# Virtual Homogeneity Learning: Defending against Data Heterogeneity in Federated Learning

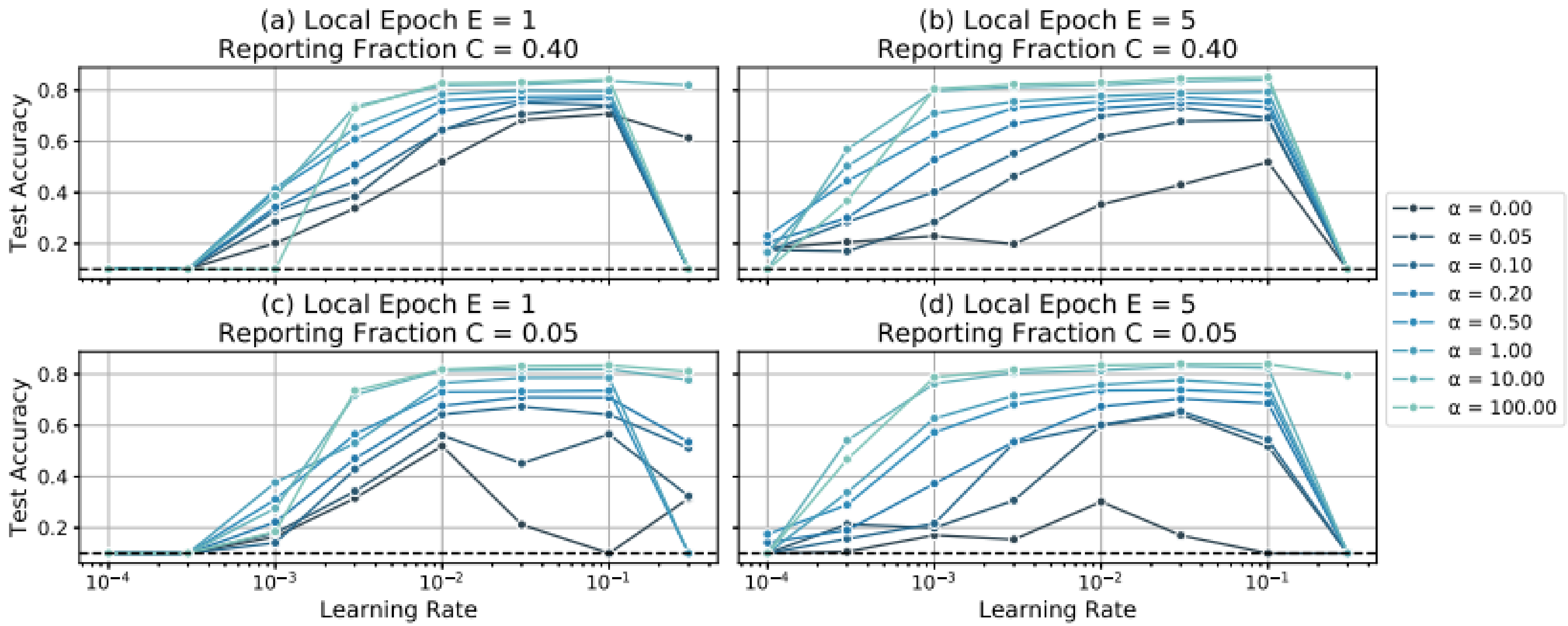
Zhenheng Tang*  Yonggang Zhang*  Shaohuai Shi  Xin He  Bo Han  Xiaowen Chu

