Supplementary Materials for SGD-Mix: Enhancing Domain-Specific Image Classification with Label-Preserving Data Augmentation

Yixuan Dong[1*] Fang-Yi Su[1,2*]
Jung-Hsien Chiang[1†]
[1]National Cheng Kung University, Tainan, Taiwan
[2]Harvard Medical School, Boston, MA, USA
radondong@iir.csie.ncku.edu.tw, fang-yi_su@hms.harvard.edu, jchiang@mail.ncku.edu.tw

Overview

This supplementary material elaborates on key aspects supporting the main text. Sections 1 and 2 provide a detailed explanation of the issues with Diff-Mix [30] and DiffuseMix [8], including Diff-Mix's limited generalizability and unstable label assignment, as well as DiffuseMix's semantic drift and limited diversity in augmentation. Section 3 details SGD-Mix's experimental setup, including datasets, backbones, and implementation specifics. Section 4 provides visualizations of the SGD-Mix process.

1. Limitations of Diff-Mix

Diff-Mix [30] proposes an innovative approach by leveraging inter-class images as reference images for a domain-specific fine-tuned diffusion model. It assigns labels to generated images using a nonlinear mixing formula:

$$\tilde{y} = (1 - s^{\gamma})y^{i} + s^{\gamma}y^{j}, \tag{1}$$

where $s \in [0, 1]$ denotes the translation strength, γ is a nonlinearity hyperparameter (empirically suggested as ≈ 0.5), and y^i and y^j represent the labels of the reference (source) and target image classes, respectively. While this method shows promise in some cases, its reliance on s as the sole determinant of label composition introduces significant limitations, as highlighted in the main text.

First, the label mixing strategy lacks generalizability. The effectiveness of Diff-Mix heavily depends on the characteristics of the underlying diffusion model, such as the noise scheduler [17] (e.g., linear or cosine), and dataset-specific factors. To illustrate this, consider the forward diffusion process [6, 26]:

$$X_s = \sqrt{\bar{\alpha}_s}\, X_0 + \sqrt{1 - \bar{\alpha}_s}\, \varepsilon, \tag{2}$$

where X_0 is the original reference image, $\varepsilon \sim \mathcal{N}(0, I)$ is Gaussian noise, and $\bar{\alpha}_s = f_{\text{scheduler}}(s)$ is a monotonically decreasing function determined by the noise scheduler. For instance, a linear scheduler defines $\bar{\alpha}_s$ as a linear decay, while a cosine scheduler [17] adopts a nonlinear, cosine-inspired mapping. As s increases, $\bar{\alpha}_s$ decreases, introducing more noise and reducing the information retained from X_0.

To quantify this retention, we analyze the mutual information $I(X_0; X_s)$. Assuming $X_0 \sim \mathcal{N}(0, \Sigma_{X_0})$ for tractability (or a local Gaussian approximation), the joint distribution of (X_0, X_s) under Eq. (2) is Gaussian, with covariance:

$$\begin{pmatrix} X_0 \\ X_s \end{pmatrix} \sim \mathcal{N}\left(0, \begin{pmatrix} \Sigma_{X_0} & \sqrt{\bar{\alpha}_s}\Sigma_{X_0} \\ \sqrt{\bar{\alpha}_s}\Sigma_{X_0} & \bar{\alpha}_s\Sigma_{X_0} + (1 - \bar{\alpha}_s)I \end{pmatrix}\right). \tag{3}$$

For simplicity, let $X_0 \sim \mathcal{N}(0, I)$[1]. The mutual information is then:

$$I(X_0; X_s) = \frac{1}{2} \log\left(1 + \frac{\bar{\alpha}_s}{1 - \bar{\alpha}_s}\right), \tag{4}$$

where $\frac{\bar{\alpha}_s}{1-\bar{\alpha}_s}$ represents the signal-to-noise ratio (SNR). Since $\bar{\alpha}_s$ decreases with s, $I(X_0; X_s)$ also decreases, but the rate of this decline varies with the scheduler's functional form. Consequently, a fixed s yields inconsistent levels of retained information across different schedulers, undermining the universal applicability of Eq. (1). This sensitivity complicates label assignment when diverse diffusion backbones or datasets are used. For example, as shown in Figure 1 of the main text, a Diff-Mix-generated image may be assigned a label composition exceeding 0.8 for the source class and below 0.2 for the target class under certain settings, resulting in unclear or uninformative labels.

