TACTFL: Temporal Contrastive Training for Multi-modal Federated Learning with Similarity-guided Model Aggregation

Guanxiong Sun, Majid Mirmehdi, Zahraa S. Abdallah, Raul Santos-Rodriguez, Ian James Craddock, Telmo M Silva Filho

Background

The EPSRC-funded TORUS(Transforming the Objective Real-world measUrement of Symptoms) research programme (£6.17M, EP/X036146/1) addresses the urgent need for **accurate and objective measures** to assess the effectiveness of new drugs for **Parkinson's disease** (PD) in clinical trials.

It aims to solve that problem through a novel platform of sensing technologies for use in patients' own homes along with an advanced data fusion and machine learning pipeline that measures changes in specific mobility-related behaviours over weeks and months.

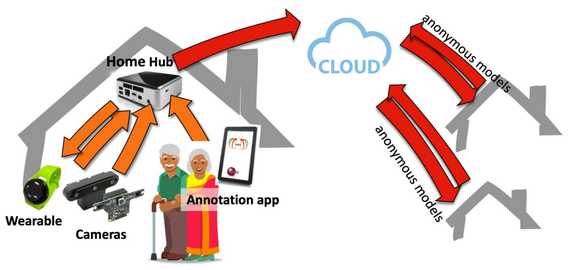

Figure 0. System overview of the TORUS project

Introduction

Motivation

Real-world federated learning faces two key challenges:

- Label Scarcity: Acquiring well-annotated data on client devices is expensive and often impractical.
- Data Heterogeneity: Clients produce diverse, multi-modal data (e.g., video, audio, sensor readings) that are not uniformly distributed.

Our Goal

Develop a unified, semi-supervised framework for multi-modal FL that learns effective representations from unlabelled, heterogeneous data while preserving privacy.

Our Solution: TACTFL

A novel framework with two key components:

- Temporal Contrastive Training (TCT): A self-supervised strategy for clients to learn from their unlabelled multi-modal data.
- Similarity-Guided Aggregation (SMA): A server-side strategy to intelligently merge client models by addressing semantic misalignment.

Extensive experiments demonstrate that TACTFL achieves state-of-the-art performance.

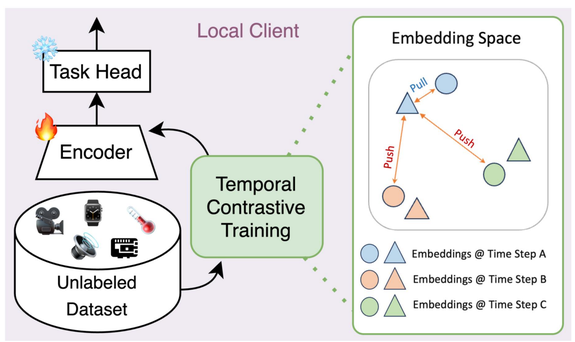

Figure 1(a). A conceptual overview of TACTFL. On each client, encoders are updated using temporal contrastive training (TCT) on unlabelled multi-modal data, while task heads remain frozen. The objective encourages embeddings from different modalities at the same time step to be pulled together in the feature space, and embeddings from different time steps to be pushed apart.

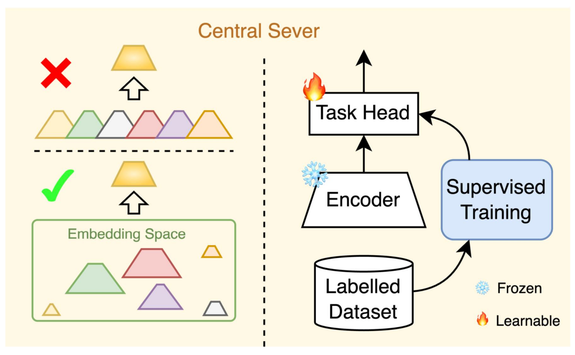

Figure 1(b). Unlike FedAvg, TACTFL adaptively weights each client model based on representational similarity. The server trains the task head on a small labelled dataset with frozen encoders.