香港浸會大學
HONG KONG BAPTIST UNIVERSITY

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

* equal contribution

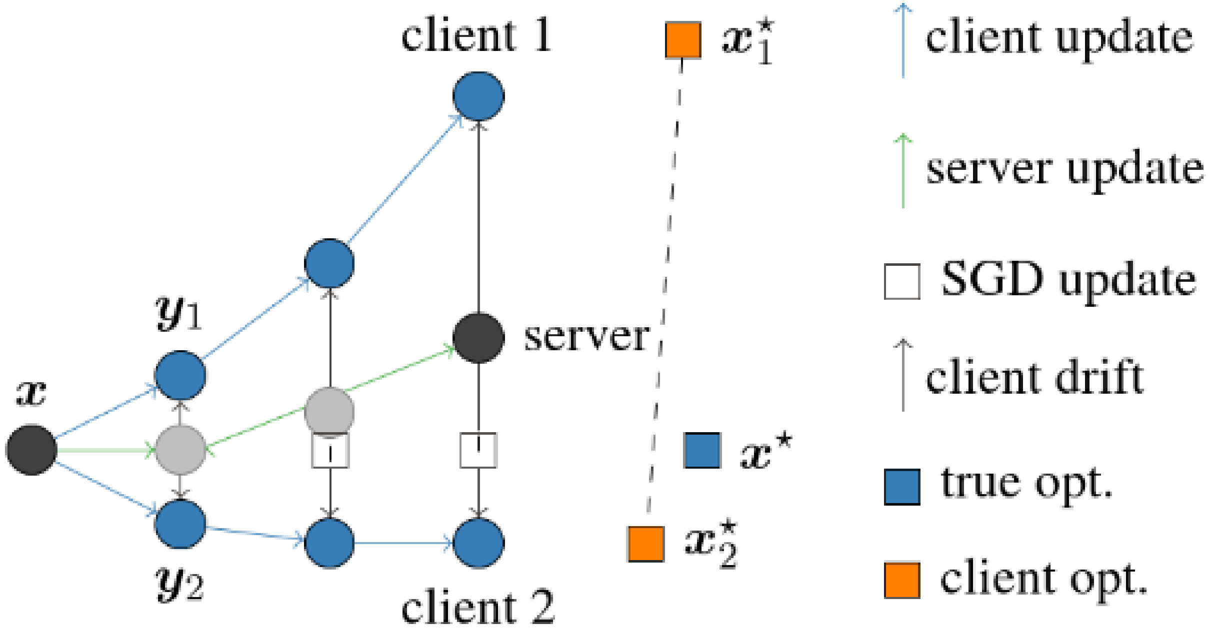
# Data Heterogeneity




Performance Drop [1]


[1] Hsu, T. H., Qi, H., and Brown, M. Measuring the effects of non-identical data distribution for federated visual classi- fication. arXiv preprint arxiv: 1909.06335, 2019.

# Data Heterogeneity




Client Drift[2]

[2] Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. SCAFFOLD: Stochastic controlled averaging for federated learning. In ICML, 2020.

# Related Work

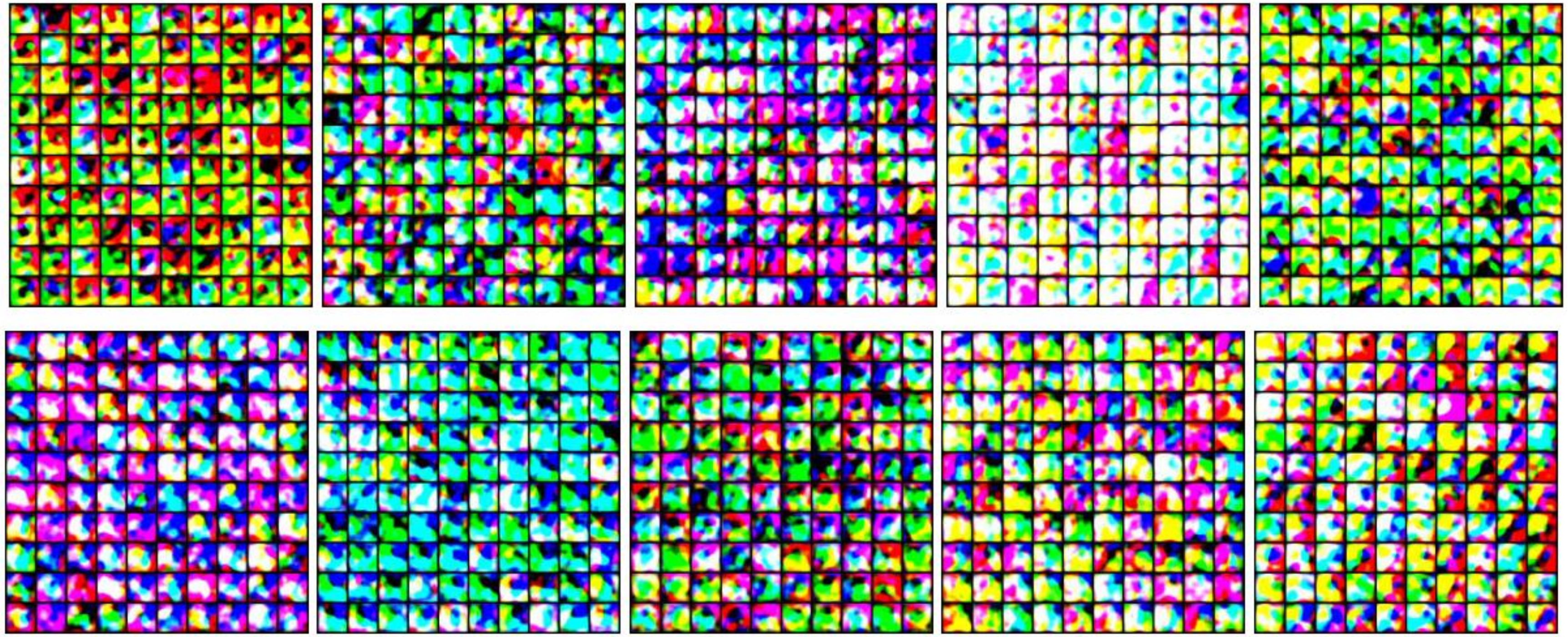
**Model Regularization.**

**Optimization Schemes**.

**Sharing Data**.

*Is it possible to defend against data heterogeneity in FL systems by sharing data containing* ***no private information****?*

