Synthetic Data is an Elegant GIFT for Continual Vision-Language Models

Supplementary Material

A. Additional Implementation Details

We use a batch size 64 for both the MTIL and CIL benchmarks. We employ AdamW [22] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and a label smoothing [24] technique for better results. Label smoothing can substitute the regularization of weight decay and achieve better performance. For MTIL and CIFAR100 [17] of CIL, label smoothing is set to 0.2, with weight decay at 0. For TinyImageNet [18] of CIL, we set label smoothing to 0 and weight decay to 0.1. Learning rates are searched among $\{10^{-5}, 10^{-6}, 10^{-7}\}$ with cosine annealing [21]. For the base class set C^0 used to prompt Stable Diffusion to generate approximate pre-training data of VLMs, we select all 1000 ImageNet [4] class names for MTIL and 100 ImageNet class names that do not overlap with downstream datasets for CIL.

B. Results on CIL Benchmarks

Benchmark Description. For the CIL setting, we conduct experiments on CIFAR100 [17] and TinyImageNet [18] datasets following [9]. The 100 classes of CIFAR100 are divided into subsets of $\{10, 20, 50\}$, while the 100 classes from TinyImageNet are divided into subsets of $\{5, 10, 20\}$ to evaluate class distribution adaptability. For metrics, we adhered to the evaluation protocol in [9], calculating the average accuracy across all datasets at all timestamps ("Avg.") and the average performance of all tasks after continual learning ("Last").

Compared Methods. We compare our method with state-of-the-art approaches in the CIL setting (methods listed above "CLIP Zero-shot" in Tab. 1 and Tab. 2). The backbone used by these methods is consistent with that in the papers where they are proposed, i.e., ViT [7] or Res-Net [11]. Like MTIL, we also implement LwF [19], iCaRL [28], Lwf-VR [6], and ZSCL [35] with CLIP as the backbone and include them in the comparison. The results of CLIP zero-shot predictions and continual fine-tuning without protection are also included, denoted as "CLIP Zero-shot" and "CLIP Fine-tune", respectively.

Result Analysis. As summarized in Tab. 1 and Tab. 2, experimental results on two CIL settings demonstrate that our method remains effective in single-domain continual learning. Although CLIP excels in zero-shot prediction under these conditions, some CLIP-based continual learning methods are outperformed by those utilizing alternative backbones. This disparity is primarily due to the more pronounced in-distribution overfitting and forgetting that occur with smaller task step sizes. Despite these challenges, our method maintains strong performance even when the number of tasks is large and the incremental step size is small.

Table 1. Comparison of state-of-the-art CL methods on CIFAR100 benchmark in class-incremental setting.

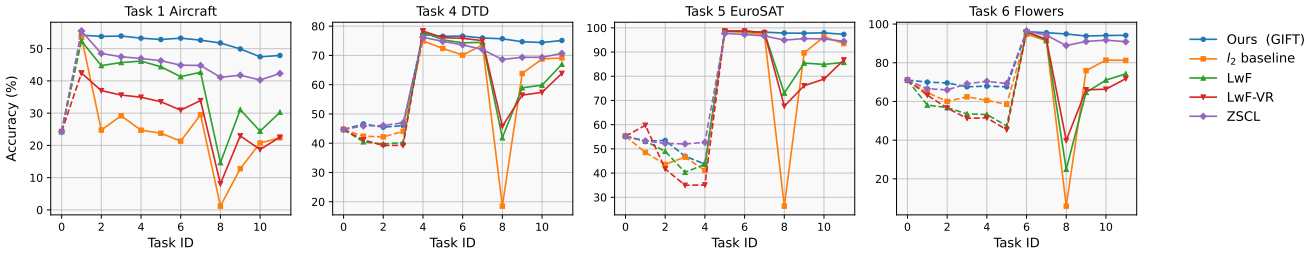
Methods	10 steps		20 steps		50 steps	
	Avg	**Last**	**Avg**	**Last**	**Avg**	**Last**
UCIR [14]	58.66	43.39	58.17	40.63	56.86	37.09
BiC [31]	68.80	53.54	66.48	47.02	62.09	41.04
RPSNet [27]	68.60	57.05	-	-	-	-
PODNet [8]	58.03	41.05	53.97	35.02	51.19	32.99
DER [33]	74.64	64.35	73.98	62.55	72.05	59.76
DyTox+ [9]	74.10	62.34	71.62	57.43	68.90	51.09
CLIP Zero-shot	74.47	65.92	75.20	65.74	75.67	65.94
CLIP Fine-tune	65.46	53.23	59.69	43.13	39.23	18.89
LwF [19]	65.86	48.04	60.64	40.56	47.69	32.90
iCaRL [28]	79.35	70.97	73.32	64.55	71.28	59.07
LwF-VR [6]	78.81	70.75	74.54	63.54	71.02	59.45
ZSCL [35]	82.15	73.65	80.39	69.58	79.92	67.36
GIFT (Ours)	**85.11**	**77.70**	**82.11**	**73.73**	**80.81**	**71.29**

Table 2. Comparison of different methods on TinyImageNet splits in class-incremental settings with 100 base classes.