Framework Overview

TACTFL Framework Overview

The training process iterates in three stages:

- Local Client Training: Each client trains its encoders on local, unlabelled multi-modal data using Temporal Contrastive Training. The task head is frozen.
- Server-side Aggregation: Clients upload their trained encoders. The server performs Similarity-Guided Aggregation to create an aligned global encoder.
- Server-side Training: The server trains the global task head on its small, labelled dataset using the new global encoder (which is kept frozen).

Algorithm 1: The overall workflow of TACTFL

Input: number of communication rounds T, number of clients C, local dataset of the c-th client D_c, labeled dataset on the server D_s
Output: The final encoders and task heads $\{E\}^T_{glob}, \{H\}^T$
System executes:
  Initialise global encoders $\{E\}^0_{glob}$ and task heads $\{H\}^0$
  for $t = 0, 1, \ldots, T-1$ **do**
    Send $\{E\}^t_{glob}$ and $\{H\}^t$ to all clients
    for $c = 0, 1, \ldots, C-1$ **do**
      $\{E\}^{t+1}_c \leftarrow$ **Temporal Contrastive Training**$(D_c, \{E\}^t_{glob})$
    end
    Receive updated encoders from clients
    $\{E\}^{t+1}_{glob} \leftarrow$ **SiGMA** $(\{E\}^t_0, \{E\}^t_1, \cdots, \{E\}^t_{C-1})$
    Train $\{H\}^{t+1}$ on D_s with $\{E\}^{t+1}_{glob}$
  end
  Return $\{E\}^T_{glob}, \{H\}^T$

Algorithm 1. A detailed pipeline of TACTFL. This design enables TACTFL to robustly learn from unlabelled multi-modal data while addressing semantic drift across clients. It is model-agnostic, supports missing modalities, and introduces minimal additional communication overhead.

Temporal Contrastive Training

To enable local learning on unlabelled multi-modal data, we introduce a self-supervised contrastive objective that leverages temporal alignment as implicit supervision. The idea is that signals captured from different modalities within overlapping time windows are likely to describe the same underlying activity and should be embedded closely in the embedding space. The temporal contrastive training process is shown as Figure 2.

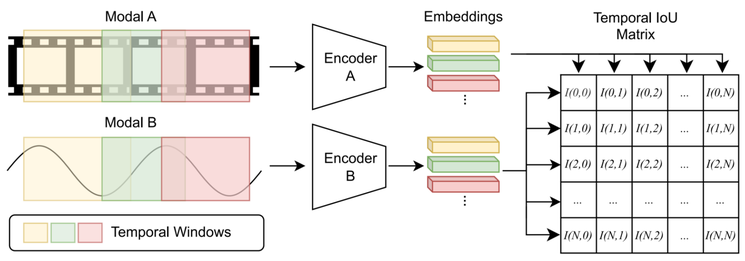

Figure 2. Details of the temporal contrastive training process

Step	Description	Purpose
1. Temporal Segmentation	Split the multi-modal input sequence into two overlapping temporal chunks.	Creates sets of segments for training pairs and acts as data augmentation.
2. Calculate tIoU Score	Compute the Temporal Intersection over Union (tIoU) score for every pair of segments across the batch.	Establishes the temporal overlap, serving as a soft pseudo-label.
3. Normalize to Soft Target	Normalize the tIoU scores to create a soft target probability distribution for each segment.	Defines the "true" target similarity distribution for the contrastive objective.
4. Contrastive Loss	Apply the loss function, which encourages high cosine similarity between encoder outputs for segment pairs with high temporal overlap.	Pulls related segment embeddings closer and pushes unrelated ones apart in the feature space.
5. Symmetrical Aggregation	Compute the total contrastive loss symmetrically, aggregating all pairs across both modalities in the mini-batch.	Ensures robust representation learning for both modalities.

Similarity-guided Model Aggregation

In contrast to canonical supervised FL, where local models are supervised by the same label distribution and tend to produce semantically aligned feature spaces, the use of self-supervised contrastive learning in TACTFL introduces a new challenge: semantic misalignment. Since each client independently optimises its encoders on local data using contrastive objectives, the resulting representations can diverge significantly across clients.

The core idea of Similarity-guided Model Aggregation is to assign higher aggregation weights to client models whose representations are more semantically consistent with the group, using a small, labeled dataset D_s on the central server.

