A. Image reconstruction from style vectors

To evaluate the safety of the style vectors, we train a generator from a SOTA GAN [32] to reconstruct the image from its style vector. We train the generator until the validation loss converges sufficiently. The best model is selected with the highest validation average PSNR. The results are shown in Figure 5. We consider three scenarios:

- First, intra-client reconstruction (train and test on data from the same client). Note that this scenario is impossible in FL unless the client has already leaked their data. This is an extreme case to examine the best results the generator can achieve.

  In Figure 5a, the diagonal image pairs show intra-client reconstruction results on PACS, and the bottom two pairs show intra-client reconstruction results on Camelyon17. We can see that the generator fails on intra-client reconstruction on the Camelyon17 dataset. For PACS, although the generator is possible to overfit the data within a single domain, this is only vulnerable when a large amount of data is leaked.

- Second, inter-client reconstruction (malicious client). It is possible that there exists a malicious client who wants to use its own data to reconstruct the images of other clients from the shared style vectors. From the results, we can hardly infer any content information except for overall color. Although (P, Photo) shows a rough shape of GT, it belongs to the intra-client scenario, which violates the FL setting.

  In Figure 5a, image pairs that are not lie at diagonal line are inter-client reconstruction results. The figure shows that the inter-client reconstruction fails on both PACS and Camelyon17 datasets.

- Third, third-party reconstruction (pre-trained on large-scale images). More generally, if an outside attacker has compromised the style vectors and wants to reconstruct the images from the shared style vectors, they can train the reconstructor on a large-scale image dataset.

  We train the generator on ImageNet and visualize the reconstruction results in Figure 5b. According to the results, the pre-trained generator totally fails to reconstruct the target images.

Therefore, in the real FL scenarios, one can hardly reconstruct the original images merely from the shared style vectors.

B. Time cost of extra computation

The extra computation time cost of our method is very low. Specifically, for the overall style computation, it takes 7 seconds for 2048 images with 256×256 resolution. For image stylization, it takes 54 seconds to stylize 2048 images of 256×256 resolution under either "Overall, K=3" or "Single, K=3" mode. The results are tested on an NVIDIA RTX 2080Ti GPU using PyTorch 1.11.0 with CUDA11.

Setting	Unseen client				Average
	P	A	C	S	
FedAvg (AISTATS'17) [34]	95.21	82.91	78.80	73.99	82.73
Jigen (CVPR'19) [4]	95.63	83.25	81.10	71.95	82.98
RSC (ECCV'20) [17]	94.55	83.20	79.99	72.79	85.31
MixStyle (ICLR'21) [47]	96.47	86.89	81.06	76.81	82.63
FedDG (CVPR'21) [33]	95.93	84.28	79.44	73.89	83.89
CCST (Overall,K=3)	**96.65**	**88.33**	**78.20**	**82.90**	**86.52**

Table 3: Compare the results of our CCST (Overall, K=3) with baselines that are trained with local iterations=3.

C. Training budget

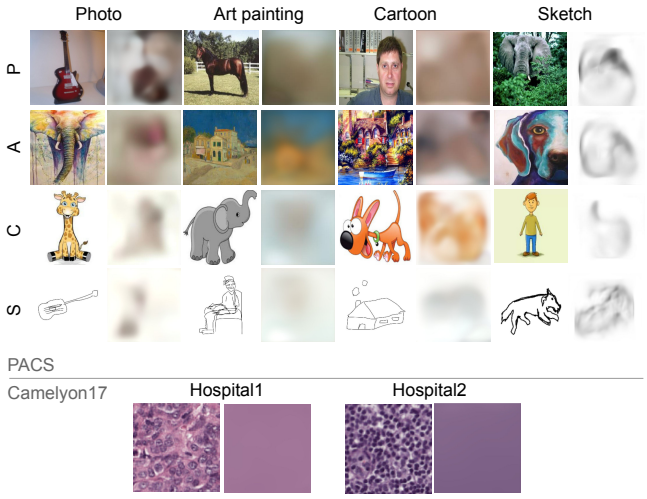
To be fairer in the training budget, we increase the local training iterations of baselines methods from 1 to 3 to compare with our overall (K=3) method. The results are shown in Table 3. According to the results, more local iterations do not lead to obvious accuracy improvement for baseline methods, and our CCST (Overall, K=3) still outperforms all the baseline methods.

