Guiding Diffusion Models with Fine-Grained Conditions and Semantics-Preserving Sampling for One-Shot Federated Learning

Supplementary Material

A. Algorithm

To provide a comprehensive description of our proposed method, we detail the complete procedures for the client-side and server-side processing of `Espresso` in Algorithm 1. It first outlines the parallel tasks performed by each client: (1) Fine-grained Condition Learning, which involves feature extraction, intra-category clustering, and the optimization of a conditional embedding for each cluster; and (2) GMM of Latent Noises, where clients model the distribution of the final-timestep latent noises using GMM. Then, it describes the subsequent server-side process, which includes generating the synthetic dataset using learned conditional embeddings, sampled initial noise from GMMs and Z-Sampling strategy. Finally, the server trains the global model on the synthetic dataset.

B. More Experimental Details

B.1. Datasets

For feature-skew settings, we select three challenging benchmark datasets. We utilize **DomainNet** [6] and **PACS** [4], which are prominent multi-domain classification datasets that primarily exhibit style-skew, where the same object category appears in different artistic styles such as photo, sketch, and cartoon. Following [1], the selected categories from DomainNet are *airplane*, *axe*, *basketball*, *bicycle*, *bird*, *bracelet*, *clock*, *flower*, *pizza* and *strawberry*. Additionally, we use **Common NICO++** [10] to simulate context-skew settings, a common real-world challenge where the object's background or environment varies significantly (e.g., an animal "on grass" versus "in water"). For Common NICO++, 10 categories are randomly selected for our experiments, including *bear*, *bus*, *cat*, *cow*, *football*, *kangaroo*, *pumpkin*, *rabbit*, *ship* and *wolf*. For all datasets, although we construct the training dataset for each client in a 16-shot manner for every selected category, we utilize the full test datasets corresponding to these categories for evaluation.

To evaluate the performance under label-skew settings, where label distributions differ across clients, we employ three datasets: (1) **UCMerced Land-Use Dataset** (UCMerced) [8], a 21-category satellite imagery dataset; and (2) **Unique NICO++** [10], a dataset designed for testing fine-grained generalization by providing object categories paired with a wide range of domains, where many domains are unique to a specific category. For all label-skew experiments, we simulate a FL system of 5 clients. For UCMerced, we partition its training dataset among the clients using Dirichlet distribution, a standard approach to simulate label-skew settings following [5]. To evaluate the model's performance under varying degrees of heterogeneity, we conduct experiments with three different concentration parameters *D(5.0)* (mildly heterogeneous), *D(0.5)* (moderately heterogeneous), and *D(0.01)*(severely heterogeneous). For the more challenging Unique NICO++, we construct a highly heterogeneous scenario that combines both label-skew and fine-grained feature-skew. Each of the 5 clients is randomly assigned a disjoint set of 5 object categories. Furthermore, for each assigned category, the client receives data from 4 randomly selected unique domains. This setup creates a complex environment where each client possesses a small, unique slice of the overall data distribution. The details of data partition for Unique NICO++ are in Tab. 3.

B.2. More Implementation Details

Hardware and Framework: All experiments are conducted on a server equipped with four NVIDIA RTX 4090 GPUs (24 GB) using PyTorch.

Traditional FL Methods: For FedAvg and FedProx, 50 communication rounds are performed. Local training utilizes an SGD optimizer with a learning rate of 0.01, momentum of 0.9, weight decay of 0.0005, and a batch size of 64.

Local Training of OSFL methods: For FedLMG, Fens and DENSE, the local models are trained for 50 epochs using the same SGD configuration as traditional FL. For local training in FedDEO, FedBiP, and `Espresso`, learnable parameters are trained for 100 epochs using an AdamW optimizer (learning rate=0.005, weight decay=0.01) with a batch size of 1 and 8 gradient accumulation steps.

Text Prompts: The diffusion-based methods that require text-like condition information guide the diffusion model with text prompts derived from templates shown in Tab. 4. Prompts-only, FedDEO, and FedLMG directly fill real domain and category names in these templates, while FedBiP fills these templates with its learned concepts.

Guidance and Initialization: The guidance scale is set to 7.0 for Prompts-only and `Espresso`, while other baselines use scales specified in their respective original papers. The learnable descriptors in FedDEO and the conditional embeddings in `Espresso` are initialized using text embeddings from the corresponding filled templates.

