UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

ANTS LAB

No Fear of Heterogeneity: Classifier Calibration for Federated Learning with Non-IID Data

Mi Luo, et al

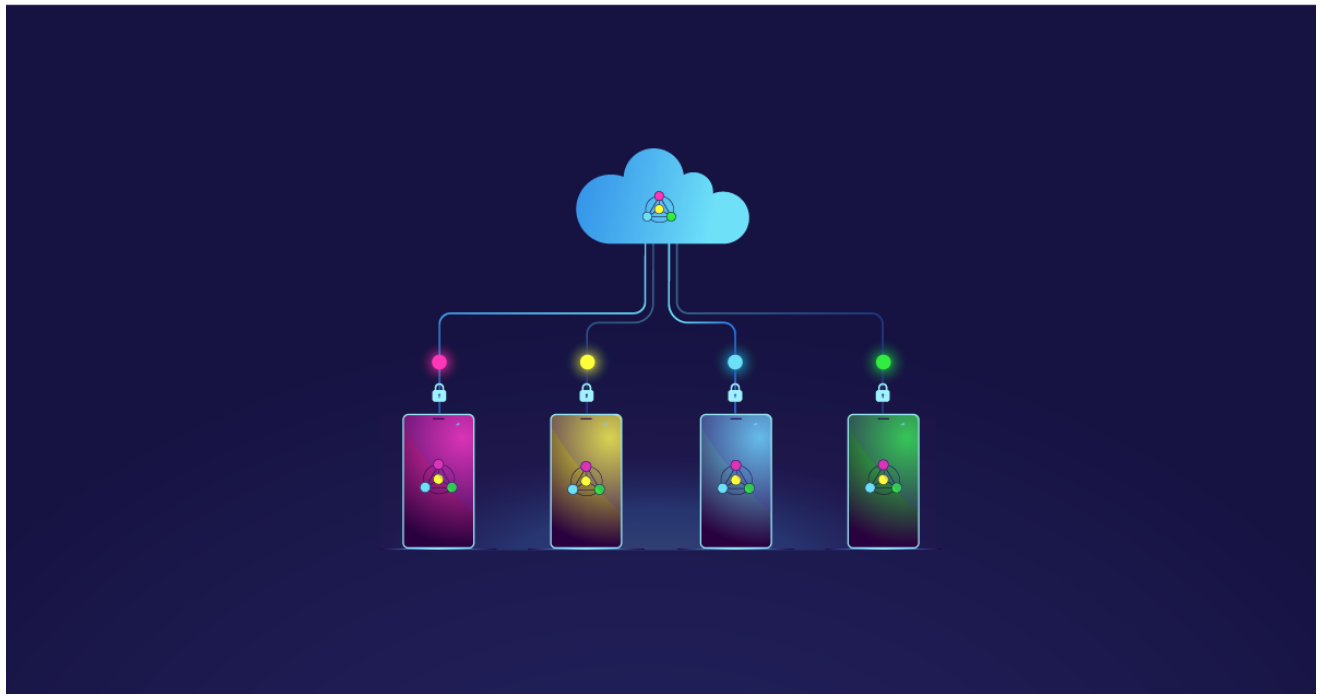
Published in NIPS' 21

Presenter: Dian Shi

Outline

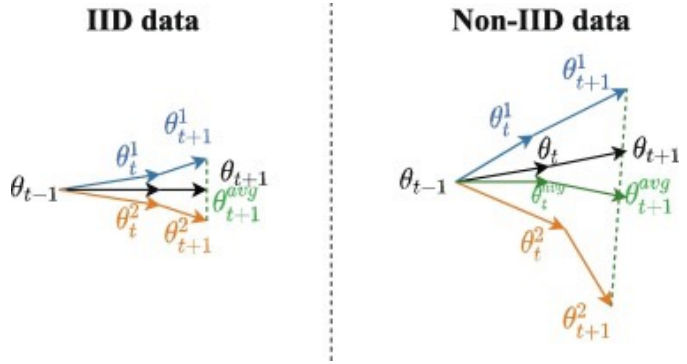
- Backgrounds of the FL with non-IID Data
- Heterogeneity in Federated Learning: The Devil Is in Classifier
- Classifier Calibration with Virtual Representations
- Performance Evaluation
- Conclusion

