the full symmetric covariance matrix $\tilde{C}_i$ by mirroring the upper triangle. The aggregated covariance is then computed as a weighted average across the M clients:

$$\tilde{C}_{\text{avg}} = \sum_{i=1}^{M} \frac{n_i}{N} \tilde{C}_i.$$

where n_i is the number of samples on client i and $N = \sum_{i=1}^{M} n_i$ is the total number of samples across all clients. The weighted aggregation preserves the global second-order structure despite quantization, enabling the server to recover an accurate estimate of the global covariance matrix.

Subspace Derivation via SVD. Once the aggregated covariance matrix $\tilde{C}_{\text{avg}}$ is obtained, the server performs a singular value decomposition (SVD) to estimate the principal directions of variation across all clients:

$$SVD(\tilde{C}_{\text{avg}}) = U \Lambda U^{\top},$$

where $U \in \mathbb{R}^{m \times m}$ contains the eigenvectors and Λ is a diagonal matrix of eigenvalues sorted in decreasing order. The top-k eigenvectors correspond to directions that capture the dominant structure of the projected embedding distribution, which we empirically observe to align with the most discriminative components of a linear classifier (see Figure 1).

We define the **global discriminative subspace** as the k-dimensional subspace spanned by these top-k eigenvectors:

$$U_k = U_{[:,1:k]} \in \mathbb{R}^{m \times k}.$$

This projection matrix is broadcast to all clients and used to map projected embeddings into the shared low-dimensional subspace. It is important to note that the elements of U_k are once again quantized to 4 bits before being sent to the clients. The communication cost of broadcasting U_k to each client per round is therefore:

$$\text{CC} = Q_4 mk \quad \text{bits.} \tag{4}$$

We acknowledge the loss of precision in U_k and $\tilde{C}_i$ during communication due to quantization; however, our empirical analysis shows that the resulting accuracy drop is negligible compared to quantizing an entire linear classifier.

Classifier Training in the Discriminative Subspace. Once the server distributes the global subspace U_k to all clients, each client projects its local embeddings into this shared low-dimensional space. Given the projected embeddings $Z_i = X_i \mathcal{R}$, the client computes:

$$h_i = Z_i U_k \in \mathbb{R}^{n_i \times k},$$

where each row of h_i represents the k-dimensional discriminative embedding of a video. Since $k \ll d$, this projection significantly reduces the input dimensionality of the classifier. Each client then trains a local linear classifier $W_i \in \mathbb{R}^{k \times Y}$ on these compact representations, where Y is the number of activity classes. Because the classifier operates in a low-dimensional space, both computation and communication are reduced. At the end of a communication round, each client sends the updated classifier parameters, trained entirely within the shared discriminative subspace. The server aggregates the parameters of all clients following the FedAvg strategy described in section 2.1.

Communication Cost Analysis. In our framework, each client uploads the upper-triangular part of its quantized covariance matrix, with communication cost given in Equation 3, downloads the quantized global subspace U_k as defined in Equation 4, and uploads/downloads the linear classifier with communication cost given in Equation 2. The total communication cost (TCC) per client per round is therefore given by:

$$\text{TCC} = 2Q_{32} |\theta| + Q_4 m \left(k + \frac{m+1}{2} \right) \quad \text{bits.} \tag{5}$$

For a linear classifier that maps k-dimensional embeddings to Y output classes, the number of trainable parameters is $|\theta| = Y \times (k+1)$, accounting for both weights and biases. Algorithm 1 provides an overview of the proposed approach.

4 Experimental Setup

4.1 Datasets

To evaluate the effectiveness of our approach, we conduct experiments on three widely used video HAR datasets: UCF101 [Soomro *et al.*, 2012], HMDB51 [Kuehne *et al.*, 2011], and Toyota Smarthome [Das *et al.*, 2019]. UCF101 and HMDB51 contain 101 and 51 activity classes, respectively. For both datasets, we report results on split-1, which consists of 9,537 training videos and 3,783 test videos for UCF101, and 3,570 training videos and 1,530 test videos for HMDB51. Toyota Smarthome comprises approximately 16,000 videos of 31 indoor activities recorded from 18 subjects in a smart-home environment. For Toyota Smarthome, we use a 70%/30% train-validation split.

