# Federated Multimodal Fusion for Action Recognition Leveraging Vision-Language Embeddings and Spatio-Temporal CNNs

**Anonymous authors**
**Paper under double-blind review**

## Abstract

Federated learning (FL) for Video Action Recognition (VAR) faces significant challenges in balancing privacy preservation, communication efficiency, and model performance. This paper introduces *FLAMeST* (*F*ederated *L*earning for *A*ction Recognition with *M*ultimodal *e*mbeddings and *S*pacio-*T*emporal Fusion), a FL framework that synergizes Vision-Language Models (VLMs) and spatiotemporal CNNs to address these challenges. Unlike existing works that use BLIP (VLM) solely for caption generation, *FLAMeST* leverages BLIP in a dual manner. To enhance temporal modeling, complementary spatiotemporal features are extracted using a pre-trained 3D CNN (Slow network). These semantic (BLIP) and motion (Slow) embeddings are concatenated into a unified representation to train a lightweight Multi-Layer Perceptron (MLP). Within the FL paradigm, only the MLP parameters are shared with the server, ensuring raw video data and generated captions remain local. *FLAMeST* employs the FedAvg algorithm for model aggregation, achieving 99%(↓) lower communication overhead compared to full-model training. Experiments on UCF101 and HMDB51 datasets demonstrate the framework's robustness, achieving improved accuracies of 5.13%(↑) and 2.71%(↑), respectively, against the baseline.

## 1 RELATED WORK

Human Action Recognition (HAR) tasks, particularly those relying on wearable sensors, generated considerable attention in both academia and industry (Gani et al. (2019); Wang et al. (2020); Hassan et al. (2018); Kalabakov et al. (2023)), subsequently motivating the extension of such techniques to video-based applications.

Early VAR methods primarily relied on hand-crafted features such as motion-energy images (MEI) and motion-history images (MHI) to capture spatiotemporal dynamics (Bobick & Davis (2001)). Despite being computationally efficient, these methods exhibited limited robustness to variations in viewpoint and complex motion patterns (Bobick & Davis (2001); Zhao et al. (2024)).

**Centralized Action Recognition:** With deep learning, 2D CNNs like AlexNet and VGGNet (Krizhevsky et al. (2012); Simonyan & Zisserman (2014)) enabled robust spatial feature extraction

from frames but lacked temporal modeling. To address this, two-stream networks emerged, combining RGB frames (spatial stream) and optical flow (temporal stream) (Alomar et al. (2024); Le et al. (2022)). While these improved accuracy, they required separate optical flow computation, thus increasing complexity. Subsequently, 3D CNNs such as C3D, I3D and SlowFast networks (Tran et al. (2015); Carreira & Zisserman (2017); Feichtenhofer et al. (2019)) enabled end-to-end spatiotemporal learning. Transformer-based models have also shown promise for VAR. Vision Transformers (ViTs) and Video Swin Transformers capture long-range temporal dependencies via attention mechanisms (Ulhaq et al. (2022); Liu et al. (2022)), while hybrids like PSO-ConvNet Transformers combine CNN and transformer features for improved accuracy (Nguyen & Ribeiro (2023)). VideoMAE extends masked autoencoding to videos using spatiotemporal tube masking (Tong et al. (2022)).

However, these models are trained in centralized settings and focus on unimodal visual features, raising concerns over privacy, data sharing, and regulatory compliance (Kazakos et al. (2021); Akbari et al. (2021)).

**Federated Learning for Action Recognition:** FL addresses privacy by enabling decentralized training. Several works have applied FL to HAR using wearable sensor data (Gani et al. (2019); Hassan et al. (2018)). For video, most methods simplify the problem to image-level classification (Doshi & Yilmaz (2022)) or apply self-supervised learning to improve generalization (Rehman et al. (2022)). Personalization and heterogeneity have been addressed through techniques like Meta-HAR (Li et al. (2021)), FedCLAR (Presotto et al. (2022)), and FedMAT (Shen et al. (2022)), which balance shared and user-specific learning. An activity recognition framework in FL is proposed in (Yang et al. (2024)), where both global (modality-agnostic) and private (modality-specific) classifiers are learned collaboratively across clients. This design effectively separates shared and unique modality characteristics through adversarial training. Architecture-aware FL approaches like FedConv (Xu et al. (2023)) study CNN configurations under heterogeneity have also been explored. Addressing the limited computational capacity of edge devices, (Jain et al. (2021)) proposed a knowledge distillation (KD) strategy involving two teacher models to facilitate efficient model deployment in an FL environment. This hierarchical distillation pipeline introduces a teaching assistant model as an intermediary, effectively bridging the gap between the complex teacher and the constrained student. For few-shot learning under FL, (Tu et al. (2024a)) employed CNNs to capture spatiotemporal cues and applied meta-learning for improved generalization. FedVision enabled FL for object detection with YOLOv3 (Liu et al. (2020)), while recent transformer-based FL work targets video anomaly detection (Doshi & Yilmaz (2023)).

Although these methods support video or multimodal learning in FL, they do not incorporate vision-language models (VLMs), missing the benefits of semantic context.

**Vision-Language Models and FL Integration:** CLIP and BLIP (Radford et al. (2021); Li et al. (2022)) demonstrate strong image-text alignment for zero-shot tasks, but their application in video tasks remains limited. A few recent efforts addressed this gap—for example, (Zhuang et al. (2023)) analyzes how foundation models can be incorporated into FL, outlining both opportunities for collaboration and the associated technical hurdles. ActionCLIP (Wang et al. (2023)), which adapts VLMs like CLIP for video VAR by leveraging the semantic alignment between visual and textual modalities. Unlike traditional methods that treat action labels as discrete classes, ActionCLIP treats them as rich textual descriptions. JoVALE (Son et al. (2024)) combines BLIP with person detection and audio-visual fusion, but the added modalities increase computational overhead.

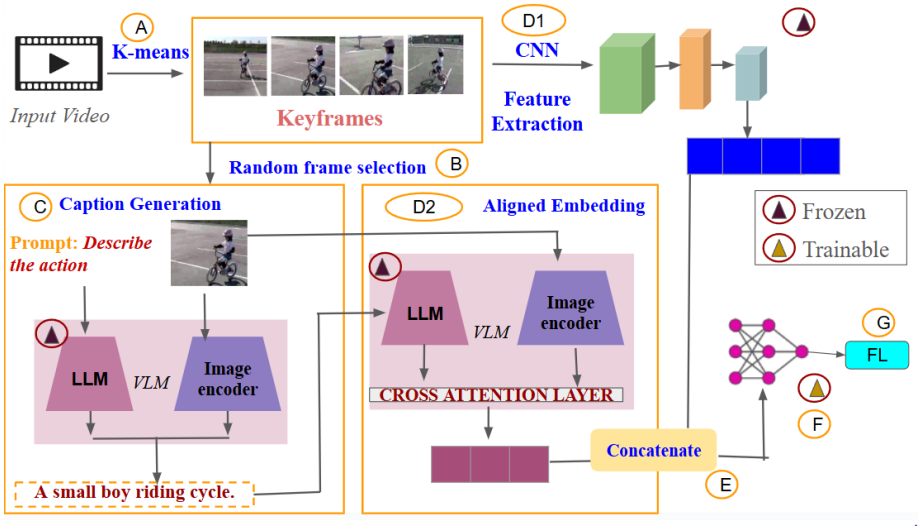



Figure 1: Overview of the *FLAMeST* pipeline. (A) Keyframes are selected using k-means clustering. (B-C) A keyframe is sampled and captioned using a VLM. (D1) All keyframes are passed through a 3D CNN; (D2) the sampled keyframe-caption pair is reprocessed for cross-modal embeddings. (E-F) CNN and VLM features are concatenated to train a lightweight MLP, which participates in the FL. Only the MLP parameters are updated and sent to the server, which aggregates via FedAvg and redistributes the global MLP (step G). The foundation models remain frozen and are not transmitted, reducing communication overhead (Supplementary Figure A-1).