Step	Description	Purpose
1. Summarize Embeddings	Server computes feature embeddings using each client model. These embeddings are averaged across Ds to produce the summary vectors V_i and V_j	Reduces complex feature space to a single comparative vector per client.
2. Compute Pairwise Similarity	Calculate the cosine similarity between all pairs of summary vectors. This forms the similarity matrix.	Quantifies the semantic alignment between every pair of client models.
3. Calculate Alignment Weights	Determine the similarity-based weight for each client, which is proportional to the sum of scores in the similarity matrix with all other clients.	Captures the relative alignment of each client's representations with the entire population.
4. Weighted Aggregation	The global model is formed as a weighted sum of the client models using the computed weights.	Creates an aggregated model biased towards better-aligned, more consistent clients.

Experiments

Dataset	α	Modalities	Metric	$r_l = 0.9$			$r_l = 0.8$		
				FedAvg [35]	FedOpt [42]	TACTFL	FedAvg [35]	FedOpt [42]	TACTFL
UCF101 [22]	0.1	Audio, Video	Acc	10.49	35.35	**68.48**	40.14	44.43	**71.67**
MiT10 [36]		Audio, Video	Acc	7.48	19.86	**48.87**	25.77	36.90	**51.77**
UCI-HAR [3]		Acce, Gyro	F1	33.78	41.26	**74.87**	35.11	42.64	**75.12**
UCF101 [22]	5.0	Audio, Video	Acc	6.41	12.56	**70.22**	40.71	46.53	**72.99**
MiT10 [36]		Audio, Video	Acc	9.21	21.04	**50.13**	27.60	38.15	**52.66**
UCI-HAR [3]		Acce, Gyro	F1	35.66	45.21	**79.49**	37.70	44.50	**76.41**

Table 1: Scenario 1: Accuracy/F1 (%) of TACTFL and two baseline methods, FedAvg/FedOpt, under two missing label rates r_l. α denotes the control factor of the Dirichlet distribution for data partitioning (best results in **bold**, second best underlined).

Datasets	Modality		FedIoT [55]	FedProx [28]	MOON [27]	CreamFL [53]	FedMEKT [25]	TACTFL
mHealth [1]	Acce-Gyro	Acce	63.83	67.86	64.47	68.58	68.28	**68.90**
		Gyro	63.62	64.36	64.12	64.34	64.60	**64.75**
	Acce-Mage	Acce	69.99	70.00	69.71	69.51	**71.16**	71.09
		Mage	68.49	69.21	68.79	69.18	71.13	**71.84**
	Gyro-Mage	Gyro	65.43	65.37	65.90	66.05	**67.10**	67.21
		Mage	68.28	68.75	68.13	68.00	69.02	**69.30**
UR-Fall [24]	Acce-RGB	Acce	61.70	65.66	65.89	66.82	**70.66**	70.43
		RGB	57.88	59.26	62.00	62.34	66.70	**66.92**
	Acce-Depth	Acce	67.24	68.79	68.08	71.61	**72.68**	72.17
		Depth	60.76	68.08	68.16	**75.33**	75.22	74.99
	RGB-Depth	RGB	69.88	75.60	73.38	**78.57**	77.87	78.18
		Depth	67.61	70.04	66.48	69.18	70.57	**70.86**
Opportunity [5]	Acce-Gyro	Acce	71.75	72.25	72.96	73.33	**73.50**	73.37
		Gyro	72.08	71.43	72.09	72.10	72.15	**73.12**

Table 2: Scenario 1: F1 scores on FedMEKT [25] benchmarks. Paired modalities are for training, and each individual modality is for testing (best in **bold**, second best underlined).

