Supplementary Materials

A. Datasets

We use 6 diverse long-tailed datasets in our experiments: CIFAR10 and CIFAR100[20], LSUN [51], Flowers [32], iNaturalist2019 [11], and AnimalFaces [44] to encompass a wide range of image domains, dataset sizes, resolutions, and imbalanced ratios ρ. In the following, we provide a detailed description of each dataset.

- **CIFAR10/100:** The original CIFAR10 and CIFAR100 datasets consist of 50,000 training images at 32×32 resolution with an equal number of images present in 10 and 100 classes, respectively. An imbalanced version of these datasets referred to as CIFAR10/100-LT, is widely used in the long-tail recognition literature [49]. We follow [5] to generate an exponentially long-tailed version with an imbalance ratio $\rho = \{50, 100\}$ for CIFAR10-LT and $\rho = 100$ for CIFAR100-LT. When calculating few-shot metrics, we follow [4] and take classes 6-9 in CIFAR10-LT and 70-99 in CIFAR100-LT as the few-shot subset in our evaluation.
- **LSUN**: Following [36, 39], we select a challenging subset of 5 classes in the LSUN Scene dataset and keep 50,000 images from the training set (250,000 in total): `dining room`, `conference room`, `bedroom`, `living room`, and `kitchen`. We make this subset long-tailed with an imbalance ratio $\rho = 1000$ and refer to it as LSUN5-LT. We take the `kitchen` class with only 50 training samples as the few-shot subset.
- **Flowers**: Oxford Flowers dataset contains 102 different flower categories. We first combine the train and validation set images to access a larger dataset for the purpose of training and evaluation. This results in a total of 7370 images across all classes. This dataset is naturally imbalanced with 234 and 34 images being present in the category with the most and least number of images ($\rho \approx 7$), respectively. To increase the skewness and learning difficulty, we further increase the imbalance ratio to $\rho = 100$. This reduces the number of training images in the tail classes to only 2 images. We use 128×128 resolution in the experiments and refer to this as Flowers-LT. For the few-shot reference, we take the 52 classes with the least number of training samples (classes 51-102), containing from 23 to 2 images.
- **iNaturalist2019**: The 2019 version of the iNaturalist dataset is a large-scale fine-grained dataset containing 1,010 species from nature. This dataset naturally follows a long-tailed distribution and we keep the training instances in each class intact. We use the training set at the 64×64 resolution in the experiments. The training set of iNaturalist2019 contains a total of 268,243 images. We take the 210 classes with the fewest training instances for the few-shot evaluation (classes 801-1010).
- **AnimalFaces**: This dataset contains the faces of 20 different animal categories (including humans). We set the resolution of the images to 64×64. The most populated categories in the dataset are `dog` and `cat` classes with 388 and 160 images, respectively. The remaining 18 classes are roughly balanced containing from 118 to 100 images. This indicates an imbalance ratio close to 4 ($\rho \approx 4$). To make it more suitable for the long-tail setup, we artificially increase the imbalance ratio to $\rho = 25$, increasing the training difficulty. We refer to this as AnimalFaces-LT in our experiments. For few-shot evaluation, we take the 10 least frequent categories (classes 11-20).

Across all datasets, we sort image categories in the decreasing order of their size, i.e., class index 0 containing the most training images and so on.

B. Effects of Weighted Sampling

Long-tailed datasets suffer from severe class imbalance problems that hinder learning, especially for the tail classes. To mitigate this issue, one common technique in the long-tail recognition literature [49] is to adjust the sampling during training by assigning higher weights to the tail classes, i.e., oversampling them. With recent advancements in data augmentation for GAN training [17, 53, 57], it is reasonable to speculate that oversampling tail instances might improve training calibration. To investigate this, we choose a simple yet effective weighted sampling (WS) method to compare against traditional random sampling methods by assigning a weight w_c for the samples from class $c \in \{1, \ldots, C\}$,

$$w_c = n_c^{-\beta}, \quad \varrho = \rho^{(1-\beta)} \tag{3}$$

where n_c is the number of samples in class c, $\rho = \max_c\{n_c\}/\min_c\{n_c\}$ is the imbalance ratio, and $\beta \in [0, 1]$ is a hyperparameter that damps the sampling imbalance. We illustrate this in Fig. 7 for CIFAR100-LT with $\rho = 100$. When $\beta = 0$, the sampling follows the original data distribution. As β increases, the *effective* imbalance ratio ϱ between the classes decreases. At $\beta = 1$, head and tail classes have the same probability of being drawn during training, i.e. $\varrho = 1$. While WS

might help with more calibrated training, it does not promote knowledge sharing among classes.

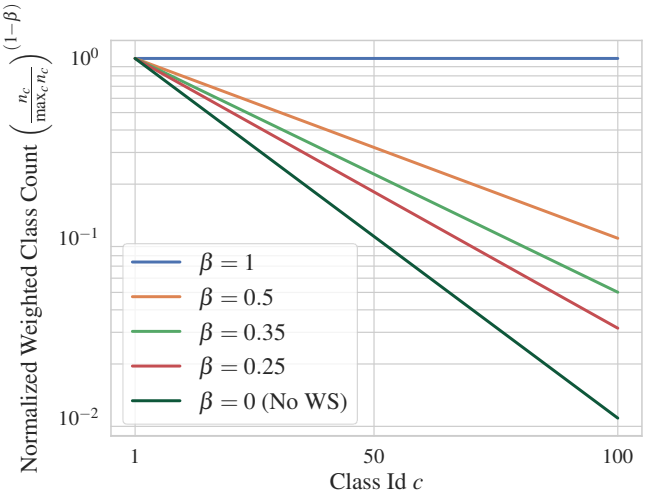

Figure 7. WS CIFAR100-LT ($\rho = 100$). When $\beta = 0$, the sampling follows the original data distribution. As β increases, the *effective* imbalance ratio ϱ between the classes decreases. At $\beta = 1$, head and tail classes have the same probability of being drawn during training, i.e. $\varrho = 1$.