To address these gaps, we propose *FLAMeST*, a framework that combines BLIP-based vision-language embeddings with spatiotemporal CNN features for efficient, privacy-preserving FL in VAR. Our approach achieves competitive accuracy on UCF101 and HMDB51 while significantly reducing communication overhead.

## 2 FLAMeST FRAMEWORK

The pipeline for the *FLAMeST* framework is shown in Figure 1. We consider an FL set-up comprising $N$ clients $\{C^1, \ldots, C^N\}$, where each client $C^i$ holds a private labeled video dataset. $\mathbf{V}_j^i$ represents the $j$th video clip of the $i$th device.

$$\begin{aligned} \mathcal{D}^i &= \{(\mathbf{V}_j^i, y_j^i)\}_{j=1}^{M^i} \\ \text{where } |D^i| &= M^i. \end{aligned} \tag{1}$$

Each video sample consists of $n_j^i$ RGB frames of spatial dimension $H \times W$ where the associated action class $y_j^i$ belongs in a label space of $K$ action classes.

$$\begin{aligned} \mathbf{V}_j^i &\in \mathbb{R}^{n_j^i \times H \times W \times 3} \\ y_j^i &\in \{1, \ldots, K\} \end{aligned} \tag{2}$$

### 2.1 Key Frame Selection via Clustering

To reduce the temporal redundancy in video sequences, we apply *k-Means* based clustering in a learned feature space to identify a reduced set of informative keyframes. Across all devices and their data elements, we set $m_j^i = k$, where $k$ is the number of clusters. We experimented with $k \in \{2, 3, 5, 8\}$ and selected k=3 using the elbow method and silhouette confidence intervals. This makes the keyframe selection systematic and reproducible, rather than heuristic. Method details are presented in Supplementary Section A.1, Algorithm A-1.

$$\begin{aligned} \mathbf{F}_{\text{key},j}^i &= \{\mathbf{f}_1^i, \ldots, \mathbf{f}_{m_j^i}^i\} \\ \text{where} \quad \mathbf{f}_l^i &\in \mathbb{R}^{H \times W \times 3}, l \in \{1, \ldots, m_j^i\} \end{aligned} \tag{3}$$

### 2.2 Random Key Frame Sampling

From the set of keyframes $\mathbf{F}_{\text{key},j}^i$ a single representative frame $\tilde{\mathbf{f}}_j^i$ is sampled uniformly at random.

$$\begin{aligned} \tilde{\mathbf{f}}_j^i &\sim \mathcal{U}(\mathbf{F}_{\text{key},j}^i) \\ \text{where} \quad P(\tilde{\mathbf{f}}_j^i) &= \frac{1}{m_j^i} \end{aligned} \tag{4}$$

### 2.3 Multimodal Embedding Generation via VLM

The sampled frame $\tilde{\mathbf{f}}_j^i$ is first passed through a VLM $\Phi_{\text{VLM}}$ to generate a descriptive caption $\mathbf{t}_j^i$.

$$\mathbf{t}_j^i = \Phi_{\text{VLM}}(\tilde{\mathbf{f}}_j^i), \tag{5}$$

Subsequently, the caption $\mathbf{t}_j^i$ and the frame $\tilde{\mathbf{f}}_j^i$ are fed into the VLM cross-attention module to produce a joint vision-language embedding, i.e, the [CLS] token embedding of the last hidden layer is extracted (Algorithm 1). Without loss of generality, we consider the BLIP VLM.

$$\begin{aligned} \mathbf{e}_{\text{VLM},j}^i &= \Phi_{\text{VLM}}^{\text{cross}}(\tilde{\mathbf{f}}_j^i, \mathbf{t}_j^i) \in \mathbb{R}^{d_e}, \\ \text{where} \quad d_e &= 1 \times 768. \end{aligned} \tag{6}$$

### 2.4 Visual Embedding via CNN Backbone

In parallel, the set of keyframes $\mathbf{F}_{\text{key},j}^i$ is processed using a 3D CNN-based backbone $\Phi_{\text{CNN}}$ (e.g., ResNet-3D, I3D, Slow) to obtain a visual-only embedding. Without loss of generality, we consider the *Slow* -3D CNN model.

$$\begin{aligned} \mathbf{e}_{\text{CNN},j}^i &= \Phi_{\text{CNN}}(\mathbf{F}_{\text{key},j}^i) \in \mathbb{R}^{d_v}, \\ \text{where} \quad d_v &= 1 \times 2048. \end{aligned} \tag{7}$$

The embeddings are obtained from the last dense layer of the CNN model (Algorithm 1).

### 2.5 Joint Feature Representation

The final multimodal feature vector is formed by concatenating the VLM-based and CNN-based embeddings. We also experimented with a gated-attention-based fusion technique. However, the performance is not any better than simple concatenation (The results are provided in Section 5.5).

$$\begin{gathered}\mathbf{e}_j^i = [\mathbf{e}_{\text{VLM},j}^i; \mathbf{e}_{\text{CNN},j}^i] \in \mathbb{R}^{d_e+d_v} \\ \text{where dimensionality of } e_j^i \text{ is } 1 \times 2816.\end{gathered} \tag{8}$$

Each client prepares its transformed data set,

$$\mathcal{D}_{\text{VLM,CNN}}^i = \{(\mathbf{e}_j^i, y_j^i)\}_{j=1}^{j=M^i} \tag{9}$$

The CNN ($\mathbf{e}_{\text{CNN},j}^i$) and VLM ($\mathbf{e}_{\text{VLM},j}^i$) embeddings often differ in dimensionality, making element-wise operations like the dot product infeasible. While projecting them to a common space is possible, it may introduce complexity and risk of information loss. To retain the full representation of both modalities without additional transformations, we adopt simple concatenation for fusion.

**Algorithm 1** High-level steps for obtaining VLM and CNN embeddings on each client

1: Determine keyframes (EQ. 3)
2: Determine a representative frame (EQ. 4)
3: Obtain caption for the frame (EQ. 5)
4: Obtain text-visual cross-embedding for the caption text and the corresponding frame (EQ. 6)
5: Obtain CNN embedding for the representative frame (EQ. 7)
6: Obtain combined embedding (EQ. 8)
7: Prepare a transformed data set (EQ. 9)

### 2.6 Local Training and Federated Aggregation at Server

On each client $C^i$ train a local multi-layer perceptron (MLP) classifier $g^i(\cdot)$ parameterized on $\mathbf{w}^i$ [1] on the transformed dataset $\mathcal{D}_{\text{VLM,CNN}}^i$.

$$g^i(\mathbf{w}^i) : \mathbb{R}^{d_e+d_v} \rightarrow \mathbb{R}^K \tag{10}$$

Each client trains $g^i(\cdot)$ by minimizing the cross-entropy loss using Stochastic Gradient Descent (SGD) as,

$$\begin{gathered}\mathcal{J}(\mathbf{w^i}) = \frac{1}{|\mathcal{D}|} \sum_{(e,y)\in\mathcal{D}} \mathcal{L}_{\text{CE}}(g(e), y) \\ \mathbf{w}_{t+1}^i = \mathbf{w^i}_t - \eta \nabla_{\mathbf{w^i}} \mathcal{J}(\mathbf{w^i}_t)\end{gathered} \tag{11}$$

where $D$ is $D_{\text{VLM,CNN}}^i$, $g$ is $g^i(\mathbf{w^i})$, $\mathcal{L}_{\text{CE}}(\cdot,\cdot)$ denotes the cross-entropy loss and $\eta$ is the learning rate. After local training, each client transmits its updated parameters $\mathbf{w}^i$ (EQ. 11) to a central

[1] $\mathbf{w}^i$ is $\mathbf{w}_{(t)}^i$ at time t

server. The server aggregates these received models by weighted averaging of the models' parameters (EQ. 12, Algorithm 2).

$$\mathbf{w}^*_{(t+1)} = \sum_{i=1}^{N} \alpha^i \mathbf{w}^i_{(t)}, \quad \alpha^i = \frac{|\mathcal{D}i|}{\sum_{j=1}^{N} |\mathcal{D}^j|} \tag{12}$$

Where $\alpha$ corresponds to the data proportion of each client. It denotes the weight given to each client during model aggregation. The aggregated global model $\mathbf{w}^*_{(t+1)}$ is broadcast to all clients, and the training proceeds for $T$ communication rounds.