D. Visualization of the FFT amplitude exchange results on the PACS

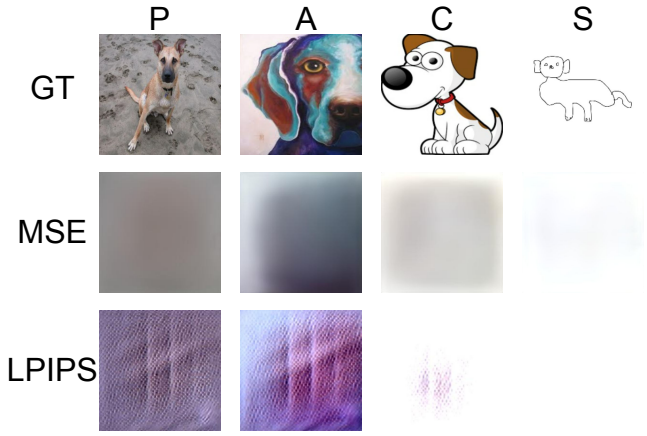
As shown in Figure 6, we visualize the results after the FFT amplitude exchange using single and overall amplitude on the PACS dataset [24]. We can see that the FFT amplitude exchange does not make a noticeable change to appearance or artistic style but only adds to some spatial repetitive texture and color patterns. This could be one of the reasons why FFT cannot outperform our CCST method. Because in the PACS dataset, we have large domain gaps such as that between photos and sketches. Simple changes in color, brightness, or background texture cannot make up the gap very well, while AdaIN style transfer can perform better by producing visually plausible artistic style transfer.

E. AdaIN style transfer vs FFT amplitude exchange

In FedDG [33], the amplitude information in the frequency space of an image can also serve as a kind of style, while we utilize the IN statistics of each feature channel as style information. To explore the differences between the FFT [35] amplitude and IN statistics as style, we made a thorough comparison under our augmentation framework. The amplitude exchange alone without episodic learning in FedDG is equivalent to our framework with the setting of the single style when K=1.



(a) Intra and inter domain reconstruction.



(b) ImageNet pre-trained reconstruction.

Figure 5: **(a)** Image reconstruction from the style vectors. (Ground truth, reconstructed image) pairs are shown. For PACS, the x-axis represents which domain the generator is trained on, the y-axis represents which domain the input style vectors are from. Diagonal image pairs show intra-client results. For Camelyon17, we show the intra-client reconstruction results only. The generator is trained with MSE loss. **(b)** We utilize the ImageNet pre-trained reconstructor to recover the PACS images from their style vectors. From top to bottom rows are the ground truth images, reconstruction results using generator trained with MSE loss, and reconstruction results using generator trained with LPIPS loss.