Other Configurations for Specific Baselines: FedD3

Algorithm 1 `Espresso`

function CLIENTUPDATE
  Input: Local client dataset $\mathcal{D}_k$, pre-trained vision model Φ, pre-trained diffusion model ϵ_θ
  for each class $y \in \mathcal{D}_k$ **do**
    ▷*1. Fine-grained Condition Learning*
    Extract features $\{f_i | y_i = y\}$ for all $x_i \in \mathcal{D}_{k,y}$
    Perform clustering on $\{f_i | y_i = y\}$ to obtain clusters $\{\mathcal{D}_{k,y}^j\}_{j=1}^J$
    for each cluster $\mathcal{D}_{k,y}^j$ **do**
      Initialize conditional embedding $c_{k,y}^j$
      Optimize $c_{k,y}^j$ using Eq. (8)
    end for
    ▷*2. GMM of Latent Noises*
    Generate a set of MixUp latent vectors $\{z_{k,y,\text{mix}}^0\}$ from all $x_i \in \mathcal{D}_{k,y}$
    Forward Diffusion from $\{z_{k,y,\text{mix}}^0\}$ to $\{z_{k,y}^T\}$ using Eq. (3)
    Fit a GMM $p(z_{k,y}^T; \{w_{k,y}^{T,m}, \mu_{k,y}^{T,m}, \Sigma_{k,y}^{T,m}\}_{m=1}^M)$ for $\{z_{k,y}^T\}$
  end for
  return $\bigcup_{y=1}^Y \left[\bigcup_{j=1}^J \{c_{k,y}^j\}\right]$ and parameters of GMMs $\bigcup_{y=1}^Y \left[\{w_{k,y}^{T,m}, \mu_{k,y}^{T,m}, \Sigma_{k,y}^{T,m}\}_{m=1}^M\right]$
end function
function SERVERPROCESSING
  Input: Client conditional embeddings $\bigcup_{k=1}^K \left\{\bigcup_{y=1}^Y \left[\bigcup_{j=1}^J \{c_{k,y}^j\}\right]\right\}$, parameters of all GMMs from clients $\bigcup_{k=1}^K \left\{\bigcup_{y=1}^Y \left[\{w_{k,y}^{T,m}, \mu_{k,y}^{T,m}, \Sigma_{k,y}^{T,m}\}_{m=1}^M\right]\right\}$
  ▷*1. Construct Synthetic Datasets*
  Initialize synthetic dataset $\mathcal{D}' \leftarrow \emptyset$
  for client $k = 1$ to K **do**
    Initialize client-specific synthetic dataset $\mathcal{D}_k' \leftarrow \emptyset$
    for category $y = 1$ to Y **do**
      Randomly select conditional embedding $c \in \{c_{k,y}^j\}_{j=1}^J$
      Sample an initial noise $z^T \sim p(z_{k,y}^T; \{w_{k,y}^{T,m}, \mu_{k,y}^{T,m}, \Sigma_{k,y}^{T,m}\}_{m=1}^M)$
      ▷*Z-sampling*
      for $t = T$ down to threshold T_z **do**
        Predict $\epsilon_1 \leftarrow \hat{\epsilon}_\theta(z^t, t, c)$ with γ_1 using Eq. (11)
        Compute $\tilde{z}^{t-1}$ using Eq. (4) with ϵ_1
        Predict $\epsilon_2 \leftarrow \hat{\epsilon}_\theta(\tilde{z}^{t-1}, t-1, c)$ with γ_2 using Eq. (11)
        Compute $\tilde{z}^t$ by inverting $\tilde{z}^{t-1}$ using Eq. (6) with ϵ_2
        Predict $\epsilon_3 \leftarrow \hat{\epsilon}_\theta(\tilde{z}^t, t, c)$ with γ_1 using Eq. (11)
        Compute z^{t-1} using Eq. (4) with ϵ_3
      end for
      for $t = T_z$ down to 1 **do**
        Predict $\epsilon_1 \leftarrow \hat{\epsilon}_\theta(z^t, t, c)$ with γ_1 using Eq. (11)
        Compute z^{t-1} using Eq. (4) with ϵ_1
      end for
      Add $(\mathcal{E}^{-1}(z^0), y)$ to $\mathcal{D}_k'$
    end for
    Add $\mathcal{D}_k'$ to $\mathcal{D}'$
  end for
  ▷*3. Train Global Model*
  Initialize global model parameters ω
  Optimize ω by Eq. (12)
  return trained global model parameters ω
end function

Table 3. Example of the data partition for Unique NICO++ under the label-skew setting. Each of the 5 clients is assigned 5 disjoint categories, and each category is assigned 4 unique domains.

Client	Categories	Unique Domains per Category
Client 1	frog	jumping, on lotus leaf, in mud, chocolate
	tortoise	on hand, green, eating earthworms, in net
	chair	lying, wooden, in classroom, circle
	bear	roaring, sitting, lying, wombat
	mailbox	green, colimnar, with flag, red
Client 2	gun	firing, air rifle, on armrack, long-barrelled
	tiger	with chain, passing the ring of fire, roaring, with shade
	umbrella	folded, in sunlight, on stand, folding
	sailboat	barque, sloop, across bridge, ketch
	lion	preying on, preying on hippo, sleeping, in cave
Client 3	seal	spotted, in aquarium, playing with ball, standing
	dolphin	black, playing with ball, through ring, white
	squirrel	lying, carrying cone, fat, eating
	spider	specimen, white, hairy, in spider web
	elephant	white elephant, baby. elephant, aside people, sleeping
Client 4	fishing rod	on hand, on railing, blue, with winding wheel
	owl	preying, sleeping, in cave, on shoulder
	racket	racket in front of face, broken wire, in bag, broken
	truck	repairing, carrying contrainer, yellow, out of tunnel
	pumpkin	holloween, green, on hand, with leaf
Client 5	butterfly	on mask, on hand, on rope, swallowtail butterfly
	bus	in gas station, trolley buses, articulated buses, aside station
	motorcycle	red, aside traffic light, on track, with container
	giraffe	in cave, white, sitting, being fed
	corn	with leaf, colorful, red, eating

Table 4. Text prompt templates used for guiding the diffusion model across different datasets. '[CLS]' is replaced by the specific category name and '[DOM]' by the specific domain name during generation.

Dataset	Text Template
DomainNet	a '[DOM]' of '[CLS]'
PACS	a '[DOM]' of '[CLS]'
Common NICO++	a '[CLS]' in '[DOM]'
UCMerced	a centered satellite photo of '[CLS]'
Unique NICO++	an image of '[DOM]' '[CLS]'

is configured to distill exactly one image per category. DENSE trains its global model via knowledge distillation for 50 epochs.

MixUp Strategy of `Espresso`: The MixUp is performed in the latent space on 2 to 4 randomly selected latent vectors. Mixing ratios are sampled from a Beta distribution with parameter 3.0 for two vectors, or a Dirichlet distribution with parameter 3.0 for more.