Each client has a pre-trained VLM and a CNN model, both frozen during training and inference. **The only trainable component is the MLP**, which participates in the FL cycle.

An alternative is to employ recent video-language captioners such as Qwen-VL or Cosmos-Reasoning. However, deploying multi-billion parameter captioners on each FL client is infeasible due to memory and latency constraints, and server-side captioning would require uploading raw video, violating privacy. Our *FLAMeST* pipeline is lightweight, client-friendly, and privacy-preserving. Exploring distilled captioners in FL is an avenue for future work.

**Algorithm 2** Federated Learning with federated averaging

1: Initialize server model to random weights $w^*_0$
2: **for** $t \in [1 \dots T]$ **do**
3:  **for** $i \in [1, \dots, N]$ **do**
4:  Initialize client weights (EQ. 10)

$$w^i = w^*_{(t-1)}$$

5:  Prepare (or update) client data set (Algorithm 1)
6:  Minimize client specific loss (EQ. 11)

$$w^i = \underset{w}{\operatorname{argmin}}\ \mathcal{J}^i(w)$$

7:  **end for**
8:  Perform federated averaging on server (EQ. 12) to obtain $w^*_{(t)}$
9: **end for**

## 3 Experimental Set-Up

We design a comprehensive experimental framework to evaluate the performance of *FLMeST* for VAR. Table 1 shows the details regarding each model and embedding size. All the experiments are done on NVIDIA A100. All references to BLIP embeddings in this work pertain to the cross-attention embeddings derived from the visual-semantic alignment within the BLIP model.

**Benchmark Datasets and Data Partitioning Strategy:** The experiments in this study are conducted on two widely recognized benchmark datasets: UCF101 (Soomro et al. (2012)) and HMDB51 (Kuehne et al. (2011)). UCF101 comprises 13,320 video clips spanning 101 action categories. The dataset is split into 10,619 training and 2,701 testing samples. HMDB51 consists

of 6,766 video clips categorized into 51 action classes. The dataset is partitioned with 5,413 videos for training and 1,353 videos for testing, following an 80:20 train-test split. To simulate real-world Non-IID conditions, we partition data across clients using a Dirichlet distribution Dir($\beta$), where a smaller $\beta$ indicates higher label skew. For baseline comparisons, we use IID splits ($\beta = 1$), while Non-IID settings use $\beta = 0.6$ to evaluate robustness under data heterogeneity.

**Benchmarking Methods and Feature Extractors:** We use two popular vision-language models (VLMs): BLIP-2 (Li et al. (2022)) and CLIP-ViT/L-14 (Radford et al. (2021)). For benchmarking, we evaluate four spatiotemporal models: (1) ResNet-3D (Tran et al. (2015)) as a 3D CNN baseline; (2) I3D (Carreira & Zisserman (2017)), which inflates 2D kernels into 3D; (3) SlowFast (Feichtenhofer et al. (2019)) using an 8-frame Slow pathway to capture appearance and motion; and (4) VideoMAE (Tong et al. (2022)), a masked autoencoder-based ViT for video representation. All models use public code and pretrained weights from PyTorchVideo (Fan et al. (2021)), Torchvision, and HuggingFace. Feature vectors are extracted from the final convolutional layer before classification to capture spatiotemporal semantics. In addition to the feature extractors discussed above, we benchmark our method against the work of Jain et al. (2021)[2] We have also considered the recent method ActionCLIP by (Wang et al. (2023)) in a non-FL setting that uses a CLIP model for action recognition on unseen classes. We do not compare our method with existing FL-based VAR approaches such as FL for Driving Action Recognition (DAR) (Doshi & Yilmaz (2022)), which targets driver-specific actions using 2D CNNs and FedGKT (He et al. (2020)) for communication efficiency. Their task-specific focus differs from our objective of general-purpose action recognition. Similarly, FSAR (Guo et al. (2023)) is another VAR method in the FL setting, but it operates on skeleton-based datasets rather than video data, making direct comparison with our approach inappropriate. Few-shot FL methods (Tu et al. (2024b)) are also excluded, as they address a different problem setting. However, we consider the extension of our framework to few-shot FL-based VAR as an interesting direction for future work.

**Client-Side Training Protocol:** In the proposed FL framework, foundation models remain frozen during training and inference and do not participate in FL communication. Each client trains a local MLP classifier, whose parameters are shared with the server during aggregation. By default, the MLP comprises two hidden layers (512 and 256 neurons) and an output layer matching the number of action classes. The input layer aligns with the dimensionality of the frozen backbone embeddings. Clients train locally for five epochs per round using a batch size of 128. Optimization is performed using Adam with a learning rate of 0.01 and weight decay of $1 \times 10^{-4}$, with cross-entropy loss as the loss objective. Additional MLP configurations are explored in Section 5. Unless stated otherwise, experiments use four clients in a cross-silo IID-FL setting over 80 rounds.

**Model Aggregation, Client Update and Evaluation Metrics:** The model aggregation is accomplished via FedAvg (McMahan et al. (2017)), which simply averages model weights across participating clients. For client-side model update, we have considered two methods, FedProx (Li et al. (2020)) and FedDyn (Acar et al. (2021)), apart from simple gradient update, which were compared in ablation studies (Section 5). Performance is assessed using two primary metrics. The top-1 classification accuracy of the *global model* and communication efficiency are measured as the number of model parameters exchanged per communication round.

[2]The code for this paper was not publicly available. Therefore, the accuracy values reported in our work are those quoted directly from the original paper.

- **Parameter reduction by using only the MLP in FL :**

$$= \frac{(\overbrace{\text{BLIP} + \text{Slow}}^{279,868,172} + \text{MLP}) - \text{MLP}}{(\text{BLIP} + \text{Slow} + \text{MLP})}$$
$$= \frac{(279,868,172 + 1,573,889) - 1,573,889}{279,868,172 + 1,573,889}$$
$$= 0.994 \quad \text{or} \quad 99.4\% \tag{13}$$

- **Parameter reduction compared to Knowledge Distillation (KD) :**

$$= \frac{\text{KD} - \text{Our}}{\text{KD}}, \quad \text{Our} = \text{MLP}$$
$$= \frac{11,689,512 - 1,573,889}{11,689,512} \tag{14}$$
$$= 0.865 \quad \text{or} \quad 86.5\%$$

**Resource Usage and Storage Requirements:** We use the VLM BLIP[3] model for generating the caption and subsequently visual-semantic cross embeddings. For the 3D-CNN backbone, we used the Slow-3D CNN model[4], which is part of Facebook AI's PySlowFast framework. An analysis of resource usage during inference mode shows that the Slow-3D CNN model recorded a maximum memory allocation of 319.51 MB per point, while the BLIP model utilized 1023.97 MB per point as inspected using *watch*[5] and *torch*.[6] The execution times for Slow-3D CNN and BLIP are 1.315 seconds and 1.362 seconds, respectively, when processing a single input point. The storage requirement[7] for BLIP is approximately 941.44 MB, the Slow-3D CNN model is 131.85 MB, and the MLP model is 6.02 MB (Refer to Table 1).

