Enhance Image Classification via Inter-Class Image Mixup with Diffusion Model

Supplementary Material

(a) Filtered samples

(b) Preserved samples

Figure 8. Examples of (a) filtered samples with the lowest 10% confidence scores and (b) preserved samples in the CUB dataset.

7. Appendix

This appendix is organized as follows:

- In Sec. 7.1, we elaborate on the details of data cleaning design for Diff-Mix.
- In Sec. 7.2, additional visualizations are presented, including the visualization of attention maps and failure examples of complex datasets.
- In Sec. 7.3, few-shot classification results on a general dataset Pascal is provided [14].
- In Sec. 7.4 and Sec. 7.5, implementation details and latency considerations are presented, respectively.

7.1. Data-Cleaning Strategy

Due to the inherent differences in contour and size between the two classes, there is a higher risk of producing less realistic images during inter-class editing. We employ a simple data cleaning strategy that utilizes CLIP [43][4] as the filtering criterion. Specifically, we construct a positive caption, "a photo with a [metaclass] on it", and a negative caption, "a photo without a [metaclass] on it", and evaluate the synthetic data's confidence score towards the positive caption. We filter out the 10% of samples with the lowest confidence scores (we do not synthesize an additional 10% samples after data cleaning), and a subset of the filtered samples is displayed in Fig. 8. The preserved samples constitute the synthetic dataset that participates in the training process of downstream classification tasks.

[4] https://huggingface.co/openai/clip-vit-base-patch32

7.2. Visualizations

Visualizations of attention maps. In Section 3.4, we have shown that Diff-Mix can perform foreground editing while preserving most of the layout of the reference image. To support the claim, we provide evidence that SD can offer weak segmentation through textual conditions and achieve realistic foreground editing. We present visualizations of attention maps in Figure 9 for different datasets. The identifier, class descriptor (e.g., "`bird`", "`car`") and the "`<eot>`" token, which contains the global semantic information, tend to attend to the foreground part in the reference image. For example, the mentioned tokens primarily emphasize the bird rather than the tree branches (refer to Row 1 in the figure). This suggests that textual guidance, can offer a robust foreground prior, aiding in effective foreground editing at intermediate strengths. We posit that this characteristic ensures the generation of challenging samples where the foreground is replaced by the target concept when employing Diff-Mix for inter-class editing.

Visualizations on more datasets. In Fig. 12, we illustrate the editing process of Diff-Mix with varying strength $s \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$ across five datasets. It is worth noting that for strength $s = 1$, the translated images still exhibit a certain degree of similarity to the reference images. This phenomenon may be attributed to the last time-step not guaranteeing a zero signal-to-noise ratio, preserving the style and layout of the reference images [31]. Particularly, when the foreground is distinct against a simple background, Diff-Mix tends to generate high-quality interpolated images. We also observe that for more complex datasets, such as Stanford Dogs, where the foreground is less clear, and there are multiple concepts in a single image, unrealistic images tend to be generated, as seen in Fig. 13 (a) and (b). For general dataset Pascal [14], the dramatic differences in contour and size between two distinct classes lead to the generation of more unrealistic images (*e.g.*, "`bus`" $\longrightarrow$ "`cat`"), especially at intermediate strengths (*e.g.*, 0.7), as seen in Fig 13 (c) and (d).

Real-Gen versus Diff-Gen. To illustrate the distribution gap between domain-specific datasets and the pre-trained T2I model, as well as to demonstrate how fine-tuning can significantly mitigate this gap, we present a comparison in Fig. 14. It is worth noting that Real-Gen sometimes fails to generate correct concepts based on the terminology name of the target class (see "`photo of a chuck`

