# Federated Multi-Task Learning under a Mixture of Distributions

Othmane Marfoq[1,2,3] Giovanni Neglia[1,2] Aurélien Bellet[1]

Laetitia Kameni[3] Richard Vidal[3]

[1]Inria and [2]Université Côte d'Azur and [3]Accenture Labs

July 26, 2021

UNIVERSITÉ
CÔTE D'AZUR

Accenture Labs

# Introduction

- A (countable) set $\mathcal{T}$ of classification (or regression) tasks which represent the set of possible clients.

# Introduction

- A (countable) set $\mathcal{T}$ of classification (or regression) tasks which represent the set of possible clients.
- Data $\mathcal{S}_t = \{s_t^{(i)} \triangleq (\mathbf{x}_t^{(i)},\ y_t^{(i)})\}_{i=1}^{n_t}$ at client $t$ is drawn from a local distribution $\mathcal{D}_t$ over $\mathcal{X} \times \mathcal{Y}$.

# Introduction

- A (countable) set $\mathcal{T}$ of classification (or regression) tasks which represent the set of possible clients.
- Data $\mathcal{S}_t = \{s_t^{(i)} \triangleq (\mathbf{x}_t^{(i)},\ y_t^{(i)})\}_{i=1}^{n_t}$ at client $t$ is drawn from a local distribution $\mathcal{D}_t$ over $\mathcal{X} \times \mathcal{Y}$.
- Client $t$ wants to learn hypothesis $h_t$

$$\underset{h_t \in \mathcal{H}}{\text{minimize}}\, \mathcal{L}_{\mathcal{D}_t}(h_t) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}\left[l\left(h_t\left(\mathbf{x}\right), y\right)\right]. \tag{1}$$

# Introduction

- A (countable) set $\mathcal{T}$ of classification (or regression) tasks which represent the set of possible clients.
- Data $\mathcal{S}_t = \{s_t^{(i)} \triangleq (\mathbf{x}_t^{(i)},\ y_t^{(i)})\}_{i=1}^{n_t}$ at client $t$ is drawn from a local distribution $\mathcal{D}_t$ over $\mathcal{X} \times \mathcal{Y}$.
- Client $t$ wants to learn hypothesis $h_t$

$$\underset{h_t \in \mathcal{H}}{\text{minimize}}\, \mathcal{L}_{\mathcal{D}_t}(h_t) \triangleq \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}_t}\left[l\left(h_t\left(\mathbf{x}\right), y\right)\right]. \tag{1}$$

- Having personalized models for each client is a necessity in many FL applications.

# Related Work

- *Model agnostic meta-learning* (MAML) based federated *multi-task learning* (MTL).
- Clustered FL.
- Model interpolation: `APFL` and `MAPPER`.
- Federated MTL via task relationships: `MOCHA`, `pFedMe`, `L2SGD` and `FedU`.

# Related Work

- *Model agnostic meta-learning* (MAML) based federated *multi-task learning* (MTL).
- Clustered FL.
- Model interpolation: `APFL` and `MAPPER`.
- Federated MTL via task relationships: `MOCHA`, `pFedMe`, `L2SGD` and `FedU`.

  **Limitation**: restrictive assumptions or complex algorithms.

# An impossibility result

Some assumption on the local data distributions $\mathcal{D}_t,\ t \in \mathcal{T}$ are needed for federated learning to be possible:

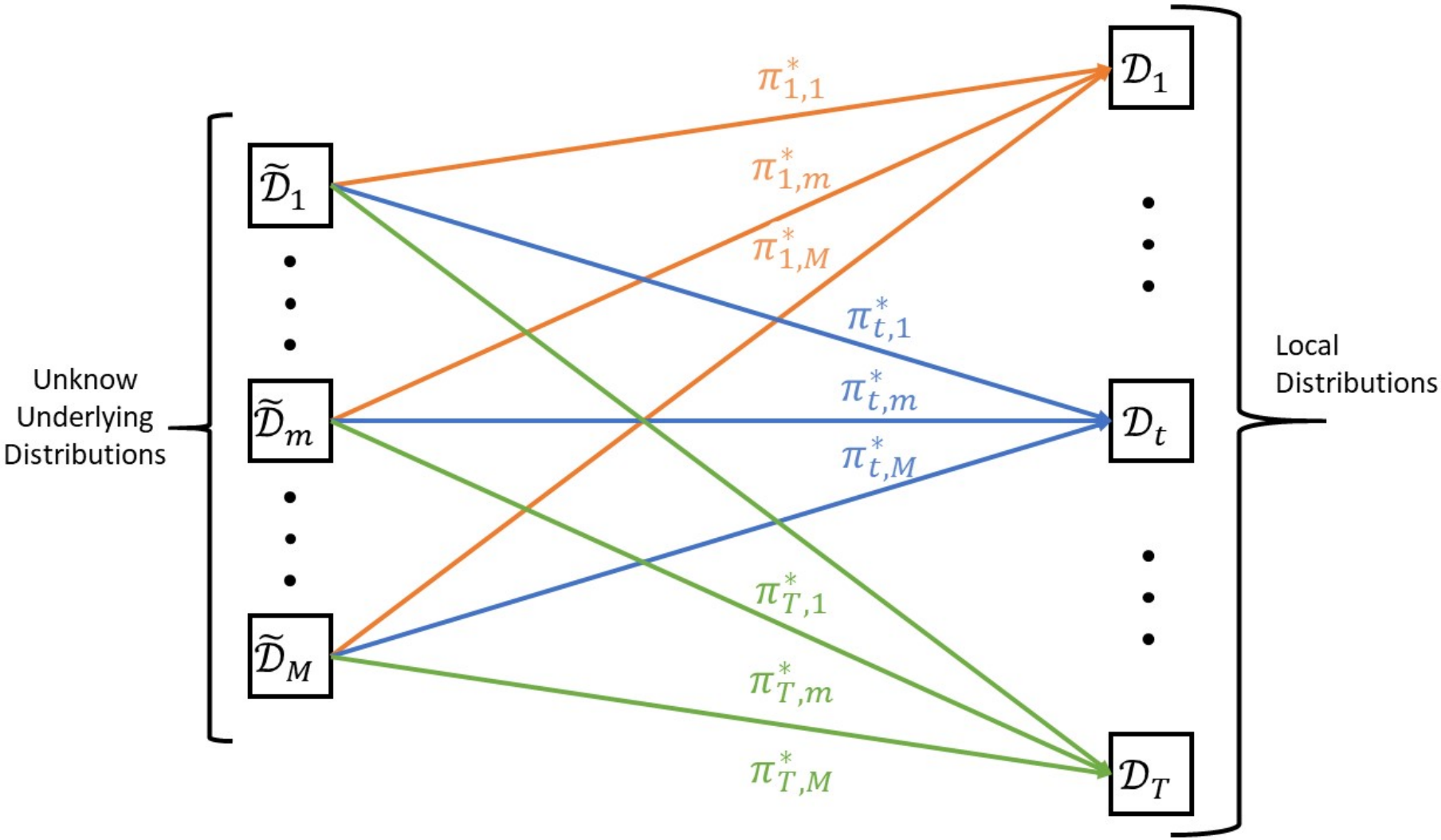
# An impossibility result

Some assumption on the local data distributions $\mathcal{D}_t,\ t \in \mathcal{T}$ are needed for federated learning to be possible:

- Federated learning with $T$ clients is equivalent to $T$ *semi-supervised learning* (SSL) problems.

- With no assumptions on the data distribution, SSL is impossible. (Ben-David et al. 2008; Darnstädt et al. 2013; Göpfert et al. 2019).

# Main assumption



# Generalizing Existing Frameworks

The generative model in the mixture assumption extends/covers some popular multi-task/personalized FL formulations in the literature.

# Generalizing Existing Frameworks

The generative model in the mixture assumption extends/covers some popular multi-task/personalized FL formulations in the literature.

**Example** (**Clustered Federated Learning**): The mixture assumption recovers this scenario considering $M = C$ and $\pi^*_{tc} = 1$ if task (client) $t$ is in cluster $c$ and $\pi^*_{tc} = 0$ otherwise.

# Main Contributions

- Flexible assumption for personalized FL (mixtures of components).

# Main Contributions

- Flexible assumption for personalized FL (mixtures of components).
- EM-like learning algorithms with convergence guarantees (both in client-server and fully-decentralized settings).