4.2 Implementation Details

For each training split, we partition the videos across $M = 20$ clients and conduct experiments under two separate scenarios: IID and non-IID. In the IID setting, each client is allocated an equal number of samples per class, ensuring identical data distributions across clients. In the non-IID setting, the class distributions are generated using a Dirichlet distribution with concentration parameter $\beta = 0.01$, producing highly skewed and imbalanced local datasets; smaller values of β correspond to greater data heterogeneity. Each client uses the same frozen pretrained model G_θ to extract video embeddings from its fully connected layer. We use either MViT or ResNet3D-18 as G_θ, both pretrained on Kinetics-400 [Kay *et al.*, 2017], a large-scale video dataset. We train a lightweight classifier on the extracted video embeddings for 100 federated rounds. In each round, clients perform 3 local training epochs using stochastic gradient descent (SGD) as the optimizer, with a learning rate of 0.01 and a batch size of 64. The classifier is a single fully connected layer that maps

IID Setting

G_θ	Method	UCF101		HMDB51		TOYOTA	
		Top@1$_\uparrow$ (%)	TCC$_\downarrow$ (×%)	Top@1$_\uparrow$ (%)	TCC$_\downarrow$ (×%)	Top@1$_\uparrow$ (%)	TCC$_\downarrow$ (×%)
MViT	Vanilla FedAvg	74.40	606.79	49.28	306.40	46.48	186.24
	FLoCoRA$_{r=45,\alpha=2r}$	67.33(-07.07)	434.50(×72)	32.55(-16.73)	287.93(×94)	39.41(-07.07)	280.90(×151)
	FLoCoRA$_{r=32,\alpha=2r}$	57.93(-16.47)	217.25(×36)	26.08(-23.20)	204.75(×67)	35.37(-11.11)	199.75(×107)
	Ours$_{m=400,k=256}$	**72.47(-01.93)**	**291.91(×48)**	**46.99(-02.29)**	**191.42(×62)**	**44.93(-01.55)**	**153.310(×82)**
	FedPAQ$_{q=8}$	68.59(-05.81)	151.70(×25)	45.95(-03.34)	76.600(×25)	40.35(-06.13)	46.5600(×25)
	Top-k	63.31(-11.09)	121.36(×20)	43.66(-05.62)	61.280(×20)	44.68(-01.80)	37.2500(×20)
	Ours$_{m=200,k=100}$	**69.09(-05.31)**	**99.230(×16)**	42.29(-06.99)	**59.770(×19)**	**42.25(-04.23)**	**43.9900(×24)**
	FedPAQ$_{q=4}$	60.54(-13.86)	75.850(×12)	38.14(-11.14)	38.300(×12)	39.24(-07.24)	23.2300(×12)
	Ours$_{m=100,k=64}$	**66.29(-08.11)**	**58.860(×09)**	**41.83(-07.45)**	**31.460(×10)**	**42.06(-04.42)**	**21.3100(×11)**
ResNet-3D-18	Vanilla FedAvg	66.69	404.79	36.67	204.40	48.19	124.24
	FLoCoRA$_{r=45,\alpha=2r}$	55.42(-11.27)	215.51(×53)	27.45(-09.22)	197.93(×97)	46.41(-01.78)	190.90(×154)
	FLoCoRA$_{r=32,\alpha=2r}$	52.41(-14.28)	153.25(×38)	24.90(-11.77)	140.75(×67)	45.28(-02.91)	135.75(×109)
	Ours$_{m=300,k=200}$	**66.29(-00.40)**	**209.87(×52)**	**41.83(+05.16)**	**131.35(×64)**	42.06(-06.13)	**99.9500(×80)**
	FedPAQ$_{q=8}$	36.16(-30.53)	101.20(×25)	20.13(-16.54)	51.100(×25)	25.34(-22.85)	31.0600(×25)
	Top-k	56.94(-09.75)	80.960(×20)	23.76(-12.91)	48.880(×20)	20.68(-27.51)	24.8400(×20)
	Ours$_{m=100,k=50}$	**39.93(-26.76)**	**45.130(×11)**	**20.20(-16.47)**	**25.200(×12)**	**38.17(-10.02)**	**17.2300(×14)**
	FedPAQ$_{q=4}$	22.54(-39.15)	50.600(×13)	16.24(-20.43)	25.550(×13)	14.34(-33.85)	15.5300(×13)
	Ours$_{m=100,k=40}$	**35.78(-30.91)**	**36.750(×09)**	**19.28(-17.39)**	**20.730(×10)**	**35.71(-12.48)**	**14.3200(×12)**