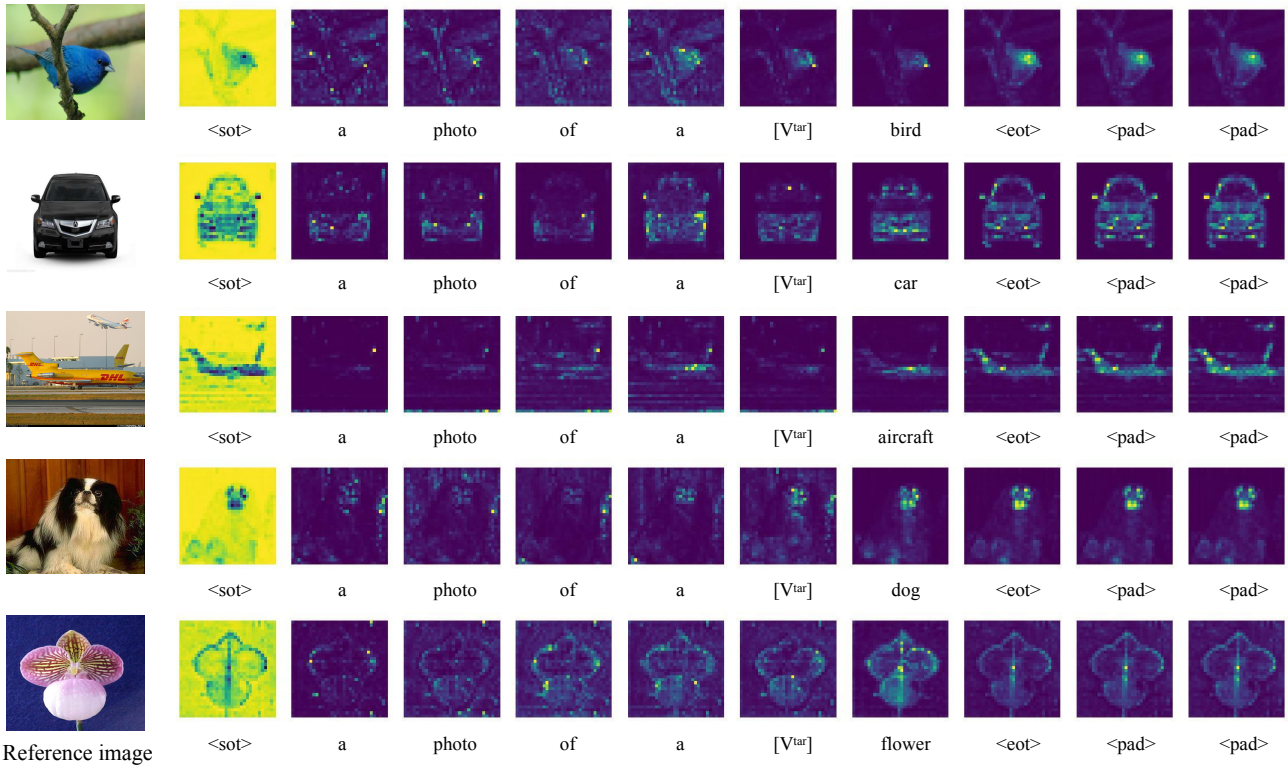


Figure 9. Visualizations of attention maps are shown in different rows for various datasets: CUB (Row 1) [61], Stanford Cars (Row 2) [27], FGVC Aircraft (Row 3) [34], Stanford Dogs (Row 4) [26], and Oxford Flowers (Row 5) [38]. These attention maps were generated during inter-class editing using Diff-Mix.

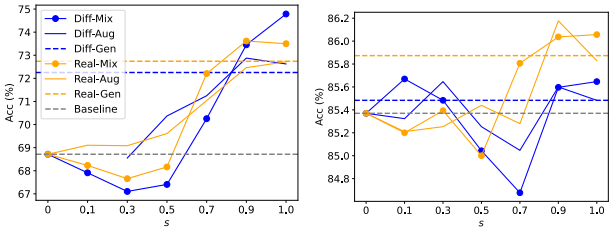
`will widow`" in panel (a)). Diff-Gen tends to generate more faithful outputs, it is noted that the majority of the generated images exhibit a similar layout and closely resemble the training samples. This resemblance is especially pronounced for those training samples characterized by a prominent foreground and a simple background.

Real-Mix versus Diff-Mix. In Fig. 15, we compare the generated samples between Real-Mix and Diff-Mix. We observe that, by conditioning on the reference image, Real-Mix accurately captures the semantic meaning of the terminology name (see "`chuck will widow`" in panel (a)). This feature of Real-Mix is consistent with its superior performance in few-shot classification, as depicted in Fig. 6. Additionally, we observe that Diff-Mix achieves more precise foreground editing (refer to Fig. 15 (c) and (d)). This enhanced accuracy can be attributed to the class descriptor maintaining its focus on the semantic content without being diverted to other extraneous information.

7.3. Experiments

Few-shot classification in Pascal.