Table. 6 presents a comparison of results obtained from the StyleGAN2-ADA baseline and the use of weighted sampling (WS). Our experiments indicate that WS improves the FID and KID metrics compared to the baseline. However, it does not enhance tail performance in terms of FID-FS and KID-FS, indicating that balancing alone is insufficient when there are very limited training examples in the tail. We also noticed that adding weighted sampling will boost overfitting. Fig. 8 illustrates the training FID-FS curves for the WS methods with different β values, compared against our method and the baseline. For both CIFAR10-LT ($\rho = 100$) and CIFAR100-LT ($\rho = 100$) datasets, WS methods exhibit overfitting behavior. As β increases, this becomes more pronounced, and training becomes unstable when $\beta = 1$ across both datasets. While WS did not show any improvements for CIFAR100-LT, we found it to demonstrate relatively better performance than the baseline at the early stages of training before overfitting sets in. While our experiments on the role of WS in training cGANs in the long-tail setup are informative, we believe that reaching a comprehensive conclusion requires further analysis.

Table 6. Effect of weighted-sampling (WS) when training on long-tailed datasets.

Dataset	CIFAR10-LT				CIFAR100-LT			
Metrics	FID ↓	FID-FS ↓	KID ↓	KID-FS ↓ ×1000	FID ↓	FID-FS ↓	KID ↓	KID-FS ↓ ×1000
StyleGAN2-ADA [17]	9.0	24.2	4.0	9.7	10.8	24.9	5.1	9.3
+ GSR[36]	8.4	24.3	3.9	11.8	11.1	25.0	5.0	8.2
+ UTLO (Ours)	**6.8**	**13.4**	2.8	**5.4**	**9.9**	**21.8**	**4.6**	**7.5**
+ WS ($\beta = 0.25$)	7.1	22.2	2.6	8.1	13.7	28.4	7.2	10.5
+ WS ($\beta = 0.35$)	7.6	22.1	2.9	8.0	14.1	28.0	7.0	9.4
+ WS ($\beta = 0.5$)	8.0	23.4	**2.5**	8.0	14.9	32.1	7.3	10.4

C. Ablation Study

Choice of Intermediate Low Resolution. In our proposed method, one of the hyperparameter choices is to select an intermediate low resolution res_{uc} for unconditional training. All layers with equal or lower resolution than res_{uc} do not have access to class-conditional information. An unconditional GAN objective is added over the images and/or features at res_{uc}. To study the impact of res_{uc}, we conducted an ablation study on the AnimalFaces-LT dataset, which contains images at

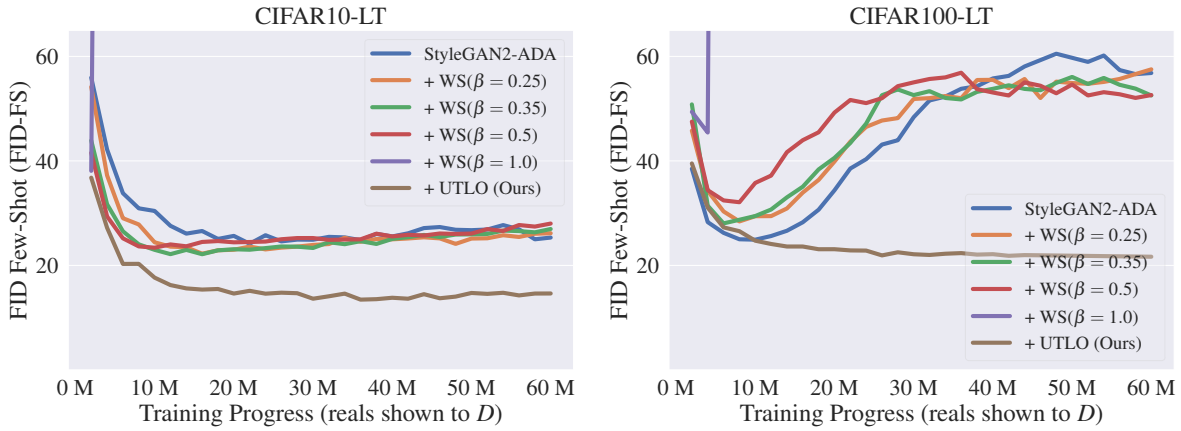

Figure 8. FID-FS curves for CIFAR10-LT and CIFAR100-LT datasets when using weighted sampling (WS) with different β values. We observe that WS leads to overfitting, and as β increases, this becomes more pronounced. Training becomes unstable when $\beta = 1$ for both datasets.

64×64. Table. 7 presents the results for selecting res_{uc} from resolutions lower than 64×64: 8×8, 16×16, and 32×32. For all resolutions, we set the unconditional objective weight $\lambda = 1$.

Studying the results obtained in Table. 7, we oberve that resolutions of 8×8 and 16 achieve relatively close performance. On the other hand, the performance degrades when layers up to 32×32 are trained unconditionally. This is anticipated as the AnimalFaces-LT is at 64×64, leaving only one layer to learn the class-conditional information which is shown to be insufficient.

To better understand the role of intermediate low resolution selected during training, we show the unconditional low-res images $\hat{\boldsymbol{x}}_l$ from the intermediate layers (see Fig.3 in main paper) for different ablated resolutions in Fig. 9. The low-res images are upsampled to the same size for better visual comparison. As resuc increases, more details are introduced to the unconditionally trained intermediate images. For the AnimalFaces-LT dataset, resolution 32×32 already includes fine and definite details, making it challenging to generate (tail) class-specific changes in a single layer before reaching the output resolution of 64×64. We demonstrate this in the bottom of Fig. 9 the final class-conditional images at 64×64 resolution from the tail class `bear` are shown. When res_{uc} is higher, the final output changes minimally from the lower-resolution images $\hat{\boldsymbol{x}}l$. On the other hand, at lower resuc, the level of change is higher.

Table 7. Ablation study on the choice of intermediate low resolution for unconditional training res_{uc} for AnimalFaces-LT dataset (64×64).