Federated Learning



- **Challenge:**

  Data heterogeneity (non-IID)



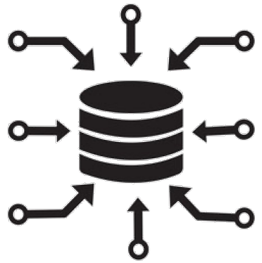
unstable and slow convergence

suboptimal or even detrimental model performance

Federated Learning with non-IID data

Four categories:

- Client drift mitigation
  modifies the local objectives; consistent with the global model



- Advanced aggregation scheme
  improves the model fusion mechanism at the server

- Data sharing
  introduces public datasets to help construct a balanced distribution

- Personalized federated learning
  train personalized models for individual clients

Federated Learning with non-IID data

Existing algorithms are still unable to achieve good performance, no better than FedAvg

- Experimental investigation: perform the layer-wise similarity measurement; the classifier has the lowest feature similarity
- Empirical trials: the classifier tends to be biased to certain classes; debias the classifier via regularizing/calibrating the classifier
- Novel approach: Classifier Calibration with Virtual Representations

Outline

- Backgrounds of the FL with non-IID Data
- Heterogeneity in Federated Learning: The Devil Is in Classifier
- Classifier Calibration with Virtual Representations
- Performance Evaluation
- Conclusion

Problem Setup

Image classification in FL K clients C classes

a sample $(\boldsymbol{x}, y)$

- the feature extractor $f_{\boldsymbol{\theta}} : \mathcal{X} \to \mathcal{Z}$
  - Output: feature vector $\boldsymbol{z} = f_{\boldsymbol{\theta}}(\boldsymbol{x}) \in \mathbb{R}^d$
- the classifier $g_{\boldsymbol{\varphi}} : \mathcal{Z} \to \mathbb{R}^C$
  - Output: probability distribution $g_{\boldsymbol{\varphi}}(\boldsymbol{z})$

Each client k:

$$\min_{\boldsymbol{w}_k^{(t)}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}^k}[\mathcal{L}(\boldsymbol{w}_k^{(t)}; \boldsymbol{w}^{(t-1)}, \boldsymbol{x}, y)]$$

Server:

$$\boldsymbol{w}^{(t)} = \sum_{k\in U^{(t)}} p_k \boldsymbol{w}_k^{(t)}, \text{ where } p_k = \frac{|\mathcal{D}^k|}{\sum_{k'\in U^{(t)}} |\mathcal{D}^{k'}|}$$

A Closer Look at Classification Model: Classifier Bias

Perform an experimental study to understand how non-IID data affect the classification model

experimental setup:

CIFAR-10; CNN model with 7 layers

non-IID: Dirichlet distribution

Similarity measurement: leverage the Centered Kernel Alignment (CKA) to measure the similarity of the output features for each layer

similarity score between 0 (not similar at all) and 1 (identical)