In Sec. 7.2, We have demonstrated that inter-class editing for general datasets tends to produce unrealistic images due to the visual gaps between two classes. Here, we present 5-shot and all-shot classification results in Figure 10 for different expansion strategies on the general dataset Pascal [14]. Originally, Pascal is an object class recognition dataset containing 11,530 images and 6,929 object segmentation masks. We construct it into a classification dataset, following the setting of Da-fusion [58], resulting in a training split of 1,464 and a validation split of 1,449 for 20 general classes (*e.g.*, cat and boat). The main observation is that inter-class augmentation tends to be less effective for this general dataset, especially as the shot number increases (compare X-Aug and X-Mix in the figure). The effective-



(a) 5-shot (b) all-shot

Figure 10. 5-shot and all-shot classification results in Pascal.

hyperparameter	DB	TI	TI+DB
Base Model	Stable Diffusion-v1.5	Stable Diffusion-v1.5	Stable Diffusion-v1.5
Optimized	U-Net(LORA)	$[V^i]$	U-Net(LORA) + $[V^i]$
Optimization Steps	35000	35000	35000
Batchsize	8	8	8
Input Resolution	512× 512	512× 512	512× 512
Learning Rate	5e-5	5e-5	5e-5
Placeholder Token	-	$[V^i]$	$[V^i]$
LORA Rank	10	-	10
# if inference steps (T)	25	25	25
Guidance Scale	7.5	7.5	7.5
Noise Scheduler	DPMsolver++[33]	DPMsolver++[33]	DPMsolver++[33]

Table 7. Hyperparameters. This tables summarizes the hyperparameter settings of different fine-tuning strategies.

Dataset	# of classes	# of training	# of val.	Source
CUB	200	5994	5794	Huggingface.co
FGVC Aircraft	100	3334	3333	Huggingface.co
Oxford Flowers	102	4070	4119	Huggingface.co
Stanford Dogs	120	12000	8580	vision.stanford.edu
Stanford Cars	196	8144	8041	Huggingface.co

Table 8. Statistics of datasets.

ness of fine-tuning also decreases, with Real-Gen consistently outperforming Diff-Gen. This suggests that the pre-trained SD is capable of generating sufficiently diverse and faithful samples for these coarse concepts. Diff-Mix excels in handling domain-specific scenarios, where smaller differences in contour and layout between two classes are presented.

7.4. Implementation Details.

Diff-Mix. Diff-Mix comprises two stages: the fine-tuning stage and the sampling stage. The implementation details of Diff-Mix for three different fine-tuning strategies are depicted in Table 7. Note that our fine-tuning strategy heavily relies on the diffuser [60] repository. For DB and TI+DB, we only fine-tune the residual LORA matrices in attention modules in the U-Net. Please note that in the original Dreambooth [46] paper, an unlearnable identifier was introduced to represent user-specific concepts in concept learning. However, in our implementation, we have opted not to use the identifier and have implemented it as a straightforward fine-tuning of text-to-image models. All fine-tuning and sampling processes are conduct on 4 RTX3090 GPUs.

Datasets. We list the statistics of the datasets involved in our experiments in Table 8.

Conventional classification. The hyperparameter settings for the CUB dataset are presented in Table 9. To reproduce Real-filtering (RF) [19], a subset is derived from Real-Gen through data cleaning, as detailed in Section 7.1. Real-guidance (RG) [19] augments the dataset with low-strength intra-class editing, akin to Real-Aug with a strength parameter $s = 0.1$. For the replication of Da-fusion [58], we fine-tune the synthetic data (SD) using our TI strategy over 35,000 steps, with translation strengths randomly selected from the set $0.25, 0.5, 0.75, 1.0$. For CutMix and Mixup, the weight decay is 1×10^{-5}, and the mixup ratios are set to 0.1 and 0.3, respectively.

hyperparameters	ResNet50	ViT-B/16
Source	torchvision	torchvision
# of parameters	25.5M	86.6M
Pre-trained	ImageNet1K	ImageNet21K
Fine-tuned	-	ImageNet1K
Input Resolution	448×448	384×384
Batchsize	64	32
Epochs	128	100
Optimizer	SGD	SGD
Learning Rate	0.02	0.001
Momentum	0.9	0.9
Weight Decay	5e-5	5e-5
Label Smoothing	0.9	0.9

Table 9. hyperparameters. This table summarizes the hyperparameter settings for CUB using two visual backbones in our conventional classification task.