Unconditional Low Resolution (res_{uc})	FID ↓	FID-FS ↓	KID ↓ ×1000	KID-FS ↓ ×1000
8×8	**26.2**	**48.4**	**12.6**	**19.6**
16×16	27.5	50.3	13.7	20.8
32×32	38.0	64.9	23.3	34.3

Contribution of Unconditional Training Objective at Low Resolutions. Another hyperparameter introduced by our method is the choice of unconditional training objective weight (λ) relative to the conditional one (see Eq. 2&3 in the main paper). Since the results for 8×8 and 16×16 resolutions were comparable in Table. 7, we conducted an ablation study on λ values for both resolutions. Table. 8 shows the results of the ablation study on the contribution of different values for the unconditional training objective ($\lambda = 0.01, 0.1, 1, 10$). We also considered the case when no unconditional training is added ($\lambda = 0$), and only the lower resolution layers $\leq res_{\text{uc}}$ do not receive class-conditional information, i.e., they are passed $\boldsymbol{w}_z$ as the style vector instead of $\boldsymbol{w}_{z,y}$ (see Figure 3 in the main paper). We find that $\lambda = 1$ achieves the best performance for both

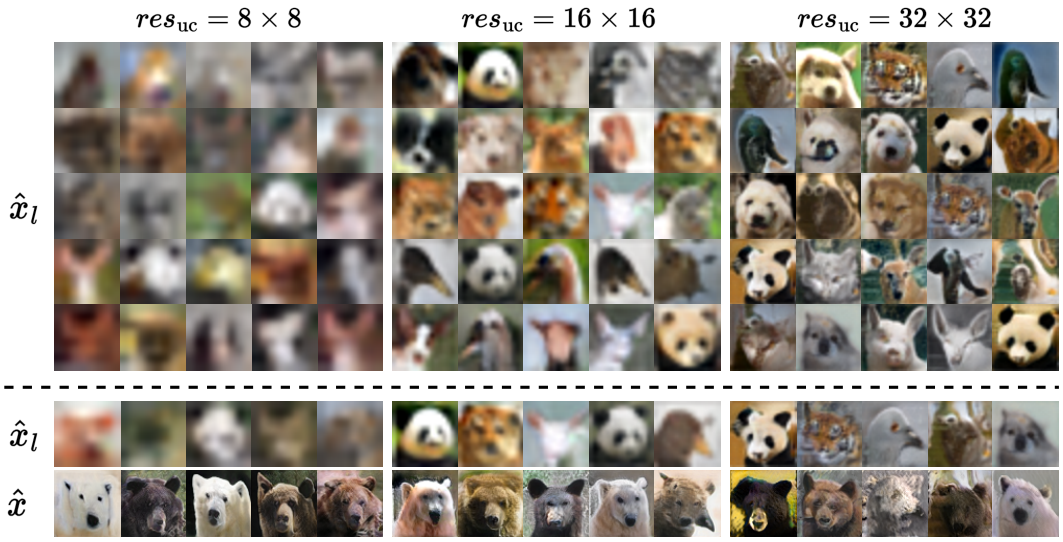

Figure 9. Visual comparison of the choice of different unconditional low-resolution (res_{uc}) in our proposed framework on the AnimalFaces-LT dataset. As (res_{uc}) increases, the unconditional low-resolution image $\hat{\boldsymbol{x}}_l$ entails finer features (top). The low-resolution images are then used to generate class-conditional images at output resolution ($\hat{\boldsymbol{x}}$) from the tail class `bear` (bottom). The images are upsampled to the same size for better comparison. (best viewed in color)

resolutions. When λ is too small (0.01) or too large (10), it disrupts the balance between the conditional and unconditional objectives, leading to performance degradation.

Table 8. Ablation on the choice of unconditional training objective weight (λ) at different low-resolutions res_{uc} for AnimalFaces-LT dataset.

res_{uc}	Unconditional Training Objective Weight (λ)	FID ↓	FID-FS ↓	KID ↓ ×1000	KID-FS ↓ ×1000
8×8	No Unconditional Training ($\lambda = 0$)	64.0	94.2	35.8	45.4
	0.01	61.0	99.9	32.4	50.0
	<u>0.1</u>	<u>28.6</u>	<u>50.0</u>	<u>13.6</u>	<u>19.8</u>
	1	**26.2**	**48.4**	**12.6**	**19.6**
	10	111.6	145.8	54.6	66.4
16×16	No Unconditional Training ($\lambda = 0$)	58.3	87.7	28.5	41.8
	0.01	64.7	88.5	30.7	35.9
	<u>0.1</u>	<u>31.4</u>	<u>53.3</u>	<u>15.7</u>	<u>22.6</u>
	1	**27.5**	**50.3**	**13.7**	**20.8**
	10	59.9	111.0	30.0	53.8

Need for unconditional layers in the discriminator ($\mathcal{L}_{uc}$) and end-to-end joint training with both unconditional and conditional objectives. In addition to the unconditional layers in the generator, we have found that unconditional layers should be explicitly present in the discriminator. Table 9 provides the ablation results where the unconditional layers are removed from UTLO (i.e., w/o $\mathcal{L}_{uc}$). This shows significantly worse performance, indicating the need for *explicit* unconditional discriminator on the low resolution.

Further, to demonstrate how the proposed method differs from other training strategies that promote coarse-to-fine learning, e.g. progressive training [41], we carefully design experiments to compare our proposed method against progressive training. We follow the progressive strategy in StyleGAN-XL [41], starting with training a stem at a low resolution. After training the stem, the training of higher-resolution layers is followed. For a more comprehensive analysis, we experimented with two stems: an unconditional stem and a conditional one.

Firstly, we observed the conditional stem exhibited mode collapse early in the training. For training the subsequent higher-resolution layers, we picked the best stem before the mode collapse. This training strategy yielded considerably worse results as shown in Table 9 (last row). Conversely, the unconditional low-resolution stem did not experience mode collapse. Indeed, Table 9 shows that using unconditional stem (second-to-last row) improved over the baseline. However, it was still significantly worse than UTLO, showing the benefit of our design: end-to-end joint training with both unconditional and conditional objectives.