## 4 Results and Evaluation

In this section, we conduct a quantitative evaluation of *FLAMeST*, focusing on its communication overhead and performance in both FL and centralized learning setups. We compare *FLAMeST* against various foundation model extractors to assess its effectiveness and efficiency in these different learning environments.

### 4.1 Communication Overhead

We assume that each participating client possesses sufficient storage and computational resources to host pre-trained foundation models, such as BLIP, CLIP, and 3D-CNN. Table 1 outlines the parameter count of the models employed. We propose a lightweight communication strategy to overcome the significant communication overhead typically associated with federated training of large-scale models. Instead of exchanging the full parameter sets of these massive foundation models, we freeze the pre-trained model's backbone during local training and extract fixed embeddings from it. A lightweight MLP classifier is trained on these embeddings. This is the only component that solely participates in the FL optimization loop. This decouples the representation learning phase from the federated aggregation step, substantially reducing communication costs.

As shown in Table 1, the number of parameters involved in FL is smaller than the size of the underlying foundation models. For instance, while the BLIP+Slow model comprises approximately 279 million parameters, only 1,573,889 parameters from the MLP are involved in FL communication, resulting in a parameter transfer reduction of approximately 99.4% (EQ. 13). Similarly,

[3]Salesforce/blip-image-captioning-base
[4]https://pytorch.org/hub/facebookresearch_pytorchvideo_resnet
[5]watch -n 1 nvidia-smi
[6]torch.cuda.max_memory_allocated()/(1024**2)
[7]The Floating Point (FP) is 32

Table (1): Model Comparison: Parameters and Input Dimensions for Feature Extractors and FL Models, where the last column refers to the learnable parameters participating in FL. Details are presented in Section 3.

| S.No | Feature Extractor Models | Input Frame | Base Model Parameters | Input Dimension | Parameters in FL |
|---|---|---|---|---|---|
| 1 | BLIP (Li et al. (2022)) | 1 | 247,414,076 | 768 | 525,311 |
| 2 | CLIP (Radford et al. (2021)) | 1 | 151,277,313 | 1024 | 656,3855 |
| 3 | KD **(Baseline)** (Jain et al. (2021)) | 8 | 49,482,360 | - | **11,689,512** |
| 4 | ResNet3D (Tran et al. (2015)) | 3 | 33,371,472 | 512 | 394,241 |
| 5 | I3D Carreira & Zisserman (2017) | 8 | 28,043,472 | 2048 | 1,180,673 |
| 6 | Slow (Feichtenhofer et al. (2019)) | 8 | 34,566,488 | 2048 | 1,180,673 |
| 7 | VideoMAE (Tong et al. (2022)) | 16 | 86304869 | 786 | 525,313 |
| 8 | BLIP (Text Only) | 1 | 247,414,076 | 768 | 525,313 |
| 9 | BLIP + I3D | 9 (1+8) | 275,457,548 | 2816 | 1,573,889 |
| 10 | BLIP + ResNet3D | 4 (1+3) | 280,785,548 | 1280 | 797,457 |
| 11 | BLIP + Slow (*FLAMeST*) | 9 (1+8) | **279,868,172** | 2816 | **1,573,889** |
| 12 | BLIP (Text Only) + ResNet3D | 4 (1+3) | 280,785,548 | 1280 | 797,457 |
| 13 | BLIP (Text Only) + I3D | 9 (1+8) | 275,457,548 | 2816 | 1,573,889 |
| 14 | BLIP (Text Only) + Slow | 9 (1+8) | 279,868,172 | 2816 | 1,573,889 |
| 15 | ActionCLIP (non-FL) (Wang et al. (2023)) | 3 | 150,000,000 | - | - |

compared to the baseline KD, the parameter transfer reduction is approximately 86.5% (EQ. 14). This selective participation significantly enhances the communication efficiency of the proposed framework without compromising representation quality. Freezing the foundation models strikes a balance between performance and efficiency, preserving the strong semantic priors of these models while avoiding the prohibitive cost of fine-tuning and communicating billions of parameters across clients.

### 4.2 Comparison of *FLAMeST* and other Feature Extractors

Table 2-(A, B) presents a comparative study of different VLMs, various CNN and transformer-based feature extractors in FL training settings on the UCF101 and HMDB51 datasets. For UCF101, the highest accuracy is achieved by the combination of embeddings extracted from the Slow model along with the cross-attention embeddings obtained from the BLIP model. Our method achieves an improvement of 5.13% against the baseline (Table 2-A, row 14, row 3). The Slow model, being a 3D CNN, effectively captures the spatial-temporal aspects of video sequences. In contrast, the BLIP model integrates visual and textual information through its vision-language alignment mechanism. Fusing these embeddings provides a richer video content representation, allowing the classifier to make more informed predictions. The integration of BLIP embeddings with the 3D CNN features leads to a significant improvement in accuracy. Specifically, there is an 18.05% increase (Table 2-A, row 1 - row 14) in accuracy for BLIP embeddings when combined with the Slow model, while the Slow model benefits from an improvement of 3.47% (Table 2-A, row 5, row 14). This trend

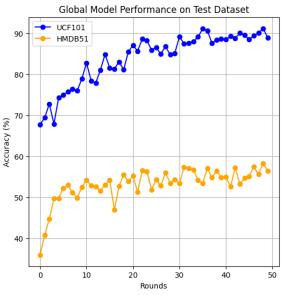



Figure 2: Accuracy achieved by the global model (*FLAMeST*) over 50 rounds (cycles) for the UCF101 and HMDB51 datasets.

underscores the advantage of utilizing *multimodal embeddings* that encapsulate textual and visual semantics. The second-best performing combination comprises the I3D model coupled with BLIP alignment embeddings. This pairing results in an accuracy improvement of 4.23% (Table 2-A, row 6 - row 13) and 17.38% (Table 2-A, row 1, row 13) for the I3D model and the BLIP embeddings, respectively. Similarly, the ResNet-3D model, when fused with cross-attention embeddings from BLIP, exhibits an increase of accuracy of 16.15% (Table 2-A, row 4, row 12) for UCF101. The highest accuracy achieved on the HMDB51 dataset is significantly lower compared to UCF101, which can be attributed to the inherent complexity and challenging nature of HMDB51.

Even in a centralized learning setup, the best performance achieved was approximately 71% (Table 3-B, row 15 ). In the FL setting, the highest accuracy of 67% (Table 2-B, row 14) was attained using the BLIP model and the Slow architecture. Our method achieves an improvement of 4.58% (Table 2-B, row 3, row 14) against the baseline. The BLIP and Slow models individually contributed to performance gains of 11.63% (Table 2-B, row 1, row 14) and 2.59% (Table 2-B, row 5, row 14), respectively, over their standalone performance. Furthermore, when BLIP embeddings were fused with the I3D model, an improvement of 4.89% (Table 2-B, row 6, row 13) in accuracy was observed, while the integration with ResNet-3D resulted in a 20.91% (Table 2-B, row 4, row 12) performance gain for the 3D CNN models. *These findings suggest that incorporating semantic-visual embeddings alongside spatial-temporal embeddings enhances model performance by enriching feature representations with complementary contextual information.* Both the VLM and the CNN model benefit from the collaboration. Regarding the training accuracy of *FLAMeST* (Figure 2), the training accuracy improves and stabilizes with the subsequent communication cycle. We also calculate the cross-validation accuracy on UCF101 and HMDB51 over 20 folds across 80 rounds to get the average estimate over different train-test splits. The mean accuracy for UCF101 was approximately 94% with a standard deviation of 0.32 (Table 2-A, row 15), whereas for HMDB51, the mean calculated was 70% with a standard deviation of 0.98 (Table 2-B, row 15).