GMM and Z-Sampling of `Espresso`: Mixed latent vectors are diffused to 2 latent noises for GMM modeling, all of which are conducted in the flattened latent space using a diagonal covariance matrix. For Z-Sampling, the guidance scale for DDIMInverse is set to 2.

Global Model Training and Evaluation: For all OSFL methods, the global model is trained for 50 epochs using an SGD optimizer with a learning rate of 0.01, momentum of 0.9, weight decay of 0.0005, and a batch size of 64. All experimental results are reported as the average of 3 independent runs.

Code repository: https://github.com/dengxj26/Espresso.

C. More Experimental Results

C.1. Comparison Results under Label-skew Settings

To evaluate the performance in label-skew settings, we perform experiments on UCMerced and Unique NICO++, with results in Tab.5. On UCMerced, as the data het-

Table 5. Performance comparison UCMerced and Unique NICO++ datasets. All values are accuracy (%). For UCMerced, we report the results under different Dirichlet Distribution settings (*D*(*)). For Unique NICO++, we report the results on client-specific (C*) test datasets and their average. The best and second performance in each column (excluding the theoretical upper-bound *Ceiling*) is in **bold** and underline.

	UCMerced			Unique NICO++					
Methods	*D(5.0)*	*D(0.5)*	*D(0.01)*	C1	C2	C3	C4	C5	Avg.
Ceiling	93.86	93.86	93.86	92.17	93.16	93.91	87.77	93.49	92.10
FedAvg	81.43	79.05	71.67	79.11	81.37	78.21	80.22	80.47	79.88
FedProx	80.48	80.00	73.33	78.59	79.85	79.49	80.94	81.07	79.99
DENSE	74.22	72.85	69.43	60.18	64.83	70.00	70.50	65.98	66.30
FedD3	67.41	65.83	61.59	60.67	62.02	60.10	59.37	61.25	60.68
Fens	80.24	78.81	74.05	-	-	-	-	-	-
Prompts-only	34.05	34.05	34.05	75.46	79.85	72.76	73.74	85.50	77.46
FedLMG	62.41	59.19	45.42	71.02	81.75	71.47	76.62	78.85	75.94
FedDEO	85.48	84.52	82.14	80.42	86.69	77.24	87.77	91.12	84.65
FedBiP	85.71	83.29	80.57	**88.25**	87.83	78.53	83.09	**91.43**	85.83
Espresso	**88.81**	**88.33**	**87.86**	86.68	**89.73**	**85.58**	**88.85**	89.65	**88.10**

erogeneity increases from *D(5.0)* to *D(0.01)*, traditional FL and non-diffusion OSFL methods such as FedAvg and Fens suffer performance degradation. In contrast, recent diffusion-based methods, particularly Espresso, demonstrate strong robustness by reconstructing balanced synthetic datasets on the server, maintaining high accuracy even under extreme label-skew setting. The poor performance of Prompts-only and FedLMG may be attributed to a semantic mismatch between the diffusion model's understanding of the simple prompted category labels and the actual visual concepts in the data. In the more challenging Unique NICO++ setting, where each client has disjoint categories, Espresso still achieves the best average performance. Notably, for Fens, since there are no overlapping categories between clients, the ensemble aggregator has no shared parameters across clients, preventing it from converging and learning shared knowledge. This highlights an advantage of diffusion-based OSFL methods like Espresso, which can create a unified training dataset and learn cohesive global models even in a fully partitioned data environment.

Method	Rounds	Comm(MB)	Time(h)	Acc(%)
FedAvg	50	2233.00	0.06	66.90
FedDEO	1	2.33	4.53	70.81
FedBiP	1	10.04	4.56	72.81
LoRA(r=16)	1	122.00	5.73	75.87
Espresso	1	9.64	6.32	75.32

Table 6. Comparison of communication cost, server computation time and accuracy on DomainNet.

C.2. Communication and Computation Efficiency

To further evaluate the efficiency of Espresso, we analyze the communication overhead, server-side computation time, and model accuracy on DomainNet. Other than existing OSFL methods including FedDEO and FedBiP, we also introduce a Parameter-Efficient Fine-Tuning (PEFT) baseline using LoRA ($r = 16$) [2] to fit each domain and each category separately. The results are summarized in Tab. 6. As an OSFL method, Espresso significantly reduces communication costs by approximately 95% compared to FedAvg. Furthermore, Espresso incurs lower communication overhead than FedBiP and is substantially more efficient than the LoRA baseline, while achieving comparable performance. For computation, although Espresso requires more server-side time due to Z-Sampling strategy, it effectively trades computation for significant performance improvement.

C.3. More Ablation Study

To provide a further understanding of Espresso, we conduct additional ablation studies focusing on the independent contributions of modules, the choice of feature extractors, and the impact of different clustering algorithms. As shown in Tab. 6, both G and Z modules consistently improve performance over Prompts-only across most datasets. Their combination (G+Z) further enhances results, demonstrating that these components provide significant benefits.

For different choices of feature extractors, we evaluate the stability of Espresso by employing various pretrained vision encoders Φ for feature extraction, including DinoV2, CLIP, and ResNet-152. The results in Fig. 7 shows

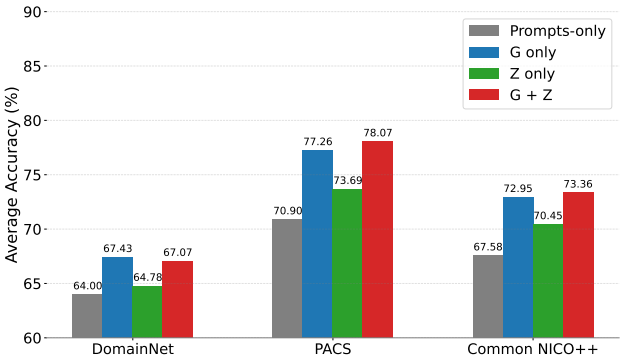


Figure 6. More Ablation study for independence of GMM of Latent Noises (G), and Z-Sampling (Z). The Prompts-only baseline represents data generation without any of our proposed components.