Table 9

Method	FID ↓	FID-FS ↓	KID ↓ ×1000	KID-FS ↓ ×1000
StyleGAN2-ADA [17]	51.4	87.1	24.7	35.9
+ UTLO (Ours)	**26.2**	48.4	12.6	19.6
+ UTLO w/o $\mathcal{L}_{uc}$	63.75	87.35	35.08	35.78
+ Progressive [41]: Uncond. Stem	37.73	64.57	21.71	31.53
+ Progressive [41]: Cond. Stem	98.35	150.86	73.97	102.20

D. Knowledge Sharing Analysis

In Figure 5 of the main paper, we presented conditionally generated images from our method trained on the CIFAR10-LT ($\rho = 100$) dataset, demonstrating knowledge sharing among the head and tail categories using a shared unconditional low-resolution image. To quantify the similarity among images that share the same input latent code but have different class-condition labels, we use the Learned Perceptual Image Patch Similarity (LPIPS) metric [54]. We first generate 1,000 random noises from different seeds. Given each sampled noise z, we generate images from all 10 classes of the CIFAR10-LT ($\rho = 100$) dataset, $c \in \{0, \ldots, 9\}$, including both head and tail classes. We then calculate the LPIPS among all class pairs. Table. 10 reports the average LPIPS score obtained from the baselines and our proposed method.

Table 10. Comparing Avg. LPIPS among all class pairs. Given a fixed noise input, we generate images from all classes (including both head and tail). We then quantify the similarities in terms of LPIPS metric among all class pairs and compare our method against baselines.

Methods	StyleGAN2-ADA [17]	+ GSR [36]	+ UTLO (Ours)
Avg. LPIPS ×1000	12.31	12.37	11.20

The results show that generated images from all classes (head and tail) in UTLO, which promotes knowledge sharing, exhibit higher similarities (lower LPIPS) compared to the baselines that do not incorporate means of knowledge-sharing. To investigate which head and tail classes share the most patch similarities, we plotted the average LPIPS for individual class pairs in Fig. 10. As somewhat expected, It is observed that tail classes such as `truck`, `ship`, and `frog` share the highest similarities with head classes `automobile`, `airplane`, and `bird`, respectively. This is intuitive as these class pairs have common attributes such as *blue/green backgrounds, wheels, etc.* Visual examples of this can also be found in Figure 5 of the main paper and Fig. 13.

E. Long-tail v.s. Limited Data Regime

Previous work on training GANs under limited data [17, 42, 57] has shown that the quality of generated images degrades as the dataset size decreases. In long-tailed datasets, on the other hand, there is an additional challenge of data imbalance across classes. To better understand their individual effects on training cGANs, we devise a setup in which we create a balanced dataset with the same size as the long-tail one. More concretely, given a long-tailed dataset with a total of n training images (across head and tail classes), we create a new balanced dataset of size n in which each class $c \in 1, \ldots, C$ contains n/C images.

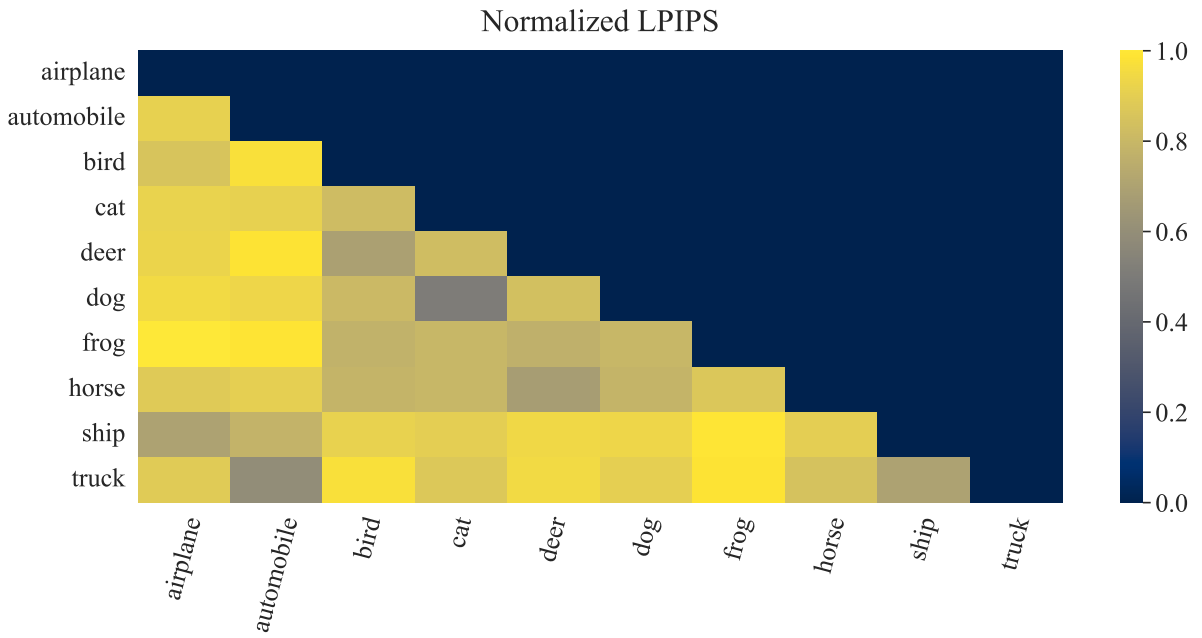

Figure 10. Normalized LPIPS among all class pairs of CIFAR10-LT dataset using our proposed method. We generate images from all classes using the same latent codes and calculate the LPIPS among all class pairs. The obtained results are normalized from 0 to 1.We observe that tail classes such as `truck`, `ship`, and `frog` share the highest similarities with head classes `automobile`, `airplane`, and `bird`, respectively. The lower LPIPS values indicate higher similarities between the generated images. (best viewed in color)

Table 11. Effect of (balanced) limited data v.s. long-tailed data in training cGANs. This table compares the quality of the generated images for different dataset sizes and distributions on CIFAR10 dataset. Baseline StyleGAN2-ADA is used **without** GSR or the proposed UTLO.