measure similarities of higher dimension rather than the paired data points

FedAvg: 100 communication rounds; 10 local epochs

A Closer Look at Classification Model: Classifier Bias

Observations: similarity between output features

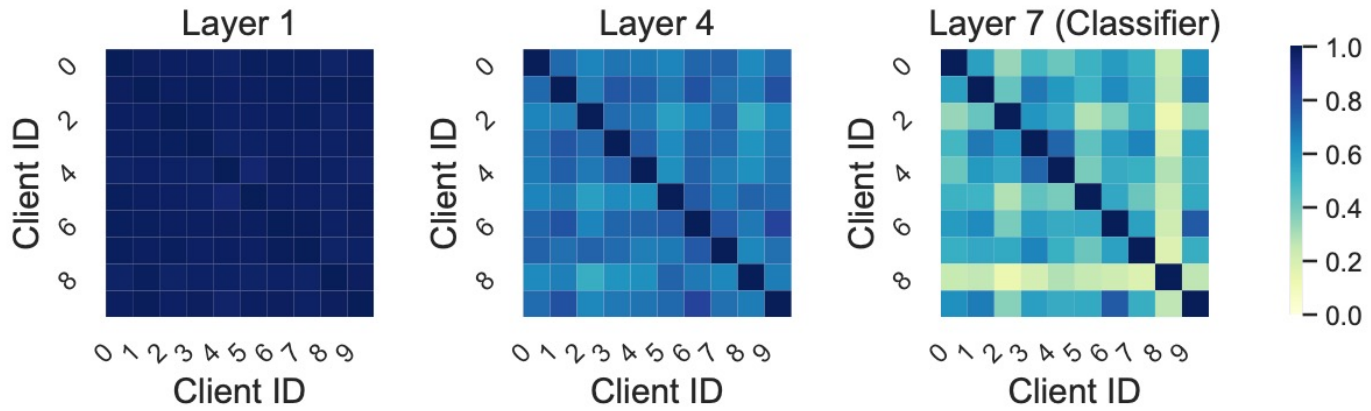


Figure 1: CKA similarities of three different layers of different 'client model-client model' pairs.

- Deeper layers have heavier heterogeneity across different clients

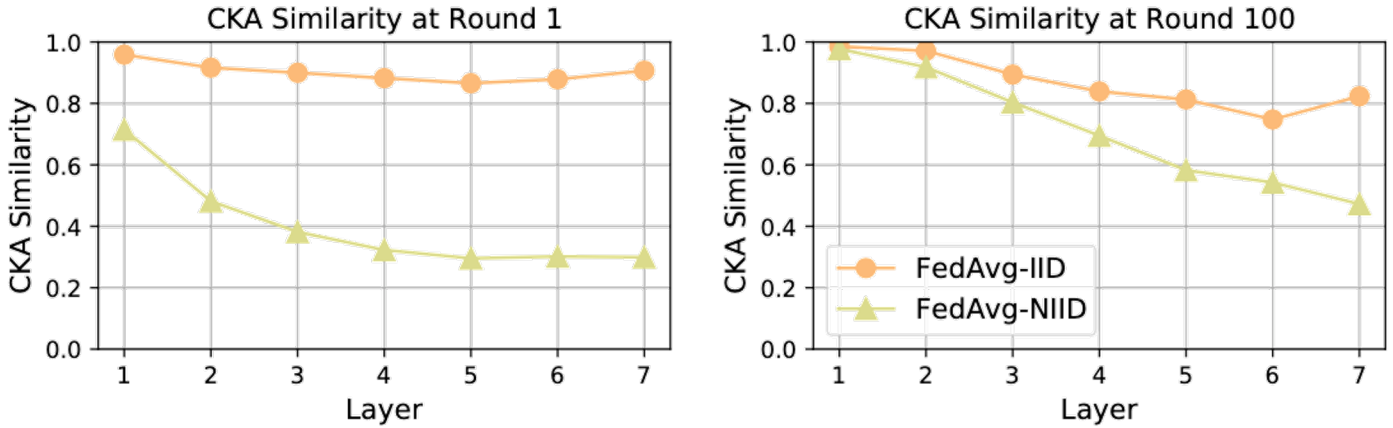


Figure 2: The means of the CKA similarities of different layers in different local models.

- Training with non-IID data have consistently lower feature similarity for all layers
- the classifier shows the lowest features similarities.