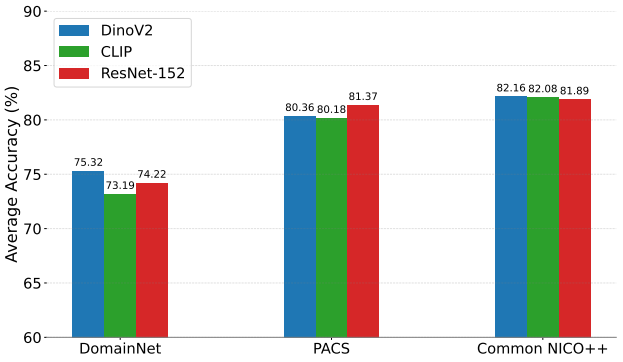


Figure 7. More Ablation study for different choices of feature extractor Φ.

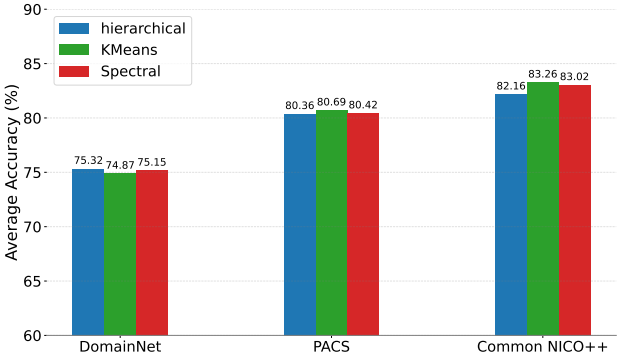
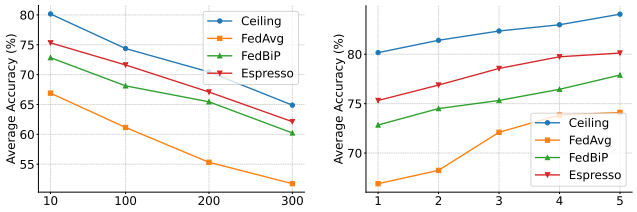


Figure 8. More Ablation study for different choices of clustering method.

that Espresso maintains stable performance regardless of the feature extractor used. While DinoV2 generally yields better results on DomainNet and Common NICO++, the performance gap remains small, indicating the compatibility of Espresso with different feature extractors.

While we use hierarchical clustering to identify intra-category patterns in most of our experiments, we also investigate its adaptability to other clustering methods including KMeans and Spectral clustering. As reported in Fig. 8, the performance of the global model remains consistent across them. This suggests that Espresso is robust to the specific implementation of fine-grained pattern discovery, as long as intra-category patterns can be captured.



(a) Number of labels (b) Number of clients per domain

Figure 9. Analysis of the scalability

C.4. Scalability Analysis

To evaluate the scalability of Espresso, we conduct experiments with different sizes of the label space and different number of participating clients per domain. For the size of label space, we use subsets of the DomainNet with varying numbers of categories. As illustrated in Fig. 9(a), the performance of Espresso and baselines consistently degrade as the number of labels increases, due to the inherent difficulty of learning large label spaces. However, the results still demonstrate that Espresso remains effective even as the global classification task becomes more complex. Moreover, following the settings of existing works [1], we investigate scenarios where the data of a single domain is distributed across multiple clients. The results in Fig. 9(b) shows that Espresso can effectively handle an increased number of clients, and outperforms other baselines in settings with different number of clients.

C.5. More Parameter Analysis

We further analyze the sensitivity of Espresso to the Z-Sampling threshold T_z and the volume of synthetic data, with results shown in Fig. 11. For T_z, which activates Z-Sampling for timesteps $t > T_z$ during the denoising process, Espresso achieves better performance when $T_z \in [200, 800]$ across all datasets. This indicates that Z-Sampling is more beneficial during the mid stage of denoising, where it helps establish the rough semantic structure of the image. However, applying it in later stages yields diminishing returns and may even degrade fine details. Regarding the volume of synthetic data, we observe that performance improves as the dataset size increases from $1\times$ to $10\times$ the original data volume. However, a clear diminishing marginal return is evident after $5\times$. This indicates that a

