Appendix

This appendix is organized as follows:

- Section A.1 gives the basic theory of spherical interpolation and the derived spherical extrapolation. Then we provide proof that the circle interpolation result of two DDIM inversions is approximately standard normal distribution.
- Section A.2 provides details of the comparison methods (*cf*., Section 4), including Mixup [17], CutMix [16], Real-Filter [3], Real-Guidance [3], Da-Fusion [14], Real-Mix [15], Diff-AUG [15], Diff-Mix [15], CMO [11] and CMO+DRW [2].
- Section A.3 provides some quantitative and qualitative analyses to support our main claim: current Intra-category DA [14] and Inter-category DA [15] can not take account of both faithfulness and diversity well.
- Section A.4 gives the implementation details of our Diff-II and the reproduction details of the comparison methods. Besides, we also give the computation cost of our Diff-II.
- Section A.5 gives some background about Diffusion, DDIM, and DDIM inversion.
- Section A.6 shows additional results of experiments. First, we give the suffixes and predefined metaclasses of each dataset in Section A.6.1 and Section A.6.2 respectively. Then, we give the more ablation results of the key components and *split ratio* (*cf*., Section 3.3) in Section A.6.3. Finally, we give additional visualizations in Section A.6.4.

A.1. Theory of Spherical Interpolation

Spherical interpolation is a method used to interpolate between two points on a sphere or in a spherical space. The main idea behind spherical interpolation is to find a point along the shortest path on the sphere's surface between two given points. The theory of spherical interpolation is grounded in spherical geometry and quaternion algebra as follows:

Shortest Path on Sphere: The shortest path between two points on the surface of a sphere is along the great circle that passes through both points. A great circle is any circle that divides the sphere into two equal hemispheres, like the equator or the meridians on a globe.

Interpolation Formula: Given two points on a sphere, represented by unit vectors A and B, and an interpolation parameter $t \in [0, 1]$, spherical interpolation calculates a third point P along the great circle from A to B using the formula:

$$P = \frac{\sin((1-t)\theta)A + \sin(t\theta)B}{\sin(\theta)}, \tag{A1}$$

where θ is the angle between A and B, found using the dot product $\cos(\theta) = A \cdot B$.

Quaternion Interpolation: When dealing with rotations in computer graphics, spherical interpolation can be applied using quaternions. Quaternions provide a way to represent orientations and rotations in three dimensions without the singularity and ambiguity problems of Euler angles. The interpolation of two quaternions q_1 and q_2 is given by:

$$q = \frac{\sin((1-t)\theta)q_1 + \sin(t\theta)q_2}{\sin(\theta)}, \tag{A2}$$

where θ is the angle between the quaternions, computed as $\cos(\theta) = \text{Re}(q_1^* q_2)$ (with q_1^* being the conjugate of q_1). Spherical interpolation can smoothly interpolate rotations and directions, ensuring that the interpolated values remain on the sphere, and thus maintaining the integrity of the rotations or directional data.

Based on the above, we can easily derive the spherical interpolation Z between two vectors (I_a and I_b) of the same length:

$$Z = \frac{sin((1-\lambda)\alpha)}{sin(\alpha)} I_a + \frac{sin(\alpha\lambda)}{sin(\alpha)} I_b, \qquad \lambda \in [0, 1], \tag{A3}$$

where λ is the interpolation strength, $\alpha = arccos(\frac{I_a^T I_b}{(||I_a||||I_b||)})$ and Z is the final interpolation result. Then we generalize to spherical extrapolation:

$$Z = \frac{sin((1+\lambda)\alpha)}{sin(\alpha)} I_a - \frac{sin(\alpha\lambda)}{sin(\alpha)} I_b, \qquad \lambda \in [0, \frac{2\pi}{\alpha} - 1] \tag{A4}$$

Spherical extrapolation can expand the trajectory along the interpolation trajectory, increasing the interpolation range while still maintaining the integrity. Based on the periodicity of trigonometric functions, we can merge spherical interpolation and extrapolation into circle interpolation:

$$Z = \frac{sin((1+\lambda)\alpha)}{sin(\alpha)} I_a - \frac{sin(\alpha\lambda)}{sin(\alpha)} I_b, \qquad \lambda \in [0, \frac{2\pi}{\alpha}] \tag{A5}$$

Then we give the proof that the circle interpolation of two DDIM inversions is approximately standard normal distribution.