In addition to action recognition, we evaluated the effectiveness of the BLIP cross-modal embeddings for image classification on the CIFAR-10 dataset. Our experiments achieved an accuracy of 90% after 30 communication rounds. Details are provided in Supplementary Section E.

### 4.3 Poor Performance of Static Text Embeddings Generated by CLIP

Unlike BLIP, which can generate captions directly for a given image, CLIP operates in a retrieval-based manner. To facilitate this process, we first generate a set of action-specific captions for each action in the UCF101 and HMDB51 datasets using large-scale language models such as ChatGPT. Each caption is then encoded using the text encoder of the CLIP model, and the resulting text embeddings are stored in a dictionary for efficient retrieval during training and inference (More details in the Supplementary A.3). Though computing embeddings in prior reduces the computational cost due to their static nature, the quality of embeddings is observed to be poor. As shown in Table 2-A, for UCF101, the accuracy obtained is 61% (Table 2-A, row 2), and for HMDB51, it is 33% (Table 2-B, row 2). Since these captions are generated before training, they are inherently generic and may fail to reflect individual video frames' unique visual and contextual attributes. *As a result, the fixed textual representation may not align well with the dynamic and heterogeneous nature of the visual embeddings, leading to suboptimal multimodal fusion and reduced classification accuracy.* As the CLIP-generated embedding was not well refined, we did not conduct further studies on it,

Table (2): Federated Learning: Accuracy Comparison of Different Feature Extractors on UCF101 and HMDB51 Datasets. The bold-faced text denotes baseline and above accuracies. The baseline and *FLAMeST* values are indicated by * and ** prefixes, respectively.

(a) UCF101 Dataset

| S.No | Feature Extractor | Accuracy (%) |
|---|---|---|
| 1 | BLIP | 76.38 |
| 2 | CLIP | 60.98 |
| 3 | **KD (Baseline)** | ***89.30** |
| 4 | ResNet-3D | 64.01 |
| 5 | Slow | **90.96** |
| 6 | I3D | **89.53** |
| 7 | Video MAE | 65.04 |
| 8 | BLIP (Text Only) | 30.28 |
| 9 | BLIP (Text Only) + ResNet-3D | 64.00 |
| 10 | BLIP (Text Only) + Slow | **91.03** |
| 11 | BLIP (Text Only) + I3D | **89.92** |
| 12 | BLIP + ResNet-3D | 80.16 |
| 13 | BLIP + I3D | **93.76** |
| 14 | **BLIP + Slow (*FLAMeST*)** | ****94.43** |
| 15 | ***FLAMeST* (Cross Validation)** | **94±0.32** |

(b) HMDB51 Dataset

| S.No | Feature Extractor | Accuracy (%) |
|---|---|---|
| 1 | BLIP | 54.75 |
| 2 | CLIP | 30.98 |
| 3 | **KD (Baseline)** | ***61.80** |
| 4 | ResNet-3D | 38.95 |
| 5 | Slow | ***63.79** |
| 6 | I3D | 59.62 |
| 7 | Video MAE | 33.45 |
| 8 | BLIP (Text Only) | 27.18 |
| 9 | BLIP (Text Only) + ResNet-3D | 40.04 |
| 10 | BLIP (Text Only) + Slow | **63.19** |
| 11 | BLIP (Text Only) + I3D | 60.13 |
| 12 | BLIP + ResNet-3D | 59.86 |
| 13 | BLIP + I3D | **64.51** |
| 14 | **BLIP + Slow (*FLAMeST*)** | ****66.38** |
| 15 | ***FLAMeST* (Cross Validation)** | **70±0.98** |

as most of the improvement would have come from the embeddings of the CNN model rather than the VLM in this joint learning.

### 4.4 Significance of Textual Embeddings

An additional set of experiments was conducted using only the textual embeddings generated by the VLM. The MLP model, in this case, was only trained on the text embeddings obtained from the VLM. Our findings resonate with recent analyses of embedding-based retrieval limits Weller et al. (2025). They highlight weaknesses of single-vector representations, consistent with our observation that BLIP-only text embeddings are weak.

For the BLIP model, when only text embeddings (captions generated) are used to train the MLP model, the accuracies obtained for UCF101 and HMDB51 are only 30.28% (Table 2-A, row 8) and 27.18% (Table 2-B, row 8), respectively.

Aligning the text embeddings with the visual embeddings of the BLIP model improved accuracy by more than 27.57% for the HMDB51 dataset (Table 2-B, row 8 to row 1) and by 46.1% (Table 2-A, row 8 to row 1) for the UCF101 dataset. From this observation, we conclude that textual information alone cannot fully capture the dynamic nature of actions in video sequences.

*While textual descriptions can introduce supplementary contextual details, they do not encompass the full range of spatial and temporal dependencies in videos.* However, textual information when used with visual features can enhance the overall representative quality (Tables 2-(A, B), row 14). An illustration of the text captions generated by BLIP are shown in Supplementary C.2.

### 4.5 Comparison with Centralized Training

Table 3-(A, B) reports results under a centralized training setting, where all data is aggregated at a single site, effectively eliminating the challenges posed by data decentralization in FL. In this

Table (3): Centralized Learning: Accuracy Comparison of Different Feature Extractors on UCF101 and HMDB51 Datasets. The bold-faced text denotes baseline and above accuracies. The baseline and *FLAMeST* values are indicated by * and ** prefixes, respectively.

(a) UCF101 Dataset

| S.No | Feature Extractor | Accuracy (%) |
|---|---|---|
| 1 | BLIP | 85.01 |
| 2 | CLIP | 62.80 |
| 3 | **KD (Baseline)** | ***91.10** |
| 4 | ResNet-3D | 78.05 |
| 5 | Slow | **91.37** |
| 6 | I3D | **92.71** |
| 7 | Video MAE | 66.38 |
| 8 | BLIP (Text Only) | 34.73 |
| 9 | BLIP (Text Only) + ResNet-3D | 79.30 |
| 10 | BLIP (Text Only) + Slow | 90.67 |
| 11 | BLIP (Text Only) + I3D | 88.78 |
| 12 | BLIP + ResNet-3D | 85.82 |
| 13 | BLIP + I3D | **93.97** |
| 14 | ActionCLIP | **95** |
| 15 | **BLIP + Slow (*FLAMeST*)** | ****94.30** |

(b) HMDB51 Dataset

| S.No | Feature Extractor | Accuracy (%) |
|---|---|---|
| 1 | BLIP | 62.36 |
| 2 | CLIP | 55.87 |
| 3 | **KD (Baseline)** | **64.10** |
| 4 | ResNet-3D | 53.87 |
| 5 | Slow | 63.17 |
| 6 | I3D | **65.24** |
| 7 | Video MAE | 39.38 |
| 8 | BLIP (Text Only) | 35.50 |
| 9 | BLIP (Text Only) + ResNet-3D | 52.92 |
| 10 | BLIP (Text Only) + Slow | **66.64** |
| 11 | BLIP (Text Only) + I3D | **65.98** |
| 12 | BLIP + ResNet-3D | **64.21** |
| 13 | BLIP + I3D | **68.04** |
| 14 | ActionCLIP | **76** |
| 15 | **BLIP + Slow (*FLAMeST*)** | ****71.94** |