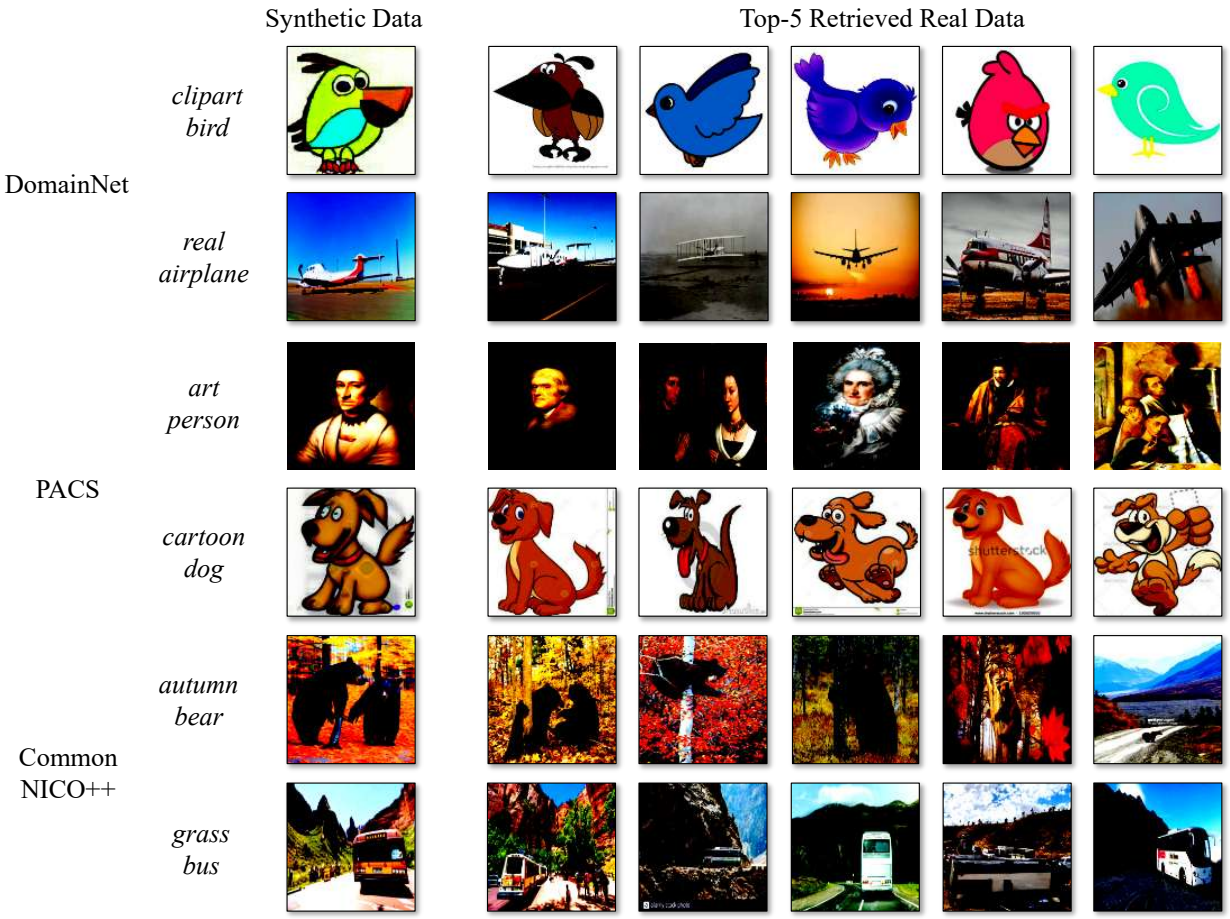
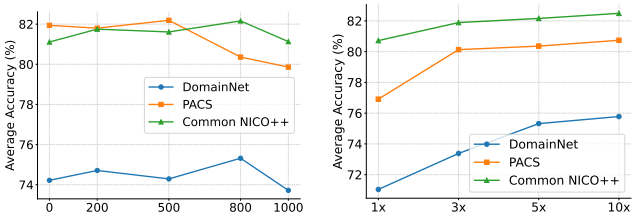


Figure 10. Visual privacy analysis via image retrieval. Each synthetic data (left) is compared against its top-5 most semantically similar neighbors (right) from the original training data.

5× dataset generated by Espresso is largely sufficient to capture the essential diversity and semantic features of the distribution of local data. As the volume further increases, the performance only slightly improves relative to the increase in computational cost.



(a) Different choices of T_z

(b) Different Volumes of Synthetic data

Figure 11. Parameter analysis of more hyperparameters.

C.6. Privacy Analysis

Visual discrepancy between synthetic and real images. A major privacy issue is the risk of data reconstruction, where synthetic images may inadvertently replicate specific samples from clients' local data. To qualitatively assess this risk for Espresso, we conduct a visual privacy analysis via a nearest-neighbor retrieval experiment. For a given synthetic data point, we use the powerful pre-trained DINOv2 [3] vision model to extract semantic features. Based on cosine similarity, we then retrieve the Top-5 most similar images from the corresponding training data with the same domain and category. As illustrated in Fig. 10, the retrieval results confirm that while the synthetic images are semantically and stylistically coherent with the real data, they are visually distinct from any single ground truth sample. The retrieved images share the same category and general style (*e.g.*, *a clipart airplane*), but the synthetic image exhibits unique characteristics in its composition, posture, and specific details. This finding demonstrates that Espresso