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

A Closer Look at Classification Model: Classifier Bias

Observations: L2 norm of local classifier weight

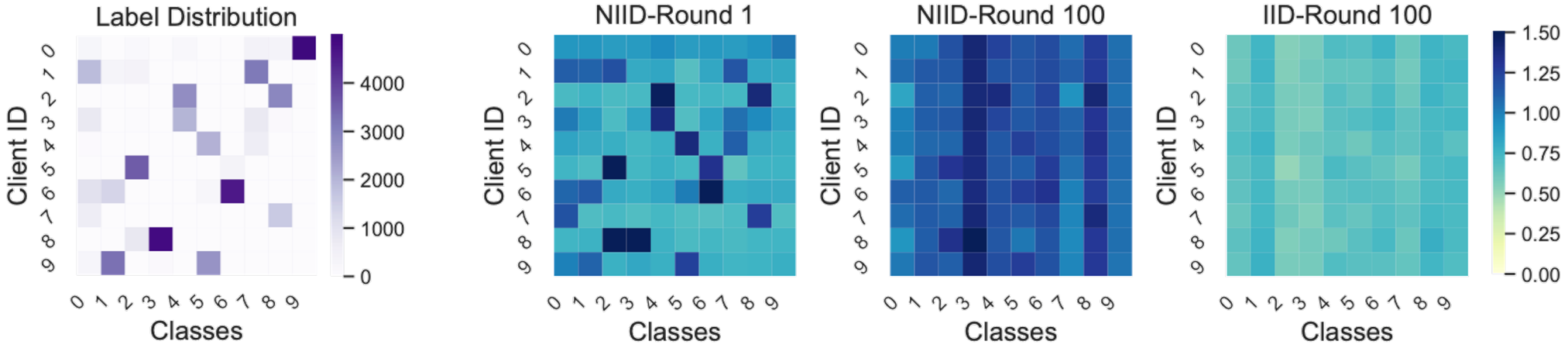


Figure 3: Label distribution of CIFAR-10 across clients (the first graph) and the classifier weight norm distribution across clients in different rounds and data partitions (the three graphs on the right).

- *at the initial stage*: the weight norms would be biased
- *at the end*: non-IID data have the heavier biased classifier than the IID data

hypothesis: Classifier can be easily biased to the heterogeneous local data, reflected by : low features similarity & biased weight norms.

Classifier Regularization and Calibration

Debiasing the classifier is promising to improve the classification performance

Debias the classifier:

Classifier Weight L2-normalization (clsnorm)

The classifier with weight φ, followed by normalization and softmax.

$$g_{\boldsymbol{\varphi}}(\boldsymbol{z})_i = \frac{e^{\boldsymbol{\varphi}_i^T \boldsymbol{z}/||\boldsymbol{\varphi}_i||}}{\sum_{i'=1}^{C} e^{\boldsymbol{\varphi}_{i'}^T \boldsymbol{z}/||\boldsymbol{\varphi}_{i'}||}}, \quad \forall i \in [C].$$

Classifier Quadratic Regularization (clsprox)

Adding a proximal term only to restrict the local classifier weights.

$$\mathcal{L}(\boldsymbol{w}_k^{(t)}; \boldsymbol{w}^{(t-1)}, \boldsymbol{x}, y) = \ell(g_{\boldsymbol{\varphi}_k^{(t)}}(f_{\boldsymbol{\theta}_k^{(t)}}(\boldsymbol{x})), y) + \frac{\mu}{2}||\boldsymbol{\varphi}_k^{(t)} - \boldsymbol{\varphi}^{(t-1)}||^2$$

Classifier Post-calibration with IID Samples

Fix the feature extractor and calibrate the classifier by IID samples.
(an experimental study, not piratical)