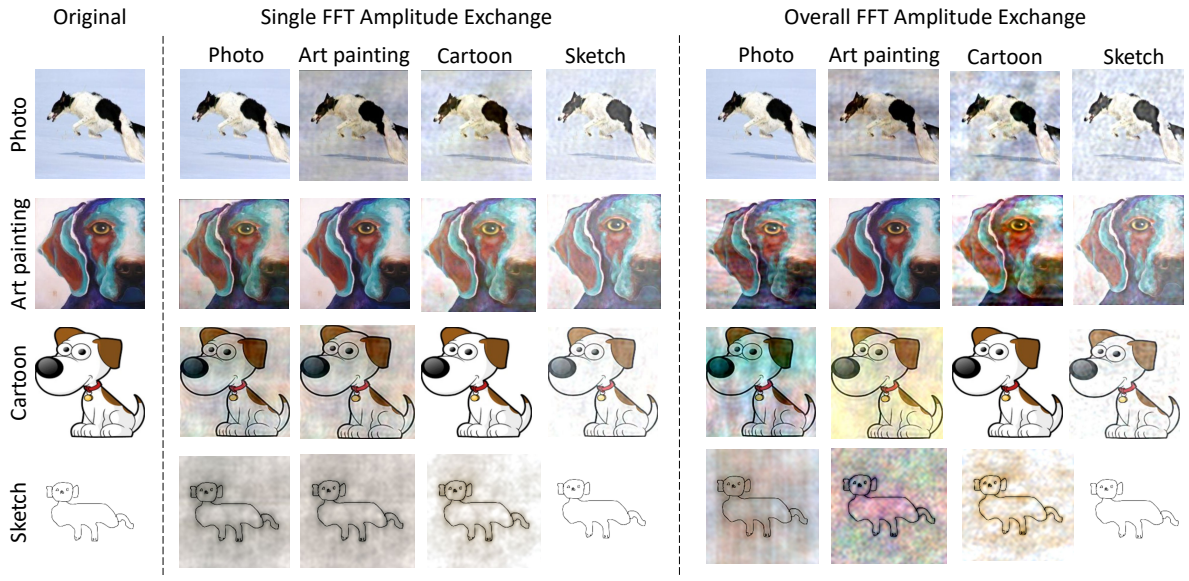


Figure 6: Visualization of images after the FFT amplitude exchange on the PACS dataset. Similar with Figure 7, we duplicate the content image if it is the same as the amplitude target image.

We show the comparison results in Table 4. Compared with our proposed method (using IN statistics as style), the FFT-based amplitude exchange method consistently performs worse under the same setting except for the setting of single style when K=3. Moreover, for the FFT-based style, the overall amplitude [1] of a domain fails to result in better results than a single image FFT amplitude. In contrast, our framework has a significant boost when using overall style.

[1] We compute the overall amplitude by averaging the amplitudes of all images in this domain.

Table 4: Comparison between using FFT amplitude and IN statistics as style in our proposed cross-client style transfer framework with single and overall style under different K values. This table reports the performance with varying hyper-parameters of our framework on the PACS dataset with ResNet50 as the backbone.

Method	CCST (Ours)					FFT Amplitude Exchange				
	P	A	C	S	Avg.	P	A	C	S	Avg.
Single(K=1)	95.75	87.5	74.66	76.56	83.62	96.71	85.69	76.19	73.76	83.09
Single(K=2)	**96.77**	86.23	75.73	80.12	84.71	96.89	87.16	79.31	74.32	84.42
Single(K=3)	96.65	86.63	74.53	81.85	84.84	96.65	**86.87**	**79.74**	77.3	**85.14**
Overall(K=1)	95.69	86.67	75.85	77.37	83.90	96.89	86.38	78.92	72.36	83.64
Overall(K=2)	96.41	**88.72**	78.03	80.91	86.02	93.95	79.79	72.18	77.96	80.97
Overall(K=3)	96.65	88.33	**78.20**	**82.90**	**86.52**	95.21	81.25	73.34	**80.27**	82.52

With the help of our framework, the best result (single, k=3) of the FFT-based method can have a 2% improvement compared with the original version in FedDG (single, K=1). In general, our second-best result (overall, K=2) outperforms the best result of FFT amplitude exchange (single, K=3) by 0.9%; our best result (overall, K=3) outperforms the best result of FFT amplitude exchange (single, K=3) by 1.4%.

The experiments show that it is only practical to use the single image amplitude for the FFT amplitude exchange method. Utilizing the single image style mode makes the communication cost high due to the uploading and downloading of the style bank, leading to inflexibility. However, our CCST method can flexibly choose between single image style and overall domain style accordingly, especially the choice of using overall domain style to decrease the communication cost.

F. PACS visualization

Figure 7 shows the visual results of cross-client style transfer with two types of styles. The overall domain style represents a more general and accurate client style, while the single image style brings more randomness.