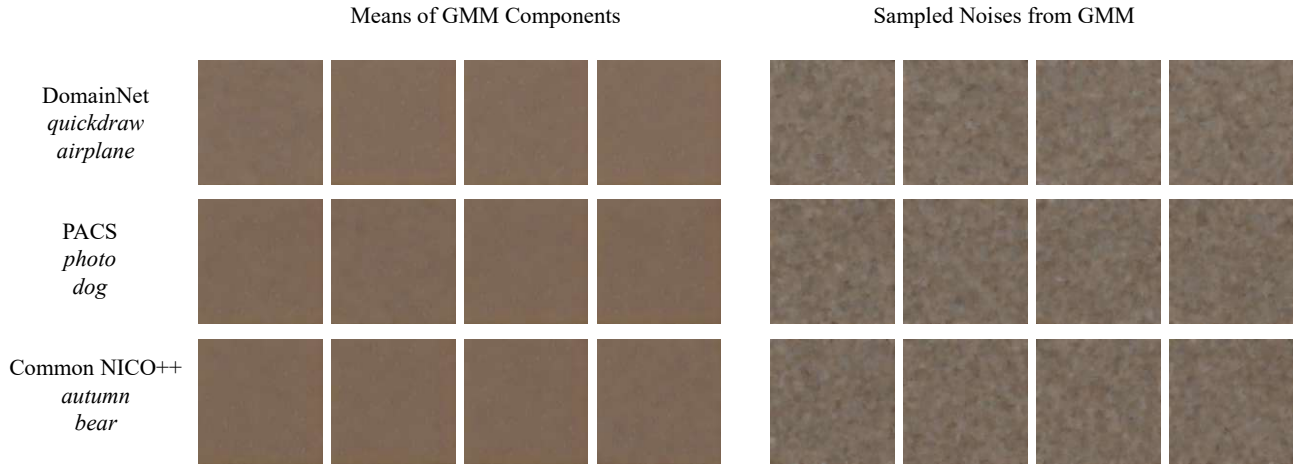


Figure 12. Visualization of the GMM latent space. Images decoded from the GMM component means (left) and from random samples of the distribution (right) appear abstract and noise-like.

does not simply memorize the training data. Instead, it learns a higher-level and abstract representation of the underlying data distribution from the client's guidance information. The diffusion model then samples from this learned distribution, effectively creating novel instances that adhere to the learned style and semantics. Therefore, this qualitative analysis suggests that `Espresso` poses a low risk of direct data leakage, as it generates new, plausible samples rather than reconstructing existing ones.

Visualization of latent noises. To further evaluate the privacy of the uploaded GMM parameters, we analyze the information content of the latent vectors they describe. Fig. 12 presents a visualization of images decoded directly from this latent space using the VAE decoder of the pretrained diffusion model. We visualize two types of samples: the means of the learned GMM components, which represent the central tendencies of the noise distribution, and random samples drawn from these distributions. The resulting images are highly abstract, bearing little to no resemblance to recognizable objects or features from the original local dataset. Even the component means, which capture the most concentrated distributional information, decode into patterns that lack any discernible semantic details. This intuitively indicates that the GMM parameter, as the unique information of the initial noise uploaded by the client, is highly obfuscated. They effectively capture a statistical prior for the generative process without encoding specific and identifiable details of the source data, thus posing a minimal risk of leaking sensitive information about the client's local data.

Membership Inference Attack (MIA). To quantitatively assess privacy, we conduct a Membership Inference Attack (MIA) against `Espresso` and FedAvg, which uses the standard cross-entropy loss of the global model as attack features, following previous works [7, 9]. The results are shown in Tab. 7. By uploading entire model parameters, FedAvg is more vulnerable with attack accuracies, indicating more risk of privacy leakage. In contrast, `Espresso` demonstrates remarkable robustness, with its attack accuracy consistently hovering near the 50% random guess baseline. This superior protection stems from the fact that the uploaded conditional embeddings represent an average over entire data clusters, and the GMM parameters describe a highly obfuscated noise distribution. This compression mechanism effectively dilutes the influence of any single data point, making it significantly more difficult for an attacker to infer membership.

Table 7. Membership Inference Attack (MIA) results. We report the attack accuracy (A), member recall (MR), and member F1-score (MF1). For all metrics, being closer to 50% indicates stronger privacy protection, with 50% representing a random guess (ideal privacy).

Dataset	Method	A(%)	MR(%)	MF1(%)
DomainNet	FedAvg	54.32	58.96	56.91
	`Espresso`	**51.44**	**52.08**	**52.33**
PACS	FedAvg	52.58	45.09	48.85
	`Espresso`	**51.79**	**49.78**	**50.91**
Common NICO++	FedAvg	56.84	60.52	59.16
	`Espresso`	**50.86**	**56.67**	**54.37**

C.7. More Visualization Results

To provide a more direct and intuitive observation of `Espresso`'s generative capabilities, we present additional qualitative results of the synthetic data produced by our

method. Fig. 13, Fig. 14, Fig. 15 and Fig. 16 showcase an array of images generated for the DomainNet, PACS, Common NICO++, and UCMerced.

References

[1] Haokun Chen, Hang Li, Yao Zhang, Jinhe Bi, Gengyuan Zhang, Yueqi Zhang, Philip Torr, Jindong Gu, Denis Krompass, and Volker Tresp. Fedbip: Heterogeneous one-shot federated learning with personalized latent diffusion models. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 30440–30450, 2025. 1, 5

[2] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. ICLR, 1(2):3, 2022. 4

[3] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. Dinov2 meets text: A unified framework for image-and pixel-level vision-language alignment. In Proceedings of the Computer Vision and Pattern Recognition Conference, pages 24905–24916, 2025. 6

[4] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In Proceedings of the IEEE international conference on computer vision, pages 5542–5550, 2017. 1

[5] Qinbin Li, Bingsheng He, and Dawn Song. Model-contrastive federated learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10713–10722, 2021. 1

[6] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In Proceedings of the IEEE/CVF international conference on computer vision, pages 1406–1415, 2019. 1

[7] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. arXiv preprint arXiv:1806.01246, 2018. 7

[8] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems, pages 270–279, 2010. 1

[9] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In 2018 IEEE 31st computer security foundations symposium (CSF), pages 268–282. IEEE, 2018. 7

[10] Xingxuan Zhang, Yue He, Renzhe Xu, Han Yu, Zheyan Shen, and Peng Cui. Nico++: Towards better benchmarking for domain generalization. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 16036–16047, 2023. 1

Figure 13. Visualization of more synthetic data for DomainNet.