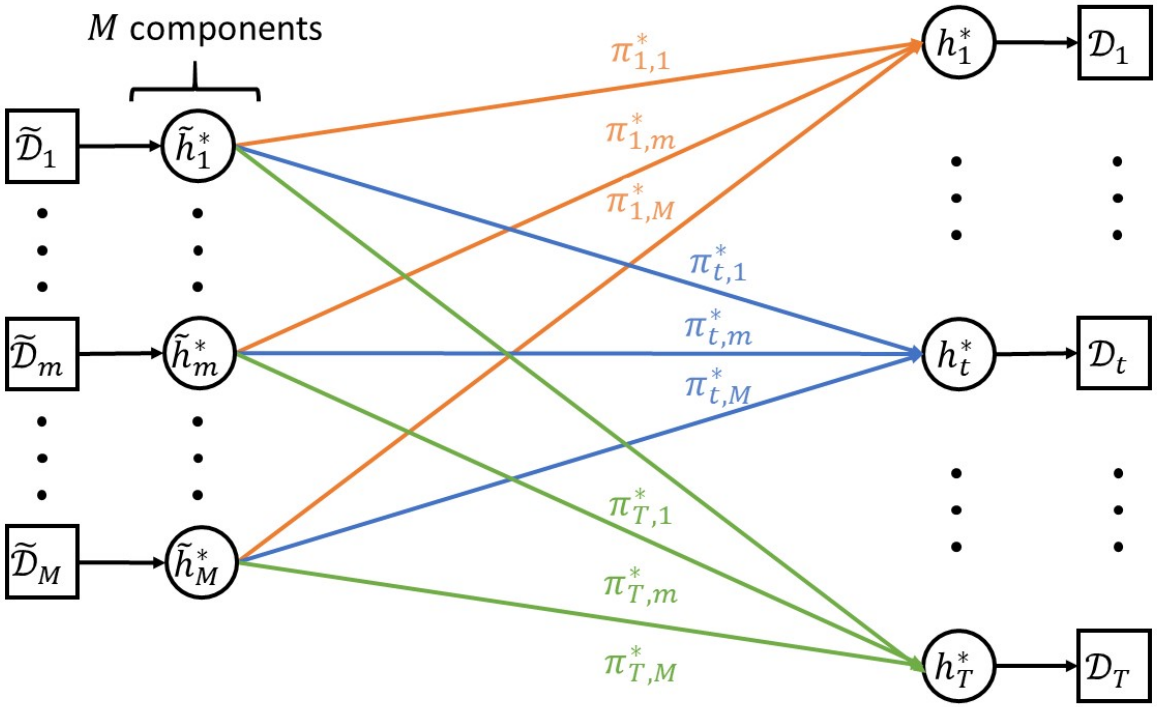
# Main Contributions

- Flexible assumption for personalized FL (mixtures of components).
- EM-like learning algorithms with convergence guarantees (both in client-server and fully-decentralized settings).
- More general *federated surrogate optimization* framework.

# Main Contributions

- Flexible assumption for personalized FL (mixtures of components).
- EM-like learning algorithms with convergence guarantees (both in client-server and fully-decentralized settings).
- More general *federated surrogate optimization* framework.
- Higher accuracy and fairness than SOTA algorithms, even for clients not present at training time.

# Learning under a mixture model




## Proposition (informal)

$$h^*_t = \sum_{m=1}^{M} \breve{\pi}_{tm} h_{\breve{\theta}_m}(\mathrm{x}), \quad \forall t \in \mathcal{T} \tag{2}$$

# Learning under a mixture model

- Estimate the parameters $\breve{\Theta}$ and $\breve{\pi}_t,\ 1 \leq t \leq T$, minimizing:

$$f(\Theta, \Pi) \triangleq -\frac{\log p(\mathcal{S}_{1:T}|\Theta, \Pi)}{n} \triangleq -\frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n_t} \log p(s_t^{(i)}|\Theta, \pi_t), \quad (3)$$

# Learning under a mixture model

- Estimate the parameters $\breve{\Theta}$ and $\breve{\pi}_t,\ 1 \leq t \leq T$, minimizing:

$$f(\Theta, \Pi) \triangleq -\frac{\log p(\mathcal{S}_{1:T}|\Theta, \Pi)}{n} \triangleq -\frac{1}{n}\sum_{t=1}^{T}\sum_{i=1}^{n_t} \log p(s_t^{(i)}|\Theta, \pi_t), \quad (3)$$

- Use Eq. (2) to get the client predictor for the $T$ clients present at training time.

- Clients $t'$ not participating at the training, learn $\pi_{t'}$ in a single shot, then use Eq. (2)

# Expectation-Maximization

A natural approach to solve problem (3) is via the

*Expectation-Maximization* (EM) algorithm

# Expectation-Maximization

A natural approach to solve problem (3) is via the

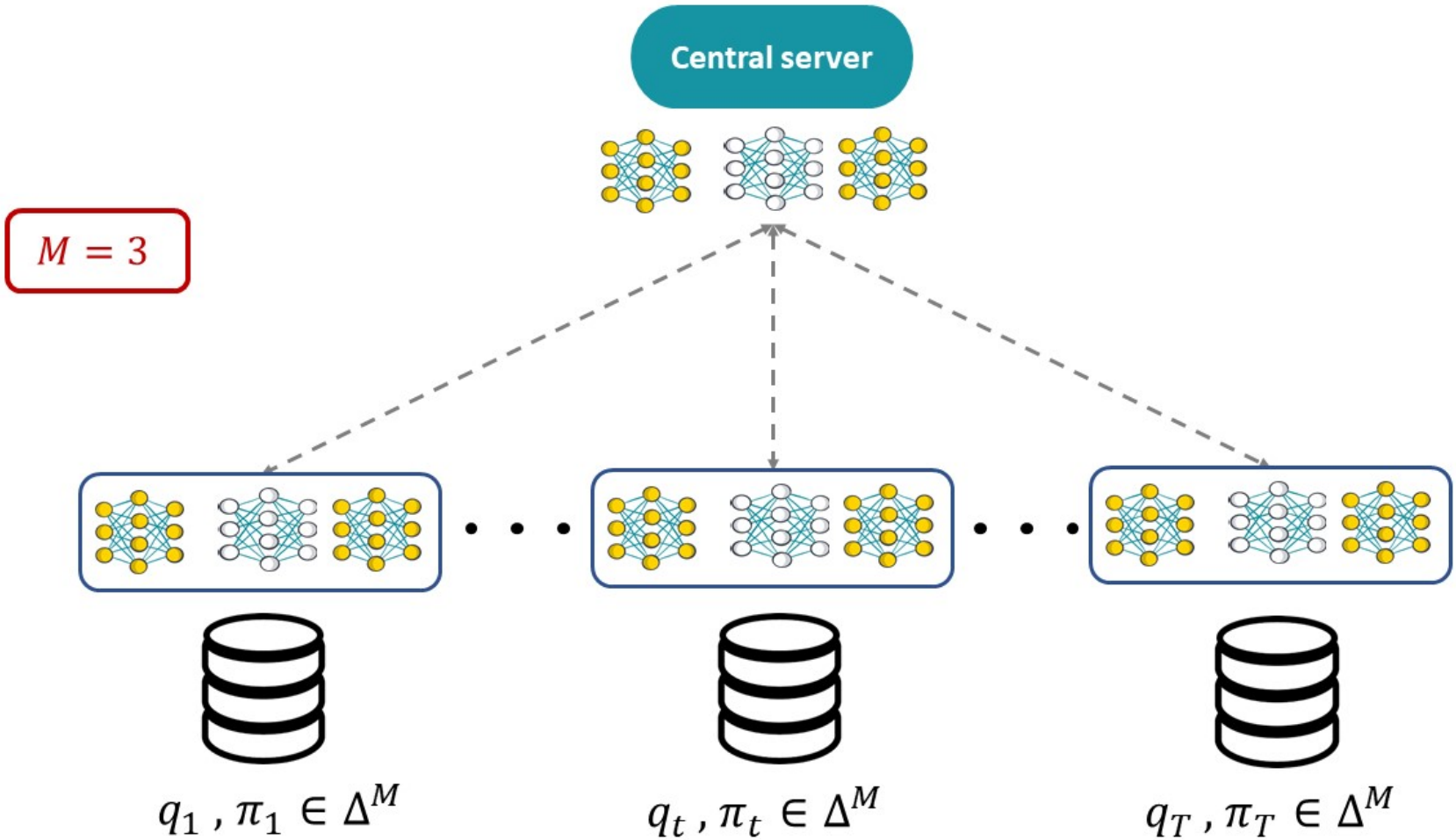
*Expectation-Maximization* (EM) algorithm