Classifier Regularization and Calibration

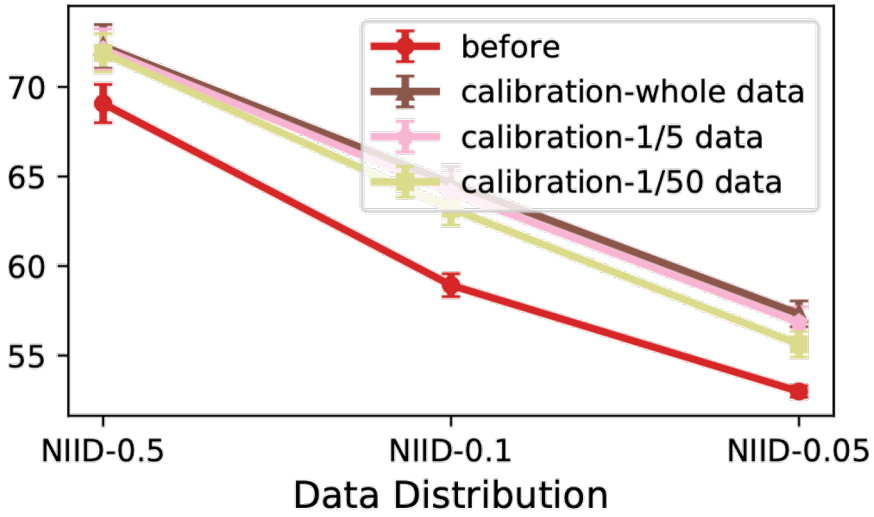
Table 1: Accuracy@1 (%) on CIFAR-10 with different degrees of heterogeneity.

Identical ⟷ not identical

Method	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.05$
FedAvg	68.62±0.77	58.55±0.98	52.33±0.43
FedAvg + clsnorm	69.65±0.35 (↑ 1.03)	58.94±0.08 (↑ 0.39)	51.74±4.02 (↓ 0.59)
FedAvg + clsprox	68.82±0.75 (↑ 0.20)	59.04±0.70 (↑ 0.49)	52.38±0.78 (↑ 0.05)
FedAvg + clsnorm + clsprox	68.75±0.75 (↑ 0.13)	58.80±0.30 (↑ 0.25)	52.39±0.24 (↑ 0.06)
FedAvg + calibration (whole data)	72.51±0.53 (↑ 3.89)	64.70±0.94 (↑ 6.15)	57.53±1.00 (↑ 5.20)

Non-IID level

- Regularizing the L2-norm (clsnorm): only effective for light data heterogeneity
- Regularizing the classifier parameters (clsprox): minor improvements
- Calibrating the classifier: significant performance improvement



These improvements verify their hypothesis, i.e., the bias is in the classifier

Outline

- Backgrounds of the FL with non-IID Data
- Heterogeneity in Federated Learning: The Devil Is in Classifier
- Classifier Calibration with Virtual Representations
- Performance Evaluation
- Conclusion

Classifier Calibration with Virtual Representations

CCVR runs on the global model at the server after the training (post processing)

Main idea: virtual features drawn from an estimated Gaussian Mixture Model (GMM)

$f_{\hat{\theta}}$: feature extractor

$g_{\hat{\varphi}}$: classifier

} Model parameter: $\hat{w} = (\hat{\theta}, \hat{\varphi})$

CCVR:

1. Feature distribution estimation (*f* : extract features ; estimate)

2. Virtual representations generation (based on estimated distribution)

3. Classifier re-train (re-train *g* with virtual representations)

Classifier Calibration with Virtual Representations

- **Feature distribution estimation**

  the features learned by DNN can be approximated with a mixture of Gaussian distribution

  Assumption: features of each class follow a Gaussian distribution

  mean $\boldsymbol{\mu}_c$
  covariance $\boldsymbol{\Sigma}_c$

1. Send global f to clients

   c: class index; **k**: client index; **j**: data index

2. Client produces features $\{\boldsymbol{z}_{c,k,1}, \ldots, \boldsymbol{z}_{c,k,N_{c,k}}\}$

$$\boldsymbol{z}_{c,k,j} = f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}_{c,k,j})$$

3. Compute local mean and covariance

$$\boldsymbol{\mu}_{c,k} = \frac{1}{N_{c,k}} \sum_{j=1}^{N_{c,k}} \boldsymbol{z}_{c,k,j}$$

$$\boldsymbol{\Sigma}_{c,k} = \frac{1}{N_{c,k}-1} \sum_{j=1}^{N_{c,k}} (\boldsymbol{z}_{c,k,j} - \boldsymbol{\mu}_{c,k})(\boldsymbol{z}_{c,k,j} - \boldsymbol{\mu}_{c,k})^T$$