Non-IID Setting ($\beta = 0.01$)

G_θ	Method	UCF101		HMDB51		TOYOTA	
		Top@1$_\uparrow$ (%)	TCC$_\downarrow$ (×%)	Top@1$_\uparrow$ (%)	TCC$_\downarrow$ (×%)	Top@1$_\uparrow$ (%)	TCC$_\downarrow$ (×%)
MViT	Vanilla FedAvg	70.80	606.79	45.10	306.40	44.21	186.24
	FLoCoRA$_{r=45,\alpha=2r}$	64.92(-05.88)	434.50(×72)	31.50(-13.60)	287.93(×94)	38.14(-06.07)	280.90(×151)
	FLoCoRA$_{r=32,\alpha=2r}$	61.70(-08.70)	217.25(×36)	28.89(-16.21)	204.75(×67)	31.19(-13.02)	199.75(×107)
	Ours$_{m=400,k=256}$	**69.55(-01.25)**	**291.91(×48)**	**43.14(-01.96)**	**191.42(×62)**	**41.84(-02.37)**	**153.310(×82)**
	FedPAQ$_{q=8}$	67.63(-03.17)	151.70(×25)	42.42(-02.68)	76.600(×25)	33.41(-10.80)	46.5600(×25)
	Top-k	65.07(-05.73)	121.36(×20)	45.62(-00.52)	61.280(×20)	43.65(-00.56)	37.2500(×20)
	Ours$_{m=200,k=100}$	**66.72(-04.08)**	**99.230(×16)**	40.07(-05.03)	**59.770(×19)**	**40.97(-03.24)**	**43.9900(×24)**
	FedPAQ$_{q=4}$	60.98(-09.82)	75.850(×12)	37.12(-07.98)	38.300(×12)	29.56(-14.65)	23.2800(×12)
	Ours$_{m=100,k=64}$	**64.43(-06.37)**	**58.860(×09)**	**39.67(-05.43)**	**31.460(×10)**	**40.35(-03.86)**	**21.3100(×11)**
ResNet-3D-18	Vanilla FedAvg	52.67	404.79	22.61	204.40	34.43	124.24
	FLoCoRA$_{r=45,\alpha=2r}$	37.54(-15.13)	215.51(×53)	21.95(-00.66)	197.93(×97)	31.94(-02.49)	190.90(×154)
	FLoCoRA$_{r=32,\alpha=2r}$	33.94(-18.73)	153.25(×38)	20.60(-02.01)	140.75(×67)	31.78(-02.65)	135.75(×109)
	Ours$_{m=300,k=200}$	**42.83(-09.84)**	**209.87(×52)**	**21.28(-01.33)**	**131.35(×64)**	**32.04(-02.39)**	**99.9500(×80)**
	FedPAQ$_{q=8}$	25.17(-27.50)	101.20(×25)	20.33(-02.28)	51.100(×25)	28.24(-06.19)	31.0600(×25)
	Top-k	26.74(-25.93)	80.960(×20)	21.67(-00.94)	48.880(×20)	28.40(-06.03)	24.8400(×20)
	Ours$_{m=100,k=50}$	**28.65(-24.02)**	**45.130(×11)**	19.28(-03.33)	**25.200(×12)**	**29.14(-05.29)**	**17.2300(×14)**
	FedPAQ$_{q=4}$	13.70(-48.97)	50.600(×13)	12.43(-10.18)	25.550(×13)	25.03(-09.40)	15.5300(×13)
	Ours$_{m=100,k=40}$	**24.08(-28.59)**	**36.750(×09)**	**14.12(-08.49)**	**20.730(×10)**	**29.17(-05.26)**	**14.3200(×12)**