# Train Images	Data Distribution	FID ↓	FID-FS ↓	KID ↓ ×1000	KID-FS ↓ ×1000
50,000	Full Dataset (Balanced)	2.5	3.3	0.4	0.4
13,996	Long-tail ($\rho = 50$)	6.5	21.4	2.4	9.0
	Limited Data (Balanced)	3.9	4.6	1.0	0.8
12,406	Long-tail ($\rho = 100$)	9.0	24.2	4.0	9.7
	Limited Data (Balanced)	4.5	5.7	1.3	1.3

To analyze the effects of dataset size v.s. the distribution of the classes in the dataset, we trained the baseline StyleGAN2-ADA [17] on different setups of CIFAR10. Table. 11 compares the quality of the generated images in terms of GAN metrics for each setup. We observe that in the balanced setup, the model achieves better scores compared to the long-tail setup. Moreover, we see that the performance gap between the full and few-shot metrics is noticeably larger in the long-tail setup. We suspect that the performance gap between the full and few-shot metrics in the balanced setup might be due to the difficulty of the selected few-shot classes or the size of the real data used in calculating the metric. A smaller gap between the KID and KID-FS, which is unbiased in design, supports the latter.

F. Implementation Details, and Choice of Hyperparameters

The baselines used in our experiments cover different designs in generator, discriminator, and data augmentation pipelines. We provide the implementation details in the following.

- **StyleGAN2 with Adaptive Data Augmentation (ADA)** [17]: We use the official PyTorch implementation [*] in our experiments. On CIFAR-LT datasets, we used the `cifar` configuration. For the rest of the datasets, we use the `auto` configuration. Adding transitional training [42], we used the official implementation [†] provided by the authors.

- **Projected GAN (PGAN)** [40]: We use the projected discriminator with both the *FastGAN* [23] and StyelGAN2 [18] generator backbones in our experiments. For the data augmentation, *Differentiable Augmentation (DA)* [57] is used. The official PyTorch implementation is provided by the authors [‡]. Default hyperparameters are used. Note, the authors of PGAN provided a liter version of FastGAN which gets to similar performance as the original FastGAN. We use the lite version in our experiments.

- **Group Spectral Regularization (GSR)** [36]: We added the GSR implementation provided by the authors [§] to both the StyleGAN2+ADA and PGAN+DA repositories. We used the default choice of hyperparameters in all experiments.

- **Noisy Twins** [37]: We use the official implementation provided by the authors [¶] and also added it to the PGAN+DA repository. We used the default choice of hyperparameters in all experiments.

- **Unconditional Training at Lower Resolutions (UTLO)**: We used the default training configuration of the baselines. Our method introduces two new hyperparameters: the intermediate low resolution and λ, the weight between the conditional and unconditional loss terms (see ablation in Sec. C). In general, we found intermediate low resolution of 8×8 and $\lambda = \{0.1, 1\}$ to be generally applicable across all datasets and resolutions, attaining competitive performance.
  For the reported results in the paper, we use 8×8 as the intermediate low-resolution across all datasets, except for the Flowers-LT dataset where 16×16 resolution shows slightly better performance. Choosing λ, on CIFAR100-LT and Flowers-LT which contain very few samples in the tail classes, we found $\lambda = 10$ performing the best. On CIFAR10-LT, λ is set to 0.1. Across the rest of the datasets, we use $\lambda = 1$.

G. Additional Evaluation Results

Naturally Imbalanced Datasets: iNaturalist2019 and Flowers-LT We evaluate our proposed method on datasets that are naturally imbalanced. This covers the iNaturalist2019 and Flowers-LT datasets which we train at 64×64 and 128×128 resolutions, respectively. Different from previous experiments, we use ProjectedGAN (StyleGAN2) + DA [40] as the baseline here. Table. 12 shows the quantitative evaluation results. The results across both datasets consolidate our findings and validate the effectiveness of our proposed method.

Additionally, we show visual examples from the training and generated images from two different tail classes in the iNaturalist2019 dataset in Fig. 11. It can be seen that the images learned from our method (UTLO) tend to generate more diverse and higher-quality images for the tail classes. Generated images from the tail classes of the Flowers-LT dataset are shown in Fig. 15.

Table 12. Comparing the proposed method against the baseline across two naturally imbalanced datasets: Flowers-LT (128×128 resolution) and iNaturalist2019 (64×64 resolution).

Dataset	Flowers-LT				iNaturalist2019			
Metrics	FID ↓	FID-FS ↓	KID ↓ ×1000	KID-FS ↓ ×1000	FID ↓	FID-FS ↓	KID ↓ ×1000	KID-FS ↓ ×1000
PGAN (StyleGAN2) + DA [40]	9.8	21.6	2.4	2.9	3.6	11.4	0.53	1.08
+ GSR[36]	8.2	17.9	1.1	**1.7**	3.5	11.1	0.51	0.99
+ NoisyTwins[37]	6.7	**15.3**	**0.9**	1.9	3.0	10.6	0.45	0.72
+ UTLO (Ours)	**6.6**	15.4	**0.9**	1.8	**2.8**	**10.1**	**0.41**	**0.60**

Combining our method with GSR and WS Although we provided a direct comparison of UTLO against GSR [36] and weighted sampling (WS), they can be integrated with our proposed training framework. In Table. 13, we present quantitative

[*] https://github.com/NVlabs/stylegan2-ada-pytorch
[†] https://github.com/mshahbazi72/transitional-cGAN
[‡] https://github.com/autonomousvision/projected-gan
[§] https://github.com/val-iisc/gSRGAN
[¶] https://github.com/val-iisc/NoisyTwins