Second, the inherent stochasticity of diffusion models exacerbates issues at high translation strengths. [15] As s grows, X_s approaches pure noise, broadening the true

*These authors contributed equally.
†Corresponding author.

[1]This assumption preserves qualitative insights, as Σ_{X_0} scaling only adjusts the SNR without altering the dependency structure.

posterior $p(X_0|X_s)$:

$$p(X_0|X_s) = \frac{p(X_s|X_0)p(X_0)}{p(X_s)}. \quad (5)$$

In practice, however, we rely on an approximated posterior $p_\theta(X_0|X_s) \approx p(X_0|X_s)$, learned by a parameterized model. The deviation between these distributions can be expressed as:

$$p_\theta(X_0|X_s) = p(X_0|X_s) + \delta_\theta(X_s), \quad (6)$$

where $\delta_\theta(X_s)$ denotes the model error. This error's impact grows with s, as:

$$\|p_\theta(X_0|X_s) - p(X_0|X_s)\| \leq f(s)\|\delta_\theta(X_s)\|, \quad (7)$$

with $f(s)$ increasing in s. At large s, even the same s can produce highly variable images due to this amplified stochasticity, resulting in semantic content that does not match the assigned labels. Assigning labels based solely on s thus risks significant mismatches with the generated images' semantics, as discussed in the main text. Tuning γ extensively cannot fully mitigate these issues, which arise from the diffusion process, scheduler choice, and model approximation limitations. While intra-class references (e.g., Diff-Aug [30]) avoid label-semantic mismatches, they sacrifice diversity by producing samples too similar to the reference.

2. Limitations of DiffuseMix

DiffuseMix [8] adopts a distinct strategy, transforming a source image I_i with a conditional prompt p_j via a diffusion model G:

$$\hat{I}_{ij} = G(I_i, p_j). \quad (8)$$

The result is split using a binary mask $M_u \in \{0,1\}^{h\times w\times c}$, combining half of $\hat{I}_{ij}$ with half of I_i, then blending with a fractal image F_v:

$$A_{ijuv} = \lambda F_v + (1-\lambda)\left[\hat{I}_{ij} \odot M_u + I_i \odot (1 - M_u)\right]. \quad (9)$$

Despite its creativity, this approach faces notable challenges, as outlined in the main text.

First, semantic drift significantly impairs DiffuseMix. The transformation $\hat{I}_{ij}$ often diverges semantically from I_i, especially under strong prompts, because the method does not explicitly disentangle the foreground and background or separate semantics from style. This drift is quantifiable as:

$$\Delta S = \|S(\hat{I}_{ij}) - S(I_i)\|_2, \quad (10)$$

where $S(\cdot)$ denotes a semantic representation, and a larger ΔS indicates greater divergence. Assigning I_i's label to A_{ijuv} without adjustment heightens the risk of mislabeling, as shown in Figure 1, and further emphasized in both the main text and DEADiff [21].

Second, diversity is constrained. Retaining half of I_i in A_{ijuv} limits the extent of transformation, while F_v adds texture but not substantial semantic variety. Additionally, similar to CutMix [32], the direct stitching of the stylized result $\hat{I}_{ij}$ with I_i using a binary mask introduces unnatural boundaries, further compromising the visual coherence of the generated images. This restricts the method's ability to balance diversity, faithfulness, and label clarity effectively.

Through this detailed analysis of Diff-Mix and DiffuseMix, we identify key shortcomings in existing diffusion-based augmentation methods. These insights motivate our proposed framework, SGD-Mix, which employs saliency-guided strategies and a fine-tuned diffusion model to ensure semantic consistency, enhance diversity, and maintain label accuracy. SGD-Mix is designed for broad applicability, addressing the instability of inter-class label mixing and semantic drift while providing a robust solution consistent with the objectives outlined in the main text.