# Virtual Dataset

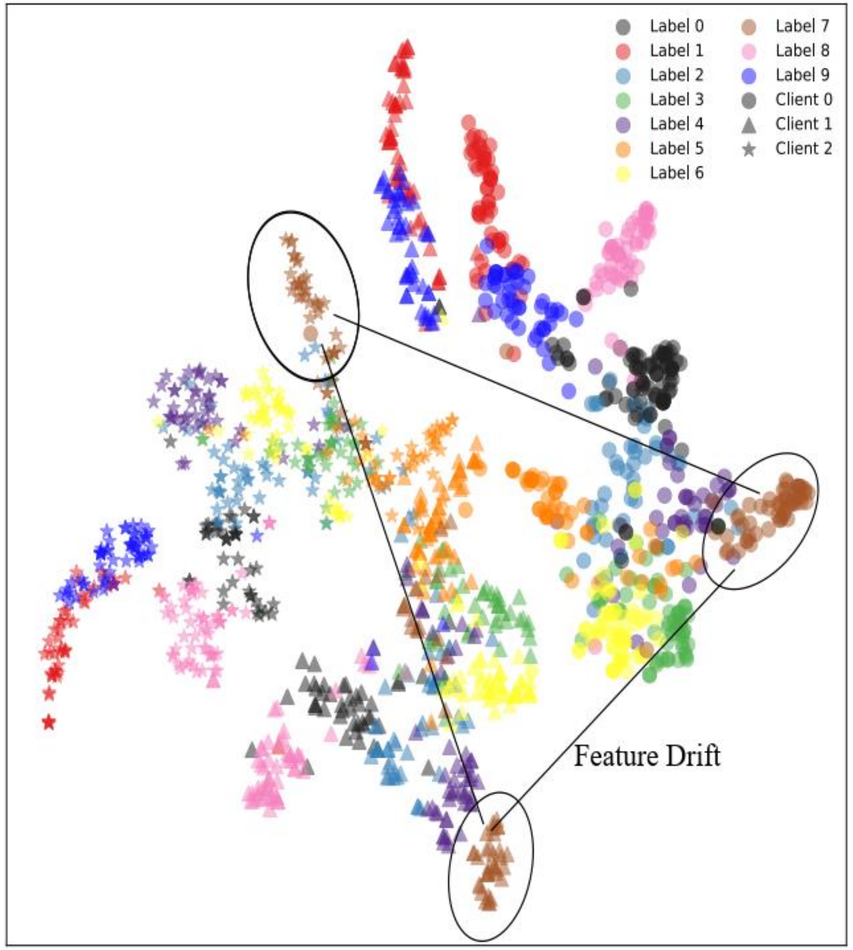


A shared noise dataset between clients.
Each figure shows 90 data samples, representing a virtual class.

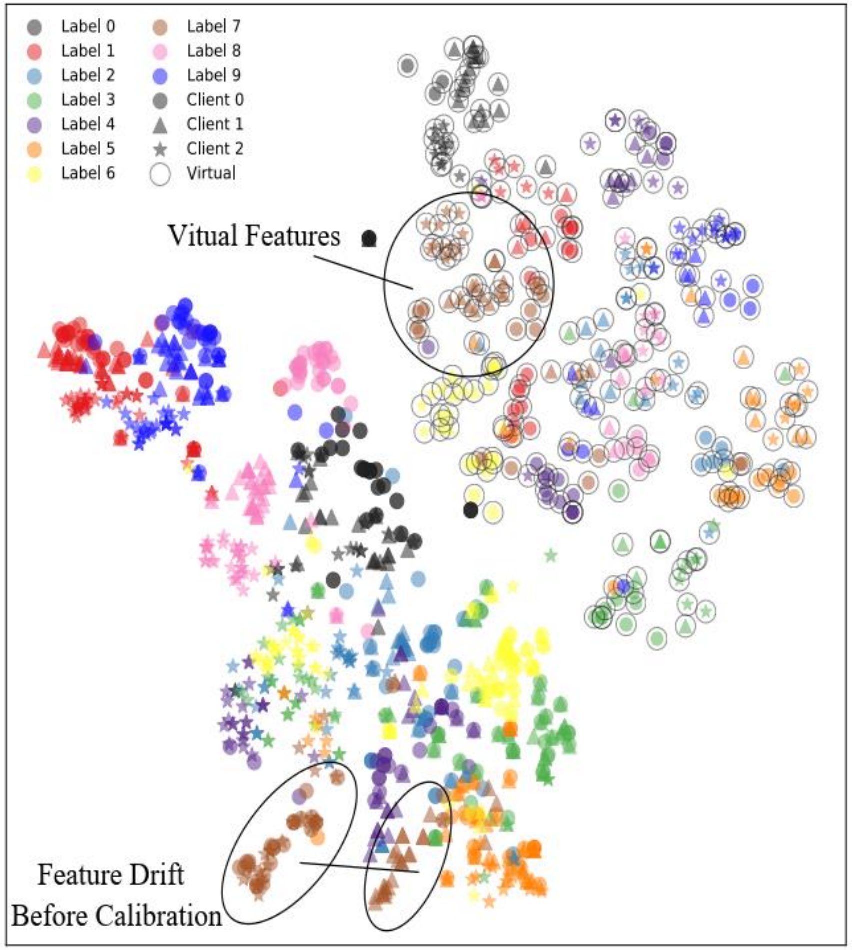
Generated by an un-trained Style-GAN.