G. Visualization of style transfer results on the Office-Home

In this section, we show the qualitative results by visualizing images before and after the AdaIN [16]-based style transfer. In Figure 8, we show images of four different target domains in the Office-Home dataset [39]. Except for the art domain, samples from the other three domains show less domain gap. For each domain, we visualize the generated images using both random single image style and overall domain style. According to our experiment results, the overall style is usually more effective than using the single image style. Random single image style sometimes may choose an image that is not representative for the whole domain. For example, in Figure 8, when transferring the clock image with the Clipart style into real-world style, the stylized image with overall style has a more colorful and representative style than that using random single image style.

H. Additional experimental results

We show the results of our CCST with ResNet [13] as the backbone network on the PACS and Office-Home dataset in Table 5. For PACS dataset, using ResNet18 (Table 5) and ResNet50 (Table 2a) as backbone have consistent results: the overall style with K=3 leads to the best performance. When using ResNet18 as the backbone, the improvement upon baseline is more significant than that of using ResNet50.

For CCST results on the Office-Home dataset, besides $K = 1$, all other settings outperform the FedAvg in terms of the average accuracy. To explore the reason for the failure of CCST with $K = 1$, we visualize the images with style transfer as shown in Figure 8. From the visualization results, we can observe that the domain shift among domains of the Office-Home is smaller than that of the PACS dataset. For example, the product images collected on websites are similar to the real-world object images taken by a regular camera. Due to the slight differences between different domain styles and the randomness in single image style transfer, Single(K=1) achieves a lower accuracy than FedAvg on average. However, the overall domain style still shows a stronger representation capability and only has a minor performance gap compared with FedAvg. Overall, our best CCST results outperform the FedAvg baseline even with less domain shift in the Office-Home dataset.

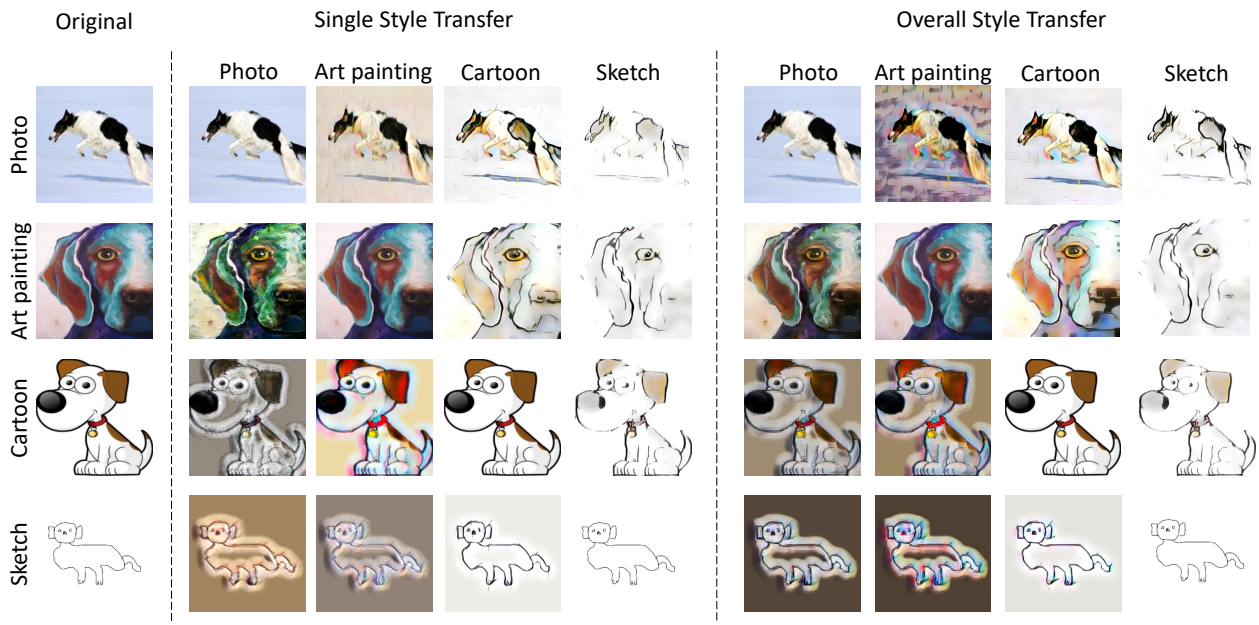


Figure 7: Visualization of stylized images on PACS. Note that if the content image is from the same domain of the style statistics, we directly copy the content image to the augmented dataset instead of transferring the same style to it.