3. Experimental Details

This section elaborates on the experimental settings for Section 6 of the main text, covering datasets, backbones, and SGD-Mix implementation specifics for four tasks: fine-grained vision classification, long-tail classification, few-shot classification, and background robustness. Experiments were run on 1 NVIDIA RTX 4090 GPU.

3.1. Dataset Sources

We tested SGD-Mix on various domain-specific datasets, detailed in Table 1. These span the tasks outlined in the main text.

3.1.1. Long-Tail Dataset Construction (CUB-LT and Flower-LT)

Consistent with the experimental setups in CMO [20] and Diff-Mix [30], we construct long-tail datasets [1, 12, 20] (CUB-LT [24], Flower-LT) derived from CUB [29] and Oxford Flowers [18], respectively, with imbalance factors (IF = 100, 50, 10) to simulate real-world class imbalance. We sort classes by their original sample counts in descending order, assigning each class an index k (where $k = 0$ for the most frequent class and $k = N - 1$ for the least frequent, N being the total number of classes). The number of real samples $\mathbf{m}_k$ for class k is determined using an exponential decay function:

$$\mathbf{m}_k = \max\left(\bar{m} \times \left(\frac{1}{\beta}\right)^{\frac{k}{N-1}}, 1\right) \quad (11)$$

Here, $\bar{m}$ is the average number of samples per class in the original dataset, and β is the imbalance factor (IF), controlling the ratio between the maximum and minimum class sizes ($\beta = \frac{\max\{\mathbf{m}_k\}}{\min\{\mathbf{m}_k\}}$). For each class, we randomly sample

Table 1. Statistics of datasets.

Dataset	# Classes	# Train	# Val	Source
CUB [29]	200	5,994	5,794	huggingface.co
FGVC Aircraft [14]	100	3,334	3,333	huggingface.co
Oxford Flowers [18]	102	4,070	4,119	huggingface.co
Stanford Cars [11]	196	8,144	8,041	huggingface.co
Stanford Dogs [10]	120	12,000	8,580	vision.stanford.edu
CUB-LT	200	{1,242, 1,798, 2,238}	5,794	Derived from CUB
Flower-LT	102	{847, 1,238, 1,532}	4,119	Derived from Oxford Flowers

$\mathbf{m}_k$ images from the original training set, ensuring at least one sample per class. This results in training set sizes of {1,242, 1,798, 2,238} for CUB-LT and {847, 1,238, 1,532} for Flower-LT under IF = {100, 50, 10}, respectively, while validation sets remain unchanged.

To address class imbalance in the long-tail experiments (Section Long-Tail Classification), we adopt the SYNAuG [31] approach to uniformize the data distribution using synthetic samples. We fix the total number of iterations per epoch to match the size of the original training set, and replace real samples with synthetic ones at a probability of 0.5. The number of synthetic samples $\mathbf{s}_k$ for class k is calculated to balance the distribution by reversing the real sample decay:

$$\mathbf{s}_k = \max\left(\bar{m} \times \left(\frac{1}{\beta}\right)^{\frac{N-1-k}{N-1}}, 1\right) \tag{12}$$

Here, $\bar{m}$ remains the average number of samples per class from the original dataset, and the exponent $\frac{N-1-k}{N-1}$ inverts the decay of Eq. (11), generating more synthetic samples for tail classes (high k) to achieve a uniform effective distribution. For evaluation, we categorize classes into three subsets based on real sample counts under IF = 100:

- For CUB-LT: Many (> 20 samples), Medium (5–20 samples), Few (< 5 samples).
- For Flower-LT: Many (> 30 samples), Medium (10–30 samples), Few (< 10 samples).