Table (4): Comparison of ActionCLIP and *FLAMeST* across different datasets and performance metrics in *Centralized Non-FL Setting*.

| Dataset | Method | Accuracy (%)↑ | Train Time (sec)↓ | Test Time (sec)↓ | Trainable Parameters↓ |
|---|---|---|---|---|---|
| **UCF101** | ActionClip | **95** | 242.993 | 96.51 | 150-155 million |
| | *FLAMeST* | 93.34 | **5.862** | **0.595** | 1 million |
| **HMDB51** | ActionClip | **75.89** | 159.170 | 15.001 | 150-155 million |
| | *FLAMeST* | 73.36 | **3.637** | **0.379** | 1 million |

scenario, a single client possesses the entire dataset, and the MLP model is trained for 80 epochs using the same optimizer and learning rate as in the FL setup (Section 3). *As expected, centralized training consistently outperforms federated learning across most feature extractor combinations for a single cycle (Tables 2 and 3)*. The highest accuracy obtained is 94.30% (Table 3-A, row 15) by the Slow and BLIP model when taken in conjunction with UCF101 and 71% (Table 3-B, row 15) for the HMDB51 dataset. This is primarily because, in FL, the model undergoes incremental updates over multiple communication rounds rather than maturing in a single training cycle. Consequently, in FL, the learning process is more gradual, and convergence takes longer compared to a centralized setting where the model has access to the complete dataset at all times. *Evaluating models in a centralized setup provides a valuable baseline for assessing the effectiveness of different feature extractors in a non-FL environment.* The results help determine whether a feature extractor is inherently strong or is hindered by federated constraints such as non-IID data distribution and communication limitations.

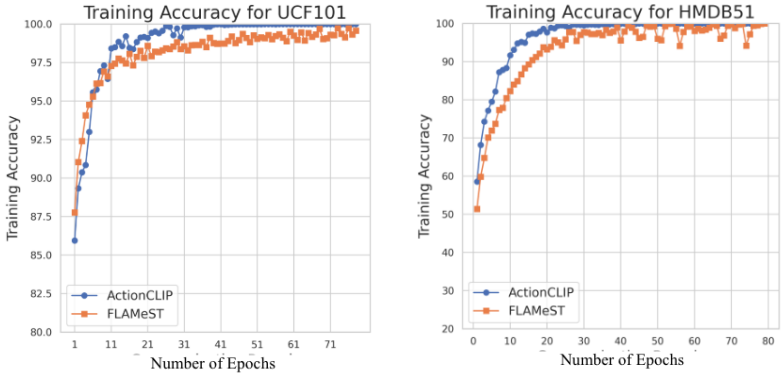



Figure 3: Centralized training accuracy trends of ActionCLIP and *FLAMeST* over 80 epochs. The graph illustrates how the models perform across training, highlighting the stability and convergence behavior of each method.

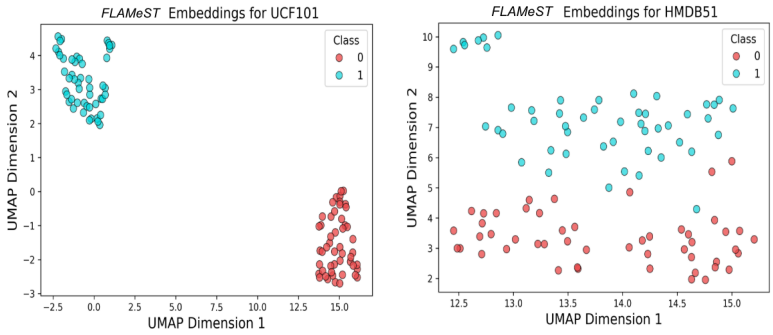
### 4.6 Comparison of ActionCLIP and *FLAMeST*

We trained the ActionCLIP[8] model from scratch, utilizing its default parameter settings, for a total of 80 epochs on our specific data partition (Section3). The results of this training are presented in Table 4, which provides a comparative analysis between *FLAMeST* and ActionCLIP. The comparison reveals that ActionCLIP outperforms *FLAMeST* by achieving a 2% higher accuracy on both UCF101 and HMDB51. However, *FLAMeST* demonstrates notable advantages in terms of efficiency, with significantly reduced training and inference times compared to ActionCLIP. Furthermore, *FLAMeST* operates with a substantially lower number of training parameters, making it a more lightweight and computationally efficient option relative to ActionCLIP. Figure 3 shows that over the epochs, *FLAMeST* also acquires comparable training accuracy to ActionCLIP, leading to the model convergence.

### 4.7 Understanding Embedding Quality through UMAP Projections

To assess the quality of the embeddings generated by the BLIP and Slow 3D CNN, we utilize UMAP—a widely used dimensionality reduction and visualization technique for high-dimensional data (McInnes et al. (2018)). For this analysis, we randomly select two classes from each dataset, UCF101 and HMDB51, and visualize the corresponding embeddings in a two-dimensional space. As illustrated in Figure 4 (a,b), the *FLAMeST* embeddings for UCF101 and HMDB51 form *well-separated and compact clusters, indicating that the combined representation effectively distinguishes between the two selected classes, demonstrating the discriminative power of the embeddings.* More UMAP Figures A-3 are provided in Supplementary C.1 .

[8]https://github.com/sallymmx/ActionCLIP




(a) *FLAMeST* (b) *FLAMeST*

Figure 4: UMAP visualization of different embedding strategies on selected classes from the UCF101 and HMDB51 datasets. For UCF101, we consider the classes 'Apply Lipstick' and 'Archery'. For HMDB51, the selected classes are 'Catch' and 'Cartwheel.'

## 5 Ablation Studies

The experimental configuration for the ablation study, including the model architecture, hyperparameters, and data partitioning strategies, remains consistent with the setup outlined in Section 3. The results quoted are the best accuracy obtained by the global model over 80 rounds.

### 5.1 Client-Side Model Update

Table 5 evaluates federated optimization algorithms—FedAvg (McInnes et al. (2018)), FedProx (Li et al. (2020)) and FedDyn (Acar et al. (2021)) under both IID and Non-IID data distributions. FedAvg (EQ. 11) is the baseline aggregation method, where client updates are averaged without accounting for data distribution differences. In contrast, FedProx introduces a proximal term in the local objective function to address client data heterogeneity (EQ. A-1). Whereas FedDyn uses a dynamic update of the local objective loss function to ensure consistency of the local model update with the global model update (EQ. A-2). Table 5 shows that all three methods perform comparably under the IID setting. Whereas for the Non-IID case, FedProx outperforms FedAvg and FedDyn for the UCF101 and HMDB51 datasets. This improvement highlights the effectiveness of *FedProx in handling the statistical challenges posed by heterogeneous data environments, making it a more suitable choice in real-world scenarios where data is often non-uniformly distributed across clients.*

### 5.2 Classifier Network Architecture

As shown in Table 6, increasing the width of hidden layers improves test performance, with the [1024, 1024, 1024](row 5) model achieving the highest accuracy of 96.3%. However, deeper networks do not always yield better results—[1024, 512, 256](row 4) underperforms compared to the simpler [1024, 512](row 3) model, suggesting that additional depth may introduce optimization difficulties or lead to diminishing returns. All architectures achieve high training accuracy (>98%), indicating minimal

Table (5): Comparison of client update methods on IID and Non-IID settings across datasets (4 clients).

| Dataset | Method | IID (%) | Non-IID (%) |
|---|---|---|---|
| **UCF101** | FedAvg | 90 | 84 |
| | FedProx | **93** | **86** |
| | FedDyn | 91 | 84 |
| **HMDB51** | FedAvg | 65 | 60 |
| | FedProx | *67* | **62** |
| | FedDyn | **69.15** | 60.81 |