4. Upload local statistics to the server

5. Compute global mean and covariance

$$\boldsymbol{\mu}_c = \frac{1}{N_c} \sum_{k=1}^{K} \sum_{j=1}^{N_{c,k}} \boldsymbol{z}_{c,k,j} = \sum_{k=1}^{K} \frac{N_{c,k}}{N_c} \boldsymbol{\mu}_{c,k}$$

$$\boldsymbol{\Sigma}_c = \frac{1}{N_c - 1} \sum_{k=1}^{K} \sum_{j=1}^{N_{c,k}} \boldsymbol{z}_{c,k,j} \boldsymbol{z}_{c,k,j}^T - \frac{N_c}{N_c - 1} \boldsymbol{\mu}_c \boldsymbol{\mu}_c^T$$

$$= \sum_{k=1}^{K} \frac{N_{c,k} - 1}{N_c - 1} \boldsymbol{\Sigma}_{c,k} + \sum_{k=1}^{K} \frac{N_{c,k}}{N_c - 1} \boldsymbol{\mu}_{c,k} \boldsymbol{\mu}_{c,k}^T - \frac{N_c}{N_c - 1} \boldsymbol{\mu}_c \boldsymbol{\mu}_c^T$$

Classifier Calibration with Virtual Representations

- **Virtual Representations Generation**

  Server generates a set G_c of virtual features from the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$

  The number $M_c := |G_c|$ for each class c could be determined by the fraction $\frac{N_c}{|\mathcal{D}|}$

- **Classifier Re-Training**

  Re-train the classifier using virtual representations

$$\min_{\widetilde{\varphi}} \mathbb{E}_{(\boldsymbol{z},y)\sim\bigcup_{c\in[C]} G_c}[\ell(g_{\widetilde{\varphi}}(\boldsymbol{z}), y)]$$

  Final classification model:
  pre-trained feature extractor $f_{\widehat{\theta}}$
  calibrated classifier $g_{\widetilde{\varphi}}$

- **Privacy Protection**

  Only uploads local Gaussian statistics
  rather than the raw representations

```
# Clients execute:
foreach client k ∈ [K] do
    foreach class c ∈ [C] do
        Produce z_{c,k,j} = f_θ̂(x_{c,k,j}) for j-th
          sample in D_c^k for j ∈ [N_{c,k}].
        Compute μ_{c,k} and Σ_{c,k} using Eq. (2).
    end
    Send {(μ_{c,k}, Σ_{c,k}) : c ∈ [C]} to server.
end
# Server executes:
foreach class c ∈ [C] do
    Compute μ_c and Σ_c using Eq. (3) and (4).
    Draw a set G_c of M_c features from
      N(μ_c, Σ_c) with ground truth label c.
end
Output: Set of virtual representations ⋃_{c∈[C]} G_c
```

Outline

- Backgrounds of the FL with non-IID Data
- Heterogeneity in Federated Learning: The Devil Is in Classifier
- Classifier Calibration with Virtual Representations
- Performance Evaluation
- Conclusion

Experiment Setup

- **Dataset:** CIFAR-10, CIFAR-100, and CINIC-10 (constructed from ImageNet & CIFAR-10)
- **Model:** 4-layer CNN with 2-layer MLP; MobileNetV2

  Apply ReLU and Tukey's transformation before classifier re-training; more Gaussian-like

- **Baselines:** 4 methods

  - **FedAvg:** typical FL method
  - **FedProx:** add a regularization term on local loss function
  - **FedAvgM:** adopt momentum update on the server-side
  - **Moon:** adopt the contrastive loss to maximize the agreement of the local/global model

Simulation results

Table 2: Accuracy@1 (%) on CIFAR-10 with different degrees of heterogeneity ($\alpha \in \{0.5, 0.1, 0.05\}$), CIFAR-100 and CINIC-10.