3.1.2. Waterbird Dataset Construction

The Waterbird dataset is an out-of-distribution test set designed to evaluate background robustness, constructed by combining foreground objects from CUB with background scenes from Places [34]. Following a similar methodology to prior work [23, 30], we segment bird foregrounds (waterbirds and landbirds) from CUB images. These foregrounds are then pasted onto randomly selected background images from Places, categorized as either "water" (e.g., oceans, lakes) or "land" (e.g., forests, fields). This process creates four compositional groups: (waterbird, water) with 642 samples, (waterbird, land) with 642 samples, (landbird, land) with 2,255 samples, and (landbird, water) with 2,255 samples. The resulting dataset contains 5,794 validation samples (642 + 642 + 2,255 + 2,255 = 5,794), with no separate training set, as models are trained on CUB and synthetic data, then evaluated on Waterbird to assess generalization under background shifts.

Table 2. Backbone settings.

Parameter	ResNet50 [4]	ViT-B/16 [2]
Source	torchvision	torchvision
Pre-trained	ImageNet1K	ImageNet21K
Fine-tuned	-	ImageNet1K
Resolution	448/224	384
Batch Size	64/256	32
Epochs	128	100
Optimizer	SGD	SGD
LR	0.02/0.05	0.001
Weight Decay	5e-5	5e-5
Momentum	0.9	0.9
Label Smoothing	0.9	0.9

3.2. Backbones

We employed ResNet50 [4] and ViT-B/16 [2], with settings in Table 2. Both share common hyperparameters unless specified.

3.3. Implementation Details of SGD-Mix

SGD-Mix comprises saliency-based target selection, saliency-guided mixing, and diffusion-based refinement (Section 5).

Table 3. Diffusion model settings.

Parameter	Setting
Base Model	Stable Diffusion v1.5 [27]
Optimized	U-Net (LoRA [7]) + $[v^i]$ [3]
Optimization Steps	35,000
Batch Size	8
Resolution	512
LR	5e-5
Guidance Scale	7.5
LoRA Rank	10
Inference Steps	25
Scheduler	DPMsolver++ [13]

3.3.1. Fine-tuning the Diffusion Model

Stable Diffusion v1.5 [27] is fine-tuned using DreamBooth [22] with LoRA [7] and Textual Inversion [3] (Table 3). Fine-tuning CUB (5,994 samples) takes roughly 3.5 hours on 1 RTX 4090 GPU with batch size 2 and resolution 512×512.

3.3.2. Data Synthesis

Synthetic data generation uses a batch size of $N = 50$. Translation strength S and replacement probability p vary by task (Section Experiments). We compute saliency maps using a gradient-based method (Grad-CAM) [25] and cache the results to facilitate processing. Sampling throughput is approximately 500 images/GPU-hour for $S = 0.5$. When $N = 10$, throughput rises to about 3,000 images/GPU-hour, and when $N = 30$, it is around 1,000 images/GPU-hour. In contrast, Diff-Mix achieves roughly 5,000 images/GPU-hour under similar conditions.

3.3.3. Saliency Maps

Gradient-based (Grad-CAM) [25] saliency maps are normalized to [0, 1] using MinMax scaling and thresholded with Otsu's [19] method, except in ablation studies (Section 7, Q3).

3.4. Sub-Experiments

3.4.1. Fine-Grained Vision Classification

We evaluated our approach on five datasets using ResNet50 (448×448) and ViT-B/16 (384×384), with $S \in \{0.5, 0.7, 0.9\}$, an expansion multiplier of 5, label smoothing [16] (0.9), and replacement probability $p = 0.1$. We compared SGD-Mix with generative methods (Diff-Mix [30], DiffuseMix [8], Real-Filtering [5], Real-Guidance [5], Da-Fusion [28]) and non-generative methods (Mixup [33], CutMix [32], GuidedMixup [9]). The settings were kept consistent with SGD-Mix, except for specific adjustments: $\gamma = 0.5$ for Diff-Mix, $S = 0.1$ for Real-Guidance, randomized $S \in \{0.25, 0.5, 0.75, 1.0\}$ for Da-Fusion, mixup ratio 0.1 for CutMix and mixup ratio 0.3 for Mixup.

