The power of adaptivity in representation learning: From meta-learning to federated learning

Sewoong Oh

Paul G. Allen School of Computer Science and Engineering
University of Washington

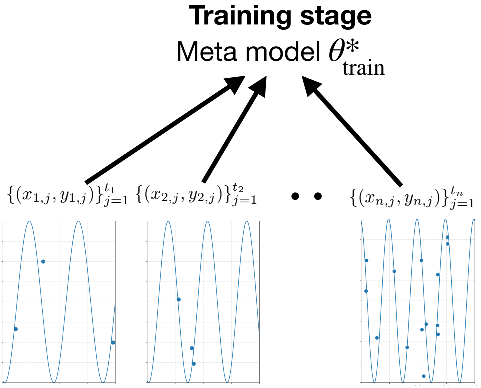
joint work with

Liam Collins
(UT Austin)

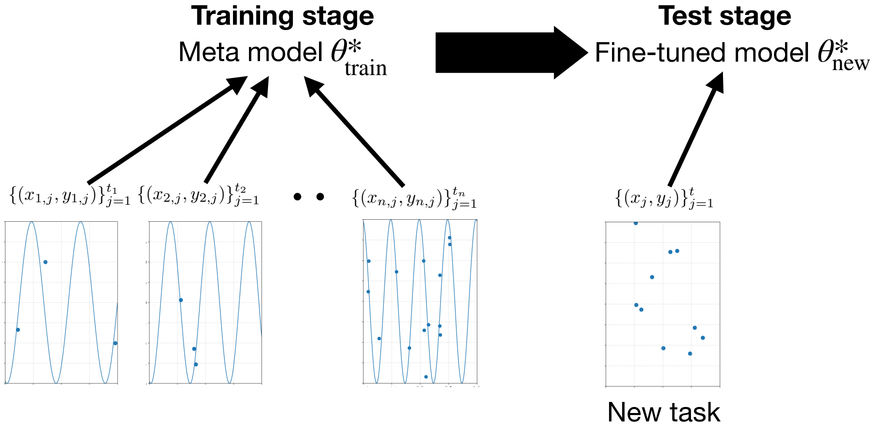
Aryan Mokhtari
(UT Austin)

Sanjay Shakkottai
(UT Austin)

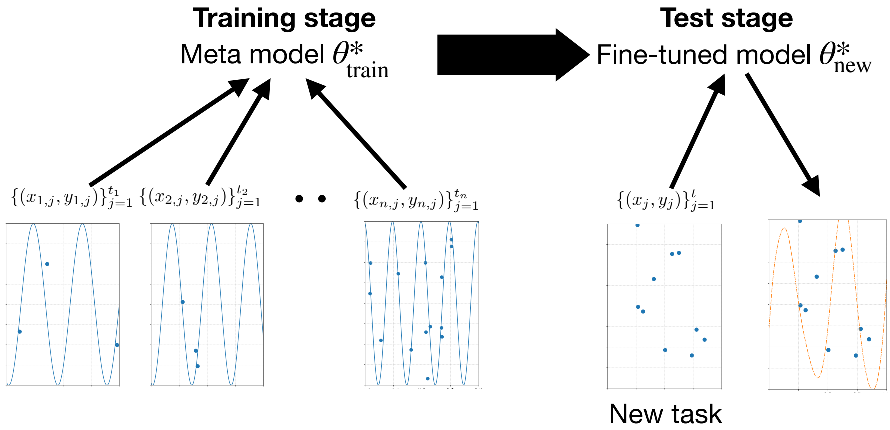
Meta-learning for few-shot learning [FAL17]



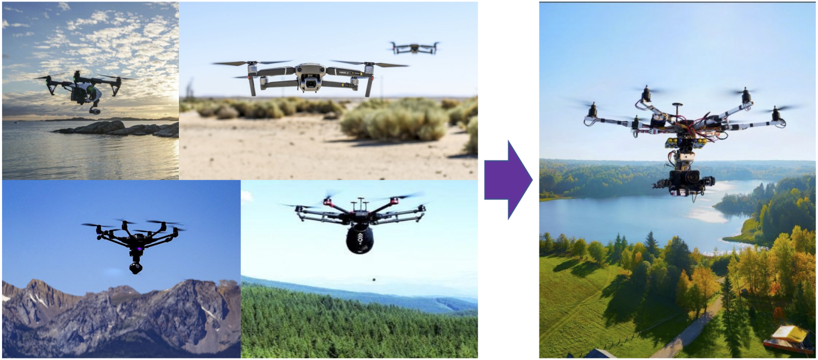
Meta-learning for few-shot learning [FAL17]



Meta-learning for few-shot learning [FAL17]



Central goal: generalize to new but similar tasks



Training Data

Task in new Scenario

Figure: Image Credits: bit.ly/3i5m8ay, bit.ly/3w723ZY, bit.ly/3KHMQ5E, bit.ly/3i7pREJ, bit.ly/34I1ytT

How to do meta-learning

- Suppose that we have access to
  (a) Training phase: large number of similar but distinct tasks each with small data
  (b) Test phase: a small amount of data available just prior to deployment from the deployment environment

- Given this setup how should we train our model?
- Possible Approach:
  (a) Build a model using data from the training phase
  (b) Fine-tune the model using the small amount of deployment data

How can we build a model that is easily fine-tunable?