Table (6): Performance of different MLP architectures on the UCF101 dataset.

| S.no | Hidden Layers | Train Acc. (%) | Test Acc. (%) |
|---|---|---|---|
| 1 | *[512, 256]* | *99.07* | *93.89* |
| 2 | [256, 256] | 98.50 | 94.30 |
| 3 | [1024, 512] | 99.37 | 96.22 |
| 4 | [1024, 512, 256] | 98.48 | 94.14 |
| 5 | [1024, 1024, 1024] | 98.80 | **96.30** |

overfitting. *A moderately deep and wide MLP is often sufficient to achieve strong generalization, while excessively deep models may not offer significant gains.*

### 5.3 Effect of Epochs on Model Performance

Table 7 presents the impact of training epochs on model accuracy. For UCF101 under IID settings, accuracy begins at 96% with 5 epochs and stabilizes near 94.8% by epoch 15, indicating early convergence. In contrast, Non-IID performance fluctuates. It rises from 84% to a peak of 86% at epoch 10, then declines to 83%, suggesting potential overfitting under data heterogeneity. On HMDB51, the IID model improves consistently from 70% to 73%, whereas the Non-IID model shows modest gains from 58% to 61%, further highlighting the challenges of learning from skewed distributions *While increasing the number of epochs helps improve or stabilize performance under IID settings, the same trend does not hold under Non-IID conditions.*

### 5.4 Scalability with Increasing Clients

**Scaling approach-1:** Table 8 presents the performance of *FLAMeST* as the number of clients increases in a cross-silo FL setting, where each client typically represents an institution. We limit the client count to 10 to reflect realistic deployment scenarios. As the number of clients increases, classification accuracy declines for both the UCF101 and HMDB51 datasets in the IID (rows 1 and 4) and Non-IID scenarios (rows 2 and 5). *The degradation is primarily due to reduced data per client as the number of clients increases, which weakens local training and limits global model generalization.*

**Scaling approach-2:** We investigate label-skewness scaling under the constraint that the per-class-per-client dataset sizes remain fixed. Each client is first assigned a balanced base dataset containing an equal number of samples from every class. Class-specific heterogeneity is then introduced through a probabilistic allocation mechanism (Supplementary Section A.4). Specifically, class proportions are drawn from a Beta distribution to determine the relative representation of each class, while a Bernoulli distribution identifies the subset of clients (takers) associated with that class. Once the participating clients for a given class are selected, the corresponding sample quotas are evenly distributed among them, thereby ensuring fairness in allocation while preserving skewness in label distribution. As observed in Table 8 (rows 3 and 6), an increase in the number of clients leads to a gradual decline in classification accuracy for both UCF101 and HMDB51, pri-

Table (7): Accuracy (%) across varying training epochs on UCF101 and HMDB51 over 80 cycles for 4 clients.

| S.no | Epochs | UCF101 | | HMDB51 | |
|---|---|---|---|---|---|
| | | IID | Non-IID | IID | Non-IID |
| 1 | 5 | **96.0** | **84.0** | 70.0 | 58.0 |
| 2 | 10 | 94.7 | 86.0 | 71.0 | 59.0 |
| 3 | 15 | 94.8 | 83.0 | **73.0** | **61.0** |

Table (8): Accuracy (%) across varying client counts on UCF101 and HMDB51 over 80 cycles with 5 local epochs.

| S.no | Setting | 4 Clients | 8 Clients | 10 Clients |
|---|---|---|---|---|
| 1 | UCF101 (IID) | **96** | 93 | 91 |
| 2 | UCF101 (Non-IID-1) | **86** | 82 | 77 |
| 3 | UCF101 (Non-IID-2) | **84.93** | 83.93 | 82.49 |
| 4 | HMDB51 (IID) | **67** | 63 | 57 |
| 5 | HMDB51 (Non-IID-1) | **55** | 47 | 37 |
| 6 | HMDB51 (Non-IID-2) | **74** | 72.96 | 72 |

Table (9): Performance comparison of Residual Gated Cross Fusion (RGCF) and simple concatenation method on UCF101 and HMDB51 datasets over 80 cycles and 4 clients.

| S.no | Dataset | RGCF (5 epochs) | RGCF (50 epochs) | Simple Concatenation |
|---|---|---|---|---|
| 1 | UCF101 | 75 | 89.5 | **94.43** |
| 2 | HMDB51 | 56 | 58.8 | **70** |

marily due to the heightened data heterogeneity. *Interestingly, for HMDB51, scaling results in an accuracy improvement that even surpasses the performance achieved under the IID setting.*

### 5.5 Fusion by Gated-Attention

In order to study the effectiveness of a more advanced fusion technique than plain concatenation, we have experimented with *residual gated cross fusion* (RGCF) between image and text embeddings (Refer to Supplementary section A.5). Table 9 shows that simple concatenation consistently outperforms RGCF on both UCF101 and HMDB51, even when RGCF is trained for more epochs. However, the accuracy of RGCF does improve with increased training—from 75% to 89.5% on UCF101 and from 56% to 58.8% on HMDB51, indicating that RGCF may benefit from longer training to stabilize. We suspect the average performance of RGCF is due to the added complexity and parameterization of RGCF, leading to overfitting on limited local data. In contrast, simple concatenation preserves the full representational capacity of CNN and VLM embeddings without additional transformations.

## 6 Failure Analysis

We observe that broadly, the error cases fall into three main categories: (A) selection of uninformative frames, (B) high inter-class similarity, and (C) noisy or incomplete caption VLM. Detailed discussion of failure cases is presented in (Supplementary D) and their mitigation strategies are discussed in (Supplementary D.1).

- **Uninformative Frame Selection:** In *FLAMeST*, frames are sampled randomly from the video clips. Consequently, the selected frame may not adequately capture the action being performed. The frames corresponding to the "kayaking" and "haircut" classes fail to depict critical visual cues such as a kayak or scissors—objects that are central to recognizing the activity (Figures A-5-A and A-6-A).

- **High Inter-Class Similarity:** Certain action classes exhibit substantial visual overlap, particularly when temporal context is omitted. For example, "kayaking" and "rafting" both involve similar water-based environments and the presence of boats, making static frame-based distinction challenging (Figures A-5-B and A-6-B).
- **Noisy or Incomplete Captions:** The VLM's inability to generate accurate and descriptive captions for the selected frames (Supplementary Figures A-5-C and A-6-C) along with the qualitative analysis of the captions generated by VLM shown in Supplementary Section B suggests that captions alone are often insufficient for complex actions as text can be noisy. Our fusion with Slow CNN embeddings mitigates this, but reliance on captions remains a limitation.

## 7 Conclusions and Future Work

This study introduces *FLAMeST*, an approach for integrating foundation models within an FL framework to enhance VAR performance. We leverage embeddings from the BLIP model alongside features extracted by a 3D CNN model called Slow. These combined representations train a lightweight MLP in the federated cycle, substantially reducing communication overhead compared to transmitting the full foundation model. The fusion of semantic and visual embeddings yields notable accuracy gains on challenging benchmarks such as HMDB51 and UCF101. Ablation studies confirm *FLAMeST's* robustness across diverse client-update schemes and its scalability under both IID and Non-IID data distributions. Moreover, the *FLAMeST* embeddings form well-clustered class representations, highlighting their discriminative richness. Comparative analyses against alternative strategies further demonstrate how *FLAMeST* effectively addresses key challenges in VAR tasks in a collaborative set-up.