3.4.2. Long-Tailed Classification

We evaluated our approach on CUB-LT and Flower-LT using ResNet50 (224×224), where SYNAuG [31] uniformizes data with $S = 0.7$ and $p = 0.5$. We compared SGD-Mix with generative methods (Diff-Mix [30], Real-Mix [30], Real-Gen [30]) and non-generative methods (CMO [20], CMO+DRW [1]). The settings were kept consistent with SGD-Mix, except for the specific adjustment of $\gamma = 0.5$ for Diff-Mix and Real-Mix.

3.4.3. Few-Shot Classification

We evaluated our approach on CUB using ResNet50 (224×224), testing 1/5/10/all-shot scenarios with $S = 0.9$, an expansion multiplier of 5, and $p \in \{0.5, 0.3, 0.2, 0.1\}$. We compared SGD-Mix with generative methods (Diff-Mix [30], Diff-Aug [30], Diff-Gen [30]). The settings were kept consistent with SGD-Mix, except for the specific adjustment of $\gamma = 0.5$ for Diff-Mix.

3.4.4. Background Robustness

We trained models on CUB and synthetic data and tested them on Waterbird using ResNet50 (224×224). We compared SGD-Mix with generative methods (Diff-Mix [30], Diff-Aug [30]) and non-generative methods (CutMix [32]). The settings were kept consistent with SGD-Mix.

4. Visualization

In this section, we present the complete versions of the visualizations referenced in the main text, specifically Figure 3 and Figure 4. These figures provide a more comprehensive view of the attention maps and generated images produced by SGD-Mix, offering additional insights into the method's ability to preserve foreground semantics and balance diversity and faithfulness across varying translation strengths.

4.1. Complete Attention Maps Before and After Saliency-Guided Mixing

Figure 2 shows the full set of attention maps before and after saliency-guided mixing, extending the examples shown in Figure 3 of the main text. For a source image I_i and target image I_j, the attention maps consistently focus on I_i's foreground region in the mixed image $I_{(i,j)}$, demonstrating the preservation of semantic consistency across multiple examples. This expanded view includes additional image pairs to highlight the robustness of the saliency-guided mixing process.

4.2. Complete Examples of Generated Images Under Varying Translation Strengths

Figure 3 provides the complete set of SGD-Mix generated images under varying translation strengths $S \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, extending the examples shown in Figure 4 of the main text. These examples illustrate how the generated images retain the source image's foreground semantics while the background evolves with increasing S, effectively balancing diversity and faithfulness. The full visualization includes additional samples to showcase the flexibility and consistency of the method across different source-target pairs.

References

[1] 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. 2, 4

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, G Heigold, S Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020. 3

[3] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit Haim Bermano, Gal Chechik, and Daniel Cohen-or. An image is worth one word: Personalizing text-to-image generation using textual inversion. In *The Eleventh International Conference on Learning Representations*, 2023. 4

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[5] 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? In *The Eleventh International Conference on Learning Representations*, 2023. 4

[6] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1

[7] 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

[8] Khawar Islam, Muhammad Zaigham Zaheer, Arif Mahmood, and Karthik Nandakumar. Diffusemix: Label-preserving data augmentation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27621–27630, 2024. 1, 2, 4

[9] Minsoo Kang and Suhyun Kim. Guidedmixup: an efficient mixup strategy guided by saliency maps. In *Proceedings of the AAAI conference on artificial intelligence*, pages 1096–1104, 2023. 4

[10] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, 2011. 3

[11] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013. 3

[12] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019. 2

[13] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems*, 35:5775–5787, 2022. 4

[14] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. 2013. 3

[15] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *International Conference on Learning Representations*, 2022. 1

[16] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 4

[17] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 1

[18] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 2, 3

[19] Nobuyuki Otsu et al. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975. 4

[20] 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. 2, 4

[21] Tianhao Qi, Shancheng Fang, Yanze Wu, Hongtao Xie, Jiawei Liu, Lang Chen, Qian He, and Yongdong Zhang. Deadiff: An efficient stylization diffusion model with disentangled representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8693–8702, 2024. 2

