Research Gap

Federated Finetuning of LLMs

state of the art: Fedkseed FlexLoRA Fed PFT

→ reduce communication and handle heterogenous client resources.

→ aggregate by arithmetic averaging

GAP → No algorithm incorporates semantic or linguistic similarity of clients →

all clients are treated equally, even if their languages or text domains are very different. This causes degraded global models and poor cross lingual transfer.

→ no mathematical or algorithmic mechanism exploits these distance aggregation

Sun Mon Tue Wed Thu Fri Sat

Date :

(A) prompt tuning & Personalization in FL

FedPGP, FedTPG, Fed BPT → focus on communication efficiency and personalization

→ Aggregation is still parameter-space averaging → no geometry or transport alignment across client.

→ Theoritical foundations:

NeurIPS'24 prompt-port f

→ no theoritical bound connecting multilingual heterogenity, privacy noise and aggregation error.

privacy vs accuracy

PrE-Text and other DP works handle privacy by adding noise or generating synthetic data.

None analyze how DP noise interacts with linguistic heterogenetity or purpose a framework that combine both optimally.

One round of Lingua-OT-FL

1. clients

compute local updates

$$\Delta_k = \nabla_\phi L_k(\phi)$$

Every ML model learns by minimizing a loss function $L(\theta)$:

$$\min_\theta E_{(x,y) \sim P_{data}} \, l(f_\theta(x), y)$$

$f_\theta(x)$ model prediction

$l(-)$ → loss function

P_{data} → underlying data distribution

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} l(f_\theta(x_i), y_i)$$

use gradient descent

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t)$$

From centralized to federated

$$L(\theta) = \sum_{k=1}^{K} \frac{n_k}{N} L_k(\theta)$$

$$L_k(\theta) = \frac{1}{n_k} \sum l(f_\theta(x), y))$$

Gradient of the global loss

$$\nabla_\theta L(\theta) = \sum_{k=1}^{K} \frac{n_k}{N} \nabla_\theta L_k(\theta)$$

3) The FedAvg Algorithm

Local $\theta_k^{t+1} = \theta_t - \eta \nabla_\theta L_k(\theta_t)$

(Often multiple mini batches

$$\theta_{t+1} = \sum_{k=1}^{K} \frac{n_k}{N} \theta_k^{t+1}$$

The NonIID challenge Formally

4). Lets call the local optimum on client k

$\theta_k^* = \arg\min L_k(\theta)$

If all clients had identical data

θ_k^* would be the same

under non IID conditions

$\|\theta_i^* - \theta_j^*\| > 0$

distance → data heterogenity

Federated optimization as a communication problem

centralized SGD

FL update only after clients train locally for E epochs.

two timescale

1) Local updates (with clients)

2) Global aggregation (server round)

This saves communication but risks model drift.

Federated fine tuning for NLP

D_k (each client text data)

diff language diff domain.

They all share pretrained backbone

f_{θ_0} (like bert)

Instead of tuning all weights θ

clients only train small parameters

ϕ_k → prompts, LoRA adaptors or prefix vectors.

$f(x; \theta_0, \phi_k) = f_{\theta_0}(x) + g_{\phi_k}(x)$

Federations happens over ϕ_k instead of huge θ_0

The server aggregates $\{\phi_k\}$

$$\phi_{global} = \sum_{k=1}^{N} \frac{n_k}{N} \phi_k$$

sends them back to client.

Thus parameter efficient FL

= small communication, low memory, faster convergence

(7) Differential privacy

each client adds a gaussian noise $\mathcal{N}(0, \sigma^2 I)$

$$\theta_k^{t+1} = \theta_k^{t+1} + N(0, \sigma^2 I)$$

the server aggregates these noisy updates.

θ_k

This ensures that any single sample's influence on the model is statistically limited.

Formally the algorithm is (ϵ, δ)-DP

Differential privacy

any two dataset differing one examp

$$Pr\,[A(D_1) \in S] \leq e^{\varepsilon}\, pr\,[A(D_2) \in S] + \delta$$

Observer cannot tell whose data is this

8) Theoritical Convergence

when all IIDs are smooth

$$E[L(\theta_T)] - L^* = O\left(\frac{1}{T}\right)$$

Fed Avg → converges roughly like → centralized Stochastic gradient Descent,

under non IID data

the rate becomes

$$O\left(\frac{1}{T} + Bias_{nonIID}\right)$$

Bias depends on how far each local optimum θ_k^* deviates from the global one.