First attempt: Build a model to minimize average training loss, and then fine-tune for deployment

Pooling all the data together

Average Risk Minimization (ARM) + Fine-tuning

- Set of tasks: $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{i=n}$ coming from distribution p
- Select a model $\boldsymbol{\theta}^*_{train}$

Average Risk (Loss) Minimization

$\boldsymbol{\theta}^*_{train} \in \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta})$

- A new task $\mathcal{T}_{test}$ is revealed, drawn according to dist. p
- Fine-tune the model: $\boldsymbol{\theta}^*_{train} \rightarrow \boldsymbol{\theta}^*_{new}$
- Performance goal: $f_{test}(\boldsymbol{\theta}^*_{new})$

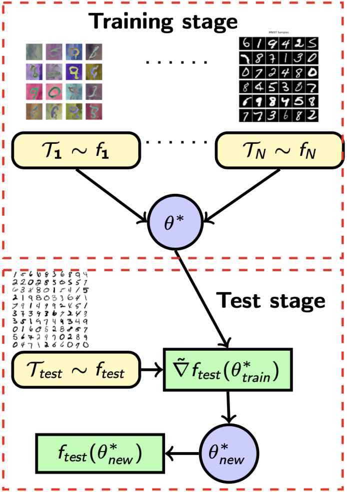


Image Credits: https://bit.ly/392pda9, https://bit.ly/3EEIElq

Pooling data has lost the structural information

- Suppose we have images from a large number of classes (e.g., Imagenet)
  - Task $=$ classifying images among a K-subset of these classes, small K

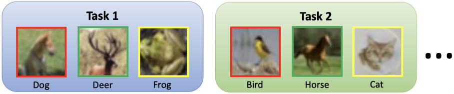
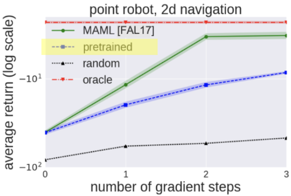


Image credits: Alex Krizhevsky, Learning Multiple Layers of Features from Tiny Images, 2009.

- ARM $+$ Fine-tuning has mixed performance [FAL17]

	5-way Accuracy	
MiniImagenet (Ravi & Larochelle, 2017)	1-shot	5-shot
fine-tuning baseline	$28.86 \pm 0.54\%$	$49.79 \pm 0.79\%$
nearest neighbor baseline	$41.08 \pm 0.70\%$	$51.04 \pm 0.65\%$
matching nets (Vinyals et al., 2016)	$43.56 \pm 0.84\%$	$55.31 \pm 0.73\%$
meta-learner LSTM (Ravi & Larochelle, 2017)	$43.44 \pm 0.77\%$	$60.60 \pm 0.71\%$
MAML, first order approx. (Finn et al., 2017)	$\mathbf{48.07 \pm 1.75\%}$	$\mathbf{63.15 \pm 0.91\%}$
MAML (Finn et al., 2017)	$\mathbf{48.70 \pm 1.84\%}$	$\mathbf{63.11 \pm 0.92\%}$

"Fine-tuning baseline": Few-shot image classification accuracy of ARM after fine-tuning (image taken from [FAL17])



"Pretrained": Fine-tuning reward for ARM on robot 2d navigation task (image taken from [FAL17])

Model-Agnostic Meta Learning (MAML) [FAL17]

- Set of tasks: $\mathcal{T} = \{\mathcal{T}_i\}_{i=1}^{i=n}$ coming from distribution p
- Select a model $\boldsymbol{\theta}^*_{train}$ such that

New objective

$$\boldsymbol{\theta}^*_{train} \in \arg\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta} - \alpha \nabla f_i(\boldsymbol{\theta}))$$

- A new task $\mathcal{T}_{test}$ is revealed, drawn according to dist. p
- Fine-tune the model: $\boldsymbol{\theta}^*_{train} \rightarrow \boldsymbol{\theta}^*_{new}$
- Performance goal: $f_{test}(\boldsymbol{\theta}^*_{new})$

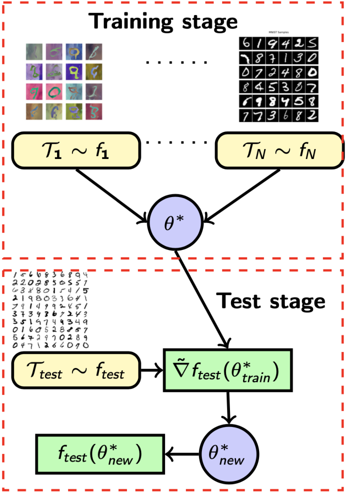


Image Credits: https://bit.ly/392pda9, https://bit.ly/3EEIElq

Original motivation: finding the right initialization for adaptation.

- **Average Risk Minimization (ARM)**: $\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta})$
- GD update for ARM: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n} \sum_{i=1}^{n} \nabla f_i(\boldsymbol{\theta}_t)$
- Gradient evaluated at same $\boldsymbol{\theta}_t$ for all tasks $\Longrightarrow$ **not adaptive**

MAML Algorithm: GD on MAML Loss

- **Average Risk Minimization (ARM)**: $\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta})$
- GD update for ARM: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n} \sum_{i=1}^{n} \nabla f_i(\boldsymbol{\theta}_t)$
- Gradient evaluated at same $\boldsymbol{\theta}_t$ for all tasks $\implies$ **not adaptive**

- **Model-Agnostic Meta-Learning (MAML)**:

$$\min_{\boldsymbol{\theta}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{\theta} - \alpha \nabla f_i(\boldsymbol{\theta}))$$

- GD update on MAML loss can be implemented as follows

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n} \sum_{i=1}^{n} (\boldsymbol{I} - \alpha \nabla^2 f_i(\boldsymbol{\theta}_t)) \nabla f_i(\boldsymbol{\theta}_t - \alpha \nabla f_i(\boldsymbol{\theta}_t))$$

MAML Algorithm: GD on MAML Loss

- **Average Risk Minimization (ARM)**: $\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta})$
- GD update for ARM: $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n} \nabla f_i(\boldsymbol{\theta}_t)$
- Gradient evaluated at same $\boldsymbol{\theta}_t$ for all tasks $\implies$ **not adaptive**

- **Model-Agnostic Meta-Learning (MAML)**:

$$\min_{\boldsymbol{\theta}} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{\theta} - \alpha\nabla f_i(\boldsymbol{\theta}))$$

- GD update on MAML loss can be implemented as follows

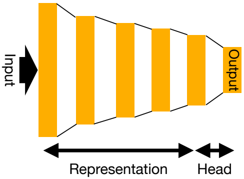
$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n}(\boldsymbol{I} - \alpha\nabla^2 f_i(\boldsymbol{\theta}_t))\nabla f_i(\boldsymbol{\theta}_t - \alpha\nabla f_i(\boldsymbol{\theta}_t))$$

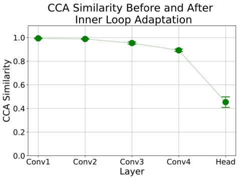
  which can be implemented via inner and outer loops