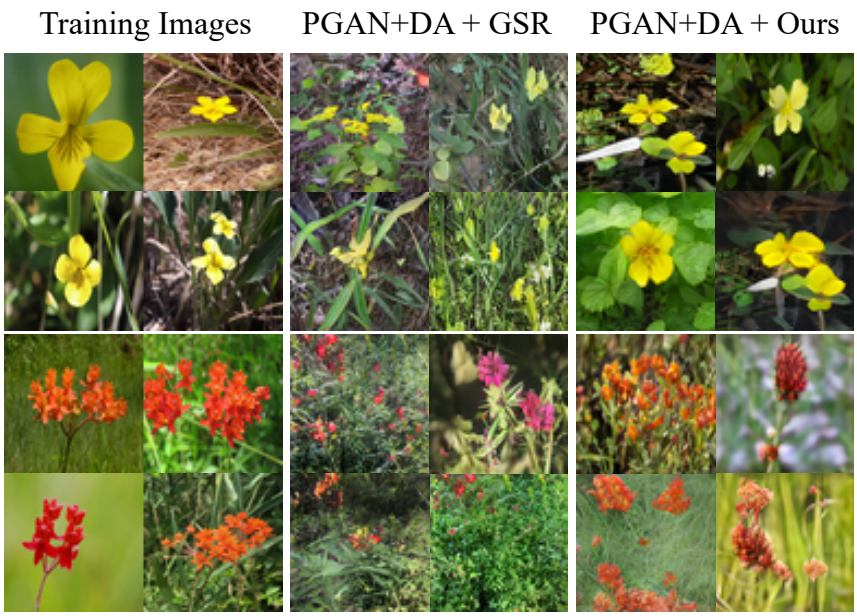

Figure 11. Examples of training and generated images contrasting our method against the baseline on two different tail classes of the iNaturalist2019 dataset at 64×64 resolution. Each row depicts a tail class.

Table 13. Integrating GSR and weighted sampling (WS) to our proposed method on the AnimalFaces-LT dataset. The obtained results from adding GR and WS do not show any noticeable improvement over solely using UTLO.

Method	FID ↓	FID-FS ↓	KID ↓ ×1000	KID-FS ↓ ×1000
StyleGAN2-ADA	51.4	87.1	24.7	35.9
+ UTLO (Ours)	26.2	48.4	12.6	19.6
+ GSR [36]	39.2	67.2	21.2	32.7
+ UTLO + GSR	<u>26.9</u>	<u>47.8</u>	**12.2**	<u>19.3</u>
+ NoisyTwins [37]	29.4	50.2	16.7	21.2
+ UTLO + NoisyTwins	<u>26.6</u>	**48.1**	<u>12.9</u>	**19.0**
StyleGAN2-ADA + UTLO + WS	27.7	47.4	13.4	18.5

results on AnimalFaces-LT when adding GSR, NoisyTwin, and WS (see Sec. B) to UTLO. We use WS with $\beta = 0.35$. Combining UTLO with regularization methods (GSR and NoisyTwins) and weighted sampling (WS) didn't show clear improvements over UTLO alone. However, it resulted in significant improvements over GSR and NoisyTwins individually.

Comparison against Unconditional Training We compare unconditional and conditional training on the AnimalFaces-LT dataset and present the results in Table. 14. The unconditional model generates samples that follow the training distribution, which is mainly dominated by head classes. This bias favors the unconditional baseline in terms of the FID and KID metrics, which do not consider the skewness in the data distribution. We recommend including FID-FS and KID-FS metrics when evaluating GANs on imbalanced datasets.

H. Additional Visual Comparison

In Fig. 12-16, we provide additional visual examples from our proposed method and compare them against baselines. Fig. 12 presents a comparison of generated images from all classes in the CIFAR10-LT dataset ($\rho = 100$). The data imbalance curve is shown in this figure where there are only 50 training samples present in the rarest tail class `truck` (top row) while

Table 14. Comparing unconditional and conditional baselines on AnimalFacs-LT dataset. The Unconditional baseline generates samples that track the training distribution which is mainly dominated by head classes. This favors it in terms of FID and KID metrics, which do not consider the skewness in the data distribution. We suggest FID-FS and KID-FS metrics should be incorporated when evaluating GANs over imbalanced datasets.

Method	FID ↓	FID-FS ↓	KID ↓ ×1000	KID-FS ↓ ×1000
StyleGAN2-ADA Unconditional	39.4	104.1	17.3	27.6
StyleGAN2-ADA Conditional	51.4	87.1	24.7	35.9
StyleGAN2-ADA Conditional + UTLO (Ours)	26.2	48.4	12.6	19.6

the most populated head class `airplane` (bottom row) has 5,000 training samples. Fig. 13 shows additional examples of knowledge sharing from the head to the tail classes in the CIFAR10-LT dataset using our proposed UTLO framework. The conditional images generated from the head (middle columns) and tail (right columns) classes share and are built on top of the same low-resolution (unconditional) images (left columns).

Fig. 14 compares the generated images from our proposed method against the baseline across classes with *only 5* training instances (shown in the top-left corner) in the CIFAR100-LT dataset ($\rho = 100$). We use StyleGAN2-ADA as the baseline in the CIFAR100-LT experiments. Further, we present additional visual comparisons of generated images from the rarest tail classes of the Flowers-LT with *only 2* training images (shown in the top-left corner) in Fig. 15. Our proposed approach enables a diverse set of features, such as backgrounds, colors, poses, and object layouts, to be infused into the tail classes with very few training images. We use ProjectedGAN (StyleGAN2) + DA as the baseline. Finally, we showcase the generated images from the 5 tail classes with the least number of training images in the AnimalFaces-LT dataset. While the diversity of the generated images is limited by the baselines, UTLO learns a more diverse set of images with very few training images. StyleGAN2-ADA serves as the baseline for the AnimalFaces-LT experiments.