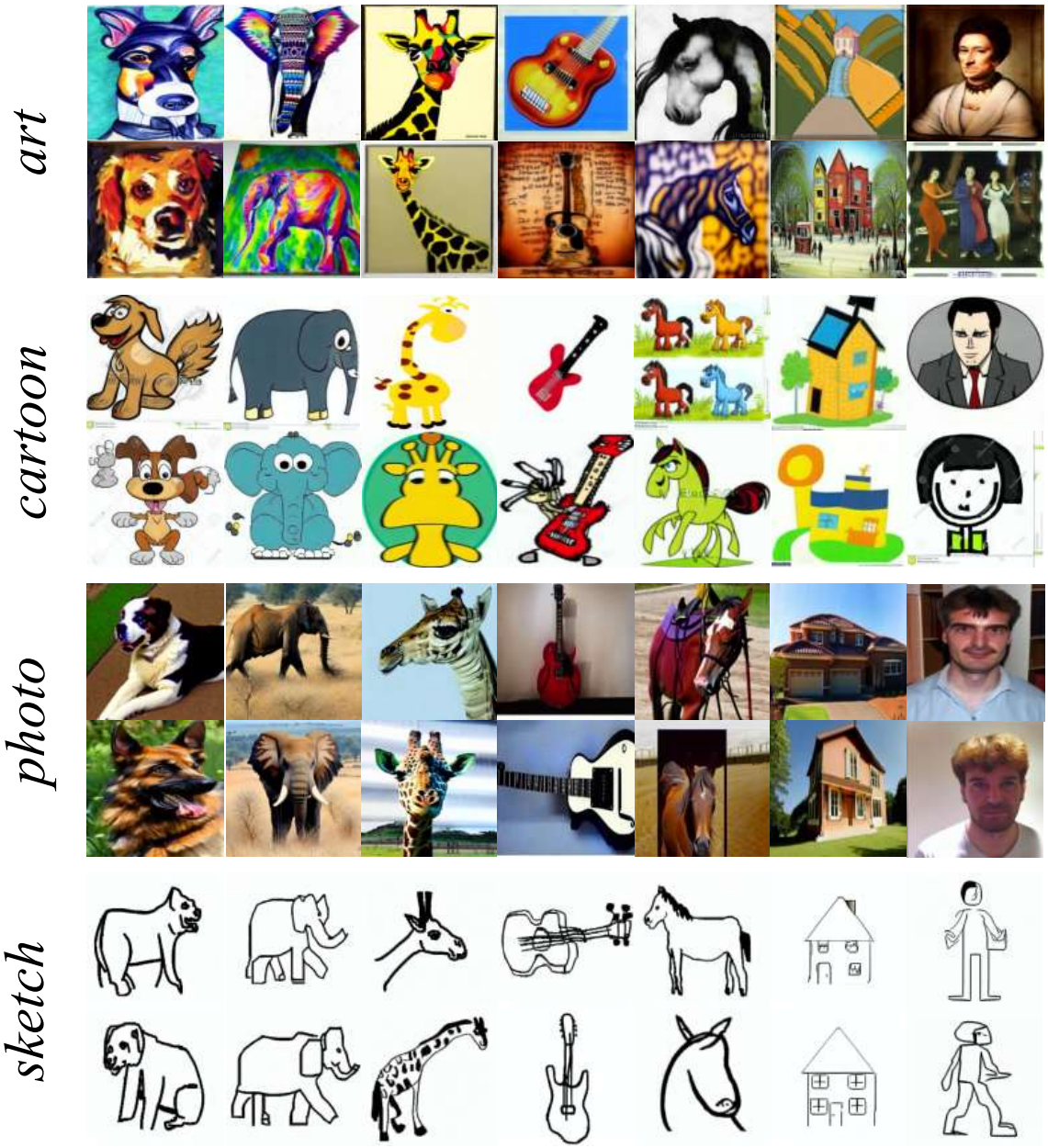


Figure 14. Visualization of more synthetic data for PACS.