**E-step:** $$q_t^{k+1}(z_t^{(i)} = m) \propto \pi_{tm}^k \cdot \exp\left(-l(h_{\theta_m^k}(\mathbf{x}_t^{(i)}), y_t^{(i)})\right).$$

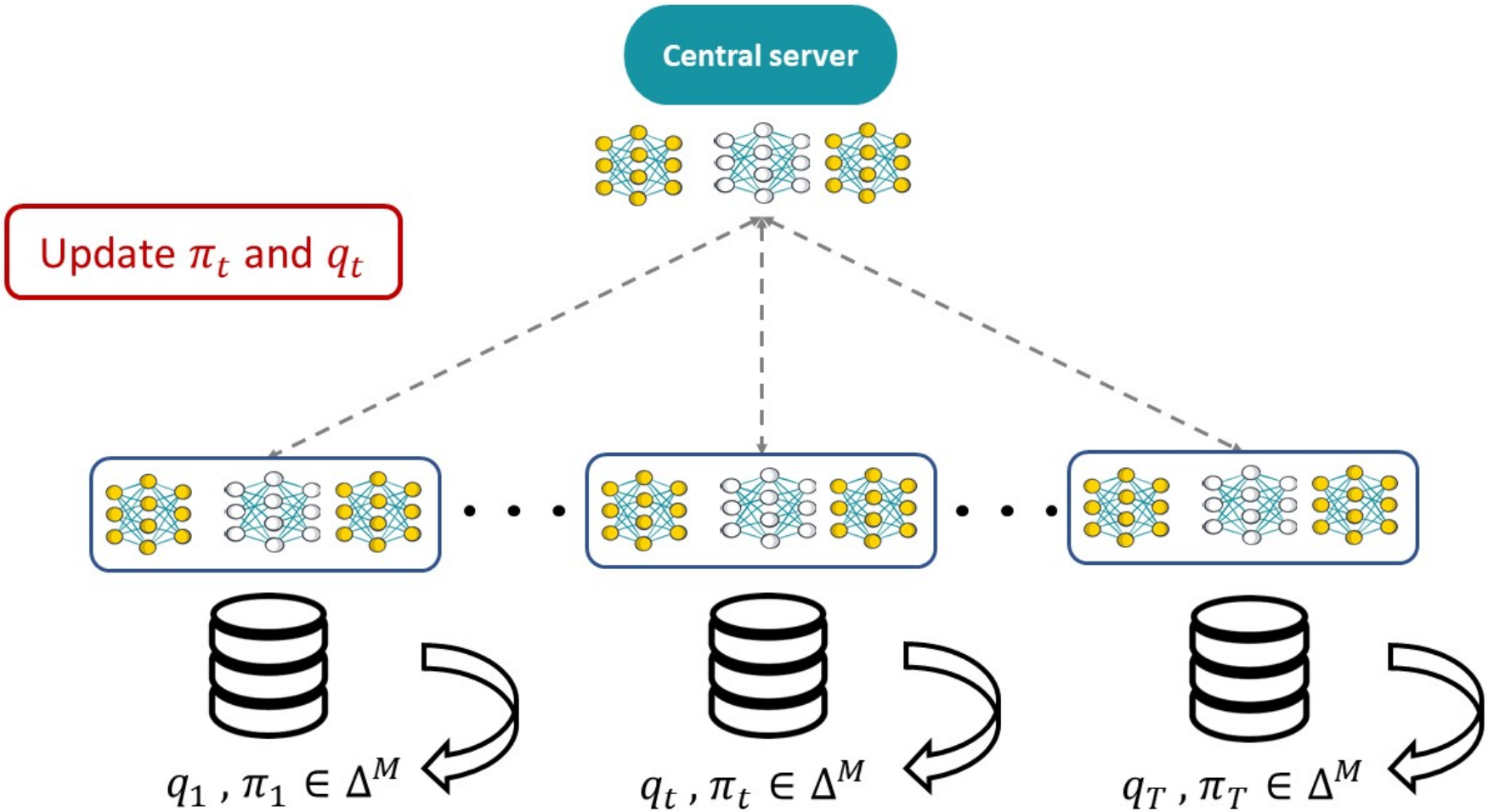
**M-step:** $$\pi_{tm}^{k+1} = \frac{\sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m)}{n_t},$$

$$\theta_m^{k+1} \in \arg\min_{\theta \in \mathbb{R}^d} \sum_{t=1}^{T} \sum_{i=1}^{n_t} q_t^{k+1}(z_t^{(i)} = m) \cdot l\big(h_\theta(\mathbf{x}_t^{(i)}), y_t^{(i)}\big).$$