[22] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023. 4

[23] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks. In

International Conference on Learning Representations, 2020. 3

[24] Dvir Samuel, Yuval Atzmon, and Gal Chechik. From generalized zero-shot learning to long-tail with class descriptors. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 286–295, 2021. 2

[25] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 4

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 1

[27] Stability AI, RunwayML, and CompVis. Stable diffusion v1.5, 2022. Accessed: March 05, 2025. 4

[28] Brandon Trabucco, Kyle Doherty, Max Gurinas, and Ruslan Salakhutdinov. Effective data augmentation with diffusion models. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023. 4

[29] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 2, 3

[30] 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, 4

[31] Moon Ye-Bin, Nam Hyeon-Woo, Wonseok Choi, Nayeong Kim, Suha Kwak, and Tae-Hyun Oh. Exploiting synthetic data for data imbalance problems: baselines from a data perspective. *arXiv preprint arXiv:2308.00994*, 6, 2023. 3, 4

[32] 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. 2, 4

[33] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 4

[34] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 3

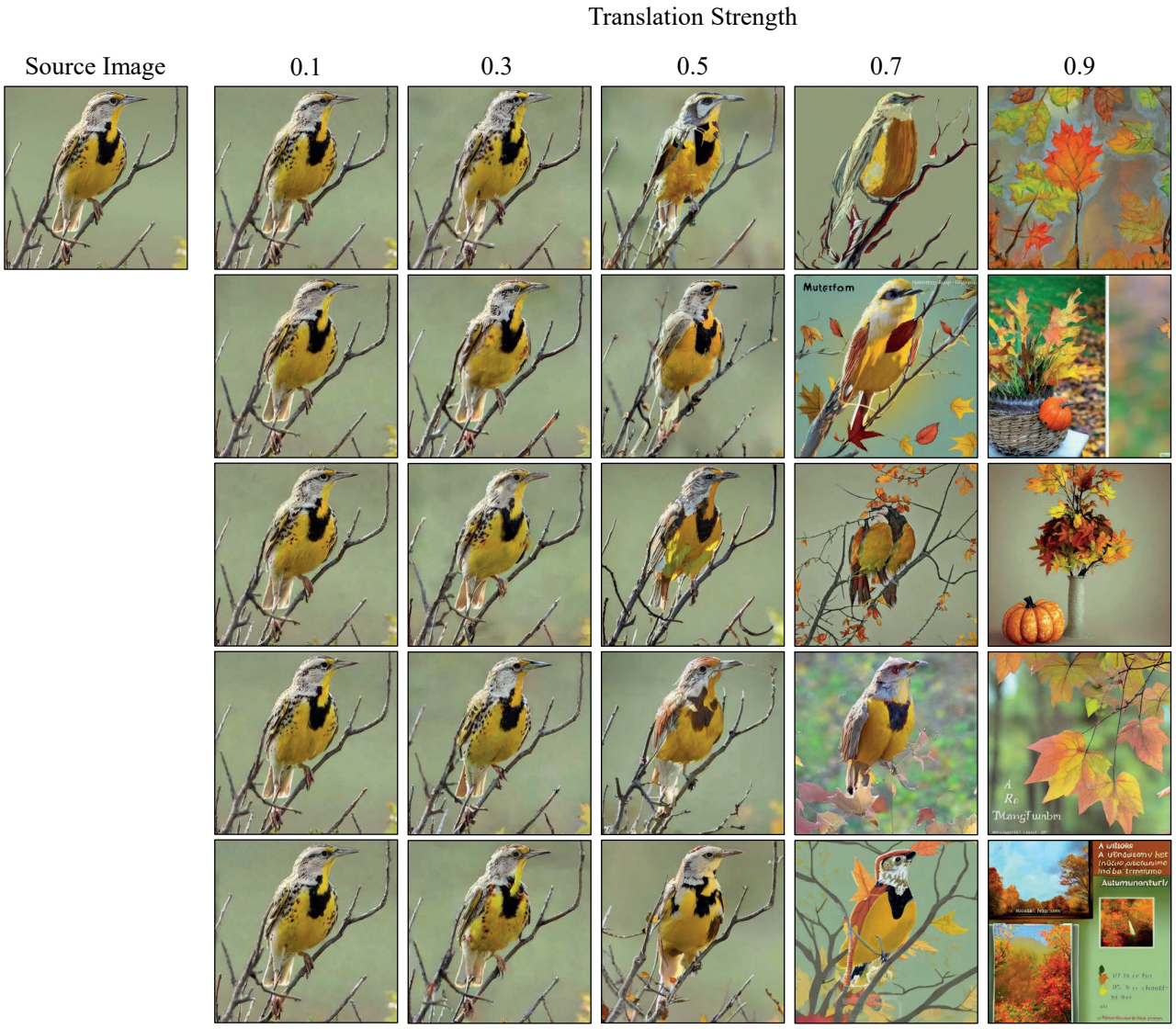

Figure 1. Visualization of semantic drift in DiffuseMix generated images using the prompt "A transformed version of image into autumn". As translation strength increases, generated images lose semantic fidelity to the source image I_i.

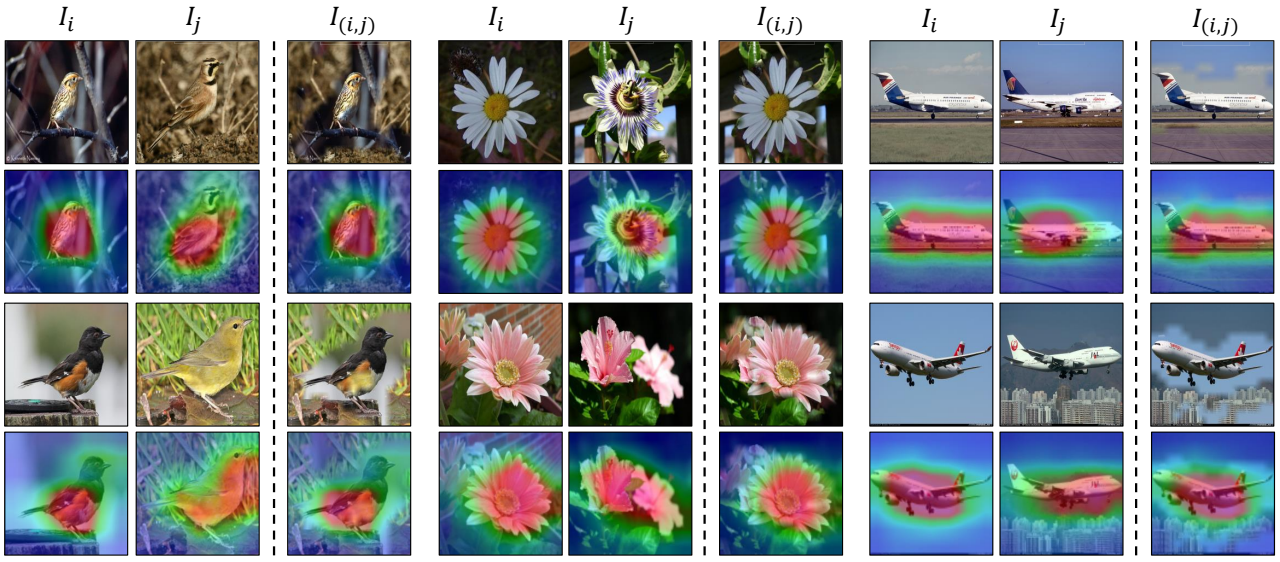

Figure 2. Attention maps before and after saliency-guided mixing in SGD-Mix. For a source image I_i and target image I_j, the attention maps (bottom row) reliably emphasize I_i's foreground region in the mixed image $I_{(i,j)}$, maintaining semantic integrity. This figure extends the examples shown in Figure 3 of the main text.