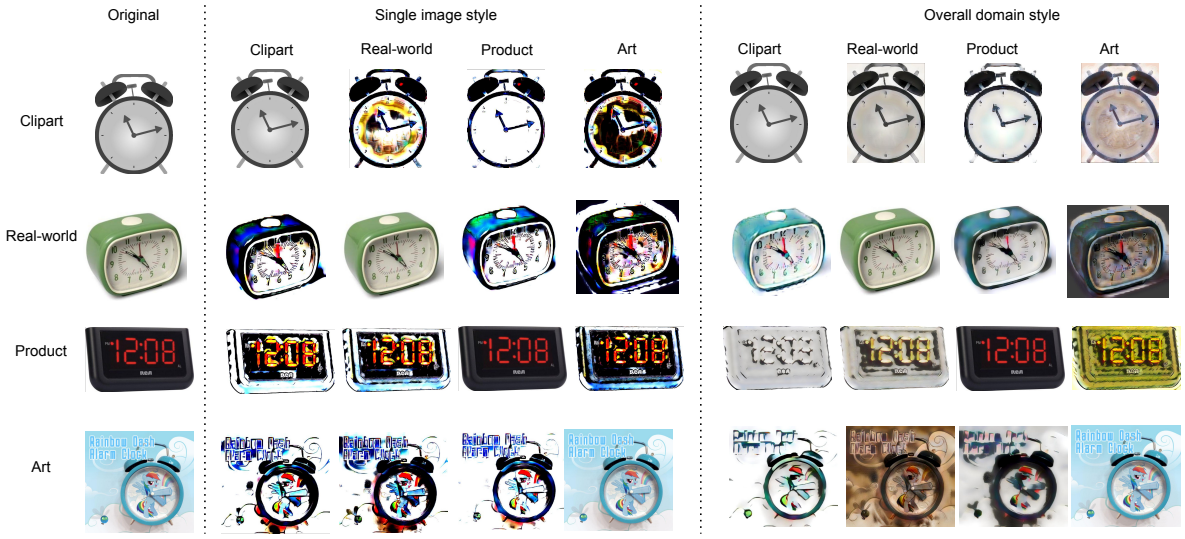


Figure 8: Visualization of the AdaIN stylized images on the OfficeHome dataset. Note that if the content image is from the same domain as that of style statistics, we directly copy the content image to the augmented dataset instead of transferring the same style to it.

Table 5: Results of our CCST with different image style types and K values under PACS and Office-Home dataset. The backbone network is ResNet18. Each column represents a single unseen target client.

Method	PACS					Office-Home				
	P	A	C	S	Avg.	A	C	P	R	Avg.
FedAvg [34]	91.44	75.98	73.21	61.08	75.43	60.08	45.59	69.48	**72.82**	61.99
Single(K=1)	94.07	77.73	70.99	72.82	78.90	55.14	43.64	68.58	68.92	59.07
Single(K=2)	**95.27**	79.05	72.82	77.88	81.26	57.61	48.68	71.17	71.44	62.23
Single(K=3)	94.79	80.27	71.72	**80.86**	81.91	58.44	45.70	72.30	71.56	62.00
Overall(K=1)	94.19	79.88	72.14	75.41	80.41	59.47	47.88	67.91	70.87	61.53
Overall(K=2)	93.95	79.79	72.18	77.96	80.97	57.82	**50.52**	71.28	70.99	62.65
Overall(K=3)	95.21	**81.25**	**73.34**	80.27	**82.52**	59.05	50.06	**72.97**	71.67	**63.44**