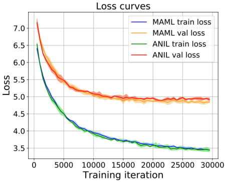
  - **Inner loop**: Compute $\boldsymbol{\theta}_{t,i} = \boldsymbol{\theta}_t - \alpha\nabla f_i(\boldsymbol{\theta}_t)$ for $i = 1, \ldots, n$
  - **Outer loop**: Compute $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \frac{\beta}{n}\sum_{i=1}^{n}(\boldsymbol{I} - \alpha\nabla^2 f_i(\boldsymbol{\theta}_t))\nabla f_i(\boldsymbol{\theta}_{t,i})$
- $\boldsymbol{\theta}_{t,i}$ adapted to each task $\implies$ **adaptive**

Empirical observations of MAML

- Original motivation: MAML learns models that **quickly adapt to new tasks** [FAL17, AES19]
- New empirical evidence suggests: MAML learns a good representation **shared across tasks** [RRBV20]
  - Even though it is not designed for representation learning!







- Can we formally prove this conjecture?

Meta-learning from linear regression tasks

Setting from multi-task learning and **linear representation learning**:

- Each task i is linear regression with ground truth parameter $\boldsymbol{\theta}_{*,i} \in \mathbb{R}^d$:

$$y_i \sim \boldsymbol{\theta}_{*,i}^\top \boldsymbol{x}_i + z_i \ ,$$

  $\boldsymbol{x}_i$ is a random input vector and $z_i \in \mathbb{R}$ is random zero-mean noise.

- Solving each task individually requires $\Omega(d)$ samples per task.

Meta-learning from linear regression tasks

Setting from multi-task learning and **linear representation learning**:

- Each task i is linear regression with ground truth parameter $\boldsymbol{\theta}_{*,i} \in \mathbb{R}^d$:

$$y_i \sim \boldsymbol{\theta}_{*,i}^\top \boldsymbol{x}_i + z_i \ ,$$

  $\boldsymbol{x}_i$ is a random input vector and $z_i \in \mathbb{R}$ is random zero-mean noise.

- Solving each task individually requires $\Omega(d)$ samples per task.

Questions in representation-based meta-learning

When does solving other tasks help you solve a new task?
What notion of similarities make meta-learning efficient for linear tasks?

Meta-learning from linear regression tasks

Setting from multi-task learning and **linear representation learning**:

- Each task i is linear regression with ground truth parameter $\boldsymbol{\theta}_{*,i} \in \mathbb{R}^d$:

$$y_i \sim \boldsymbol{\theta}_{*,i}^\top \boldsymbol{x}_i + z_i \ ,$$

  $\boldsymbol{x}_i$ is a random input vector and $z_i \in \mathbb{R}$ is random zero-mean noise.

- Solving each task individually requires $\Omega(d)$ samples per task.

- Now suppose the $\boldsymbol{\theta}_{*,i}$ lie in a shared k-dimensional subspace, $k \ll d$
- Let the columns of $\boldsymbol{B}_* \in \mathbb{R}^{d \times k}$ span this subspace, that is, for each task there exists a corresponding low-dimensional $\boldsymbol{w}_{*,i} \in \mathbb{R}^k$ such that

$$\boldsymbol{\theta}_{*,i} = \underbrace{\boldsymbol{B}_*}_{\text{Representation}} \underbrace{\boldsymbol{w}_{*,i}}_{\text{Head}}$$

- If we know $\mathrm{col}(\boldsymbol{B}_*)$, we can solve new tasks with only $O(k)$ samples

Meta-learning from linear regression tasks

Setting from multi-task learning and **linear representation learning**:

- Each task i is linear regression with ground truth parameter $\boldsymbol{\theta}_{*,i} \in \mathbb{R}^d$:

$$y_i \sim \boldsymbol{\theta}_{*,i}^\top \boldsymbol{x}_i + z_i \ ,$$

  $\boldsymbol{x}_i$ is a random input vector and $z_i \in \mathbb{R}$ is random zero-mean noise.

- Solving each task individually requires $\Omega(d)$ samples per task.

- Now suppose the $\boldsymbol{\theta}_{*,i}$ lie in a shared k-dimensional subspace, $k \ll d$
- Let the columns of $\boldsymbol{B}_* \in \mathbb{R}^{d \times k}$ span this subspace, that is, for each task there exists a corresponding low-dimensional $\boldsymbol{w}_{*,i} \in \mathbb{R}^k$ such that

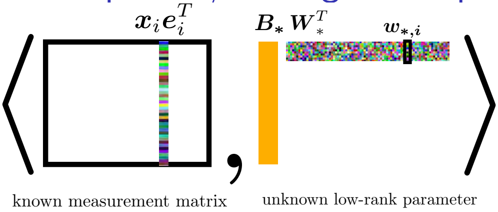
$$\boldsymbol{\theta}_{*,i} = \underbrace{\boldsymbol{B}_*}_{\text{Representation}} \underbrace{\boldsymbol{w}_{*,i}}_{\text{Head}}$$

- If we know $\mathrm{col}(\boldsymbol{B}_*)$, we can solve new tasks with only $O(k)$ samples

Does GD on ARM learn $\boldsymbol{B}_*$? Does GD on MAML learn $\boldsymbol{B}_*$?

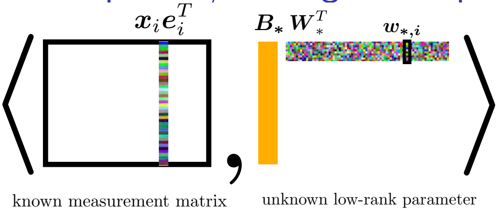
Prior work use matrix completion/sensing techniques

$$y_i \simeq \boldsymbol{x}_i^T \boldsymbol{B}^* \boldsymbol{w}_{*,i} =$$

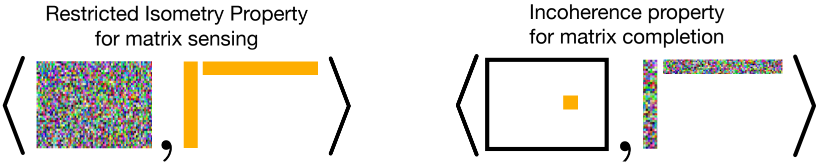


Prior work use matrix completion/sensing techniques

$$y_i \simeq x_i^T B^* w_{*,i} =$$



- [TJJ21,CHMS21,TJNO21] show that although the standard assumptions are not satisfied, i.e.,



  sample efficient learning is possible as long as we have **task diversity** = small condition number of W_*.
- Can a **single parameter** algorithm, such as ARM and MAML, learn the ground truth (linear) representation?