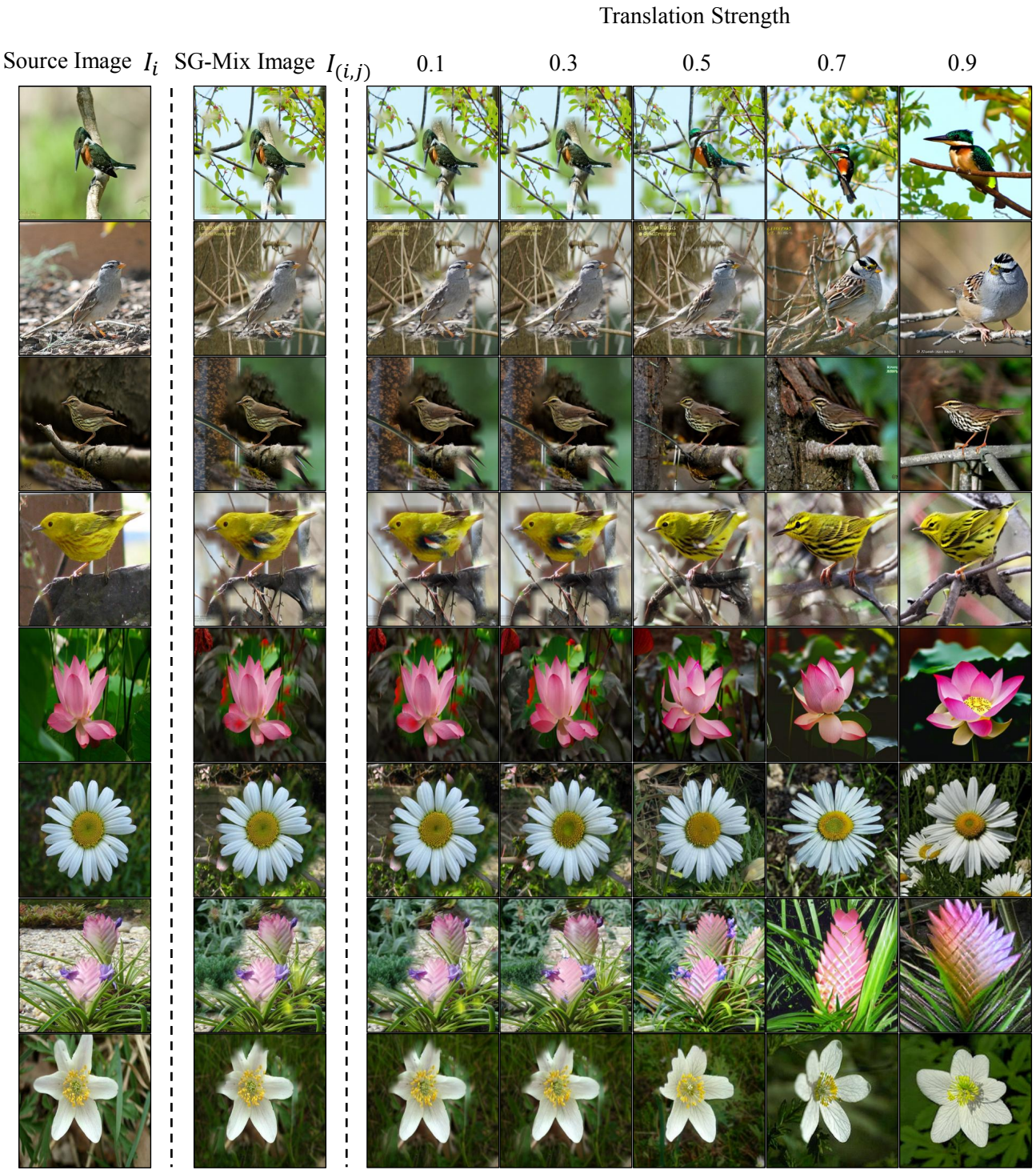

Figure 3. SGD-Mix generated images with translation strengths $S \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. The output preserves the foreground semantics of the source image, with the background varying as S increases, effectively managing diversity and fidelity. This figure extends the examples shown in Figure 4 of the main text.