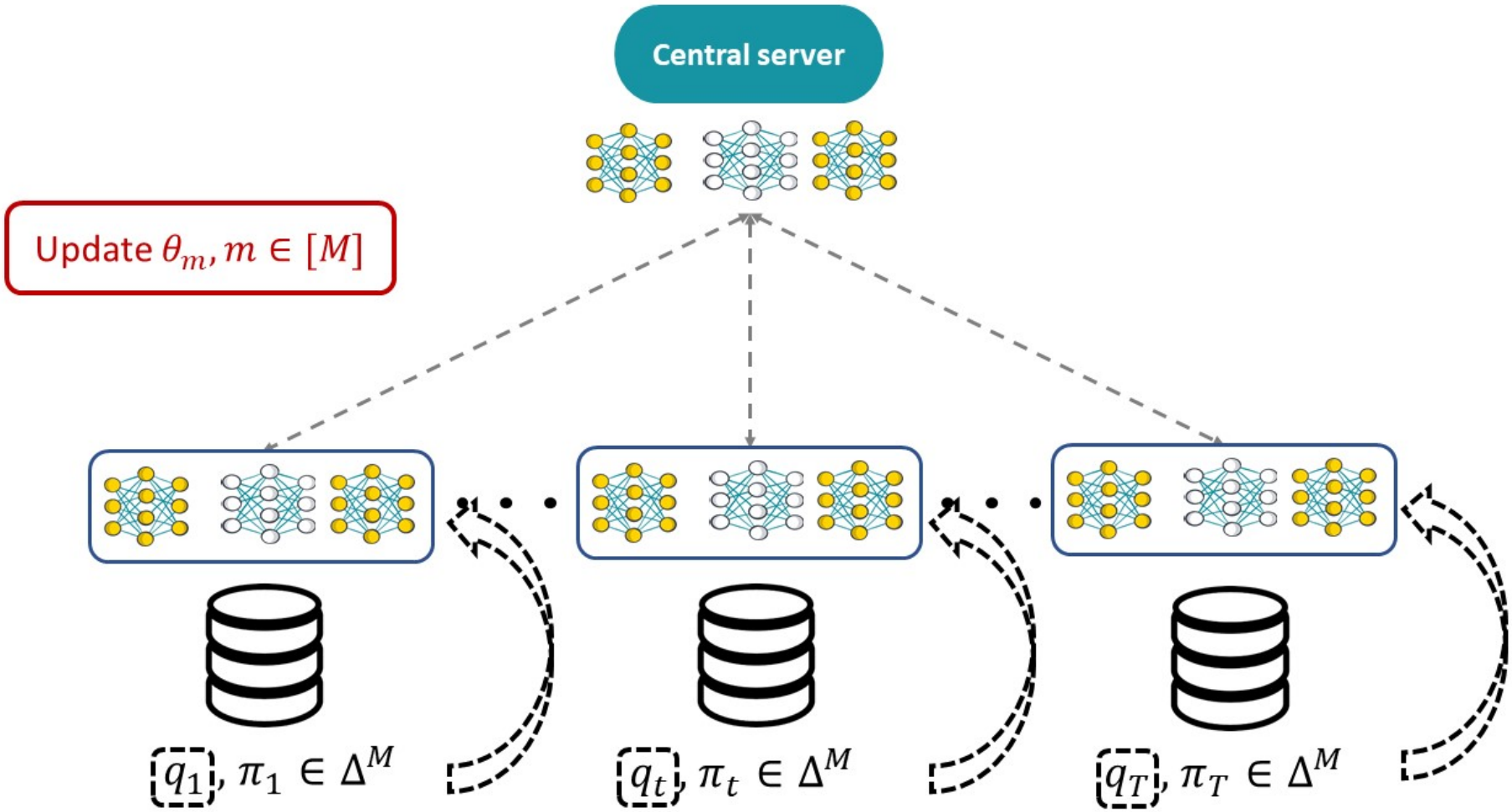
# Federated Expectation-Maximization



# Federated Expectation-Maximization



# Federated Expectation-Maximization



# Federated Expectation Maximization

## Theorem

*Under Assumptions 1–3 and some other mild assumptions, when clients use SGD as local solver with learning rate* $\eta = \frac{a_0}{\sqrt{K}}$*, after a large enough number of communication rounds* $K$*,* `FedEM`*'s iterates satisfy:*

$$\frac{1}{K}\sum_{k=1}^{K}\mathbb{E}\left\|\nabla_{\Theta} f\left(\Theta^k, \Pi^k\right)\right\|_F^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{K}}\right),$$

$$\frac{1}{K}\sum_{k=1}^{K}\Delta_{\Pi} f(\Theta^k, \Pi^k) \leq \mathcal{O}\left(\frac{1}{K^{3/4}}\right),$$

*where the expectation is over the random batches samples, and*

$$\Delta_{\Pi} f(\Theta^k, \Pi^k) \triangleq f\left(\Theta^k, \Pi^k\right) - f\left(\Theta^k, \Pi^{k+1}\right) \geq 0.$$

# Surrogate Federated Optimization

- `FedEM` can be seen as a particular instance of a more general framework that we call *federated surrogate optimization*.