First, we consider that I_a and I_b are two DDIM inversions, which are approximately in standard normal distribution:

$$I_a \sim N(\mu_a, \sigma_a^2), \qquad \mu_a \simeq 0, \sigma_a \simeq 1 \tag{A6}$$

$$I_b \sim N(\mu_b, \sigma_b^2), \qquad \mu_b \simeq 0, \sigma_b \simeq 1 \tag{A7}$$

According to the superposition of normal distribution, we can get the distribution of Z:

$$\begin{aligned} Z \sim N(&\frac{sin((1+\lambda)\alpha)}{sin(\alpha)}\mu_a - \frac{sin(\alpha\lambda)}{sin(\alpha)}\mu_b, \\ &(\frac{sin((1+\lambda)\alpha)}{sin(\alpha)})^2\sigma_a^2 + (\frac{sin(\alpha\lambda)}{sin(\alpha)})^2\sigma_b^2) \end{aligned} \tag{A8}$$

For the mean term of Eq. (A8):

$$\frac{sin((1+\lambda)\alpha)}{sin(\alpha)}\mu_a - \frac{sin(\alpha\lambda)}{sin(\alpha)}\mu_b \simeq 0 \tag{A9}$$

For the variance term, the $\alpha \simeq \pi/2$ due to I_a and I_b are two independent high-dimension vectors. Thus, $sin\alpha \simeq 1$ and $cos\alpha \simeq 0$. Then, we can simplify the variance term:

$$\begin{aligned}
&(\frac{sin((1+\lambda)\alpha)}{sin(\alpha)})^2\sigma_a^2 + (\frac{sin(\alpha\lambda)}{sin(\alpha)})^2\sigma_b^2 \\
\simeq &sin^2((1+\lambda)\alpha) + sin^2(\alpha\lambda) \\
= &sin^2(\alpha+\alpha\lambda) + sin^2(\alpha\lambda) \\
= &(sin(\alpha)cos(\alpha\lambda) + cos(\alpha)sin(\alpha\lambda))^2 + (sin(\alpha\lambda))^2 \\
\simeq &cos^2(\alpha\lambda) + sin^2(\alpha\lambda) = 1
\end{aligned} \tag{A10}$$

Thus,

$$Z \sim N(\mu, \sigma^2), \qquad \mu \simeq 0, \sigma \simeq 1 \tag{A11}$$

Proof completed.

A.2. Comparison Methods

In this section, we introduce all the comparison methods of experiments.

Conventional Data Augmentation Methods:

- *Mixup* [17]: Conduct linear interpolation on the RGB space between two images. The interpolation strength will decide the soft label of the mixed image.
- *CutMix* [16]: Mix two images into one mixed image in this way: randomly generate a cropping box, crop the corresponding position of one image, and then use the corresponding position of another mage to put it into the cropped area to form a new sample. The crop size ratio will decide the soft label of the mixed image.

Diffusion-based Data Augmentation Methods:

- *Real-Filter* [3]: Directly generate some synthetic images with prompts containing their corresponding category labels. Then, leverage a pre-trained perception network to extract the features of both images of the original training set and synthetic images. Finally, filter all the synthetic images that are far from images of the original training set and only maintain those that are closed to the original training images.
- *Real-Guidance* [3]: Given an image from the original training set, add T timesteps noise to the image and use the noised one to replace the random noise at the beginning of the generation. Finally, denoise it with a prompt containing its category label.
- *Da-Fusion* [14]: Firstly, set a few learnable token embeddings to learn an accurate concept for each category with the original training set. Then for a given image of the original training set, add random timesteps noise and denoise the noised image with a prompt containing its learned category concept.
- *Real-Mix* [15]: Given an image from the original training set, add random timesteps noise to the image. Then denoise the noised image with a prompt containing other-category labels. This will lead to a synthetic image with intermediate semantics between the two categories. Design a calculation mechanism to decide the soft label for this synthetic image.
- *Diff-AUG* [15]: Firstly, set a few learnable token embeddings and insert some learnable low-rank matrixes into the U-Net to learn an accurate concept for each category with the original training set. Then for a given image, add T timesteps noise and denoise with a prompt containing its learned category concept.
- *Diff-Mix* [15]: Firstly, set a few learnable token embeddings and insert some learnable low-rank matrixes into the U-Net to learn an accurate concept for each category with the original training set. Then for a given image, add random timesteps noise and denoise with a prompt containing learned other-category concepts. This will lead to a synthetic image with intermediate semantics between the two categories. Design a calculation mechanism to decide the soft label for this synthetic image.