Few-shot classification. The few-shot classification is conducted on CUB with varying shot numbers: 1, 5, 10, and all. The comparison methods encompass: (1) inter-class augmentation strategies, namely Diff-Mix and Real-Mix, (2) intra-class augmentation strategies, namely Diff-Aug and Real-Aug, and (3) distillation-based methods, Diff-Gen and Real-Gen. The backbone model used is ResNet50 with an input resolution of 224^2. We employ the same hyperparameters as in the conventional setting, as detailed in Table 9, albeit with a larger batch size (256) and a higher learning rate (0.05). All experiments are conducted with three trials, and the average results are reported.

Long-tail classification. Thanks to the authors of CMO [39], the reproduced long-tail results are built upon its open-

Dataset	# of classes	# of training	Imbalance Factor (IF)
CUB-LT	200	{1242, 1798, 2238}	{100,50,10}
Flower-LT	102	{847, 1238, 1532}	{100,50,10}

Table 10. Statistics of long-tail datasets CUB-LT and Flower-LT.

source git repository [5]. To construct the imbalanced dataset, an imbalance factor is introduced to control the imbalance level. The imbalance factor ρ is defined as $\rho = \frac{\max_k\{n_k\}}{\min_k\{n_k\}}$, where n_k is the number of samples in the k-th class. Specifically, given a normal dataset, we first sort the classes based on the number of images within classes in descending order, and use k' to denote the sorted class index. A subset of images is randomly sampled from each class to achieve the desired imbalance, ensuring that the number of images for each class corresponds to the calculated target. The number of sampled images is determined by,

$$n_{k'} = \max\left(\bar{n} \times \left(\frac{1}{\rho}\right)^{\frac{k}{N-1}}, 1\right) \tag{4}$$

where $\bar{n}$ is the averaged number of images for each class, and N is the total number of classes. The statistics of constructed CUB-LT and Flower-LT datasets can be found in Table 10. To uniformize the distribution of imbalanced real data, we first fix the number of iterative samples within each epoch and replace real samples with synthetic data with a 50% probability. Note that the synthetic data conforms to the distribution specified by Eq. 4 with reversed class indices, thereby generating more synthetic images for tail classes. We maintain a constant number of training epochs and learning rate for both synthesis-free and synthesis-based approaches to ensure a fair comparison. We further present accuracy results for three distinct subsets when IF is 100: Many-shot classes (classes with over 20/30 training samples for CUB-LT/Flower-LT), medium-shot classes (classes with 5-20/10-30 samples for CUB-LT and Flower-LT), and few-shot classes (classes with fewer than 5/10 samples for CUB-LT/Flower-LT).

7.5. Latency

Compared to non-generative augmentation methods, Diff-Mix's implementation incurs additional computational overhead during fine-tuning and data sampling. For instance, when working with the CUB dataset, which contains approximately 6,000 training samples, the fine-tuning process is completed in about 3 hours. This duration is achieved using an input resolution of 512×512 on 4 NVIDIA RTX 3090 GPUs with a total batch size of 8. During sampling, synthetic samples are generated at the same resolution with a total of $T = 25$ reverse steps. The throughput across various translation strengths is evaluated

[5] https://github.com/naver-ai/cmo

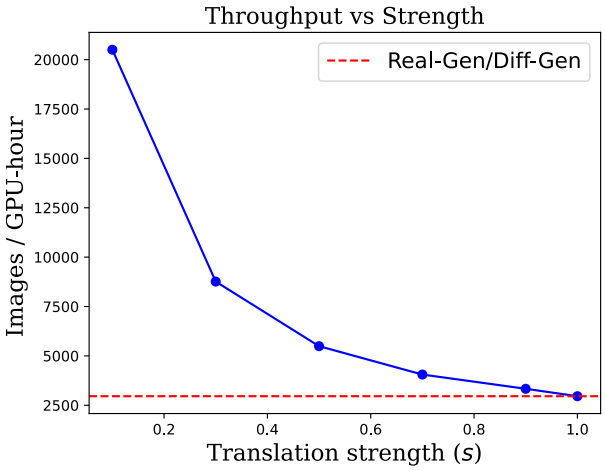


Figure 11. Sampling throughput across various translation strengths in a single RTX 3090 GPU.