Table 1: Comparative analysis of our method (**Ours**) against baseline approaches on UCF101, HMDB51, and Toyota Smarthome. We use MViT and ResNet-3D-18 as embedding extractors (G_θ) and assume full client participation ($M = 20$).We highlight the best-performing results of our method in **green**, and the second-best results are **underlined**. Results are reported for both IID and non-IID settings.

the video embeddings to logits over the Y activity classes ($Y \in \{101, 51, 31\}$ depending on the dataset). For the results in Table 1, we assume full client participation in all federated learning rounds (100% client participation rate).

Federated Subspace Discovery. Following the approach described in Section 3 and Algorithm 1, we vary the dimension of the random projection matrix m and the number of selected eigenvectors k corresponding to the largest eigenvalues. Table 1 reports results for $m \in \{100, 200, 300, 400\}$ and $k \in \{40, 50, 64, 100, 200, 256\}$. Under this setting, the classifier is trained directly in the resulting k-dimensional subspace, mapping the reduced embeddings to the class logits. At each communication round, only participating clients compute and transmit a quantized covariance sketch of their projected embeddings. Non-participating clients remain idle. This design accommodates partial participation and statistical heterogeneity across clients, as illustrated in Table 3. We conduct experiments only with a 4-bit quantized covariance matrix, as this yields the minimum communication cost defined in Equation 3.

Comparison with Baselines. We compare our approach against several baselines: (i) vanilla FedAvg [McMahan *et al.*, 2017]; (ii) the low-rank adaptation method FLoCoRA [Grativol *et al.*, 2024] with ranks $r \in \{32, 45\}$ and scaling factor $\alpha = 2r$; (iii) the Top-k gradient sparsification method [Alistarh *et al.*, 2018] with $k = 10\%$, where only the top 10% of classifier gradients are transmitted each round; and (iv) the FedPAQ quantization scheme [Reisizadeh *et al.*, 2020] using $q \in \{4, 8\}$ quantization bits, with model aggregation performed after every round. Although our main experiments use a single fully connected layer as the classifier, we also analyze our approach with deeper classifiers in the appendix and compare it against the baselines described above.

IID Setting

Method	Required for global subspace U_k				UCF101	
	$\mathcal{R}$	C	$\tilde{T}$	SVD($\tilde{C}$)	Top@1$_\uparrow$(%)	TCC$_\downarrow$
Vanilla FedAvg	✗	✗	✗	✗	74.40	606.79
Projection$_{m=400}$	✓	✗	✗	✗	70.86(-03.54)	316.41(×52)
ours$_{m=400,k=256}$	✓	✓	✓	✓	**72.47(-01.93)**	**291.91(×48)**

Non-IID Setting ($\beta = 0.01$)

Vanilla FedAvg	✗	✗	✗	✗	70.80	606.79
Projection$_{m=400}$	✓	✗	✗	✗	67.75(-03.05)	316.41(×52)
ours$_{m=400,k=256}$	✓	✓	✓	✓	**69.55(-01.25)**	**291.91(×48)**

Table 2: Our method benefits from sharing quantized covariance statistics (upper-triangular $\tilde{T}$) and performing SVD to identify a discriminative subspace. Using random projection alone (**Projection**) to reduce embedding dimensionality is insufficient to achieve comparable accuracy-communication trade-offs. Here, G_θ is MViT.