MAML for linear representation learning

- Loss function for task i at round t:

$$f_i(\boldsymbol{B}, \boldsymbol{w}) := \tfrac{1}{2}\mathbb{E}_{\boldsymbol{x}_i, y_i}[(\langle \boldsymbol{B}\boldsymbol{w}, \boldsymbol{x}_i\rangle - y_i)^2]$$

- MAML is called a **gradient-based meta-learning** algorithm (as opposed to representation-based meta-learning)

Algorithm (MAML)

- **(Outer loop)** For $t = 1, \ldots, T$:
  - Sample n linear tasks
  - **(Inner loop)** For each task $i \in \{1, \ldots, n\}$:
    - Adapt: $\begin{bmatrix} \boldsymbol{w}_{t,i} \\ \boldsymbol{B}_{t,i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_t \\ \boldsymbol{B}_t \end{bmatrix} - \alpha \begin{bmatrix} \nabla_{\boldsymbol{w}} f_i(\boldsymbol{B}_t, \boldsymbol{w}_t) \\ \nabla_{\boldsymbol{B}} f_i(\boldsymbol{B}_t, \boldsymbol{w}_t) \end{bmatrix}$
  - $\begin{bmatrix} \boldsymbol{w}_{t+1} \\ \boldsymbol{B}_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_t \\ \boldsymbol{B}_t \end{bmatrix} - \frac{\beta}{n} \sum_{i=1}^{n} (\boldsymbol{I} - \alpha \nabla^2_{\boldsymbol{w}, \bar{\boldsymbol{B}}} f_i(\boldsymbol{B}_t, \boldsymbol{w}_t)) \begin{bmatrix} \nabla_{\boldsymbol{w}} f_i(\boldsymbol{B}_{t,i}, \boldsymbol{w}_{t,i}) \\ \nabla_{\boldsymbol{B}} f_i(\boldsymbol{B}_{t,i}, \boldsymbol{w}_{t,i}) \end{bmatrix}$

MAML vs. ANIL (Almost No Inner Loop)

- Loss function for task i at round t:

$$f_i(\boldsymbol{B}, \boldsymbol{w}) := \frac{1}{2}\mathbb{E}_{\boldsymbol{x}_i, y_i}[(\langle \boldsymbol{B}\boldsymbol{w}, \boldsymbol{x}_i\rangle - y_i)^2]$$

- MAML is a **gradient-based meta-learning** algorithm
- ANIL is a **representation-based meta-learning** algorithm

Algorithm (MAML and ANIL)

- **(Outer loop)** For $t = 1, \ldots, T$:
  - Sample n linear tasks
  - **(Inner loop)** For each task $i \in \{1, \ldots, n\}$:
    - MAML adapts both: $\begin{bmatrix} \boldsymbol{w}_{t,i} \\ \boldsymbol{B}_{t,i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_t \\ \boldsymbol{B}_t \end{bmatrix} - \alpha \begin{bmatrix} \nabla_{\boldsymbol{w}} f_i(\boldsymbol{B}_t, \boldsymbol{w}_t) \\ \nabla_{\boldsymbol{B}} f_i(\boldsymbol{B}_t, \boldsymbol{w}_t) \end{bmatrix}$
    - ANIL adapts only head: $\begin{bmatrix} \boldsymbol{w}_{t,i} \\ \boldsymbol{B}_{t,i} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_t \\ \boldsymbol{B}_t \end{bmatrix} - \alpha \begin{bmatrix} \nabla_{\boldsymbol{w}} f_i(\boldsymbol{B}_t, \boldsymbol{w}_t) \\ 0 \end{bmatrix}$
  - $\begin{bmatrix} \boldsymbol{w}_{t+1} \\ \boldsymbol{B}_{t+1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}_t \\ \boldsymbol{B}_t \end{bmatrix} - \frac{\beta}{n}\sum_{i=1}^{n} \boldsymbol{H}_{t,i,\mathrm{Alg}}(\boldsymbol{B}_t, \boldsymbol{w}_t) \begin{bmatrix} \nabla_{\boldsymbol{w}} f_i(\boldsymbol{B}_{t,i}, \boldsymbol{w}_{t,i}) \\ \nabla_{\boldsymbol{B}} f_i(\boldsymbol{B}_{t,i}, \boldsymbol{w}_{t,i}) \end{bmatrix}$
    where $\boldsymbol{H}_{t,i,\mathrm{Alg}}(\cdot)$ is a Hessian that differs between MAML and ANIL

MAML: Evidence of representation learning

- We consider four meta-learning algorithms:
  - ANIL (representation-based meta-learning),
  - MAML (gradient-based meta-learning),
  - their first-order approximations (FO-MAML and FO-ANIL).

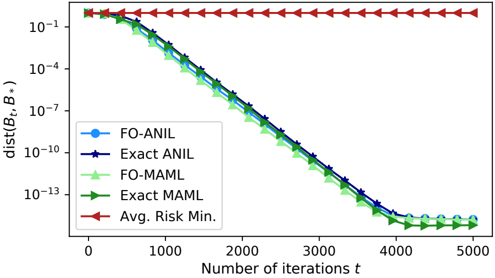


Figure: MAML learns the true (linear) representation, $\mathrm{col}(\boldsymbol{B}_*)$, while ARM does not.

- We only evaluate the training phase, assuming that failure to learn the representation leads to failure in few-shot fine-tuning.

