# 6 Appendix

## 6.1 Code Release & Implementation details

FedX is open-sourced at https://github.com/Sungwon-Han/FEDX. Algorithm 1 shows the overall training process.

**Algorithm 1** Local update process of FedX

**Input:** the number of local epochs $E$, global model $F$, local model $f^m$, projection head $h^m$, local dataset $\mathcal{D}^m$, and temperature $\tau$
**Output:** a trained local model $f^m$ and a projection head $h^m$

```
In each client m,
f^m ← F                          // Replace the local model with the global model
F ← F.detach()        // Fix the global model for a given communication round
for epoch e ∈ {1, 2, ..., E} do
    for batch B, B_r ∈ D^i do
        B̃ ← Augment(B)
        for x_i ∈ B, x̃_i ∈ B̃, and x_j ∈ B_r do
            /*  Local KD  */
            z_i, z_j, z̃_i ← f^m(x_i), f^m(x_j), f^m(x̃_i)
            // Local relationship vectors
            r_i^j = exp(sim(z_i, z_j)/τ) / Σ_{k∈B_r} exp(sim(z_i, z_k)/τ)
            r̃_i^j = exp(sim(z̃_i, z_j)/τ) / Σ_{k∈B_r} exp(sim(z̃_i, z_k))/τ)
            L_c^local ← Calculate from Eq. 3 or 5
            L_r^local ← Calculate from Eq. 7
            L_local-KD = L_c^local + L_r^local
            /*  Global KD  */
            z_i^l, z̃_i^l ← h^m ∘ f^m(x_i), h^m ∘ f^m(x̃_i)
            z_i^g, z_j^g, z̃_i^g ← F(x_i), F(x_j), F(x̃_i)
            // Global relationship vectors
            r'_i^j = exp(sim(z_i^l, z_j^g)/τ) / Σ_{k∈B_r} exp(sim(z_i^l, z_k^g)/τ)
            r̃'_i^j = exp(sim(z̃_i^l, z_j^g)/τ) / Σ_{k∈B_r} exp(sim(z̃_i^l, z_k^g)/τ)
            L_c^global ← Calculate from Eq. 9
            L_r^global ← Calculate from Eq. 11
            L_global-KD = L_c^global + L_r^global
        end
        L_total-KD = L_local-KD + L_global-KD         // Calculate the total loss
        Update f^m, h^m via back-propagation    // Update local model parameters
    end
end
return f^m, h^m
```

## 6.2 Further Results on Performance Evaluation

**Statistical errors in Table 1.** Table 5 reports standard errors from various training iterations (i.e., last five epochs & best five epochs), which shows that our model's accuracy converges stably.

Table 5: Adding FedX increases the performance of various federate learning models. The final round accuracy and the best accuracy are reported with standard errors.

| Method | CIFAR-10 | | SVHN | | F-MNIST | |
|---|---|---|---|---|---|---|
| | Last | Best | Last | Best | Last | Best |
| FedSimCLR | 51.31±0.25 | 52.88±0.06 | 75.19±3.14 | 76.50±0.04 | 77.66±0.75 | 79.44±0.06 |
| + FedX | **56.88±0.72** | **57.95±0.03** | **77.19±1.67** | **77.70±0.09** | **81.98±0.74** | **82.47±0.02** |
| FedMoCo | 56.74±1.63 | 57.82±0.02 | 70.69±3.03 | 70.99±0.07 | 82.31±0.18 | 83.58±0.06 |
| + FedX | **58.23±1.22** | **59.43±0.09** | **73.57±2.79** | **73.92±0.02** | **83.62±0.32** | **84.65±0.04** |
| FedBYOL | 52.24±0.61 | 53.14±0.06 | 65.95±1.62 | 67.32±0.24 | 81.45±0.27 | 82.37±0.06 |
| + FedX | **56.49±0.72** | **57.79±0.12** | **68.94±1.13** | **69.05±0.07** | **83.18±0.31** | **84.30±0.07** |
| FedProtoCL | 51.33±1.03 | 52.12±0.03 | 49.85±0.77 | 50.19±0.11 | 81.76±0.22 | **83.57±0.04** |
| + FedX | **55.36±0.98** | **56.76±0.01** | **69.31±1.72** | **69.75±0.15** | **82.74±0.35** | 83.34±0.04 |
| FedU | 50.79±0.47 | 50.79±0.05 | 66.02±1.83 | 66.22±0.17 | 80.59±0.42 | 82.03±0.05 |
| + FedX | **56.15±0.58** | **57.26±0.05** | **68.13±1.17** | **68.39±0.07** | **83.73±0.20** | **84.12±0.03** |

Table 6: Performance improvement with three different algorithms on classification accuracy. Both the final round accuracy and the best accuracy show that FedX brings nontrivial improvement over the baseline algorithm.

| Method | CIFAR-10 | | SVHN | | F-MNIST | |
|---|---|---|---|---|---|---|
| | Last | Best | Last | Best | Last | Best |
| FedSimCLR | 51.31 | 52.88 | 75.19 | 76.50 | 77.66 | 79.44 |
| + FedCA | 47.46 | 48.54 | 59.40 | 59.86 | 81.51 | 82.05 |
| + MOON-unsup | 51.78 | 52.84 | 75.36 | 76.03 | 80.58 | 80.93 |
| + FedX (ours) | **56.88** | **57.95** | **77.19** | **77.70** | **81.98** | **82.47** |

Table 7: Performance improvement on ImageNet-10. Both last and best round accuracy are reported.