Method	Translation strength s	Images per GPU-hour
Real-filtering	$\in\{1.0\}$	2,957
Real-guidance	$\in\{0.1\}$	20,502
Da-fusion	$\in\{0.25,0.5,0.75,1.0\}$	4,952
Diff-Mix	$\in\{0.5,0.7,0.9\}$	4,179

Table 11. Comparison of sampling throughput of different expansion strategies.

in Fig. 11, and a throughput comparison with other synthesis strategies is provided in Table 11. While Diff-Mix is more efficient than generating data from scratch, it is less so than low-strength editing (*e.g.*, Real-guidance). For generating synthetic data for the CUB dataset with a 5x multiplier (resulting in approximately 30,000 images), the process requires roughly 2.5 hours using 4 NVIDIA RTX 3090 GPUs.

8. Limitations

Our inter-class augmentation method shows less effective when applied to general datasets that encompass a broad spectrum of concepts. We are optimistic, however, that integrating an image inpainting strategy or confining Diff-Mix to operate among adjacent classes could address this limitation. Moreover, the current annotation strategy is determined empirically and lacks a robust theoretical foundation, which may limit the generalizability of the strategy.

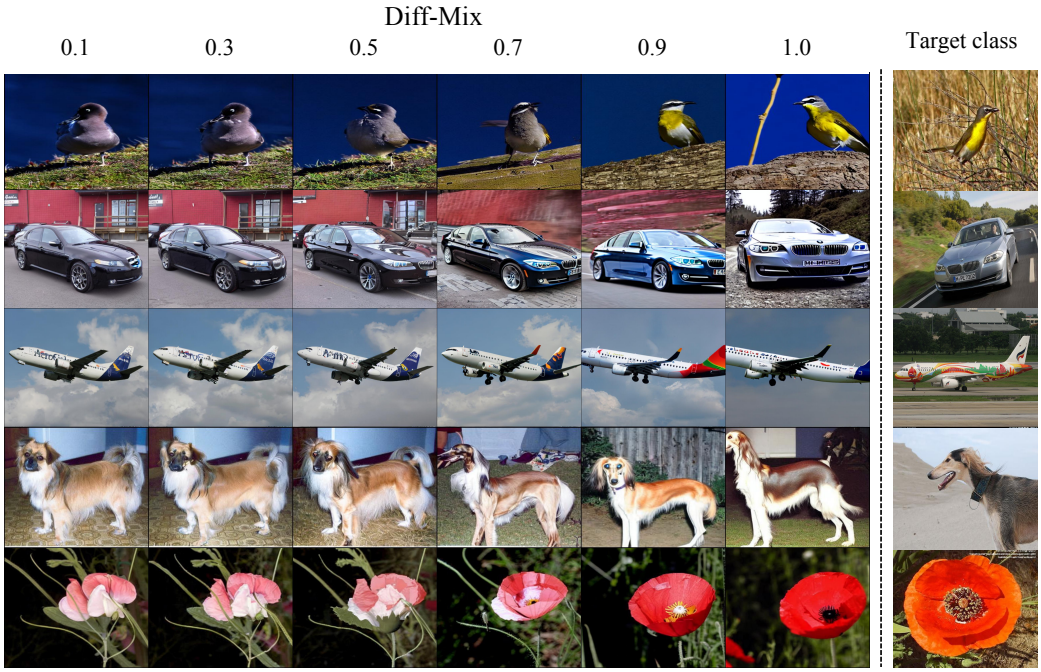


Figure 12. Examples of image generated using Diff-Mix with varying translation strengths in CUB (Row 1), Stanford Cars (Row 2), FGVC Aircraft (Row 3), Stanford Dogs (Row 4) , Oxford Flowers (Row 5).