Main Results (informal)

- Under the linear representation learning setting

Informal theorem

- *Under standard assumptions, MAML, ANIL and their first-order analogues recover* $\mathrm{col}(\boldsymbol{B}_*)$ *exponentially fast when run on the task population losses.*
- *ANIL and FO-ANIL require* $m = \Omega((\frac{d}{n} + 1)k^3) \ll d$ *samples per task to recover* $\mathrm{col}(\boldsymbol{B}_*)$*.*
- *The key is that MAML and ANIL's adaptation of the head harnesses* **task diversity** *to improve the representation in all directions.*

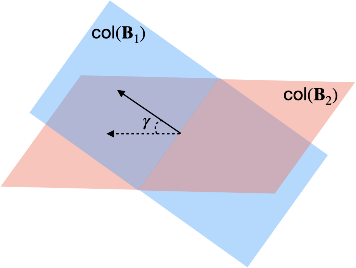
- First results showing that MAML and ANIL provably learn effective representations!

Informal negative result from [CHMS22]

There exist problems for which ARM fails to learn $\mathrm{col}(\boldsymbol{B}_*)$*.*

Principal Angle Distance

- We use the **principal angle distance** to measure the distance between representations.



- Formally,

$$\mathrm{dist}(\boldsymbol{B}_1, \boldsymbol{B}_2) := \|\hat{\boldsymbol{B}}_{1,\perp}^\top \hat{\boldsymbol{B}}_2\|_2,$$

where $\hat{\boldsymbol{B}}_{1,\perp}$ and $\hat{\boldsymbol{B}}_2$ are orthonormal matrices s.t. $\mathrm{col}(\hat{\boldsymbol{B}}_{1,\perp}) = \mathrm{col}(\boldsymbol{B}_1)^\perp$ and $\mathrm{col}(\hat{\boldsymbol{B}}_2) = \mathrm{col}(\boldsymbol{B}_2)$.

Average Risk Minimization (ARM) fails to recover $\mathrm{col}(B_*)$

- Let's focus on the population case to simplify the expressions

$$\text{ARM:} \quad \min_{\boldsymbol{B},\boldsymbol{w}} \frac{1}{n}\sum_{i=1}^{n} f_i(\boldsymbol{B},\boldsymbol{w})$$

- Dynamics of GD on ARM:

$$\boldsymbol{B}_{t+1} \leftarrow \boldsymbol{B}_t \Big(\underbrace{\boldsymbol{I}_k - \beta \boldsymbol{w}_t \boldsymbol{w}_t^\top}_{\text{prior weight}} \Big) + \beta\, \boldsymbol{B}_* \underbrace{\Big(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_{*,i} \Big) \boldsymbol{w}_t^\top}_{\text{signal weight}}$$

- Two issues:
  1. Prior weight only reduces $\boldsymbol{B}_t$ in one direction $\Rightarrow$ slow in forgetting $\boldsymbol{B}_0$
  2. Column space of signal weight is rank one and does not change over time $\Rightarrow$ we only improve in one fixed direction of the true signal $\boldsymbol{B}_*$.

Average Risk Minimization (ARM) fails to recover $\mathrm{col}(B_*)$

- Let's focus on the population case to simplify the expressions

$$\text{ARM:} \quad \min_{\boldsymbol{B},\boldsymbol{w}} \frac{1}{n} \sum_{i=1}^{n} f_i(\boldsymbol{B}, \boldsymbol{w})$$

- Dynamics of GD on ARM:

$$\boldsymbol{B}_{t+1} \leftarrow \boldsymbol{B}_t \Big(\underbrace{\boldsymbol{I}_k - \beta \boldsymbol{w}_t \boldsymbol{w}_t^\top}_{\text{prior weight}} \Big) + \beta\, \boldsymbol{B}_* \underbrace{\Big(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{w}_{*,i} \Big) \boldsymbol{w}_t^\top}_{\text{signal weight}}$$

Formal Theorem from [CHMS22]

For any $\delta \in (0., 0.5], \alpha, T, \{w_{*,i}\}$ and full rank $\boldsymbol{B}_0$, there exists a $\boldsymbol{B}_*$ whose column space is δ-close to $col(\boldsymbol{B}_0)$, i.e., $\mathrm{dist}(\boldsymbol{B}_0, \boldsymbol{B}_*) = \delta$, while its distance from the representation learned by ARM is at least 0.7δ, i.e., $\mathrm{dist}(\boldsymbol{B}_T^{ARM}, \boldsymbol{B}_*) > 0.7\delta$.

Dynamics of ANIL, MAML, and FO variations

- For FO-ANIL under population loss, we have

$$\boldsymbol{B}_{t+1} \leftarrow \boldsymbol{B}_t \underbrace{\left(\boldsymbol{I}_k - \frac{\beta}{n} \sum_{i=1}^{n} \boldsymbol{w}_{t,i} \boldsymbol{w}_{t,i}^\top \right)}_{\text{prior weight}} + \beta \, \boldsymbol{B}_* \underbrace{\left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{w}_{*,i} \boldsymbol{w}_{t,i}^\top \right)}_{\text{signal weight}}$$

- Suppose
  - $\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{w}_{*,i} \boldsymbol{w}_{*,i}^\top$ has small condition number (**task diversity**), and
  - $\boldsymbol{w}_{t,i}$'s are close to $\boldsymbol{w}_{*,i}$'s (**head adaptation**), then:

Key observation

Prior weight reduces energy from $\boldsymbol{B}_t$, and signal weight boosts energy from $\boldsymbol{B}_*$ in all directions.

$\Longrightarrow$ **Head adaptation** and **task diversity** are critical!

Challenges in proving representation learning

- Need to show **head adaptation**, that the $\boldsymbol{w}_{t,i}$'s are close to the true heads $\boldsymbol{w}_{*,i}$'s
- From the inner loop of ANIL/MAML:

$$\boldsymbol{w}_{t,i} \leftarrow \overbrace{\underbrace{(\boldsymbol{I}_k - \alpha \boldsymbol{B}_t^\top \boldsymbol{B}_t)}_{\text{}} }^{\text{prior weight}} \underbrace{}_{\text{shared for all tasks}} \boldsymbol{w}_t + \overbrace{\underbrace{\alpha \boldsymbol{B}_t^\top \boldsymbol{B}_* \, \boldsymbol{w}_{*,i}}_{\text{unique for each task } i}}^{\text{signal weight}}$$

- In order to show the unique part dominates, we must show three things hold for all t:
  1. $\|\boldsymbol{I}_k - \alpha \boldsymbol{B}_t^\top \boldsymbol{B}_t\|_2$ is small
  2. $\|\boldsymbol{w}_t\|_2$ is small
  3. $\sigma_{\min}(\boldsymbol{B}_t^\top \boldsymbol{B}_*)$ is lower bounded
- Difficult because the algorithms lack explicit regularization and a normalization step.
- Leads to an intricate 6-way induction....