Long-tail Classification Methods:

- *CMO* [11]: To balance the number of different categories's training samples. CMO crops the objects from the rare-category images and pastes them to rich-category images to get some new images with rare-category objects and rich-category images' backgrounds. These new images will be used to expand the rare-category images.
- *CMO+DRW* [2]: Except on oversampling-based CMO, DRW gives different weights to the loss of different categories. Specifically, the rare categories get a large loss weight while rich categories get a smaller loss weight.

Methods	CLIP Score (↑)	LPIPS (↑)	Acc (↑)
Da-Fusion [14]	30.47	31.3%	-2.65
Diff-Mix [15]	-	**59.3%**	-4.02
Ours	**30.60**	52.7%	**+5.52**

Table A1. **Quantitative analyses of diversity and faithfulness.**

A.3. Analyses of Diversity and Faithfulness

A.3.1. Quantitative Analyses

We investigated the synthetic set of 5-shot Aircraft (same setting as Sec. 4.1 with ResNet) and reported: **CLIP Score** [4] of the synthetic set; average **LPIPS** [19] between images of the synthetic set, and classification **accuracy**. The CLIP score can reflect the faithfulness of the synthetic set while the LPIPS can indicate the diversity. As shown in A1, we compared with two typical methods: Da-Fusion [14] and Diff-Mix [15] which is Intra-category DA and Inter-category DA of Figure 2 respectively.

We can see that Da-Fusion had a decent CLIP Score, which means good faithfulness. However, its LPIPS is quite low, indicating low diversity. For Diff-Mix, although its diversity is quite outstanding, faithfulness can not be guaran-

teed since the soft labels are quite difficult to decide, bootstrapping a bad classification accuracy. Thus, we can conclude that current Intra-category DA and Inter-category DA methods can not take account of both faithfulness and diversity well.

A.3.2. Qualitative Analyses

As shown in Figure 1: (a) is the synthetic images of Da-Fusion [14], (b) is Diff-Mix [15] and (c) is our Diff-II. We can get an intuitive perception that Da-Fusion and Diff-Mix can not take account of both faithfulness and diversity well. In comparison, our Diff-II can generate both faithful and diverse images.

A.4. Implementation Details

In this section, we give all the implementation details of our Diff-II and reproduction details of comparison methods.

Details of Our Diff-II:

- *Category concept learning*: We follow the implementations of [15][1].
- *Inversion interpolation*: We use DDIM inversion [12] with 25 steps and 1.0 guidance scale [5] to calculate the inversion for each image. Then for each category, we randomly sample inversion pairs until the number of inversion pairs reaches five times the number of samples in the original training set. After that, we conduct circle interpolation on these pairs with random strength $\lambda \in [0, 2\pi/\alpha]$ (*cf*., Section 3.2.2).
- *Two-stage denosing*: We used BLIP-caption [7] to get captions of all images. Then, we used GPT-4-turbo [1] to summarize the captions into suffixes with the prompt:
  "I have a set of image captions that I want to summarize into objective descriptions that describe the scenes, actions, camera pose, zoom, and other image qualities present.
  My captions are: {captions}
  I want the output to be a <= 10 of captions that describe a unique setting, of the form {prefix}.
  Here are 3 examples of what I want the output to look like:
  - {prefix} standing on a branch.
  - {prefix} flying in the sky with the Austin skyline in the background.
  - {prefix} playing in a river at night.
  Based on the above captions, the output should be:"
  Then, for each denoising, we randomly sampled a suffix for the first stage. For 5-shot classification, the split ratio was 0.3; for the 10-shot classification, the split ratio was 0.1; for the long-tail classification, the split ratio was 1.0. For the sample, we used the DDIM sampler with 25 steps and 7.5 guidance scale.

[1] https://github.com/Zhicaiwww/Diff-Mix

Details of Comparison Methods: For few-shot classification, we followed the reproduction implementations (*i.e*., the timesteps of adding noise) of [15]. The translation strengths of Real-Guidance, Real-Mix, Da-Fusion, Diff-AUG, and Diff-Mix are 0.1, random one of [0.5, 0.7, 0.9], random one of [0.25, 0.5, 0.75, 1.0], 1.0, and random one of [0.5, 0.7, 0.9]. For CutMix [16] and Mixup [17], the weight decay is 1e-5, and the mixup ratios are set to 0.1 and 0.3, respectively. For long-tail classification, we directly report the results from [15].

Details of Classifier Training: For fairness, we used 0.5 as the replacement probability for all methods. Besides, we followed [15] for other settings and hyperparameters.