# Feature Drift

**FedAvg** with private data




**FedAvg** with Both shared noise data and private data

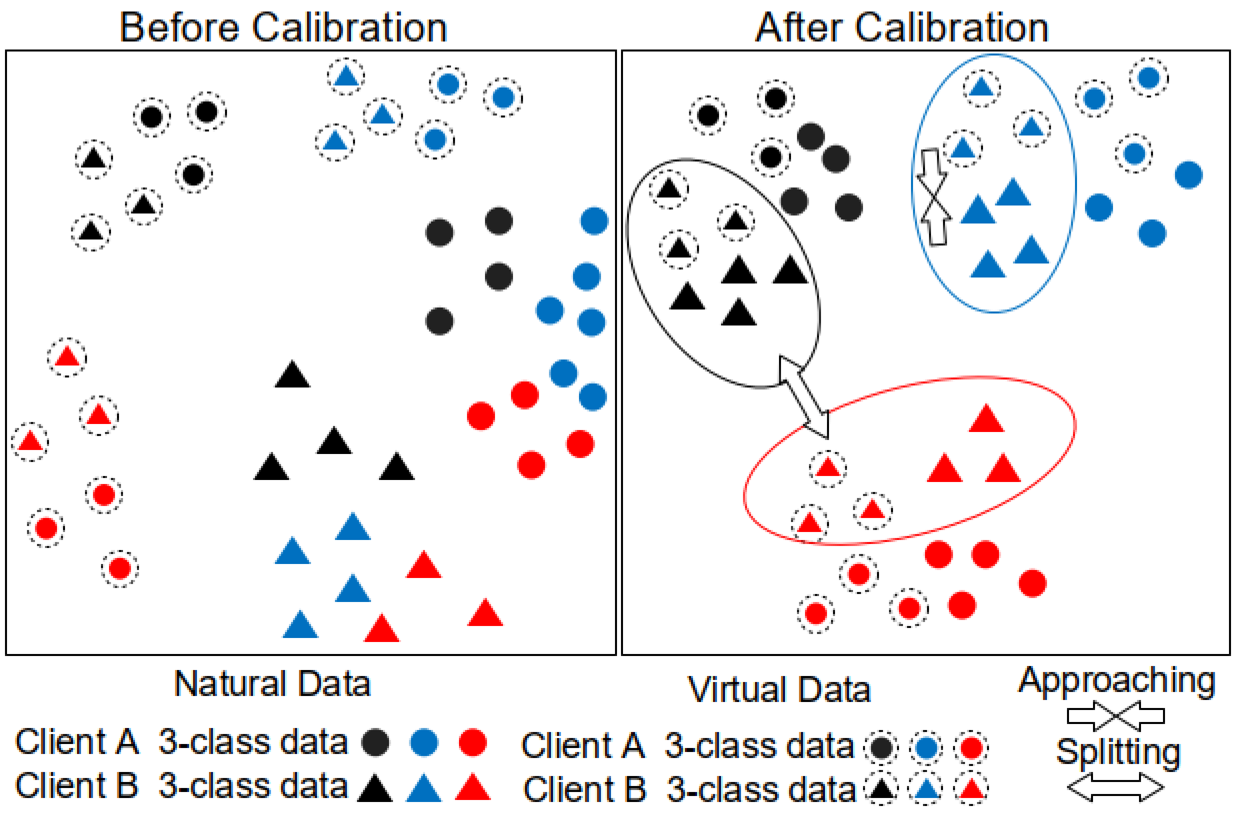



The Private Data of the same class has **divergent feature distribution** between clients

The Shared Noise Data of the same class has **similar feature distribution** between clients

# Feature Calibration

Pull the samples of the **same label** (Virtual and Natural) together.