Identical ⟷ not identical

	Method	$\alpha = 0.5$	$\alpha = 0.1$	$\alpha = 0.05$	CIFAR-100	CINIC-10
No Calibration	FedAvg	68.62±0.77	58.55±0.98	52.33±0.43	66.25±0.54	60.20±2.04
	FedProx	69.07±1.07	58.93±0.64	53.00±0.32	66.31±0.39	60.52±2.07
	FedAvgM	69.00±1.68	59.22±1.14	51.98±0.91	66.43±0.23	60.46±0.73
	MOON	70.48±0.36	57.36±0.85	49.91±0.38	67.02±0.31	65.67±2.10
CCVR (Ours.)	FedAvg	71.03±0.40 (↑ 2.41)	**62.68±0.54** (↑ 4.13)	54.95±0.61 (↑ 2.62)	66.60±0.63 (↑ 0.35)	69.99±0.54 (↑ 9.79)
	FedProx	70.99±1.21 (↑ 1.92)	62.60±0.43 (↑ 3.67)	**55.79±1.07** (↑ 2.79)	66.61±0.48 (↑ 0.30)	70.05±0.66 (↑ 9.53)
	FedAvgM	**71.49±0.88** (↑ 2.49)	62.64±1.07 (↑ 3.42)	54.57±0.58 (↑ 2.59)	66.71±0.16 (↑ 0.28)	**70.87±0.61** (↑ 10.41)
	MOON	71.29±0.11 (↑ 0.81)	62.22±0.70 (↑ 4.86)	55.60±0.63 (↑ 5.69)	**67.17±0.37** (↑ 0.15)	69.42±0.65 (↑ 3.75)
Oracle	FedAvg	72.51±0.53 (↑ 3.89)	64.70±0.94 (↑ 6.15)	57.53±1.00 (↑ 5.20)	66.84±0.50 (↑ 0.59)	**73.47±0.30** (↑ 13.27)
	FedProx	72.26±1.22 (↑ 3.19)	64.63±0.93 (↑ 5.70)	57.33±0.72 (↑ 4.33)	66.68±0.43 (↑ 0.37)	73.10±0.57 (↑ 12.58)
	FedAvgM	**73.30±0.19** (↑ 4.30)	64.24±1.32 (↑ 5.02)	57.11±1.08 (↑ 5.13)	66.94±0.32 (↑ 0.51)	72.88±0.37 (↑ 12.42)
	MOON	72.05±0.16 (↑ 1.57)	**64.94±0.58** (↑ 7.58)	**58.14±0.47** (↑ 8.23)	**67.56±0.44** (↑ 0.54)	73.38±0.23 (↑ 7.71)

whole data are available for classifier calibration

- CCVR consistently improves all baseline methods
- The improvement on CIFAR-100 is small (biased classifier & feature quality)
- The improvement on CINIC-10 is huge