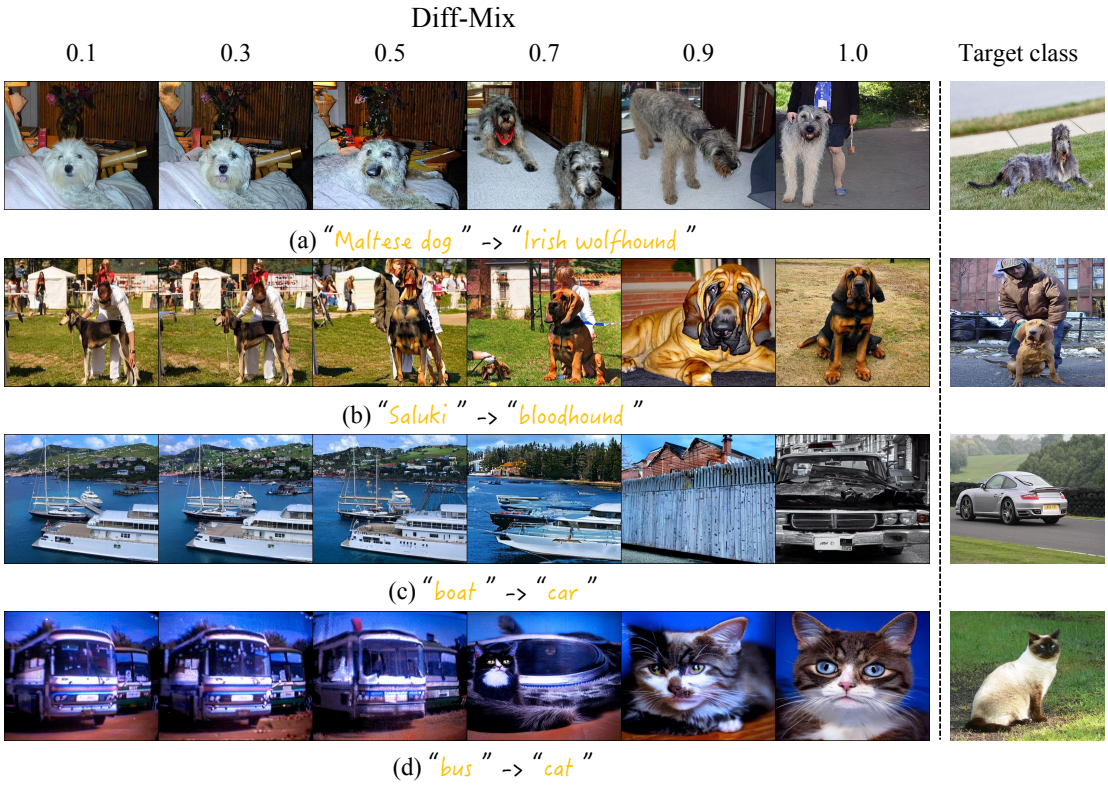


Figure 13. Failure examples generated using Diff-Mix with varying translation strengths are shown in panels (a) and (b) for the complex dataset Stanford Dogs, and in panels (c) and (d) for the general dataset Pascal [14].