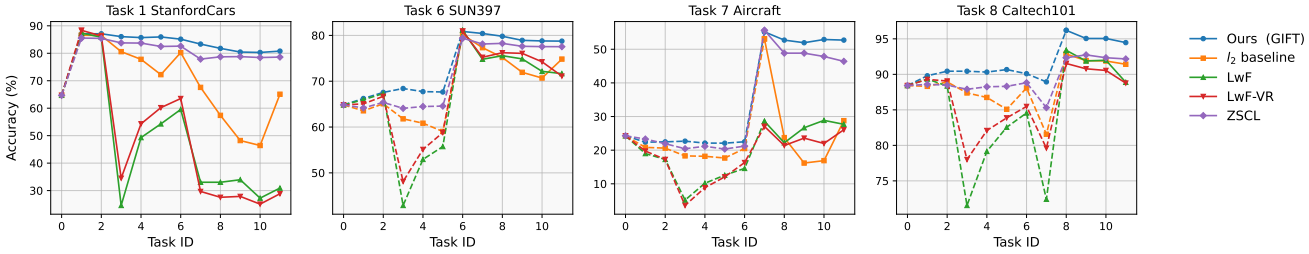
Methods	5 steps		10 steps		20 steps	
	Avg	**Last**	**Avg**	**Last**	**Avg**	**Last**
EWC [15]	19.01	6.00	15.82	3.79	12.35	4.73
EEIL [2]	47.17	35.12	45.03	34.64	40.41	29.72
UCIR [14]	50.30	39.42	48.58	37.29	42.84	30.85
MUC [20]	32.23	19.20	26.67	15.33	21.89	10.32
PASS [36]	49.54	41.64	47.19	39.27	42.01	32.93
DyTox [9]	55.58	47.23	52.26	42.79	46.18	36.21
CLIP Zero-shot	69.62	65.30	69.55	65.59	69.49	65.30
CLIP Fine-tune	61.54	46.66	57.05	41.54	54.62	44.55
LwF [19]	60.97	48.77	57.60	44.00	54.79	42.26
iCaRL [28]	77.02	70.39	73.48	65.97	69.65	64.68
LwF-VR [6]	77.56	70.89	74.12	67.05	69.94	63.89
ZSCL [35]	80.27	73.57	78.61	71.62	77.18	68.30
GIFT (Ours)	**81.16**	**77.04**	**80.20**	**75.51**	**79.32**	**74.87**

C. Detailed Results on MTIL Benchmark

Additional Benchmark Description. MTIL [35] comprises 11 datasets from diverse domains, organized into two task sequences to introduce different domain shifts. **The first sequence**, referred to as Order I, follows alphabetical order: Aircraft [23], Caltech101 [10], CIFAR100 [17], DTD [3], EuroSAT [12], Flowers [25], Food [1], MNIST [5], OxfordPet [26], StanfordCars [16], SUN397 [32]. **The second sequence**, Order II, is randomly arranged: StanfordCars, Food, MNIST, OxfordPet, Flowers, SUN397, Air-



(a) Accuracy Changes of MTIL Datasets in Order I.



(b) Accuracy Changes of MTIL Datasets in Order II.

Figure 1. Illustration of the classification accuracy changes as tasks are being learned on the MTIL benchmark in two orders. The dashed lines represent the results of zero-shot predictions for an unlearned task. At task 0, the initial CLIP model's zero-shot accuracy is evaluated.

craft, Caltech101, DTD, EuroSAT, CIFAR100.

Additional Metric Formulation. We further clarify the calculation of metrics used in the main text. Consider the accuracy matrix $[a_{i,j}]_{n\times n}$ where each element $a_{i,j}$ represents the test accuracy on task j after the model has learned task i, evaluated across all n tasks. In traditional CL, only the lower triangular portion of this matrix is relevant, as the model cannot predict for tasks it has not yet encountered. However, for VLMs, the upper triangular matrix offers valuable insight into the degradation of the model's zero-shot capability, indicating the extent of forgetting pre-training knowledge. The metrics are computed as follows:

$$\text{Transfer} = \frac{1}{n-1}\sum_{j=2}^{n}\frac{1}{j-1}\sum_{i=1}^{j-1}a_{i,j}, \tag{1}$$

$$\text{Last} = \frac{1}{n}\sum_{j=1}^{n}a_{n,j}, \tag{2}$$

$$\text{Avg.} = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}a_{i,j}. \tag{3}$$

Detailed Results. Tab. 3 and Tab. 4 present the detailed results of Transfer, Avg, and Last metrics on each dataset of the MTIL benchmark in Order I and Order II respectively. Task-specific metric values are calculated as described in [35], with their averages reported in the "Average" column, which is also presented in the main text. "Zero-shot" denotes the zero-shot prediction performance of the initial CLIP model, and "Fine-tune" represents the direct fine-tuning accuracy on each dataset, both of which can be seen as an upper bound where no forgetting phenomenon happens. "Continual Finetuning" refers to the naive continual learning method that fine-tunes the model on the new task without any protection, indicating the lower bound suffering from most significant forgetting. Evaluated under both orderings, our method can achieve the best performance on most datasets.