# Virtual Homogeneity Learning

Virtual Homogeneity Learning – We calibrate private features based on the virtual features

$$\bar{J}_k(w) = \mathbb{E}_{\substack{(x,y)\sim\mathcal{P} \\ (\tilde{x},\tilde{y})\sim\tilde{\mathcal{P}}}} \ell(f(x;w),y) + \ell(f(\tilde{x};w),\tilde{y}) + \lambda \mathbb{E}_y d(\mathcal{P}(\phi|Y=y), \mathcal{P}(\tilde{\phi}|Y=y)),$$

$x$ Natural Data

$\tilde{x}$ Virtual Data

$\phi$ Feature

**Algorithm 1** FedAvg with **VHL**

**server input:** initial $w^0$, maximum communication round $R$
**client** $k$**'s input:** local epochs $E$
**Initialization:** server distributes the initial model $w^0$ to all clients, as well as the virtual dataset $\tilde{D}$ .

**Server_Executes:**
**for** each round $r = 0, 1, \cdots, R$ **do**
server samples a set of clients $\mathcal{S}_r \subseteq \{1, ..., K\}$
server **communicates** $w_r$ to all clients $k \in \mathcal{S}$
**for** each client $k \in \mathcal{S}^r$ **in parallel do do**
$w^{r+1}_{k,E-1} \leftarrow \text{ClientUpdate}(k, w^r)$
**end for**
$w^{r+1} \leftarrow \sum_{k=1}^{K} p_k w^r_{k,E-1}$
**end for**

**Client_Training**$(k, w)$**:**
**for** each local epoch $j$ with $j = 0, \cdots, E-1$ **do**
$w_{k,j+1} \leftarrow w_{k,j} - \eta_{k,j} \nabla_w \bar{J}_k(w)$, i.e., Eq. 6
**end for**
**return** $w$ to server

# Experiment

**Datasets**: CIFAR-10, Fashion-MNIST, SVHN, CIFAR-100

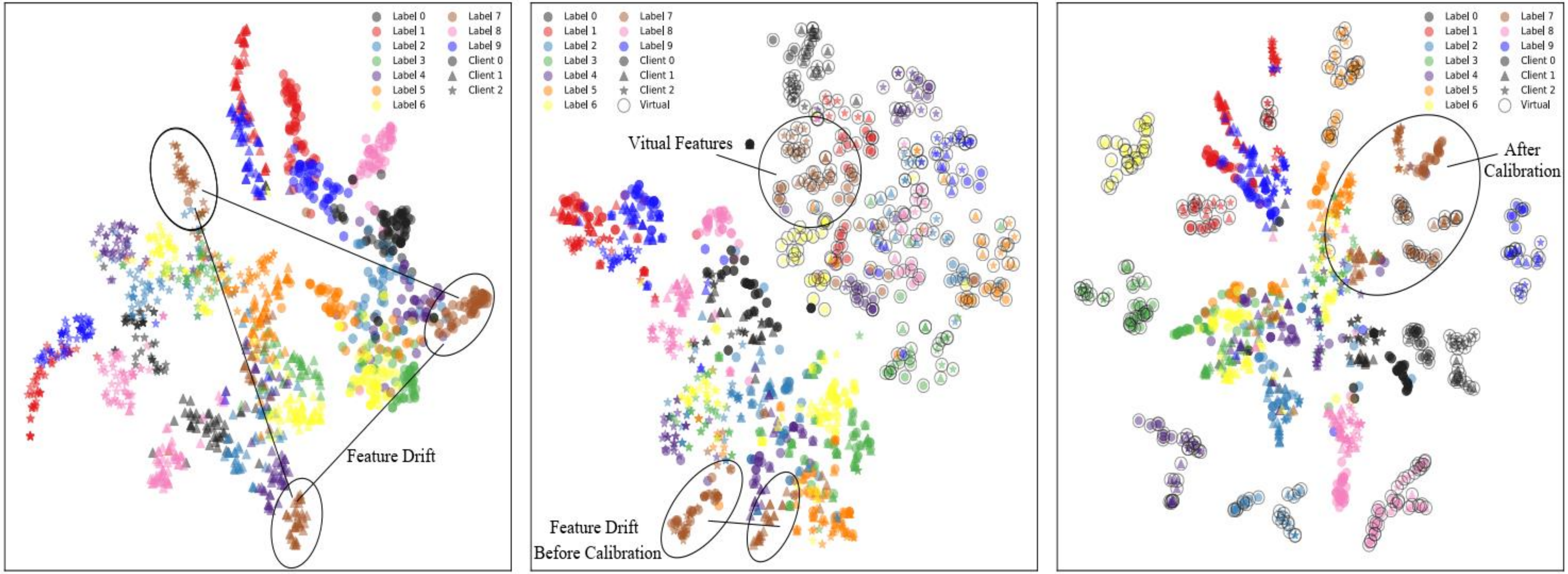
**FL settings**: 10 clients or 100 clients, LDA partition, a=0.1 or 0.05, Local epoch = 1 or 5.