# Surrogate Federated Optimization

- `FedEM` can be seen as a particular instance of a more general framework that we call *federated surrogate optimization*.
- This framework minimizes an objective function $\sum_{t=1}^{T} \omega_t f_t(\mathbf{u}, \mathbf{v}_t)$
- Each client $t \in [T]$ can compute a partial first order surrogate of $f_t$.

# Experiments

| Dataset | Local | FedAvg | FedProx | FedAvg+ | clustered FL | pFedMe | FedEM (Ours) |
|---|---|---|---|---|---|---|---|
| FEMNIST | 71.0 / 57.5 | 78.6 / 63.9 | 78.9 / 64.0 | 75.3 / 53.0 | 73.5 / 55.1 | 74.9 / 57.6 | **79.9** / **64.8** |
| EMNIST | 71.9 / 64.3 | 82.6 / 75.0 | 83.0 / 75.4 | 83.1 / 75.8 | 82.7 / 75.0 | 83.3 / 76.4 | **83.5** / **76.6** |
| CIFAR10 | 70.2 / 48.7 | 78.2 / 72.4 | 78.0 / 70.8 | 82.3 / 70.6 | 78.6 / 71.2 | 81.7 / 73.6 | **84.3** / **78.1** |
| CIFAR100 | 31.5 / 19.9 | 40.9 / 33.2 | 41.0 / 33.2 | 39.0 / 28.3 | 41.5 / 34.1 | 41.8 / 32.5 | **44.1** / **35.0** |
| Shakespeare | 32.0 / 16.6 | **46.7** / 42.8 | 45.7 / 41.9 | 40.0 / 25.5 | 46.6 / 42.7 | 41.2 / 36.8 | **46.7** / **43.0** |
| Synthetic | 65.7 / 58.4 | 68.2 / 58.9 | 68.2 / 59.0 | 68.9 / 60.2 | 69.1 / 59.0 | 69.2 / 61.2 | **74.7** / **66.7** |

Table: Test accuracy: average across clients / bottom decile.

# Experiments

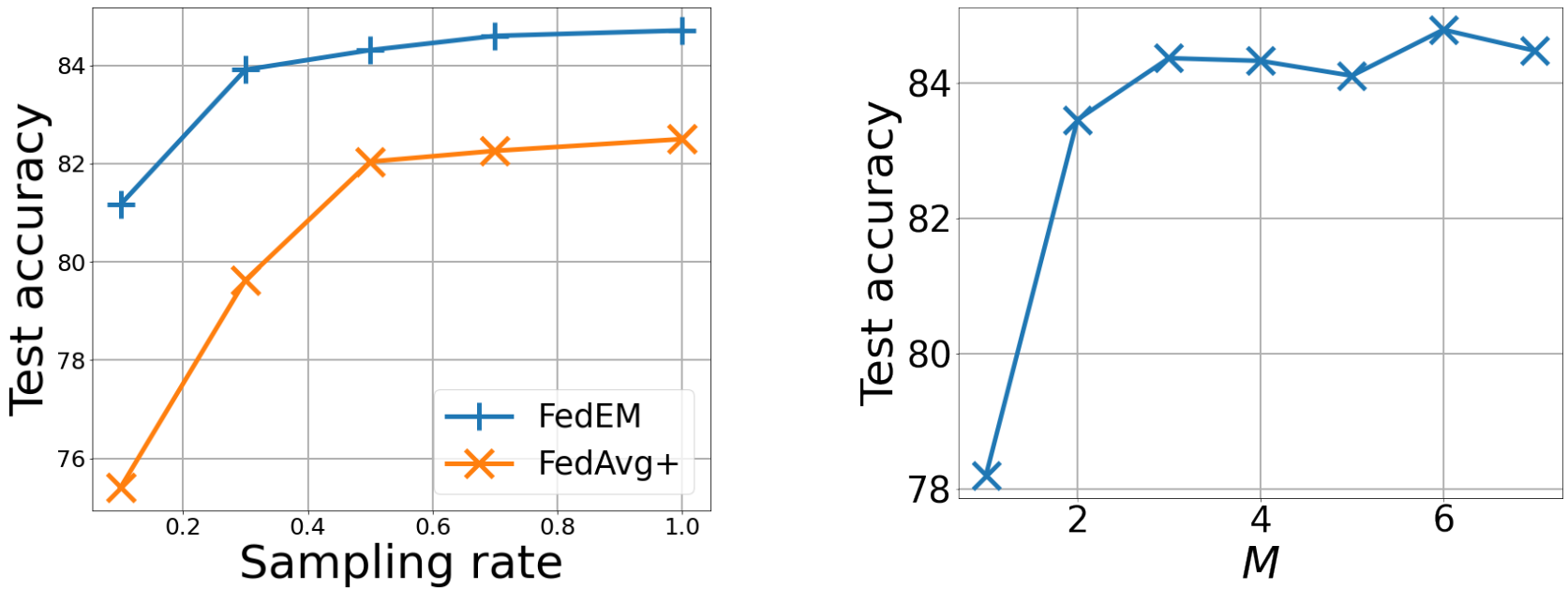



Figure: Effect of client sampling rate (left) and `FedEM` number of mixture components $M$ (right) on the test accuracy for CIFAR10.