Evaluation Metrics. We evaluate activity recognition using the percentage Top@1 accuracy metric, following prior work [Nken *et al.*, 2025; Hara *et al.*, 2018; Luo *et al.*, 2024]. We report the total communication cost (TCC) per client per round in kilobytes (KB). Throughout our evaluation, the symbol $\uparrow$ indicates that higher values are better, while $\downarrow$ indicates that lower values are better. For each method, we also report in parentheses (i) the percentage drop in Top@1 accuracy relative to vanilla FedAvg, and (ii) the percentage of communication cost used by the method relative to vanilla FedAvg. For example, if a method achieves 67.33 Top@1 accuracy while FedAvg achieves 74.40, we report this as "67.33(–7.07)". Likewise, if its communication cost is 434.50 KB while FedAvg uses 606.79 KB, we report this as "434.50 (×72)", indicating that it uses 72% of FedAvg's communication cost.

4.3 Results Analysis

For each video embedding extractor G_θ, we organize the results in Table 1 into three groups based on the magnitude of TCC. For example in the IID setting, with G_θ= MViT, the **high-communication** group includes FedAvg, FLoCoRA, and our method with random projection dimension $m = 400$ and subspace dimension $k = 256$ (**Ours**$_{m=400,k=256}$). The **medium-communication** group includes Top-k (with k=10%), FedPAQ with 8-bit quantization (FedPAQ$_{q=8}$), and **Ours**$_{m=200,k=100}$. Finally, the **low-communication** group consists of FedPAQ$_{q=4}$ and **Ours**$_{m=100,k=64}$.

Our approach consistently outperforms most baselines in terms of Top@1 accuracy and TCC (highlighted in **dark green**). It also ranks second in several cases, indicated by **underline** values. Specifically;

Communication regimes. Across all datasets, our method consistently achieves favorable accuracy-communication trade-offs compared to competing baselines across the considered communication regimes. While aggressive compression leads to some degradation in Top@1 accuracy, our ap-

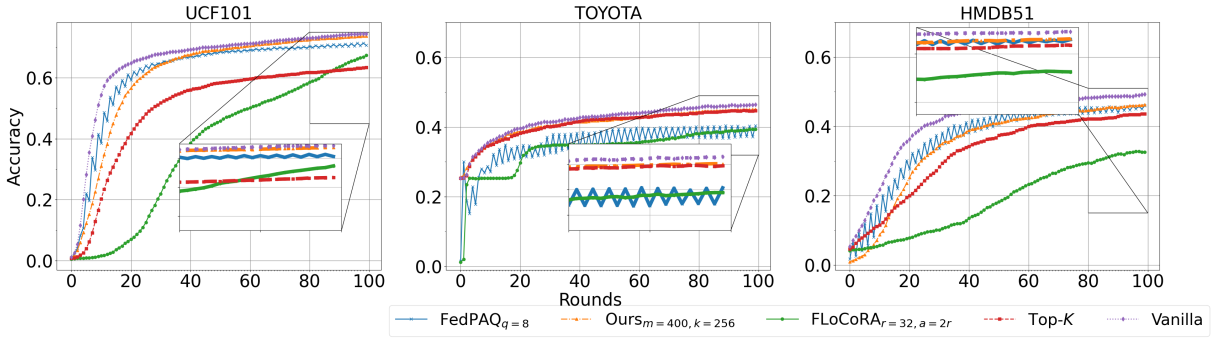

Figure 3: Evolution of Top@1 accuracy over 100 communication rounds with M=20 clients under the IID setting. The embedding extractor is G_θ=MViT. For our method (**Ours**), we use a random projection dimension of m=400 and a discriminative subspace dimension of k=256.

proach maintains competitive performance even at extreme communication budgets. For example, using an MViT backbone under the IID setting, our most compact configuration (**Ours**$_{m=100,k=64}$) reduces the total communication cost to approximately 9% of vanilla FedAvg (i.e 91% communication saving), while incurring an 8.11% absolute drop in Top@1 accuracy on UCF101. We also observe that, the Top@1 accuracy on HMDB51 and Toyota SmartHome is lower than on UCF101. This is because HMDB51 is more challenging and noisier, with substantial camera motion, occlusions, and background clutter from diverse web sources, while Toyota SmartHome contains fine-grained daily activities with high intra-class variation (different people, viewpoints, and environments), making classes harder to separate.