Hardware: All experiments are conducted on 8 NVIDIA GeForce RTX 3090 GPUs.

Computation Cost: For our Diff-II, the main computation cost is related to the number of synthetic images. Thus, consider generating one image: The interpolation calculations are very simple and cheap, which can be ignored. The two-stage denoising process just uses different prompts in two stages and does not need extra time steps. Consequently, compared with naive text-2-image generation, our Diff-II has almost the same time complexity. Specifically, it takes 9 seconds to generate one augmentation image on a single NVIDIA GeForce RTX 3090 GPU.

A.5. Background of Diffusion and DDIM Inversion

A.5.1. Diffusion

The recent surge of visual generation benefits from Diffusion models [6, 13], and high-quality images and videos are generated by sampling from Gaussian noises. Meanwhile, the downstream tasks include editing [10, 18], composing [8], and erasing [9] are also frequently researched.

A.5.2. DDIM

Denoising Diffusion Implicit Models (DDIM) [12] are a class of generative models that extend Denoising Diffusion Probabilistic Models (DDPM) by introducing a non-Markovian diffusion process. This results in a deterministic mapping between the latent variables and the data, allowing for faster sampling without compromising sample quality.

The DDIM sampling process is defined by the following iterative update rule:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}}\left(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t, c, t)}{\sqrt{\bar{\alpha}_t}}\right) + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_\theta(x_t, c, t), \tag{A12}$$

where x_t denotes the latent variable at time step t, α_t is the noise schedule parameter, $\epsilon_\theta(x_t, c, t)$ represents the noise predicted by the model at time t.

This formulation allows DDIM to generate samples in fewer steps compared to traditional diffusion models.

A.5.3. DDIM Inversion

DDIM inversion refers to the process of mapping a data sample x_0 back to its corresponding latent representation x_T through the reverse diffusion process. This is particularly useful for tasks like image editing and interpolation in the latent space.

The inversion process employs the following update equation in reverse:

$$x_t \simeq \frac{\sqrt{\bar{\alpha}_t}}{(\sqrt{\bar{\alpha}_t}-1)}(x_{t-1} - \sqrt{1-\bar{\alpha}_{t-1}}\epsilon_\theta(x_{t-1}, c, t)) + \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_{t-1}, c, t) \quad \text{(A13)}$$

starting from $t = 0$ up to $t = T$. By iteratively applying this equation, we can recover the latent code x_T corresponding to the original data sample x_0.

A.6. Additional results

A.6.1. Suffixes

The suffixes of each dataset are listed as follows:

5-shot CUB:
- standing on a tree branch.
- flying around flowers.
- standing on a post by the water.
- flying over water.
- standing on the ground.
- swimming in the water.
- sitting on a rock with a blue sky.
- perched on a branch in a tree.
- flying over water with wings spread.
- perched on a tree branch.

10-shot CUB:
- flying over water.
- standing on the ground.
- sitting on a rock.
- swimming in water.
- sitting on a bird feeder.
- standing on the beach near water.
- perched on a wire with a blue sky in the background.
- standing on a branch with tall grass in the background.
- flying in the sky with its wings spread.

5-shot Car:
- parked on a city street.
- on a white background.
- parked in a lot with green trees in the background.
- parked on a gravel road with mountains in the background.
- driving down a tree-lined road.
- parked on a black floor.
- with its doors open.
- charging at a station.
- driving on a racing track.

10-shot Car:
- parked on a road.
- parked on a gravel road.
- parked with trees in the background.
- driving down a forested road.
- driving on a dirt road in desert area.
- on display at a show.
- driving on a city street.
- parked in a garage.
- parked in front of a store with other cars.

5-shot Aircraft:
- parked on the runway.
- flying in the sky with the landing gear down.
- landing with another plane in the background.
- on the runway at an airport.
- on the tarmac with mountains in the background.
- flying in the air with the landing gear down.
- parked in a hangar with the door open.
- flying in the sky with palm trees in the background.
- flying in the sky against a blue background.
- lined up on the runway at the airport.

10-shot Aircraft:
- flying in the sky with landing gear down.
- taking off from the airport with a city in the background.
- at the tarmac of an airport with a building in the background.
- with passengers, flying in the sky.
- propeller plane on a runway, with a Honeywell sign in the background.
- airplane where workers are seen working on it in a hangar.
- plane with a green stripe on the runway.
- with people on board, floating in water.
- with a red cross on its tail and landing gear.
- jet on the runway with smoke coming out of it.