We select several tasks from both two orderings and plot the accuracy curves against the task ID, as shown in Fig. 1 (a) and 1 (b). An ideal continual learning method for VLMs should produce an accuracy curve resembling a mirrored "Z" shape, indicating that the zero-shot accuracy before learning the task is maintained and there is almost no forgetting after learning. Our approach aligns with this standard. However, most methods struggle with catastrophic forgetting of both pre-training knowledge and downstream task knowledge, causing significant fluctuations in the accuracy curve. This issue is particularly evident when there is a substantial domain shift, such as with the MNIST dataset (i.e., Task 8 in Fig. 1 (a) and Task 3 in Fig. 1 (b)), where the CLIP model's feature space nearly collapses without strong protection. A comparison between Fig. 1 (a) and Fig. 1 (b) shows that th the l_2 baseline offers some resistance to drastic domain shifts but weakens as the task sequence progresses. Since the MNIST dataset appears later in Order I than in Order II, the l_2 baseline can resist this domain shift in Order II but fails to do so in Order I. The strong performance of our method demonstrates that integrating knowledge distillation with the l_2 constraint effectively resists domain shifts and tolerates longer task sequences.

Table 3. Detailed Transfer, Avg., and Last scores (%) of different continue training methods on MTIL benchmark in **Order I**. The highest single score of each metric in each column is highlighted in **bold**, while multiple top scores are underlined.

Method	Aircraft [23]	Caltech101 [10]	CIFAR100 [17]	DTD [3]	EuroSAT [12]	Flowers [25]	Food [1]	MNIST [5]	OxfordPet [26]	Cars [16]	SUN397 [32]	Average
Zero-shot	24.3	88.4	68.2	44.6	54.9	71.0	88.5	59.4	89.0	64.7	65.2	65.3
Fine-tune	62.0	95.1	89.6	79.5	98.9	97.5	92.7	99.6	94.7	89.6	81.8	89.2
Transfer												
Continual Finetune		67.1	46.0	32.1	35.6	35.0	57.7	44.1	60.8	20.5	46.6	44.6
l_2 baseline		83.2	63.5	42.9	44.9	61.2	79.5	63.8	71.9	43.9	54.6	61.0
LwF [19]		74.5	56.9	39.1	51.1	52.6	72.8	60.6	75.1	30.3	55.9	56.9
iCaRL [28]		56.6	44.6	32.7	39.3	46.6	68.0	46.0	77.4	31.9	60.5	50.4
LwF-VR [6]		77.1	61.0	40.5	45.3	54.4	74.6	47.9	76.7	36.3	58.6	57.2
WiSE-FT [30]		73.5	55.6	35.6	41.5	47.0	68.3	53.9	69.3	26.8	51.9	52.3
ZSCL [35]		86.0	67.4	45.4	**50.4**	69.1	87.6	61.8	86.8	60.1	66.8	68.1
MoE-Adapter [34]		87.9	68.2	44.4	49.9	**70.7**	**88.7**	59.7	89.1	**64.5**	65.5	68.9
GIFT (Ours)		**88.5**	**69.8**	**46.0**	49.4	68.5	87.1	**69.9**	**88.9**	57.7	**67.7**	**69.3**
Avg.												
Continual Finetune	25.5	81.5	59.1	53.2	64.7	51.8	63.2	64.3	69.7	31.8	49.7	55.9
l_2 baseline	24.0	82.3	68.2	58.2	70.9	67.0	76.2	57.1	77.5	51.4	56.9	62.7
LwF [19]	36.3	86.9	72.0	59.0	73.7	60.0	73.6	74.8	80.0	37.3	58.1	64.7
iCaRL [28]	35.5	89.2	72.2	60.6	68.8	70.0	78.2	62.3	81.8	41.2	62.5	65.7
LwF-VR [6]	29.6	87.7	74.4	59.5	72.4	63.6	77.0	66.7	81.2	43.7	60.7	65.1
WiSE-FT [30]	26.7	86.5	64.3	57.1	65.7	58.7	71.1	70.5	75.8	36.9	54.6	60.7
ZSCL [35]	45.1	92.0	80.1	64.3	79.5	81.6	**89.6**	75.2	88.9	64.7	68.0	75.4
MoE-Adapter [34]	50.2	91.9	**83.1**	**69.4**	78.9	**84.0**	89.1	73.7	89.3	**67.7**	66.9	76.7
GIFT (Ours)	**51.9**	**93.9**	81.4	67.7	**80.3**	82.8	89.3	**80.6**	**90.3**	63.1	**68.9**	**77.3**
Last												
Continual Finetune	31.0	89.3	65.8	67.3	88.9	71.1	85.6	99.6	92.9	77.3	81.1	77.3
l_2 baseline	22.4	91.1	80.8	69.2	93.5	81.2	90.5	49.4	92.7	83.8	80.1	75.9
LwF [19]	26.3	87.5	71.9	66.6	79.9	66.9	83.8	99.6	92.1	66.1	80.4	74.6
iCaRL [28]	35.8	93.0	77.0	70.2	83.3	88.5	90.4	86.7	93.2	81.2	**81.9**	80.1
LwF-VR [6]	20.5	89.8	72.3	67.6	85.5	73.8	85.7	99.6	93.1	73.3	80.9	76.6
WiSE-FT [30]	27.2	90.8	68.0	68.9	86.9	74.0	87.6	99.6	92.6	77.8	81.3	77.7
ZSCL [35]	40.6	92.2	81.3	70.5	94.8	90.5	**91.9**	98.7	93.9	85.3	80.2	83.6
MoE-Adapter [34]	**49.8**	92.2	**86.1**	**78.1**	95.7	**94.3**	89.5	98.1	89.9	81.6	80.0	85.0
GIFT (Ours)	47.9	**95.6**	82.8	75.1	**97.3**	94.2	91.7	99.2	**94.2**	**87.0**	80.9	**86.0**