| Method | FedSimCLR | | FedSimCLR+FedX | |
|---|---|---|---|---|
| | Last | Best | Last | Best |
| ImageNet-10 | 81.50 | 81.50 | **86.17** | **86.57** |

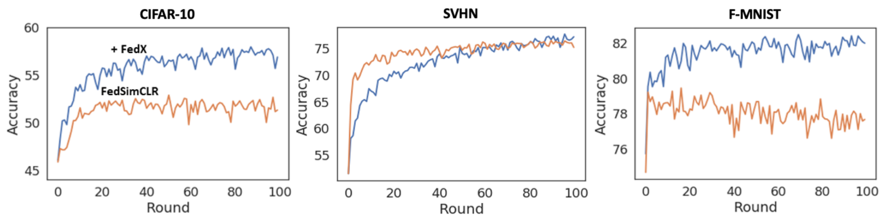
**Comparison with other relevant baselines.** We compared FedX with other contrastive learning methods: FedCA (Zhang *et al.*)[6] and MOON (Li *et al.*). We followed the same implementation guidelines as in the original work, only substituting the InfoNCE objective for the supervised loss in MOON (calling this MOON-unsup). FedSimCLR with FedAvg served as the base framework in both cases. Our method continues to outperform, as shown in Table 6, demonstrat-

[6] Note that the dictionary module has been removed from FedCA for fair comparison as it directly shares the local data information of all clients.

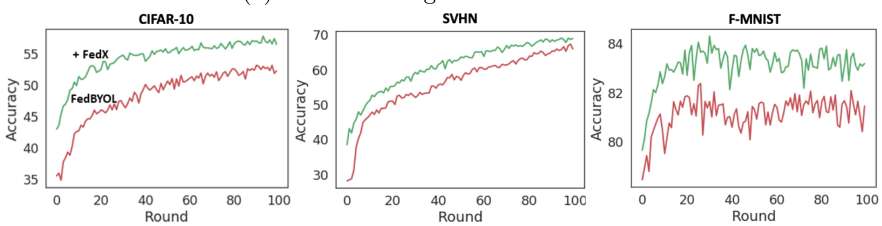
ing the benefit of the proposed knowledge distillation strategy in unsupervised federated learning.

**Performance on ImageNet.** To verify our model's applicability to the large-scale dataset, we run FedX on ImageNet-10 benchmark, a 10-class subset of ImageNet. FedX was trained for 10 local epochs in each communication round, with a total of 50 rounds. Table 7 shows that adding FedX to FedSimCLR improves the classification accuracy by 5pp. Extended results for the larger number of the class will also be released.

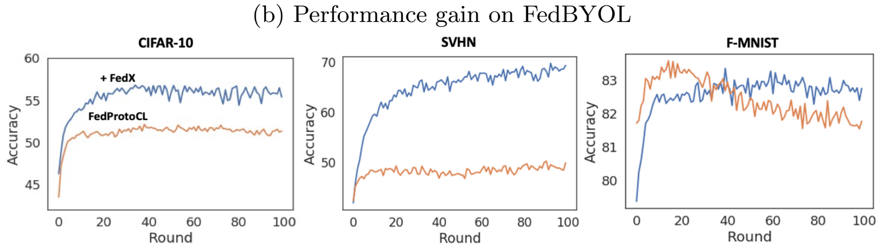
**Full comparison results over communication rounds.** FedX brings meaningful performance improvements than when using the baseline algorithms alone. Figure 7 shows trajectories including three additional baselines – FedProtoCL, FedMoCo, and FedU that were omitted in Fig. 4. We can check how quickly the model benefits baselines over the varying communication rounds. Especially, in FedSimCLR and FedProtoCL (F-MNIST) experiments, FedX prevents the local bias degrading the performance during the entire training phase and thereby stops such deterioration in performance.



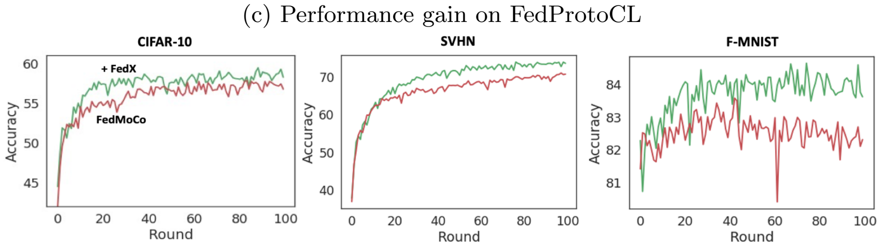

(a) Performance gain on FedSimCLR



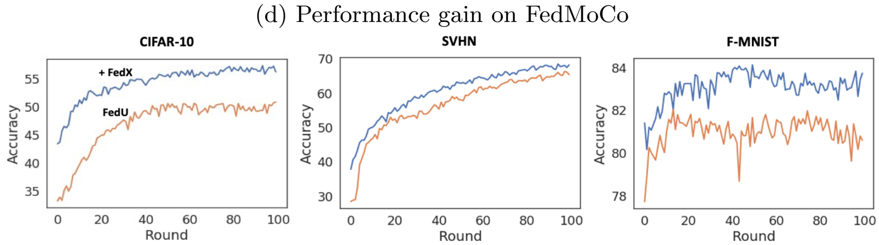

(b) Performance gain on FedBYOL




(c) Performance gain on FedProtoCL




(d) Performance gain on FedMoCo




(e) Performance gain on FedU

Fig. 7: Performance comparison between all baselines (i.e., FedSimCLR, FedBYOL, FedProtoCL, FedMoCo, and FedU) and FedX-enhanced versions over communication rounds. FedX brings extra performance on baseline unsupervised learning models in three benchmark datasets.