5-shot Pet:
- lying on a pillow on the floor.
- playing with a toy on the floor.
- in the grass, looking at the camera with a leash.
- on a window sill looking out.
- sitting on a couch with a stuffed animal.
- on a rock beside a person.
- running in the grass with a frisbee.

10-shot Pet:
- laying down in the grass.
- sitting on brown leather furniture.
- sitting on a couch with a dark background.
- playing with a ball in the grass.
- sitting on a windowsill, looking outdoors.
- standing on a wooden deck.
- sitting inside a cage.

- sitting on a chair with its mouth open.
- laying on a couch with a white background.

CUB-LT/IF=10:
- flying over the ocean.
- sitting on a rock by the water.
- standing in the grass.
- flying in the sky with its wings spread.
- swimming in the water.
- standing on a sandy beach.
- sitting on a wire fence.
- perched on a bird feeder in the snow.
- standing on a tree stump.

CUB-LT/IF=20:
- sitting on a rock in the water.
- perched on a branch in a tree.
- standing on the ground in the grass.
- sitting on a post by the water.
- standing on a ledge near the water.
- flying in the sky with its wings spread.
- sitting on a branch with a blurred background.

CUB-LT/IF=100:
- flying in the sky with its wings spread.
- standing on the ground in the dirt.
- sitting on a branch of a tree.
- swimming in the water.
- perched on a hand in a grassy field.
- standing on the shore of the water.
- sitting on a branch of a tree.
- sitting on a ledge by water.
- standing in the water with its reflection.

Flower-LT/IF=10:
- with water droplets on it.
- growing in a garden.
- in a close-up view.
- with a bee on it.
- in front of a water body.
- against a brick wall.
- with a butterfly on it.
- with mixed colors in a bush.

Flower-LT/IF=20:
- close-up with a dark background.
- blooming in a garden.
- growing in a pot.
- floating on water in a pond.
- arranged on a table.
- with a bee on it in the garden.
- on a tree.
- in a field with a rocky surface.
- against a blue sky.
- with a blurry background in a field.

Flower-LT/IF=100:
- with water droplets on it.
- growing in a garden.

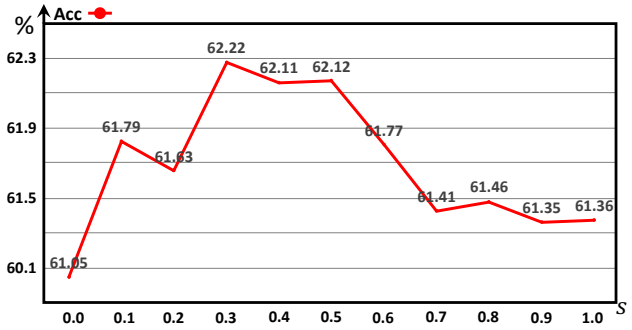

Figure A1. Classification accuracy for different split ratios. Except for the split ratio, all other settings and hyperparameters are the same with 5-shot CUB classification with ResNet50.

II	TD	CLIP Score (↑)	LPIPS (↑)	Acc (↑)
		30.73	47.9%	+2.11
✓		30.65	51.5%	+3.86
	✓	30.63	50.0%	+4.15
✓	✓	30.60	**52.7%**	**+5.52**

Table A2. **Components Ablation.** "II" is Inversion Interpolation and "TD" is Two-stage Denosing. "Acc" is the increase relative to no DA.

- in a close-up view.
- with a bee on it.
- in front of a water body.
- against a brick wall.
- with a butterfly on it.
- with mixed colors in a bush.

A.6.2. Prefdefined Metaclasses

We list the metaclass of each dataset here: CUB→"bird", Aircraft→"aircraft", Cars→"car", Pet→"animal", CUB-LT→"bird", Flower-LT→"flower".

A.6.3. More Ablation Results

Components: We ablated our key components: Inversion Interpolation (II) and Two-stage Denoising (TD). We investigated the synthetic set of 5-shot Aircraft (same setting as Sec. 4.1 with ResNet) and reported: **CLIP Score** [4] of the synthetic set; average **LPIPS** [19] between images of the synthetic set, and classification **accuracy**. The CLIP score can reflect the faithfulness of the synthetic set while the LPIPS can indicate the diversity. As shown in Table 4, the first row (w/o both II and TD) directly denoise a random noise with a prompt without suffix in one stage(*cf*., Sec. 3.3). We can see that: independently adding II or adding TD both can increase the LPIPS while nearly maintaining the CLIP Score. After adding both components together, the LPIPS further increased. This indicates that each component can significantly benefit the diversity with negligible harm to faithfulness, thus boosting higher accuracies. Then