Different embedding extractors G_θ. Our approach remains competitive across different embedding extractors G_θ, including MViT and ResNet-3D-18. In most settings, it outperforms or matches communication-efficient baselines, indicating that the proposed discriminative subspace discovery is largely model-agnostic and does not rely on backbone-specific properties.

IID vs. non-IID settings. As expected, all methods experience a reduction in Top@1 accuracy under non-IID data distributions due to increased statistical heterogeneity across clients. Nevertheless, our method consistently exhibits stronger robustness to non-IID data, outperforming competing approaches across datasets and communication regimes. This suggests that learning a shared discriminative subspace helps mitigate the adverse effects of client heterogeneity. Figure 3 illustrates the Top@1 accuracy of each baseline and our approach across all three datasets.

4.4 Ablation Study

Importance of sharing the covariance matrix. Applying a random projection locally already reduces the dimensionality of video embeddings, thereby lowering communication cost. One may therefore consider training classifiers directly on the projected embeddings without sharing any covariance statistics. However, as shown in Table 2, this baseline (**Projection**) achieves inferior performance compared to our method. While random projection preserves distances, it does not align the embedding space with discriminative directions. Sharing covariance statistics enables the server to identify a global subspace that captures the dominant discriminative structure across clients, leading to improved accuracy at comparable communication cost.

IID Setting					
Selection rate	FedAvg	Top-k	FedPAQ$_{q=8}$	FLoCoRA$_{r=45,\alpha=2r}$	**Ours**$_{m=400,k=256}$
20%	69.26	54.78(-14.48)	**65.76(-03.50)**	59.29(-09.97)	**65.76(-03.50)**
40%	73.38	60.28(-13.10)	68.75(-04.63)	62.51(-10.87)	**70.46(-02.92)**
60%	74.10	62.04(-12.06)	69.68(-04.42)	63.14(-10.96)	**72.15(-01.95)**
80%	74.38	63.37(-11.01)	70.21(-04.17)	63.52(-10.38)	**73.21(-01.17)**
Non-IID Setting ($\beta = 0.01$)					
20%	27.91	27.21(-00.70)	26.13(-01.73)	26.59(-01.32)	**27.34(-00.57)**
40%	43.25	**43.23(-00.02)**	41.58(-01.67)	43.17(-00.08)	42.57(-00.68)
60%	58.25	57.89(-00.36)	54.38(-03.87)	55.82(-02.43)	**57.95(-00.30)**
80%	66.31	63.95(-02.36)	62.97(-03.34)	62.31(-04.00)	**64.47(-01.84)**

Table 3: For different client selection rates, our method consistently outperforms the baselines in Top@1 accuracy on the UCF101 dataset. Here, G_θ is MViT.

Varying the client selection rate. We conduct experiments in which we randomly sample X clients from the M=20 clients used in our setup. The participation rate (i.e., the selection rate reported in Table 3) varies from 20% to 80% of M. In both the IID and non-IID settings, our approach consistently outperforms all baselines, demonstrating its robustness to partial client participation.

Other ablations. Additional ablation studies on the random projection dimension m and the discriminative subspace dimension k are provided in the appendix. We also analyze the time complexity of Algorithm 1 and propose some insights explaining why eigenvectors preserve linear separability.

5 Conclusion

We propose a communication-efficient federated learning framework for video human activity recognition based on discriminative subspace discovery. Motivated by the observation that only a subset of video embedding dimensions is necessary for accurate recognition, each client computes low-dimensional covariance statistics of its embeddings, which are aggregated at the server to derive a shared subspace via singular value decomposition. Subsequent classifier training is performed in this shared low-dimensional space, significantly reducing communication cost while maintaining competitive recognition accuracy.