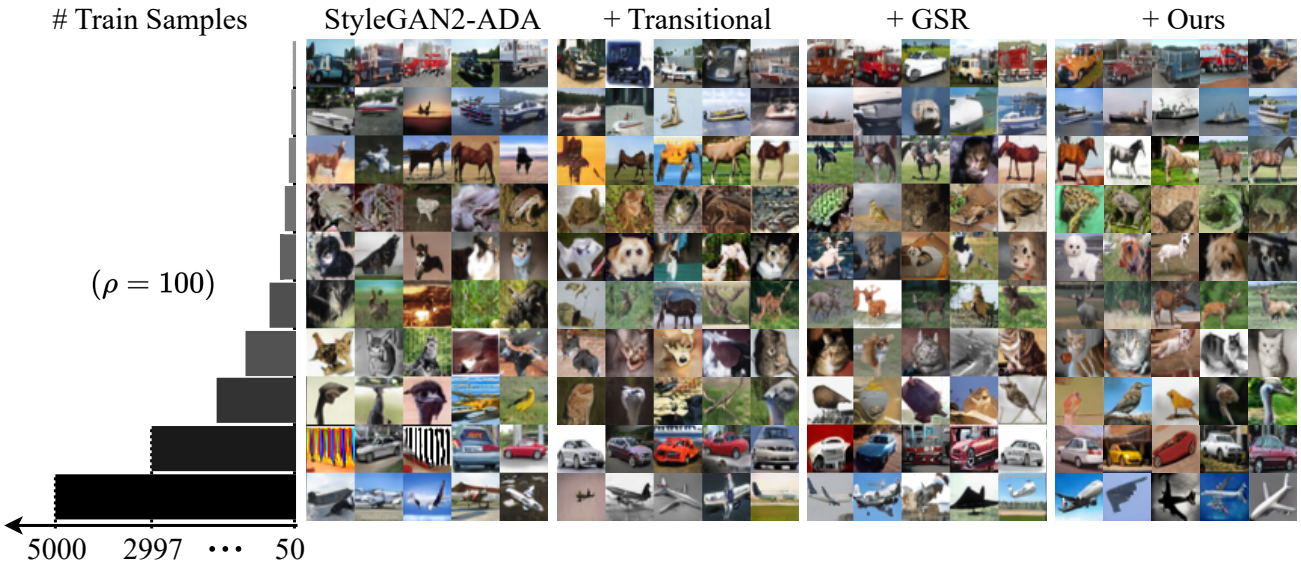

Figure 12. Qualitative comparison of the generated images from all classes in the CIFAR10-LT dataset ($\rho = 100$). There are only 50 training samples present in the rarest tail class `truck` (top row) while the most populated head class `airplane` (bottom row) has 5,000 training samples.

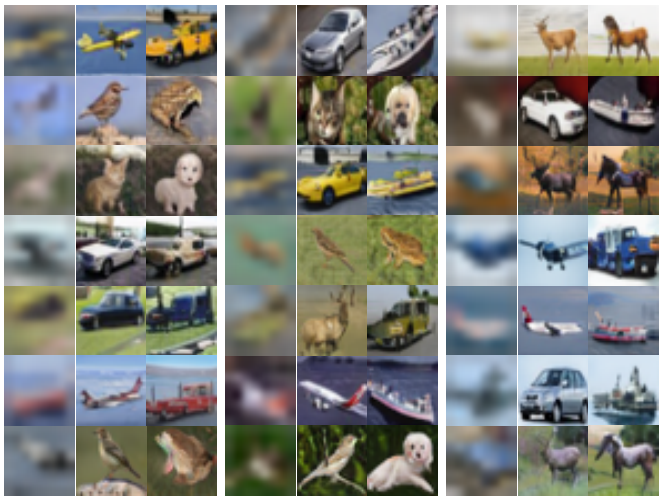

Figure 13. Additional examples of knowledge sharing from the head to the tail classes in CIFAR10-LT dataset using our proposed UTLO framework. The conditional images generated from the head (middle columns) and tail (right columns) classes share and are built on top of the same low-resolution (unconditional) images (left columns). Low-resolution images (8×8) are upsampled to that of CIFAR10- LT (32×32) for improved visualization. (best viewed in color.)

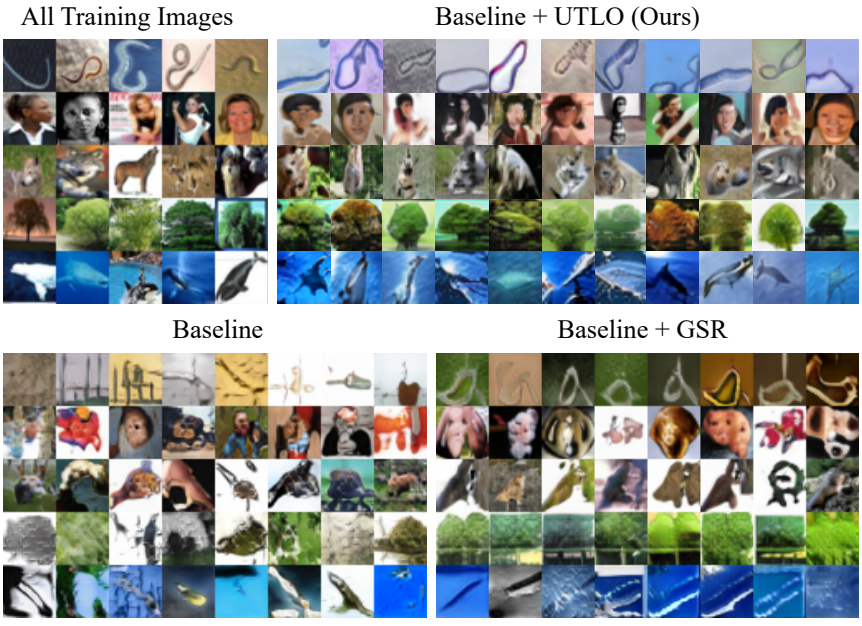

Figure 14. Comparing the generated images from our proposed method against the baseline across classes with only 5 training instances in the CIFAR100-LT dataset ($\rho = 100$). The baseline used is StyleGAN2-ADA. Training images are shown in the top-left corner.