Datasets	r_m	FedOpt [42]	MOON [27]	FedProx [28]	FPL [20]	FedIoT [55]	FedMSplit [6]	MFCPL [26]	TACTFL
UCI-HAR [3]	0.5	74.79	74.87	70.14	75.16	71.57	74.76	77.44	**77.92**
	0.7	66.17	73.63	69.37	74.28	70.58	73.32	75.61	**75.83**
	0.8	56.45	72.78	68.77	73.46	68.62	72.77	**75.19**	75.12
Hateful Memes [23]	0.5	54.62	52.67	52.75	54.97	53.98	55.57	56.30	**56.89**
	0.7	51.61	51.40	50.49	53.79	52.72	53.61	**55.69**	55.31
	0.8	49.58	50.68	50.23	52.94	51.31	52.65	55.04	**59.47**
MELD [40]	0.5	51.58	50.66	51.92	52.11	51.81	52.97	54.71	**55.26**
	0.7	49.27	47.51	51.83	50.09	49.48	51.65	53.62	**54.10**
	0.8	47.54	46.94	50.48	49.49	49.61	50.57	52.18	**53.25**

Table 3: Scenario 2: Performance for various missing-modality rates (r_m) on UCI-HAR, Hateful Memes, MELD datasets (best in **bold**, second best underlined).

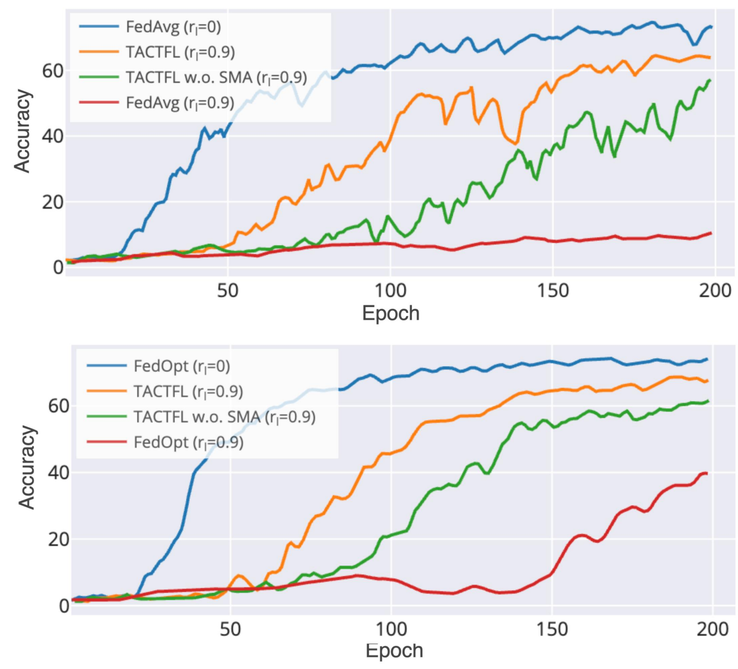

Figure 3: Test accuracy curve under four settings: 1. SSFL ($r_l = 0.9$); 2. TACTFL without SMA ($r_l = 0.9$); 3. Full TACTFL ($r_l = 0.9$); 4. Fully-supervised FL ($r_l = 0$).

Setting	Method	Rate				
		0.1	0.3	0.5	0.7	0.9
r_l	FedAvg [35]	71.02	70.81	67.28	60.25	10.49
	TACTFL	71.31	71.15	69.73	67.08	63.12
r_m	FedOpt [42]	74.48	73.20	71.46	68.56	56.32
	TACTFL	75.01	74.76	73.52	72.84	72.22

Table 4: Accuracy under varying levels of missing label and missing modality rates, r_l and r_m.

Temporal Window Size	90%	80%	70%	60%	50%
Accuracy	69.26	71.34	72.22	71.28	70.86

Table 5: Effect of temporal window size on performance (accuracy %).

Acknowledgements

This work was supported by the TORUS Project, EPSRC grant number EP/X036146/1. RSR is funded by the UKRI Turing AI Fellowship EP/V024817/1.

Contact Information

Guanxiong Sun[1]
guanxiong.sun@bristol.ac.uk
Majid Mirmehdi[2]
m.mirmehdi@bristol.ac.uk
Zahraa Abdallah[1]
zahraa.abdallah@bristol.ac.uk
Raul Santos-Rodriguez[1]
enrsr@bristol.ac.uk
Ian Craddock[1,3]
ian.craddock@bristol.ac.uk
Telmo de Menezes e Silva Filho[1]
telmo.silvafilho@bristol.ac.uk

[1] School of Engineering Mathematics and Technology
University of Bristol
Bristol, UK
[2] School of Computer Science
University of Bristol
Bristol, UK
[3] School of Civil, Aerospace and Design Engineering
University of Bristol
Bristol, UK