This → slower convergence for heterogenic data.

For smarter aggregation

language structure introduces another kind of heterogeneity

Demelon

Co-dopa

. Adding language distance

$d_{ij} = |S_i - S_j|$

instead of plain averaging weight updates using a decreasing function of distance.

$$\omega_k = \frac{e^{-\alpha d(k, global)}}{\sum_j e^{-\alpha d(j, global)}}$$

Aligning updates with optimal transport

→ minimizes the effort to move one distribution into another

Date :

Sun Mon Tue Wed Thu Fri Sat

In lingua-OT-FL →

align updates from different language so that semantically similar gradients correspond before averaging

server solves

$$\bar{\Delta}^{OT} = \arg\min_{\Delta} \sum \omega_k W_2^2(\Delta, \Delta_k)$$

W_2 is wasserstein barycenter.

→

clients → compute local update

$\Delta_k = \nabla_\phi L_k(\phi)$

and language signature S_k

server computes pairwise distance

$d_{ij} = \|S_i - S_j\|$

compute OT barycenter

$$\bar{\Delta}^{OT} = \arg\min \sum w_k W_2^2(\Delta, \Delta_k)$$

3. add Differential privacy

$$\bar{\Delta}^{OT-DP} = \bar{\Delta}^{OT} + \mathcal{N}(0, \sigma^2 I)$$

4. Global update

$$\phi_{global} \leftarrow \phi_{global} - \eta \bar{\Delta}^{OT-DP}$$

personalization

$$\phi_k = \lambda_k \phi_{global} + (1-\lambda) \phi_k$$

Optimal Transport Theory.

Standard Fed Avg → just takes a weighted mean of client updates

→ Thats fine if everyone's updates "point" roughly in the same direction

But multilingual or multidomain

NLP :

CA → Bangla update "focus on vowel normalization"

CB → English update - handle tense consistency

CC → Arabic handle script diff

Their gradients point in diff direction in parameter space.

plain averaging can cancel useful signals out.

so we need a smarter way to align these updates before combining them.

OT enters

The transport analogy

Imagine each client's update as a pile of sand spread across a 2D landscape. (The parameter space)

To merge them → we want to move each pile into a single "average" → but moving sand costs effort

The question OT answers:

→ what is the most efficient way to move (each pile into a single average) mass from one distribution to another?

→ The effort is measured by how far and how much sand is moved.

Say we have two probability distributions $p(x)$ and $Q(y)$

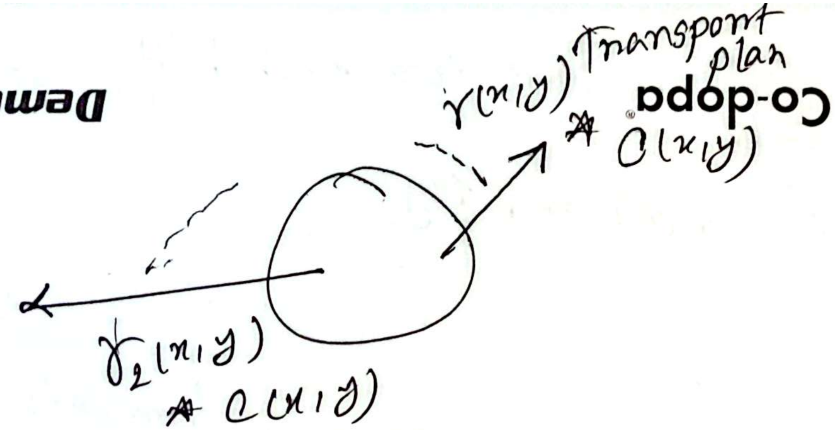
A transport plan $\gamma(x,y)$ tells us how much mass to move from x to y.

The transport cost is:

$$\text{cost}(P,Q,\gamma) = \iint c(x,y)\,\gamma(x,y)\,dx\,dy$$

$c(x,y) \rightarrow$ cost of moving unit mass (usually euclidan distance)

The optimal transport distance (also called as wasserstein distance) is the minimum cost over all valid γ.