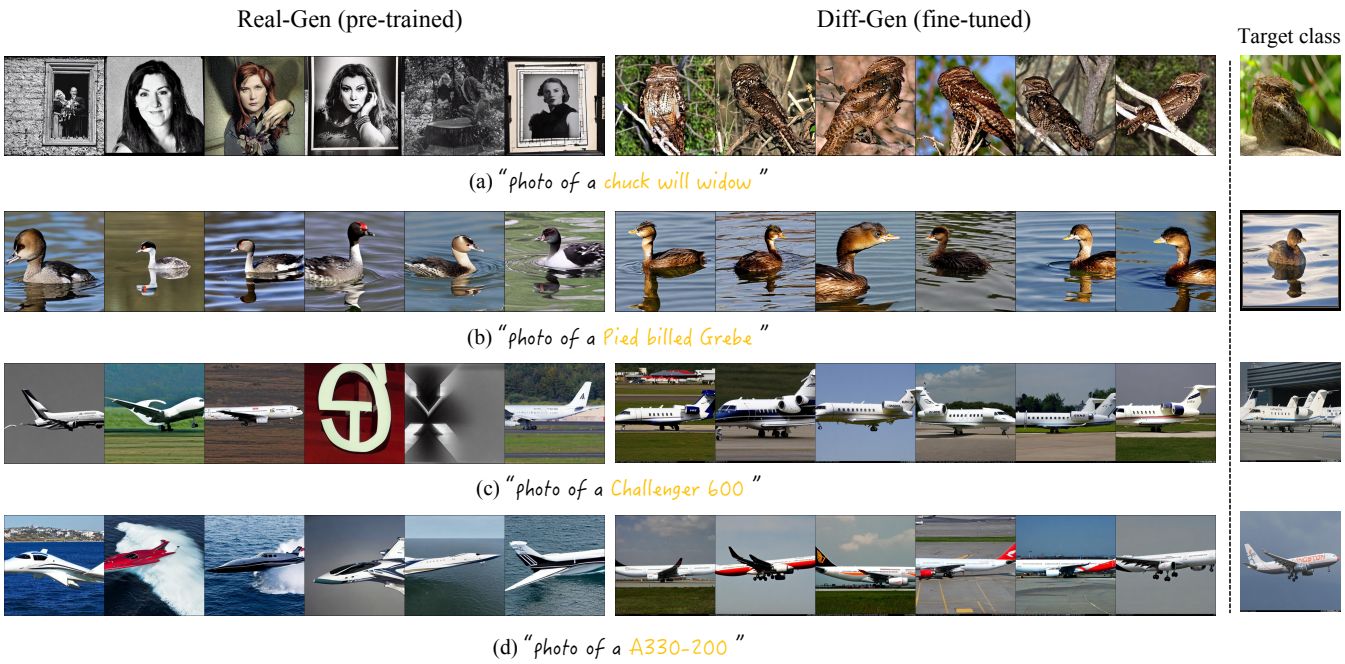


Figure 14. Examples of image generated using Real-Gen and Diff-Gen. The prompts are formatted as "`photo of a [terminology name]`" for Real-Gen and "`photo of a [V^i] [metaclass]`" for Diff-Gen. Panels (a) and (b) depict the samples of CUB dataset, while panels (c) and (d) depict the samples of FGVC Aircraft dataset.

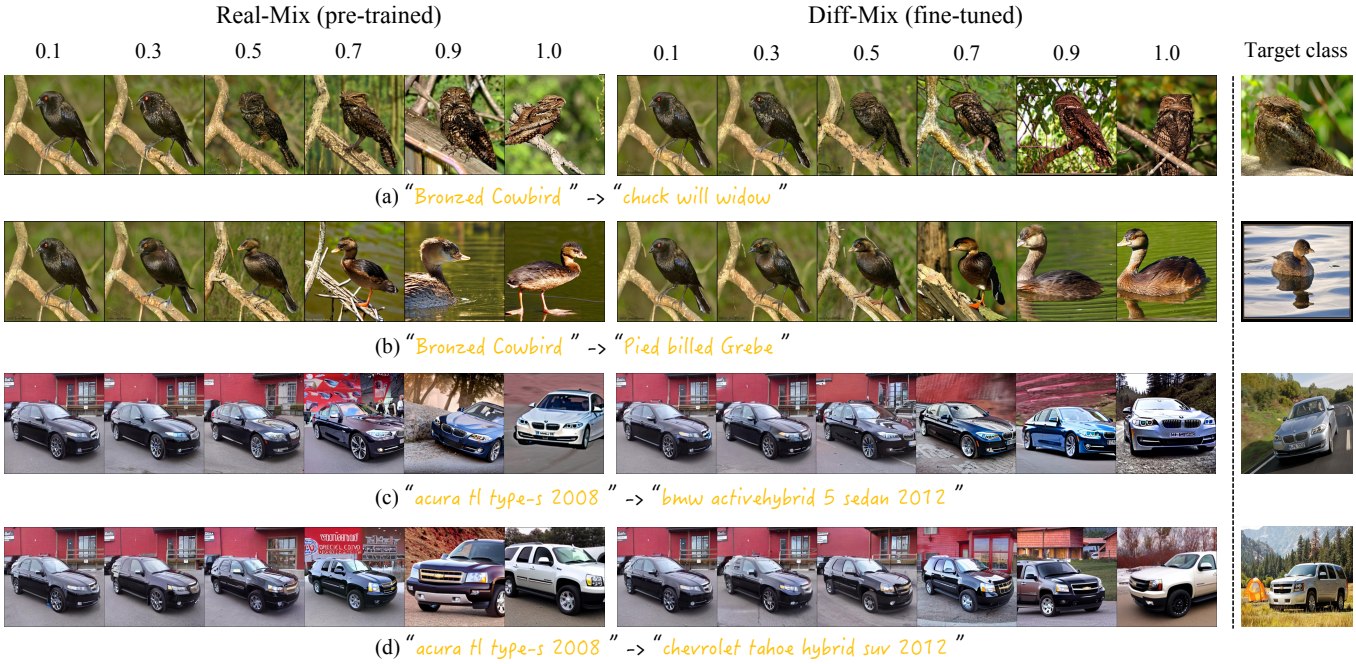


Figure 15. Examples of image generated using Real-Mix and Diff-Mix with varying translation strengths. The prompts are formatted as "`photo of a [terminology name]`" for Real-Mix and "`photo of a [V^i] [metaclass]`" for Diff-Mix. Panels (a) and (b) depict the samples of CUB dataset, while panels (c) and (d) depict the samples of Stanford Car dataset.