# Experiments

| Dataset | FedAvg | FedAvg+ | FedEM |
|---|---|---|---|
| FEMNIST | 78.3 (80.9) | 74.2 (84.2) | **79.1** (81.5) |
| EMNIST | 83.4 (82.7) | 83.7 (92.9) | **84.0** (83.3) |
| CIFAR10 | 77.3 (77.5) | 80.4 (80.5) | **85.9** (90.7) |
| CIFAR100 | 41.1 (42.1) | 36.5 (55.3) | **47.5** (46.6) |
| Shakespeare | **46.7** (47.1) | 40.2 (93.0) | **46.7** (46.6) |
| Synthetic | 68.6 (70.0) | 69.1 (72.1) | **73.0** (74.1) |

Table: Average test accuracy across **clients unseen at training** (train accuracy in parenthesis).

# Conclusion

Thank you for your attention

# References I


- Ben-David, Shai, Tyler Lu, and D. Pál (2008). "Does Unlabeled Data Provably Help? Worst-case Analysis of the Sample Complexity of Semi-Supervised Learning". In: *COLT*.
- Darnstädt, Malte, H. U. Simon, and Balázs Szörényi (2013). "Unlabeled Data Does Provably Help". In: *STACS*.
- Dinh, Canh T, Nguyen H Tran, and Tuan Dung Nguyen (2020). "Personalized Federated Learning with Moreau Envelopes". In: *arXiv preprint arXiv:2006.08848*.
- Göpfert, Christina et al. (2019). "When can unlabeled data improve the learning rate?" In: *Conference on Learning Theory*. PMLR, pp. 1500–1518.
- Mansour, Yishay et al. (2020). "Three approaches for personalization with applications to federated learning". In: *arXiv preprint arXiv:2002.10619*.

# References II


Sattler, Felix, Klaus-Robert Müller, and Wojciech Samek (2020). "Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints". In: *IEEE Transactions on Neural Networks and Learning Systems*.

Smith, Virginia et al. (2017). "Federated Multi-Task Learning". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17. Long Beach, California, USA: Curran Associates Inc., pp. 4427–4437. ISBN: 9781510860964.

# Main assumptions

## Assumption

*There exist $M$ underlying (independent) distributions $\tilde{\mathcal{D}}_m$, $1 \leq m \leq M$, such that for $t \in \mathcal{T}$, $\mathcal{D}_t$ is mixture of the distributions $\{\tilde{\mathcal{D}}_m\}_{m=1}^M$ with weights $\pi_t^* = [\pi_{t1}^*, \ldots, \pi_{tm}^*] \in \Delta^M$, i.e.*

$$z_t \sim \mathcal{M}(\pi_t^*), \quad ((x_t, y_t) \,|z_t = m) \sim \tilde{\mathcal{D}}_m, \quad \forall t \in \mathcal{T}, \tag{4}$$

*where $\mathcal{M}(\pi)$ is a multinomial (categorical) distribution with parameters $\pi$.*

# Main assumptions

## Assumption

*For all* $m \in [M]$, *we have* $\tilde{\mathcal{D}}_m(\mathrm{x}) = \mathcal{D}(\mathrm{x})$.

## Assumption

$\tilde{\mathcal{H}} = \{h_\theta\}_{\theta \in \mathbb{R}^d}$ *is a set of hypotheses parameterized by* $\theta \in \mathbb{R}^d$, *whose convex hull is in* $\mathcal{H}$. *For each distribution* $\tilde{\mathcal{D}}_m$ *with* $m \in [M]$, *there exists a hypothesis* $h_{\theta_m^*}$, *such that*

$$l\left(h_{\theta_m^*}(\mathrm{x}), y\right) = -\log p_m(y|\mathrm{x}) + c, \tag{5}$$

*where* $c \in \mathbb{R}$, *is a normalization constant.* $l(\cdot, \cdot)$ *is then the log loss associated to* $p_m(y|\mathrm{x})$.

# Generalizing Existing Frameworks

The generative model in the mixture assumption extends/covers some popular multi-task/personalized FL formulations in the literature.

# Generalizing Existing Frameworks

The generative model in the mixture assumption extends/covers some popular multi-task/personalized FL formulations in the literature.

- **Clustered Federated Learning** Sattler et al. 2020: The mixture assumption recovers this scenario considering $M = C$ and $\pi^*_{tc} = 1$ if task (client) $t$ is in cluster $c$ and $\pi^*_{tc} = 0$ otherwise.

# Generalizing Existing Frameworks

The generative model in the mixture assumption extends/covers some popular multi-task/personalized FL formulations in the literature.