Totalcost $(P,Q,\gamma) = \sum \gamma(x,y) * c(x,y) + \gamma_2(x,y) * c(x,y)$

So minimum cost OT distance

$$W_2^2(P,Q) = \min_{\gamma \in \Pi(P,Q)} \iint \|x-y\|^2 \gamma(x,y)\, dx\, dy$$

This gives a geometry aware notion "distance" between ~~distance~~ distributions.

From two Distributions to many:

The Barycenter:

If you have several clients with distr $P_1, P_2 \dots P_K$.

$$P^* = \arg\min_P \sum_k w_k W_2^2(P, P_k)$$

minimizes the weighted transport cost

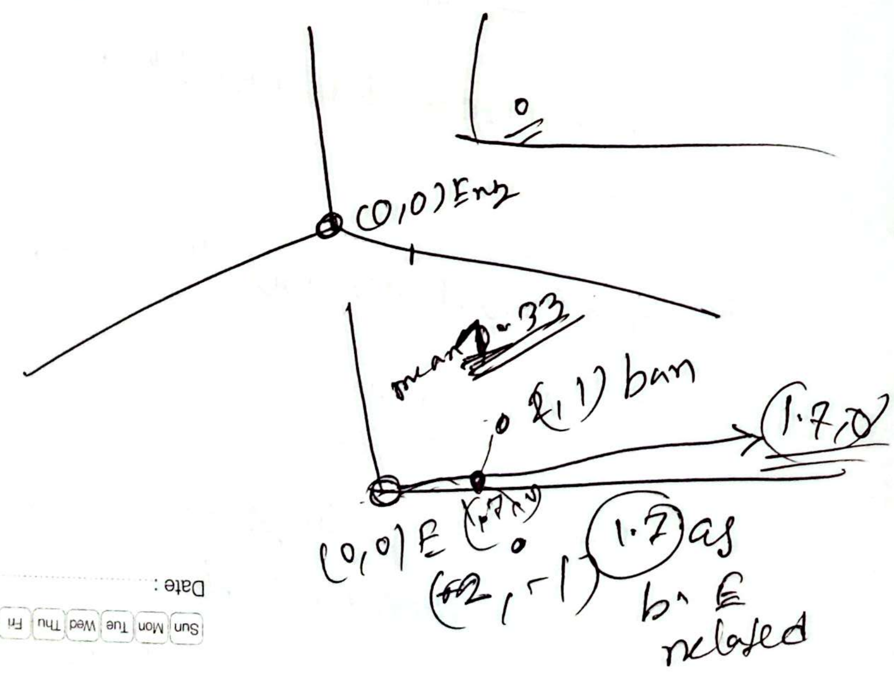
It's the "most central" distribution in the OT sense.

The one requiring the least overall movement from all clients.

Intution

mean → best if all data live on same line.

Wasserstein barycenter best if your data live on a curved manifold.



Applying OT in federated learning

$$\bar{\Delta}^{OT} = \arg\min_{\Delta} \sum w_k W_2^2(\Delta, \Delta_k)$$

This finds a global update that minimizes total transport cost across all client updates.

Find the update closest to everyon's considering how far apart their update geometry are

Demelon

Co-dopa

Linguistic weights.

we assign each client a weight

$$w_k = \frac{e^{-\alpha d(P_k, P_{global})}}{\sum_j e^{-\alpha d(P_j, P_{global})}}$$

where $d(\cdot,\cdot)$ is the language distance

→ computed from embeddings, phoneme stats, or perplexity between LMs (language models)

Languages that are closer to the global center influence aggregation more.

this lingua part of Lingua OTFL

Differential privacy in OT aggregation

→ After computing OT barycenter

$$\bar{\Delta}^{OT\text{-}DP} = \Delta^{OT} + \mathcal{N}(0, \sigma^2 I)$$

noise scale σ controls the privacy level ϵ

personalization After OT

$$\phi_k^{new} = \lambda_k \bar{\phi}_{global} + (1-\lambda_k)\phi_k^{local}$$

$$\lambda_k = \frac{1}{1+\beta d(P_k, P_{global})}$$

computing OT efficiency