**Model**: ResNet18 for CIFAR-10, Fashion-MNIST, SVHN, ResNet50 for CIFAR-100.

**FL algorithms**: FedAvg, FedProx, SCAFFOLD, FedNova.

# Experiment

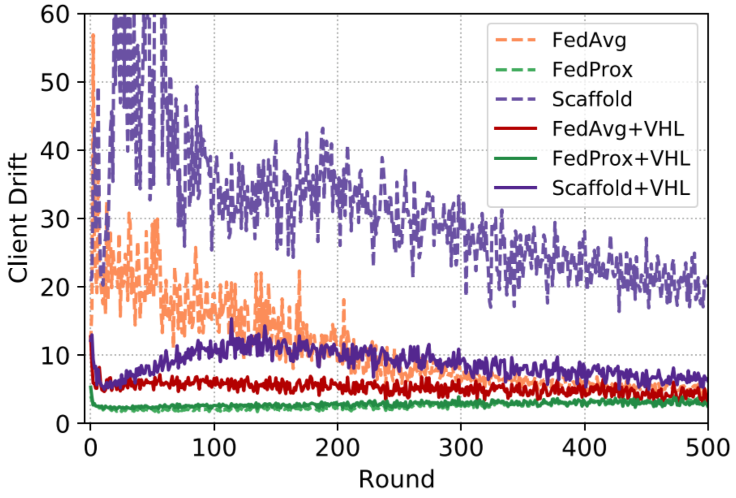
**Feature Distribution After Calibration**




(a) FedAvg without VHL (b) FedAvg with Naive VHL (c) FedAvg with VHL

# Experiment

$$\frac{1}{|\mathcal{S}^r|}\sum_{i\in\mathcal{S}^r}\|\bar{w}-w_i\|$$




(b) Model Divergence

Measuring Empirical Client Drift (Model Divergence)

# Experiment

## Metrics: Test Accuracy, Target Round

*Table 1.* Results with/without VHL on CIFAR-10.

| | ACC ↑ | ROUND ↓ | Speedup ↑ |
|---|---|---|---|
| | w/ (w/o) **VHL** | | |
| centralized training ACC = **92.88%** (92.53%) | | | |
| $a = 0.1, E = 1, K = 10$ ( Target ACC = 79% ) | | | |
| FedAvg | **87.82** (79.98) | **128** (287) | × **2.2** (× 1.0) |
| FedProx | **87.30** (83.56) | **128** (188) | × **2.2** (× 1.5) |
| SCAFFOLD | **84.87** (83.58) | **90** (291) | ×**3.2** (× 1.0) |
| FedNova | **87.56** (81.35) | **128** (351) | × **2.2** (× 0.8) |
| $a = 0.05, E = 1, K = 10$ ( Target ACC = 69% ) | | | |
| FedAvg | **79.23** (69.02) | **112** ( 411 ) | × **3.7** (× 1.0) |
| FedProx | **80.84** (78.66) | **151** ( 201 ) | × **2.7** (× 2.0) |
| SCAFFOLD | **55.73** (38.55) | **Nan** (Nan) | **Nan** (Nan) |
| FedNova | **80.59** (64.78) | **247** (Nan) | × **1.66** (Nan) |
| $a = 0.1, E = 5, K = 10$ ( Target ACC = 84% ) | | | |
| FedAvg | **89.93** (84.79) | **91** (255 ) | × **2.8** (× 1.0) |
| FedProx | **86.41** (82,18) | **255** (Nan) | × **1.0** (Nan) |
| SCAFFOLD | **87.27** (86.20) | **45** (66) | × **5.7** (× 2.0) |
| FedNova | **90.24** (86.09) | **67** (127) | × **3.8** (× 1.0) |
| $a = 0.1, E = 1, K = 100$ ( Target ACC = 49% ) | | | |
| FedAvg | **70.20** (49.61) | **385** (957) | × **2.5** (× 1.0) |
| FedProx | **73.90** (49.97) | **325** (842) | × **2.9** (× 1.1) |
| SCAFFOLD | **59.66** (52.24) | **479** (664) | × **2.0** (× 1.4) |
| FedNova | **61.59** (46.53) | **554** (Nan) | × **1.7** (Nan) |

"ROUND" represents the communication rounds that need to attain the target accuracy. The notion ↓ (↑) indicates smaller (larger) values are preferred.