Acknowledgments

This study was funded by the Research Ireland Centre for Research Training in Digitally-Enhanced Reality (D-real), under Grant No. 18/CRT/6224, and with the financial support of Insight Research Ireland Centre for Data Analytics under Grant Number 12/RC/2289_P2. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

References

[Alistarh *et al.*, 2017] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. *Advances in neural information processing systems*, 30, 2017.

[Alistarh *et al.*, 2018] Dan Alistarh, Torsten Hoefler, Mikael Johansson, Nikola Konstantinov, Sarit Khirirat, and Cédric Renggli. The convergence of sparsified gradient methods. *Advances in Neural Information Processing Systems*, 31, 2018.

[Amiri *et al.*, 2020] Mohammad Mohammadi Amiri, Deniz Gunduz, Sanjeev R Kulkarni, and H Vincent Poor. Federated learning with quantized global model updates. *arXiv preprint arXiv:2006.10672*, 2020.

[Azar *et al.*, 2019] Sina Mokhtarzadeh Azar, Mina Ghadimi Atigh, Ahmad Nickabadi, and Alexandre Alahi. Convolutional relational machine for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7892–7901, 2019.

[Benfold and Reid, 2011] Ben Benfold and Ian Reid. Stable multi-target tracking in real-time surveillance video. In *CVPR 2011*, pages 3457–3464. IEEE, 2011.

[Das *et al.*, 2019] Srijan Das, Rui Dai, Michal Koperski, Luca Minciullo, Lorenzo Garattoni, Francois Bremond, and Gianpiero Francesca. Toyota smarthome: Real-world activities of daily living. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 833–842, 2019.

[Doshi and Yilmaz, 2022] Keval Doshi and Yasin Yilmaz. Federated learning-based driver activity recognition for edge devices. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3338–3346, 2022.

[Fan *et al.*, 2021] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6824–6835, 2021.

[Fanì *et al.*, 2024] Eros Fanì, Raffaello Camoriano, Barbara Caputo, and Marco Ciccone. Accelerating heterogeneous federated learning with closed-form classifiers. *Proceedings of the International Conference on Machine Learning*, 2024.

[Gautam *et al.*, 2024] Vinay Gautam, Himani Maheshwari, Raj Gaurang Tiwari, Ambuj Kumar Agarwal, and Naresh Kumar Trivedi. Federated learning empowered violence recognition in cctv footage: A yolo and resnet-50 fusion approach. In *2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, pages 01–06. IEEE, 2024.

[Gondara and Wang, 2020] Lovedeep Gondara and Ke Wang. Differentially private small dataset release using random projections. In *Conference on Uncertainty in Artificial Intelligence*, pages 639–648. PMLR, 2020.

[Grammenos *et al.*, 2020] Andreas Grammenos, Rodrigo Mendoza Smith, Jon Crowcroft, and Cecilia Mascolo. Federated principal component analysis. *Advances in neural information processing systems*, 33:6453–6464, 2020.

[Grativol *et al.*, 2024] Lucas Grativol, Mathieu Leonardon, Guillaume Muller, Virginie Fresse, and Matthieu Arzel. Flocora: Federated learning compression with low-rank adaptation. In *2024 32nd European Signal Processing Conference (EUSIPCO)*, pages 1786–1790. IEEE, 2024.

[Guo *et al.*, 2024] Mingzhao Guo, Dongzhu Liu, Osvaldo Simeone, and Dingzhu Wen. Efficient wireless federated learning via low-rank gradient factorization. *IEEE Transactions on Vehicular Technology*, 2024.

[Han *et al.*, 2015] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28, 2015.

[Hara *et al.*, 2018] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.

[He *et al.*, 2020] Chaoyang He, Murali Annavaram, and Salman Avestimehr. Group knowledge transfer: Federated learning of large cnns at the edge. *Advances in neural information processing systems*, 33:14068–14080, 2020.

[Horvóth *et al.*, 2022] Samuel Horvóth, Chen-Yu Ho, Ludovit Horvath, Atal Narayan Sahu, Marco Canini, and Peter Richtárik. Natural compression for distributed deep learning. In *Mathematical and Scientific Machine Learning*, pages 129–141. PMLR, 2022.