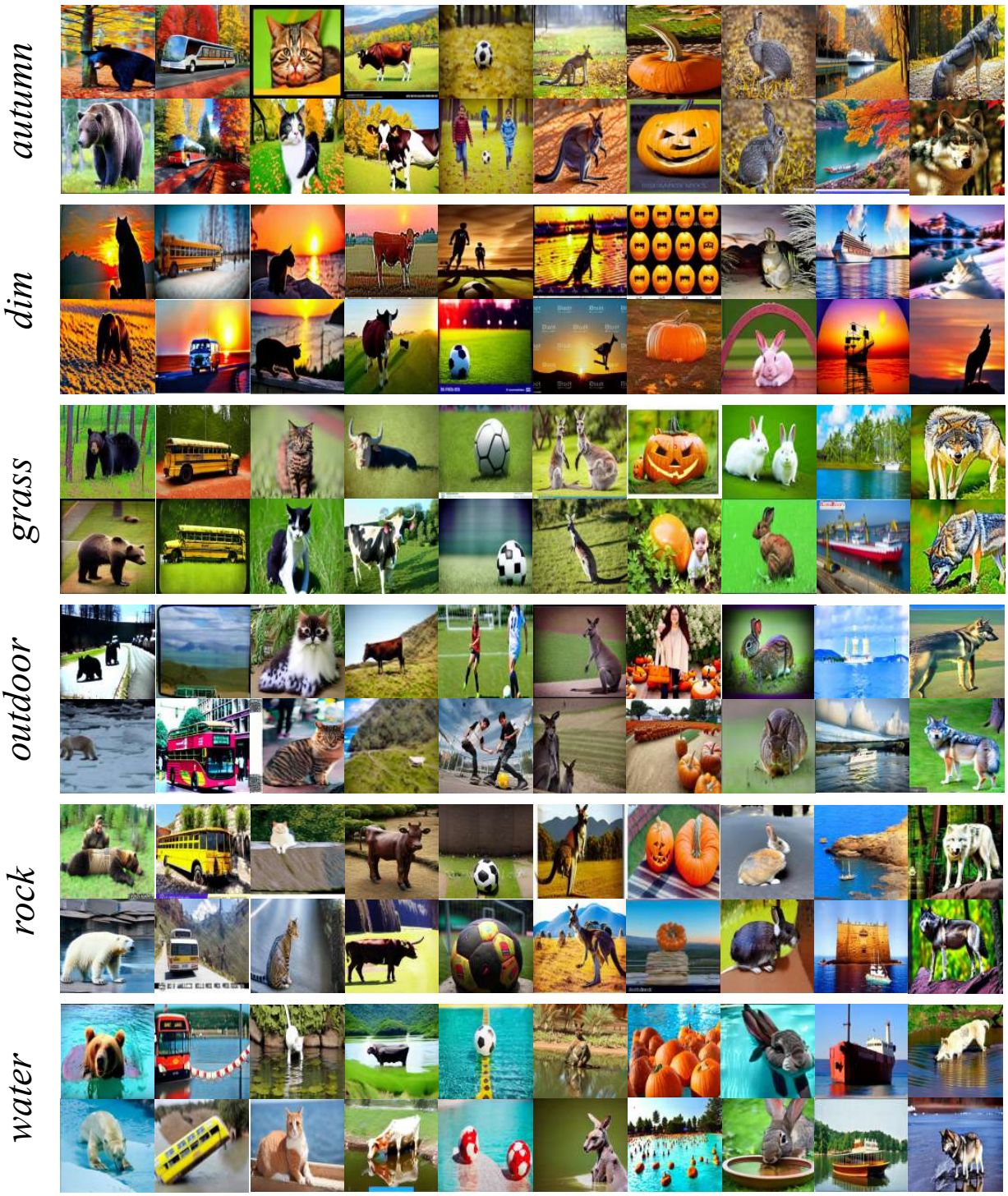


Figure 15. Visualization of more synthetic data for Common NICO++.

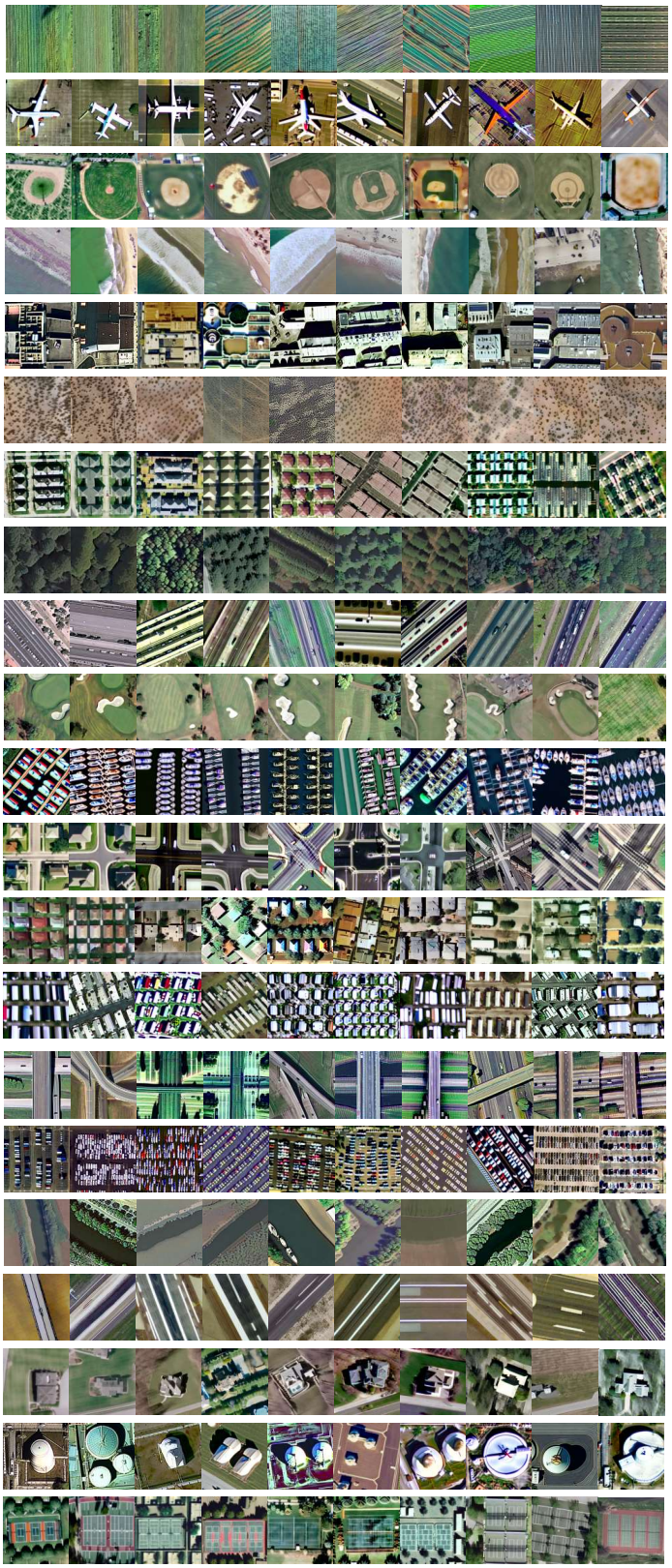


Figure 16. Visualization of more synthetic data for UCMerced.