- **Clustered Federated Learning** Sattler et al. 2020: The mixture assumption recovers this scenario considering $M = C$ and $\pi^*_{tc} = 1$ if task (client) $t$ is in cluster $c$ and $\pi^*_{tc} = 0$ otherwise.

- **Personalization via model interpolation** Mansour et al. 2020; Dinh et al. 2020: Each client model can be seen as a linear combination of $M = T + 1$ models $h_m = h_{\mathrm{loc},m}$ for $m \in [T]$ and $h_0 = h_{\mathrm{glob}}$ with specific weights $\pi^*_{tt} = \alpha_t$, $\pi^*_{t0} = 1 - \alpha_t$, and $\pi^*_{tt'} = 0$ for $t' \in [T] \setminus \{t\}$.

# Generalizing Existing Frameworks

- **Federated MTL via task relationships** Smith et al. 2017 When predictors $h_{\theta_m^*}$ are linear and have bounded norm, our framework leads to the same ASO formulation used in Smith et al. 2017, i.e.,

$$\min_{W,\Omega} \sum_{t=1}^{T} \sum_{i=1}^{n_t} l(h_{w_t}(\mathbf{x}_t^{(i)}), y_t^{(i)}) + \lambda \operatorname{tr}(W\Omega W^{\mathsf{T}}),$$

# Learning under a mixture model

## Proposition

*Let $l(\cdot, \cdot)$ be the mean squared error loss, the logistic loss or the cross-entropy loss, and $\breve{\Theta}$ and $\breve{\Pi}$ be a solution of the following optimization problem:*

$$\underset{\Theta,\Pi}{\text{minimize}} \underset{t\sim D_{\mathcal{T}}}{\mathbb{E}} \underset{(\mathrm{x},y)\sim\mathcal{D}_t}{\mathbb{E}} \left[-\log \mathcal{D}_t(\mathrm{x}, y|\Theta, \pi_t)\right], \tag{6}$$

*where $D_{\mathcal{T}}$ is any distribution with support $\mathcal{T}$. Under Assumptions 1, 2, and 3, the predictors*

$$h_t^* = \sum_{m=1}^{M} \breve{\pi}_{tm} h_{\breve{\theta}_m}(\mathrm{x}), \quad \forall t \in \mathcal{T} \tag{7}$$

*minimize $\mathbb{E}_{(\mathrm{x},y)\sim\mathcal{D}_t}[l(h_t(\mathrm{x}), y)]$ and thus solve Problem (1).*

# Federated Expectation Maximization

**Algorithm 1** `FedEM`

**Input:** data $\mathcal{S}_{1:T}$; number of mixture distributions $M$; number of communication rounds $K$
**for** iterations $k = 1, \ldots, K$ **do**
  server broadcast $\theta_m^{k-1},\ 1 \leq m \leq M$ to the $T$ clients
  **for** tasks $t = 1, \ldots, T$ in parallel over $T$ clients **do**
    **for** component $m = 1, \ldots, M$ **do**
      update $q_t^k(z_t^{(i)} = m)\ \forall i \in \{1, \ldots, n_t\}$
      update $\pi_{tm}^k$
      $\theta_{m,t}^k \leftarrow \texttt{LocalSolver}(m,\ \theta_m^{k-1},\ q_t^k,\ \mathcal{S}_t)$
  client $t$ sends $\theta_{m,t}^k,\ 1 \leq m \leq M$, to the server
  **for** component $m = 1, \ldots, M$ **do**
    $\theta_m^k \leftarrow \sum_{t=1}^T \frac{n_t}{n} \times \theta_{m,t}^k$

# Surrogate Federated Optimization

## Definition (Partial first-order surrogate)

A function $g(\mathbf{u}, \mathbf{v}) : \mathbb{R}^{d_u} \times \mathcal{V} \to \mathbb{R}$ is a partial first-order surrogate of $f(\mathbf{u}, \mathbf{v})$ wrt $\mathbf{u}$ near $(\mathbf{u}_0, \mathbf{v}_0) \in R^{d_u} \times \mathcal{V}$ when the following conditions are satisfied:

- $g(\mathbf{u}, \mathbf{v}) \geq f(\mathbf{u}, \mathbf{v})$ for all $\mathbf{u} \in \mathbb{R}^{d_u}$ and $\mathbf{v} \in \mathcal{V}$;
- $r(\mathbf{u}, \mathbf{v}) \triangleq g(\mathbf{u}, \mathbf{v}) - f(\mathbf{u}, \mathbf{v})$ is differentiable and $L$-smooth with respect to $\mathbf{u}$. Moreover, we have $r(\mathbf{u}_0, \mathbf{v}_0) = 0$ and $\nabla_{\mathbf{u}} r(\mathbf{u}_0, \mathbf{v}_0) = 0$.
- $g(\mathbf{u}, \mathbf{v}_0) - g(\mathbf{u}, \mathbf{v}) = d_{\mathcal{V}}(\mathbf{v}_0, \mathbf{v})$ for all $\mathbf{u} \in \mathbb{R}^{d_u}$ and $\mathbf{v} \in \arg\min_{\mathbf{v}' \in \mathcal{V}} g(\mathbf{u}, \mathbf{v}')$, where $d_{\mathcal{V}}$ is non-negative and $d_{\mathcal{V}}(\mathbf{v}, \mathbf{v}') = 0 \iff \mathbf{v} = \mathbf{v}'$.