As part of future work, we intend to investigate distilled variants of these foundation models to reduce storage and computation costs, thereby improving the practicality of edge-device FL. Another direction involves enabling client-side customization, wherein users can record and label video clips. This introduces new challenges around server-side training and privacy preservation, which merit deeper investigation. As future work, we plan to explore distilled or lightweight video-language captioners that could be feasible for deployment in federated learning without compromising privacy.

## References

Durmus Alp Emre Acar, Yue Zhao, Ramon Matas Navarro, Matthew Mattina, Paul N. Whatmough, and Venkatesh Saligrama. Federated learning based on dynamic regularization. CoRR, abs/2111.04263, 2021. URL `https://arxiv.org/abs/2111.04263`.

Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. Advances in neural information processing systems, 34:24206–24221, 2021.

Khaled Alomar, Halil Ibrahim Aysel, and Xiaohao Cai. Rnns, cnns and transformers in human action recognition: A survey and a hybrid model. arXiv preprint arXiv:2407.06162, 2024.

Aaron F. Bobick and James W. Davis. The recognition of human movement using temporal templates. IEEE Transactions on pattern analysis and machine intelligence, 23(3):257–267, 2001.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299–6308, 2017.

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, pp. 3338–3346, 2022.

Keval Doshi and Yasin Yilmaz. Privacy-preserving video understanding via transformer-based federated learning. In 2023 IEEE Conference on Dependable and Secure Computing (DSC), pp. 1–8. IEEE, 2023.

Haoqi Fan, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, et al. Pytorchvideo: A deep learning library for video understanding. In Proceedings of the 29th ACM international conference on multimedia, pp. 3783–3786, 2021.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 6202–6211, 2019.

Md Osman Gani, Taskina Fayezeen, Richard J Povinelli, Roger O Smith, Muhammad Arif, Ahmed J Kattan, and Sheikh Iqbal Ahamed. A light weight smartphone based human activity recognition system with high accuracy. Journal of Network and Computer Applications, 141:59–72, 2019.

Jingwen Guo, Hong Liu, Shitong Sun, Tianyu Guo, Min Zhang, and Chenyang Si. Fsar: federated skeleton-based action recognition with adaptive topology structure and knowledge distillation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 10400–10410, 2023.

Mohammed Mehedi Hassan, Md Zia Uddin, Amr Mohamed, and Ahmad Almogren. A robust human activity recognition system using smartphone sensors and deep learning. Future Generation Computer Systems, 81:307–313, 2018.

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.

Pranjal Jain, Shreyas Goenka, Saurabh Bagchi, Biplab Banerjee, and Somali Chaterji. Federated action recognition on heterogeneous embedded devices. arXiv preprint arXiv:2107.12147, 2021.

Stefan Kalabakov, Borche Jovanovski, Daniel Denkovski, Valentin Rakovic, Bjarne Pfitzner, Orhan Konak, Bert Arnrich, and Hristijan Gjoreski. Federated learning for activity recognition: A system level perspective. IEEE Access, 11:64442–64457, 2023.

Evangelos Kazakos, Jaesung Huh, Arsha Nagrani, Andrew Zisserman, and Dima Damen. With a little help from my temporal context: Multimodal egocentric action recognition. arXiv preprint arXiv:2111.01024, 2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25, 2012.

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, pp. 2556–2563. IEEE, 2011.

Viet-Tuan Le, Kiet Tran-Trung, and Vinh Truong Hoang. A comprehensive review of recent deep learning techniques for human activity recognition. Computational Intelligence and Neuroscience, 2022(1):8323962, 2022.

Chenglin Li, Di Niu, Bei Jiang, Xiao Zuo, and Jianming Yang. Meta-har: Federated representation learning for human activity recognition. In Proceedings of the web conference 2021, pp. 912–922, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In International conference on machine learning, pp. 12888–12900. PMLR, 2022.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems, 2:429–450, 2020.

Yang Liu, Anbu Huang, Yun Luo, He Huang, Youzhi Liu, Yuanyuan Chen, Lican Feng, Tianjian Chen, Han Yu, and Qiang Yang. Fedvision: An online visual object detection platform powered by federated learning. In Proceedings of the AAAI conference on artificial intelligence, volume 34, pp. 13172–13179, 2020.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 3202–3211, 2022.

S. Lloyd. Least squares quantization in pcm. IEEE Transactions on Information Theory, 28(2): 129–137, 1982. doi: 10.1109/TIT.1982.1056489.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. arXiv preprint arXiv:1802.03426, 2018.

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, pp. 1273–1282. PMLR, 2017.

Huu Phong Nguyen and Bernardete Ribeiro. Video action recognition collaborative learning with dynamics via pso-convnet transformer. Scientific Reports, 13(1):14624, 2023.

Riccardo Presotto, Gabriele Civitarese, and Claudio Bettini. Fedclar: Federated clustering for personalized sensor-based human activity recognition. In 2022 IEEE international conference on pervasive computing and communications (PerCom), pp. 227–236. IEEE, 2022.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In International conference on machine learning, pp. 8748–8763. PmLR, 2021.

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, pp. 506–522. Springer, 2022.

Qiang Shen, Haotian Feng, Rui Song, Stefano Teso, Fausto Giunchiglia, Hao Xu, et al. Federated multi-task attention for cross-individual human activity recognition. In IJCAI, pp. 3423–3429. International Joint Conferences on Artificial Intelligence, 2022.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.

Taein Son, Soo Won Seo, Jisong Kim, Seok Hwan Lee, and Jun Won Choi. Jovale: Detecting human actions in video using audiovisual and language contexts. arXiv preprint arXiv:2412.13708, 2024.

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.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. Advances in neural information processing systems, 35:10078–10093, 2022.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision, pp. 4489–4497, 2015.

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, pp. 270–279, 2024a.

Nguyen Anh Tu, Nartay Aikyn, Nursultan Makhanov, Assanali Abu, Kok-Seng Wong, and Min-Ho Lee. Benchmarking federated few-shot learning for video-based action recognition. IEEE Access, 2024b.

Anwaar Ulhaq, Naveed Akhtar, Ganna Pogrebna, and Ajmal Mian. Vision transformers for action recognition: A survey. arXiv preprint arXiv:2209.05700, 2022.

Mengmeng Wang, Jiazheng Xing, Jianbiao Mei, Yong Liu, and Yunliang Jiang. Actionclip: Adapting language-image pretrained models for video action recognition. IEEE Transactions on Neural Networks and Learning Systems, 2023.

Qu Wang, Haiyong Luo, Hao Xiong, Aidong Men, Fang Zhao, Ming Xia, and Changhai Ou. Pedestrian dead reckoning based on walking pattern recognition and online magnetic fingerprint trajectory calibration. IEEE Internet of Things Journal, 8(3):2011–2026, 2020.

Orion Weller, Michael Boratko, Iftekhar Naim, and Jinhyuk Lee. On the Theoretical Limitations of Embedding-Based Retrieval. arXiv preprint arXiv:2508.21038, 2025.

Peiran Xu, Zeyu Wang, Jieru Mei, Liangqiong Qu, Alan Yuille, Cihang Xie, and Yuyin Zhou. Fedconv: Enhancing convolutional neural networks for handling data heterogeneity in federated learning. arXiv preprint arXiv:2310.04412, 2023.

Xiaoshan Yang, Baochen Xiong, Yi Huang, and Changsheng Xu. Cross-modal federated human activity recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

Lanfei Zhao, Zixiang Lin, Ruiyang Sun, and Aili Wang. A review of state-of-the-art methodologies and applications in action recognition. Electronics, 13(23):4733, 2024.

Weiming Zhuang, Chen Chen, and Lingjuan Lyu. When foundation model meets federated learning: Motivations, challenges, and future directions. arXiv preprint arXiv:2306.15546, 2023.