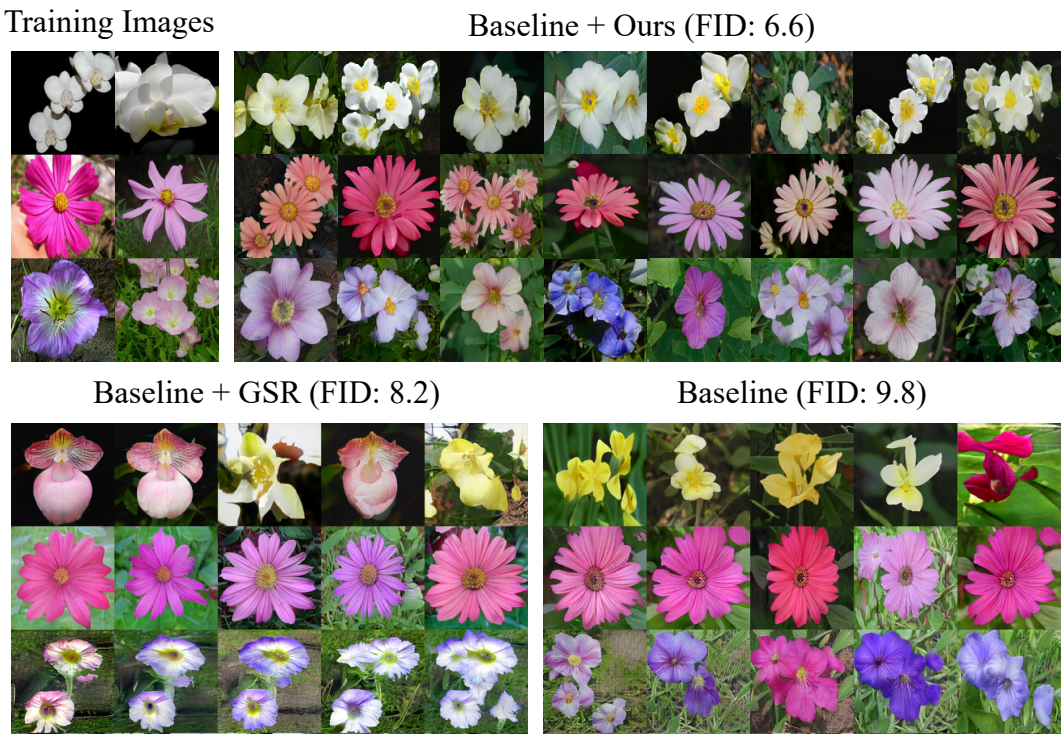

Figure 15. Additional visual examples when generating images from the rarest tail classes in the Flowers-LT with only 2 training images. Our proposed approach allows for a diverse set of features such as backgrounds, colors, poses, object layouts, etc. to be infused into the tail classes with very few training images. Training images are shown in the top-left corner. ProjectedGAN (StyleGAN2) + DA is used as the baseline.

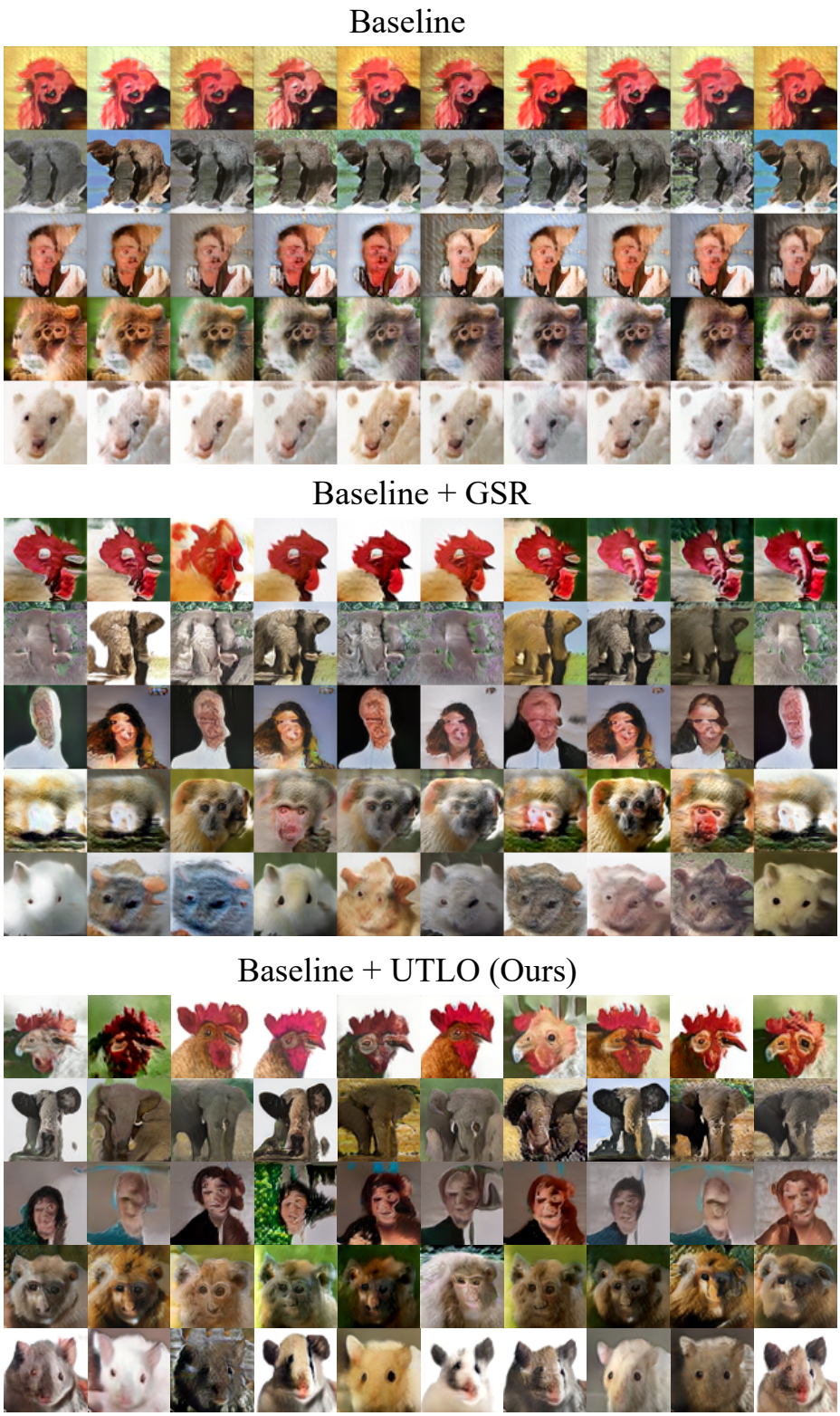

Figure 16. Generated images from 5 tail classes with the least number of training images in the AnimalFaces-LT dataset. While the diversity of the generated images is limited by baselines, UTLO learns a set of more diverse images with very few training images. StyleGAN2-ADA is used as the baseline.