CCVR on CINIC-10

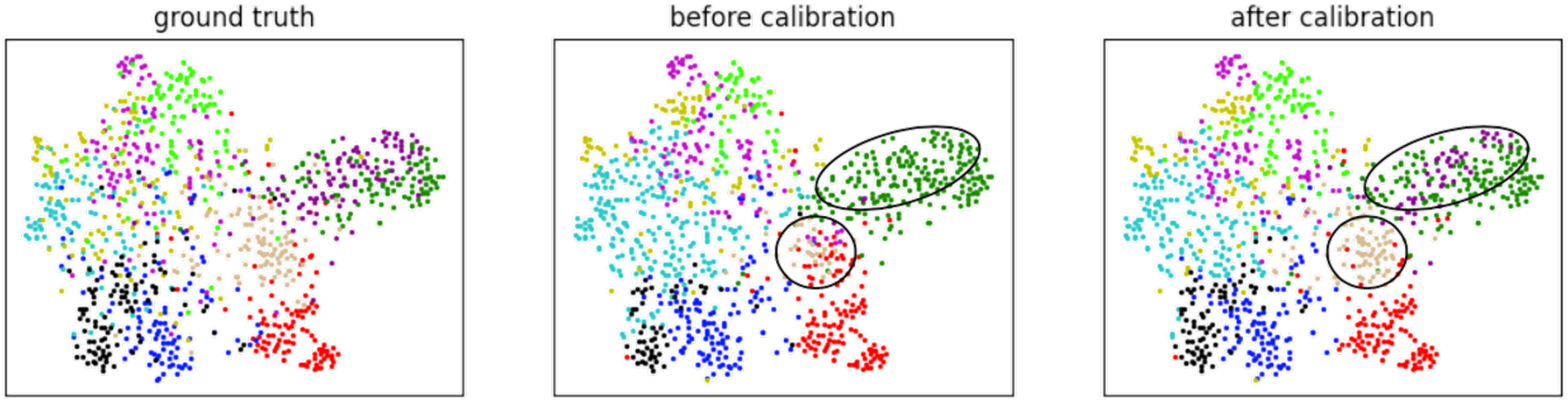


Figure 5: t-SNE visualization of the features learned by FedAvg on CINIC-10. The features are colored by the ground truth and the predictions of the classifier before and after applying CCVR. Best viewed in color.

- Fig.1 & 2, some classes dominate the classification results
- There exists a great bias in the classifier
- Mistakes are basically made when identifying features that are close to the decision boundary

CCVR effective in models with good feature representations but serious classifier biases

Other observations

- hyperparameter in CCVR: the number of virtual features M_c for each class c to generate

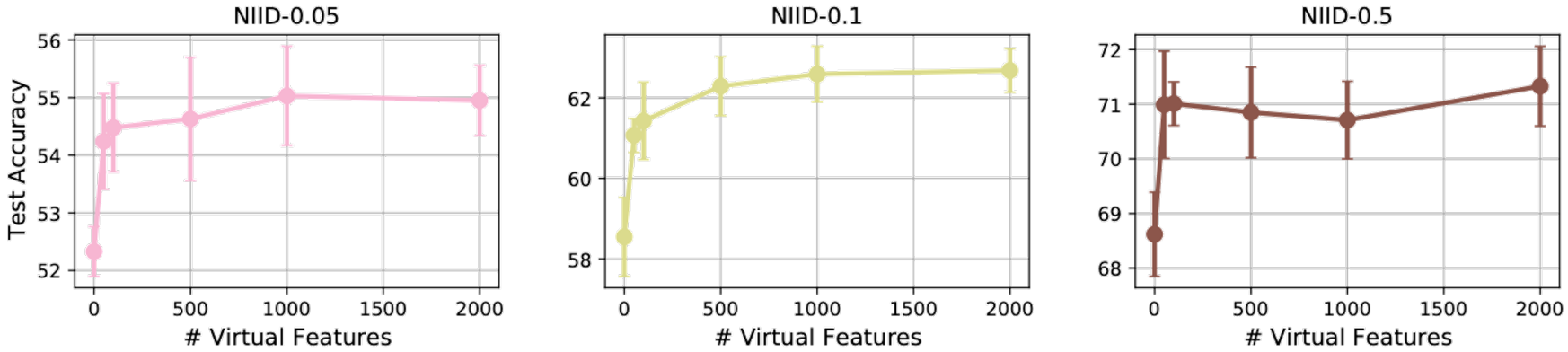


Figure 7: Accuracy@1 (%) of CCVR on CIFAR-10 with different numbers of virtual samples.

- How does CCVR improve the classifier?

L2-norms of the classifier weight vectors before and after applying CCVR

Conclusion

- Observe that the classifiers of different local models are less similar than other layers, and there exists bias among classifiers
- Propose a novel method CCVR, which samples virtual features from an approximated GMM to calibrate the classifier
- Experimental results show that CCVR improves over several popular federated learning algorithms


THANK YOU
UNIVERSITY of HOUSTON | ENGINEERING