[Ijaz *et al.*, 2022] Momal Ijaz, Renato Diaz, and Chen Chen. Multimodal transformer for nursing activity recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2065–2074, 2022.

[Jiang *et al.*, 2022] Yuang Jiang, Shiqiang Wang, Victor Valls, Bong Jun Ko, Wei-Han Lee, Kin K Leung, and Leandros Tassiulas. Model pruning enables efficient federated learning on edge devices. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):10374–10386, 2022.

[Kannan *et al.*, 2014] Ravi Kannan, Santosh Vempala, and David Woodruff. Principal component analysis and higher correlations for distributed data. In *Conference on Learning Theory*, pages 1040–1057. PMLR, 2014.

[Kay *et al.*, 2017] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[Kuehne *et al.*, 2011] 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.

[Luo *et al.*, 2024] Zelun Luo, Yuliang Zou, Yijin Yang, Zane Durante, De-An Huang, Zhiding Yu, Chaowei Xiao, Li Fei-Fei, and Animashree Anandkumar. Differentially private video activity recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6657–6667, 2024.

[McMahan *et al.*, 2017] 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.

[Molchanov *et al.*, 2019] Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272, 2019.

[Narayanamurthy *et al.*, 2022] Praneeth Narayanamurthy, Namrata Vaswani, and Aditya Ramamoorthy. Federated over-air subspace tracking from incomplete and corrupted data. *IEEE Transactions on Signal Processing*, 70:3906–3920, 2022.

[Nken *et al.*, 2025] Allassan Tchangmena A Nken, Susan McKeever, Peter Corcoran, and Ihsan Ullah. Videodprp: A differentially private approach for visual privacy-preserving video human activity recognition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 345–362. Springer, 2025.

[Rehman *et al.*, 2022] Yasar Abbas Ur Rehman, Yan Gao, Jiajun Shen, Pedro Porto Buarque de Gusmao, and Nicholas Lane. Federated self-supervised learning for video understanding. In *European Conference on Computer Vision*, pages 506–522. Springer, 2022.

[Reisizadeh *et al.*, 2020] Amirhossein Reisizadeh, Aryan Mokhtari, Hamed Hassani, Ali Jadbabaie, and Ramtin Pedarsani. Fedpaq: A communication-efficient federated learning method with periodic averaging and quantization. In *International conference on artificial intelligence and statistics*, pages 2021–2031. PMLR, 2020.

[Sanyal *et al.*, 2018] Amartya Sanyal, Varun Kanade, Philip HS Torr, and Puneet K Dokania. Robustness via deep low-rank representations. *arXiv preprint arXiv:1804.07090*, 2018.

[Soomro *et al.*, 2012] 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.

[Tu *et al.*, 2024] Nguyen Anh Tu, Assanali Abu, Nartay Aikyn, Nursultan Makhanov, Min-Ho Lee, Khiem Le-Huy, and Kok-Seng Wong. Fedfslar: A federated learning framework for few-shot action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 270–279, 2024.

[Wu *et al.*, 2024] Xinghao Wu, Xuefeng Liu, Jianwei Niu, Haolin Wang, Shaojie Tang, and Guogang Zhu. Fedlora: When personalized federated learning meets low-rank adaptation. 2024.

[Xu *et al.*, 2023] Jian Xu, Xinyi Tong, and Shao-Lun Huang. Personalized federated learning with feature alignment and classifier collaboration. *arXiv preprint arXiv:2306.11867*, 2023.

[Yang *et al.*, 2022] Xiaoshan Yang, Baochen Xiong, Yi Huang, and Changsheng Xu. Cross-modal federated human activity recognition via modality-agnostic and modality-specific representation learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 3063–3071, 2022.

[Yang *et al.*, 2025] Yiyuan Yang, Guodong Long, Qinghua Lu, Liming Zhu, Jing Jiang, and Chengqi Zhang. Federated low-rank adaptation for foundation models: A survey. *arXiv preprint arXiv:2505.13502*, 2025.