Population case result

Main Theorem [Collins-Mokhtari-O-Shakkottai, ICML 2022]

*Suppose there are $m = \infty$ samples/task, the ground-truth heads satisfy $\mu_*^2 \boldsymbol{I}_k \preceq \frac{1}{n}\sum_{i=1}^n \boldsymbol{w}_{*,i}\boldsymbol{w}_{*,i}^\top \preceq L_*^2 \boldsymbol{I}_k$ (**Task Diversity**), and the step sizes α, β are sufficiently small. Then after T iterations, ANIL, FO-ANIL, MAML, and FO-MAML learn a representation $\boldsymbol{B}_T$ that satisfies:*

$$\mathrm{dist}(\boldsymbol{B}_T, \boldsymbol{B}_*) \leq \left(1 - \Omega(\beta\alpha\mu_*^2)\right)^{T-1}$$

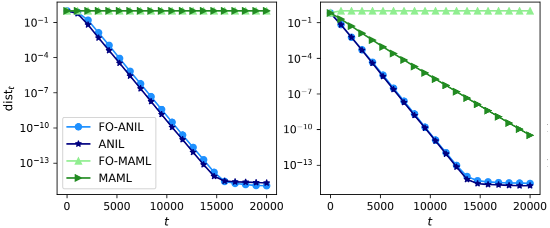
as long as:

- *ANIL, FO-ANIL:* $\mathrm{dist}(\boldsymbol{B}_0, \boldsymbol{B}_*) \leq c$ *for a constant* c.
- *MAML:* $\mathrm{dist}(\boldsymbol{B}_0, \boldsymbol{B}_*) = O((L_*/\mu_*)^{-0.75})$.
- *FO-MAML:* $\mathrm{dist}(\boldsymbol{B}_0, \boldsymbol{B}_*) = O((L_*/\mu_*)^{-1})$ *and* $\|\frac{1}{n}\sum_{i=1}^n \boldsymbol{w}_{*,t,i}\|_2 = O((L_*/\mu_*)^{-1.5})$.

- We also show finite-sample results in the paper.

MAML vs. ANIL

- Recall that our result requires
  1. stronger initialization for MAML and FO-MAML than for ANIL and FO-ANIL, and
  2. for FO-MAML, $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_{*,i} \approx 0$.
- We empirically show these conditions are tight:



- (Left) Random initialization leads MAML and FO-MAML to fail
- (Right) Even with good initialization, $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_{*,i}$ far from zero leads FO-MAML to fail
  $\implies$ MAML/FO-MAML's updates $\boldsymbol{B}_t$ in the inner loop, which can inhibit representation learning.

Proof sketch - FO-ANIL (1/4)

$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t \Big(\underbrace{\boldsymbol{I}_k - \beta \boldsymbol{\Psi}_t}_{\text{prior weight}} \Big) + \beta\, \boldsymbol{B}_* \left(\frac{1}{n} \sum_{i=1}^{n} \boldsymbol{w}_{*,i} \boldsymbol{w}_{t,i}^\top \right),$$

$$\boldsymbol{w}_{t,i} = \underbrace{\boldsymbol{\Delta}_t}_{\text{prior weight}} \boldsymbol{w}_t + \alpha \boldsymbol{B}_t^\top \boldsymbol{B}_* \boldsymbol{w}_{*,i}$$

- Inductive hypotheses:

Bounded Head Weight

$A_1(t) := \{\|\boldsymbol{w}_t\|_2 = O(\sqrt{\alpha})\}$

Small Head Prior Weight

$A_2(t) := \{\|\boldsymbol{\Delta}_t\|_2 = \rho \|\boldsymbol{\Delta}_{t-1}\|_2 + O(\beta^2 \alpha^2 \operatorname{dist}_{t-1}^2)\}$

$A_3(t) := \{\|\boldsymbol{\Delta}_t\|_2 = O(1)\}$

Small Representation Prior Weight

$A_4(t) := \{\kappa(\boldsymbol{\Psi}_t) = O(1)\}$

Progress

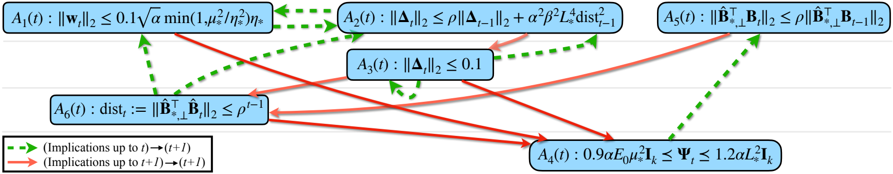
$A_5(t) := \{\|\boldsymbol{B}_{*,\perp}^\top \boldsymbol{B}_t\|_2 = \rho \|\boldsymbol{B}_{*,\perp}^\top \boldsymbol{B}_{t-1}\|_2\}$

$A_6(t) := \{\operatorname{dist}_t \leq \rho^{t-1}\}$

where, $\boldsymbol{\Delta}_t := \boldsymbol{I}_k - \alpha \boldsymbol{B}_t^\top \boldsymbol{B}_t$, $\boldsymbol{\Psi}_t := \frac{1}{n}\sum_{i=1}^n \boldsymbol{w}_{t,i}\boldsymbol{w}_{t,i}^\top$, and $\rho := 1 - \Omega(\beta\alpha)$

Proof sketch - FO-ANIL (2/4)

- Inductive logic:



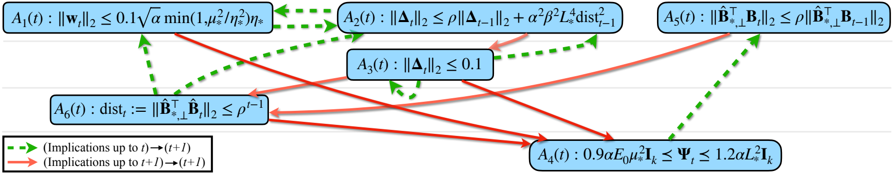
$$\boldsymbol{B}_{t+1} = \boldsymbol{B}_t \underbrace{\left(\boldsymbol{I}_k - \beta \boldsymbol{\Psi}_t\right)}_{\text{prior weight}} + \beta\, \boldsymbol{B}_* \underbrace{\left(\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_{*,i} \boldsymbol{w}_{t,i}^\top\right)}_{\text{signal weight}}, \quad \boldsymbol{\Delta}_t := \boldsymbol{I}_k - \alpha \boldsymbol{B}_t^\top \boldsymbol{B}_t$$

Notable implications (1/3):

- $A_4(t) \implies A_5(t+1) \overset{A_3(t+1)}{\implies} A_6(t+1)$
  - well-conditioned $\boldsymbol{\Psi}_t$ implies small prior weight and hence per-step improvement
  - per-step improvements imply geometric convergence

Proof sketch - FO-ANIL (3/4)

- Inductive logic:



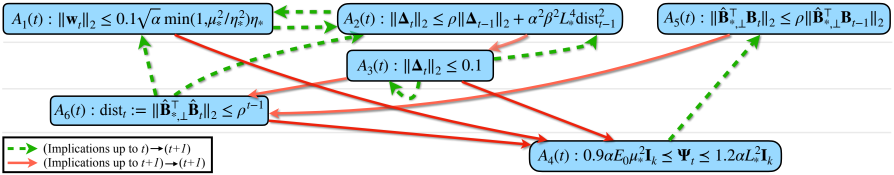
$$\boldsymbol{w}_{t,i} = \underbrace{\boldsymbol{\Delta}_t \boldsymbol{w}_t}_{\text{shared for all } i} + \underbrace{\alpha \boldsymbol{B}_t^\top \boldsymbol{B}_* \boldsymbol{w}_{*,i}}_{\text{unique for each } i}, \qquad \boldsymbol{\Psi}_t := \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}_{t,i}\boldsymbol{w}_{t,i}^\top$$

Notable implications (2/3):

- $A_1(t+1), A_3(t+1), A_6(t+1) \implies A_4(t+1)$
  - Small $\|\boldsymbol{\Delta}_t\|_2$, $\|\boldsymbol{w}_t\|_2$, and $\text{dist}_t(\boldsymbol{B}_t, \boldsymbol{B}_*)$ implies adapted heads are diverse

Proof sketch - FO-ANIL (4/4)

- Inductive logic:



Notable implications (3/3):

- $A_2(t) + A_6(t) \implies A_1(t+1)$
  - This is tricky as it relies on showing that $\|\mathbf{\Delta}_t\|_2$ and dist_t are summable to show that $\|\boldsymbol{w}_t\|$ is bounded

Discussion

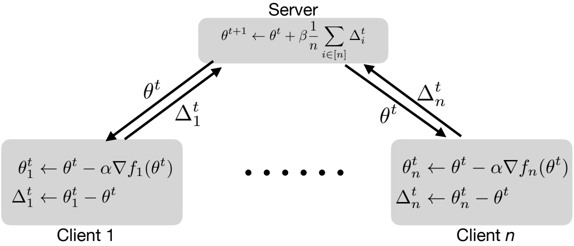
- We have obtained the first results showing that ANIL and MAML learn effective representations.*
- Inner loop **adaptation of the head** is key to MAML and ANIL's ability to learn representations.
- Inner loop adaptation of the representation restricts representation learning for MAML.

*L. Collins, A. Mokhtari, S. Oh, S. Shakkottai. MAML and ANIL Provably Learn Representations, ICML 2022

Connections to federated learning

Connections to federated learning†

- **Distributed Stochastic Gradient Descent (D-SGD)**



- Federated implementation of Average Risk Minimization (ARM):

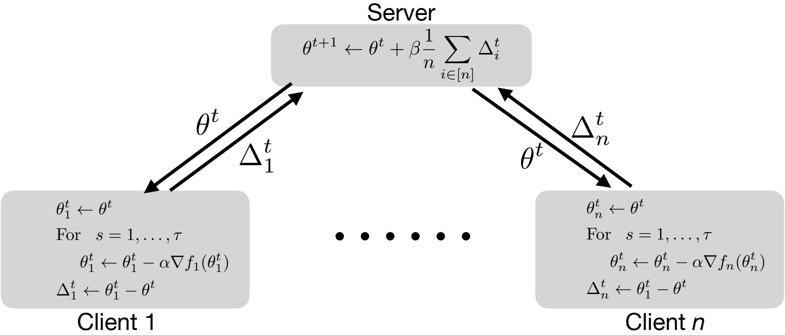
$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \alpha\beta \frac{1}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} f_i(\boldsymbol{\theta}^t)$$

- Major difference: Data never leaves the client device for privacy

†L. Collins, A. Mokhtari, H. Hassani, S. Shakkottai. "FedAvg with Fine-tuning: Local Updates Lead to Representation Learning", NeurIPS 2022

Connections to federated learning

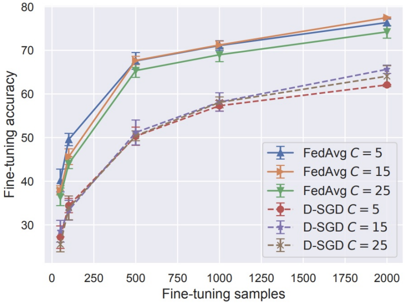
- **Federated Average (FedAvg)**[‡] performs multiple local updates similar to MAML

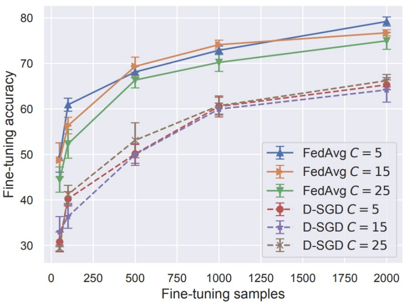


- Original motivation: communication rounds $\ll$ number of gradient updates
- New observation: effective representation learner