*Table 2.* Results with/without VHL on FMNIST.

| | ACC ↑ | ROUND ↓ | Speedup ↑ |
|---|---|---|---|
| | w/ (w/o) **VHL** | | |
| centralized training ACC = **93.8%** (93.7%) | | | |
| $a = 0.1, E = 1, K = 10$ ( Target ACC = 86% ) | | | |
| FedAvg | **92.05** (86.81) | **52** (119) | × **2.3** (× 1.0) |
| FedProx | **90.68** (87.12) | **31** (135) | × **3.8** (× 0.9) |
| SCAFFOLD | **90.27** (86.21) | **14** (143) | × **8.5** (× 0.8) |
| FedNova | **91.88** (86.99) | **52** (83) | × **2.3** (× 1.4) |
| $a = 0.05, E = 1, K = 10$ ( Target ACC = 78% ) | | | |
| FedAvg | **89.06** (78.57) | **53** (425) | × **8.0** (× 1.0) |
| FedProx | **87.76** (81.96) | **30** (41) | × **14.2** (× 10.4) |
| SCAFFOLD | **80.68** (76.08) | **58** (Nan) | × **7.3** (Nan) |
| FedNova | **87.25** (79.06) | **30** (538) | × **14.2** (× 0.8) |
| $a = 0.1, E = 5, K = 10$ ( Target ACC = 87% ) | | | |
| FedAvg | **91.52** (87.45) | **51** (278) | × **5.5** (× 1.0 ) |
| FedProx | **88.27** (86.07) | **74** (Nan) | × **3.8** (Nan) |
| SCAFFOLD | **91.82** (87.10) | **20** (105) | × **13.9** (× 2.7) |
| FedNova | **91.86** (87.53) | **51** (193) | × **5.5** (× 1.4) |
| $a = 0.1, E = 1, K = 100$ ( Target ACC = 90% ) | | | |
| FedAvg | **91.14** (90.11) | **436** (658) | × **1.5** (× 1.0) |
| FedProx | **91.37** (90.71) | **283** (491) | × **2.3** (× 1.3) |
| SCAFFOLD | **87.91** (85.99) | **Nan** (Nan) | **Nan** (Nan) |
| FedNova | **88.34** (87.09) | **Nan** (Nan) | **Nan** (Nan) |

*Table 4.* Results with/without VHL on SVHN.

| | ACC ↑ | ROUND ↓ | Speedup ↑ |
|---|---|---|---|
| | w/ (w/o) **VHL** | | |
| centralized training ACC = 95.01% (**95.27%**) | | | |
| $a = 0.1, E = 1, K = 10$ ( Target ACC = 88% ) | | | |
| FedAvg | **93.49** (88.56) | **75** (251) | × **3.3** (× 1.0) |
| FedProx | **91.70** (86.51) | **271** (Nan) | × **0.9** (Nan) |
| SCAFFOLD | **87.54** (80.61) | Nan (Nan) | Nan (Nan) |
| FedNova | **93.35** (89.12) | **75** (251) | × **3.3** (× 1.0) |
| $a = 0.05, E = 1, K = 10$ ( Target ACC = 82% ) | | | |
| FedAvg | **92.26** (82.67) | **94** (357) | × **3.8** (× 1.0) |
| FedProx | **89.30** (78.57) | **320** (Nan) | × **1.1** (Nan) |
| SCAFFOLD | **83.89** (74.23) | **147** (Nan) | × **2.4** (Nan) |
| FedNova | **91.82** (82.22) | **128** (741) | × **2.8** (× 0.5) |
| $a = 0.1, E = 5, K = 10$ ( Target ACC = 87% ) | | | |
| FedAvg | **90.52** (87.92) | **145** (131) | × **0.9** (× 1.0) |
| FedProx | **87.20** (78.43) | **351** (Nan) | × **0.4** (Nan) |
| SCAFFOLD | **88.04** (81.07) | **210** (Nan) | × **0.6** (Nan) |
| FedNova | **90.99** (88.17) | **75** (162) | × **1.7** (× 0.8) |
| $a = 0.1, E = 1, K = 100$ ( Target ACC = 89% ) | | | |
| FedAvg | **92.05** (89.44) | **362** (618) | × **1.7** (× 1.0) |
| FedProx | **92.08** (89.51) | **356** (618) | × **1.7** (× 1.0) |
| SCAFFOLD | 89.21 (**89.55**) | 968 (**643**) | × 0.6 ( × **1.0** ) |
| FedNova | **92.01** (82.08) | **676** (Nan) | × **0.9** (Nan) |