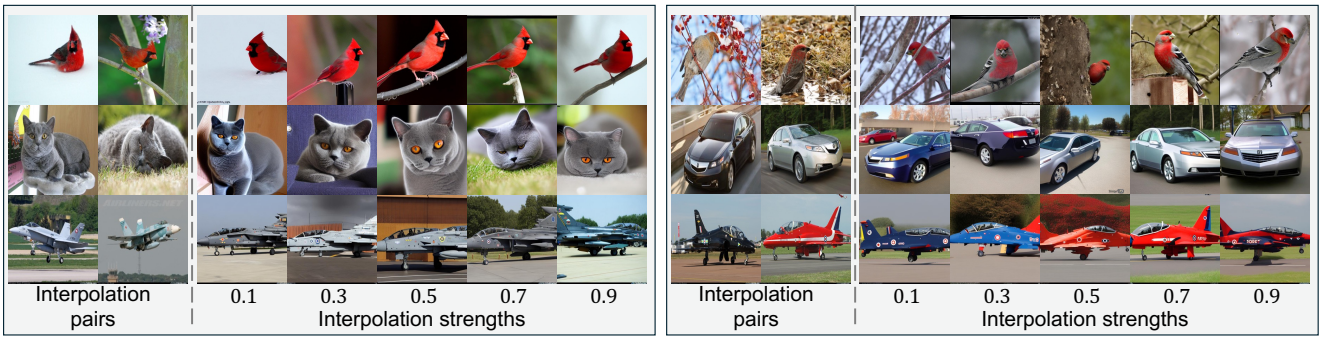

Figure A2. Synthetic images regarding different interpolation strengths (The unit is $2\pi/\alpha$).

we provide some explanations why starting with an interpolation result (the second row in Table A2) is better than a random noise (the first row in Table A2): Interpolation can not only sample some points in latent space that are not easy to sampled by standard normal distribution, but also the relative distance between these points will not be too close. This ensures the improvement of diversity. Besides, according to the characteristics of circle interpolation, these points are in the position with relatively dense semantics of the pre-trained diffusion model, thus ensuring faithfulness. Therefore, the inversion interpolation results tend to generate more diverse samples than random Gaussian noise and can finally bootstrap better classification results.

Split Ratio We ablated the *split ratio* $s \in [0, 1]$ in Figure A1. we can see that: the value of s will influence final classification accuracy. We get the best balance (when $s = 0.3$) between faithfulness and diversity.

A.6.4. Additional Visualizations

Visualizations across Different Interpolation Strength λ: As shown in Figure A2, we give our synthetic images regarding different interpolation strengths (*cf*., Sec. 3.2.2). We can see that our Diff-II can generate samples with new context while maintaining the category concept characteristics. The interpolation strengths λ can control the relative similarity between the synthetic sample and two samples of interpolation pair.

Synthetic Images in Few-shot and Long-tail Classification: we gave more synthetic images of our Diff-II used in few-shot and long-tail classification (*cf*., Figure A3).

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 3

[2] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019. 1, 2

[3] Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022. 1, 2

[4] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021. 2, 5

[5] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[7] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 3

[8] Shilin Lu, Yanzhu Liu, and Adams Wai-Kin Kong. Tf-icon: Diffusion-based training-free cross-domain image composition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2294–2305, 2023. 3

[9] Shilin Lu, Zilan Wang, Leyang Li, Yanzhu Liu, and Adams Wai-Kin Kong. Mace: Mass concept erasure in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6430–6440, 2024. 3

[10] Shilin Lu, Zihan Zhou, Jiayou Lu, Yuanzhi Zhu, and Adams Wai-Kin Kong. Robust watermarking using generative priors against image editing: From benchmarking to advances. *arXiv preprint arXiv:2410.18775*, 2024. 3

[11] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6887–6896, 2022. 1, 2

[12] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 3

[13] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 3

[14] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. *arXiv preprint arXiv:2302.07944*, 2023. 1, 2, 3

[15] Zhicai Wang, Longhui Wei, Tan Wang, Heyu Chen, Yanbin Hao, Xiang Wang, Xiangnan He, and Qi Tian. Enhance image classification via inter-class image mixup with diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17223–17233, 2024. 1, 2, 3

[16] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 1, 2, 3

[17] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 1, 2, 3

[18] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3836–3847, 2023. 3

[19] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2, 5

Figure A3. More synthetic images of our Diff-II in few-shot and long-tail classification.