[‡]introduced in [MMRHA17]

Local updates help in personalization [CHMS22]

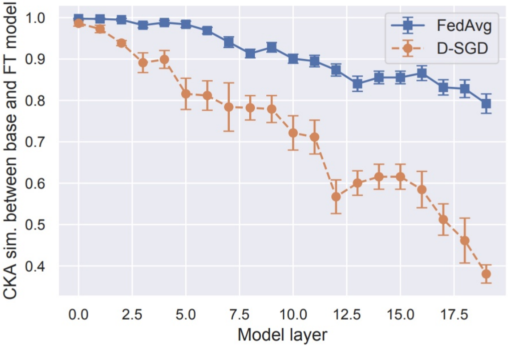




- Left plot: Models trained on 80 classes from CIFAR-100 (with C classes/client) and fine-tuned on new clients from 20 new classes from CIFAR-100
- Right plot: Models trained on CIFAR-100 (with C classes/client) and fine-tuned on new clients from CIFAR-10
- $T\tau = 125000$ for both.
  (FedAvg $\tau = 50$, $T = 2500$, DSGD $\tau = 1$, $T = 125000$)

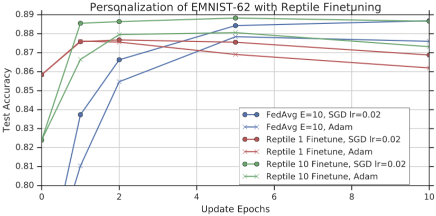
Representation learned by FedAvg changes less in fine-tuning [CHMS22]

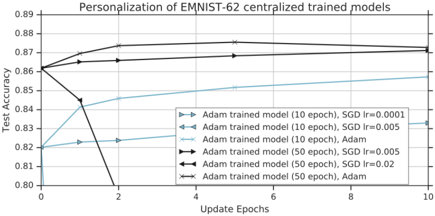
- The early layers of FedAvg's pre-trained model (corresponding to the representation) change much less than those of D-SGD



- Local updates enable learning the common representation across the clients.

Local updates help in personalization [JKRK19]





- Personalization in FL: Federated trained model is further fine-tuned on client data and evaluated on client data
- FedAvg (left) achieves higher personalization accuracy compared to D-SGD (right)

FedAvg provably learns representations

Theorem (informal) [CHMS22]

Under the linear representation learning setting, if the number of local updates is more than one, i.e., $\tau \geq 2$, FedAvg recovers $col(\boldsymbol{B}^*)$ exponentially fast when run on the task population losses.

- The key insight is that FedAvg local updates harness **task diversity** to improve the representation in all directions.

$$\mathbf{B}_{t+1} \approx \mathbf{B}_t \underbrace{\left(\boldsymbol{I}_k - \frac{\alpha}{n} \sum_{i=1}^{n} \sum_{s=0}^{\tau-1} \mathbf{w}_{t,i,s} \mathbf{w}_{t,i,s}^\top \right)}_{\text{prior weight}} + \mathbf{B}_* \underbrace{\left(\frac{\alpha}{n} \sum_{i=1}^{n} \sum_{s=0}^{\tau-1} \mathbf{w}_{*,i} \mathbf{w}_{t,i,s}^\top \right)}_{\text{signal weight}}$$

- Prior weight reduces energy from $\boldsymbol{B}_t$, and signal weight boosts energy from $\boldsymbol{B}_*$ in all directions
- Local updates and task diversity are critical.

Discussion

- We have obtained the first results showing that ANIL and MAML learn effective representations.[§]
- Inner loop **adaptation of the head** is key to MAML and ANIL's ability to learn representations.
- Inner loop adaptation of the representation restricts representation learning for MAML.
- Follow-up work by [CMHS22][¶] shows that Federated Averaging also learns effective representations.

[§] L. Collins, A. Mokhtari, S. Oh, S. Shakkottai. MAML and ANIL Provably Learn Representations, ICML 2022

[¶] L. Collins, A. Mokhtari, H. Hassani, S. Shakkottai. "FedAvg with Fine-tuning: Local Updates Lead to Representation Learning", NeurIPS 2022

References

[FAL17] Chelsea Finn, Pieter Abbeel, Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Neural Networks, *International Conference on Machine Learning*, 2017.

[AES19] Antreas Antoniou, Harrison Edwards, Amos Storkey. How to Train Your MAML, *International Conference on Learning Representations*, 2019.

[RRBV19] Aniruddh Raghu, Maithra Raghu, Samy Bengio, Oriol Vinyals. Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML, *International Conference on Learning Representations*, 2020.

[HRJ21] Mike Huisman, Jan N. van Rijn, Aske Plaat. A Survey of deep Meta-Learning, *Artificial Intelligence Review* Volume 54, pages 4483–4541, 2021.

[TJNO21] Kiran K. Thekumparampil, Prateek Jain, Praneeth Netrapalli, Sewoong Oh. Statistically and Computationally Efficient Linear Meta-representation Learning, *Advances in Neural Information Processing Systems*, 2021.

[TJJ21] Nilesh Tripuraneni, Chi Jin, Michael I Jordan, Provable Meta-Learning of Linear Representations, *International Conference on Learning Representations*, 2021

References

[CHMS21] Liam Collins, Hamed Hassani, Aryan Mokhtari, Sanjay Shakkottai, Exploiting Shared Representations for Personalized Federated Learning, *International Conference on Learning Representations*, 2021

[CHMS22] Liam Collins, Hamed Hassani, Aryan Mokhtari, Sanjay Shakkottai, FedAvg with Fine Tuning: Local Updates Lead to Representation Learning, *Advances in Neural Information Processing Systems*, 2022.

[MMRHA17] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Agüera y Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, *AISTATS*, 2017.

[JKRK19] Yihan Jiang, Jakub Konecny, Keith Rush, Sreeram Kannan, Improving federated learning personalization via model agnostic meta learning, arXiv:1909.12488, 2019.