*Table 5.* Results with/without VHL on CIFAR-100.

| | ACC ↑ | ROUND ↓ | Speedup ↑ |
|---|---|---|---|
| | w/ (w/o) **VHL** | | |
| centralized training ACC = 71.90 % (**74.25 %**) | | | |
| $a = 0.1, E = 1, K = 10$ ( Target ACC = 67% ) | | | |
| FedAvg | **70.04** (67.95) | **384** (497) | × **1.3** (× 1.0) |
| FedProx | **68.29** (65.29) | **617** (Nan) | × **0.8** (Nan) |
| SCAFFOLD | **67.88** (67.14) | **294** (766) | × **1.7** (× 0.6) |
| FedNova | **69.58** (68.26) | **384** (472) | × **1.3** (× 1.1) |
| $a = 0.05, E = 1, K = 10$ ( Target ACC = 62% ) | | | |
| FedAvg | **65.61** (62.07) | **354** (514) | × **1.5** (× 1.0) |
| FedProx | **64.39** (61.52) | **482** (Nan) | × **1.1** (Nan) |
| SCAFFOLD | **60.67** (59.04) | Nan (Nan) | Nan (Nan) |
| FedNova | **66.45** (60.35) | **320** (Nan) | × **1.6** (Nan) |
| $a = 0.1, E = 5, K = 10$ ( Target ACC = 69% ) | | | |
| FedAvg | **69.85** (69.81) | **327** (283) | × **0.9** (× 1.0) |
| FedProx | **63.83** (62.62) | Nan (Nan) | Nan (Nan) |
| SCAFFOLD | 69.43 (**70.68**) | 291 (**171**) | × 1.0 (× **1.7**) |
| FedNova | 68.86 (**70.05**) | Nan (**292**) | Nan (× **1.0**) |
| $a = 0.1, E = 1, K = 100$ ( Target ACC = 48% ) | | | |
| FedAvg | **53.45** (48.33) | **717** (967) | × **1.3** (× 1.0) |
| FedProx | **52.68** (48.14) | **717** (955) | × **1.3** (× 1.0) |
| SCAFFOLD | **54.93** (51.63) | **656** (827) | × **1.5** (× 1.2) |
| FedNova | **53.50** (48.12) | **797** (967) | × **1.2** (× 1.0) |

# Ablation Study

| | | | | |
|---|---|---|---|---|
| Baselines | 79.98 | 83.56 | 83.58 | 81.35 |
| VHL | 87.82 | 87.30 | 84.87 | 87.56 |
| VFTL | 80.38 | 82.20 | 83.83 | 80.63 |
| Naive VHL | 86.50 | 85.66 | 85.70 | 85.74 |
| VFA | 85.14 | 84.75 | 85.31 | 86.59 |

- Virtual Feature Transfer Learning (VFTL): Pretrained on noise data
- Naive VHL: Training with both private data and noise data without feature calibration
- Virtual Feature Alignment (VFA): Feature calibration based on random features of different classes.

| | | | | |
|---|---|---|---|---|
| Pure Noise | 87.01 | 86.46 | 85.57 | 87.81 |
| Simple-CNN | 84.87 | 85.30 | 84.15 | 85.25 |
| Tiny-ImageNet | 84.05 | 83.62 | 81.57 | 85.41 |
| $B_v = 64$ | 87.36 | 86.36 | 82.39 | 87.20 |
| $B_v = 128$ | 87.82 | 87.30 | 84.87 | 87.56 |
| $B_v = 256$ | 88.95 | 86.82 | 84.68 | 87.89 |
| $B_v = 384$ | 89.69 | 86.59 | 85.87 | 88.73 |
| $\lambda = 0.2$ | 87.04 | 86.02 | 84.41 | 87.65 |
| $\lambda = 0.5$ | 87.02 | 85.99 | 84.29 | 87.15 |
| $\lambda = 1.0$ | 87.82 | 87.30 | 84.87 | 87.56 |
| $\lambda = 2.0$ | 87.87 | 86.86 | 84.81 | 88.71 |
| $\lambda = 5.0$ | 83.52 | 88.34 | 85.58 | 88.47 |
| $h_{shallow}$ | 86.10 | 85.59 | 85.19 | 84.95 |
| $h_{middle}$ | 87.06 | 86.30 | 87.97 | 87.27 |
| $h_{deep}$ | 89.26 | 87.54 | 86.00 | 88.42 |
| $h_{last}$ | 87.82 | 87.30 | 84.87 | 87.56 |

- Different Noise
- Different Batch Size
- Different weight of calibration loss
- Different depth of calibration

Thank You