Table 4. Detailed Transfer, Avg., Last accuracy (%) of different continue training methods on MTIL benchmark in **Order II**. The highest single score of each metric in each column is highlighted in **bold**, while multiple top scores are underlined.

Method	Cars [16]	Food [1]	MNIST [5]	OxfordPet [26]	Flowers [25]	SUN397 [32]	Aircraft [23]	Caltech101 [10]	DTD [3]	EuroSAT [12]	CIFAR100 [17]	Average
Zero-shot	64.7	88.5	59.4	89.0	71.0	65.2	24.3	88.4	44.6	54.9	68.2	65.3
Fine-tune	89.6	92.7	94.7	94.7	97.5	81.8	62.0	95.1	79.5	98.9	89.6	89.2
Transfer												
Continual Finetune		85.9	59.6	57.9	40.0	46.7	11.1	70.0	30.5	26.6	37.7	46.6
l_2 baseline		87.0	62.3	83.7	60.6	62.1	19.3	86.6	42.7	41.4	60.1	60.6
LwF [19]		87.8	58.5	71.9	46.6	57.3	12.8	81.4	34.5	34.5	46.8	53.2
iCaRL [28]		86.1	51.8	67.6	50.4	57.9	11.0	72.3	31.2	32.7	48.1	50.9
LwF-VR [6]		88.2	57.0	71.4	50.0	58.0	13.0	82.0	34.4	29.3	47.6	53.1
WiSE-FT [30]		87.2	57.6	67.0	45.0	54.0	12.9	78.6	35.5	28.4	44.3	51.0
ZSCL [35]		88.3	57.5	84.7	68.1	64.8	21.1	88.2	45.3	**55.2**	68.2	64.2
MoE-Adapter [34]		**88.8**	59.5	**89.1**	69.9	64.4	18.1	86.9	43.7	54.6	68.2	64.3
GIFT (Ours)		88.3	**63.4**	88.1	**70.8**	**67.7**	**22.8**	**90.4**	**46.7**	51.8	**68.8**	**65.9**
Avg.												
Continual Finetune	42.1	70.5	92.2	80.1	54.5	59.1	19.8	78.3	41.0	38.1	42.3	56.2
l_2 baseline	69.9	86.2	91.9	89.0	74.0	69.1	23.2	88.6	51.1	50.8	62.6	68.8
LwF [19]	49.0	77.0	92.1	85.9	66.5	67.2	20.9	84.7	44.6	45.5	50.5	62.2
iCaRL [28]	52.0	75.9	77.4	74.6	58.4	59.3	11.7	79.6	42.1	43.2	51.7	56.9
LwF-VR [6]	44.9	75.8	91.8	85.3	63.5	67.6	16.9	84.9	44.0	40.6	51.3	60.6
WiSE-FT [30]	52.6	79.3	91.9	83.9	63.4	65.2	23.3	83.7	45.4	40.0	48.2	61.5
ZSCL [35]	81.7	**91.3**	91.1	91.0	82.9	72.5	33.6	89.7	53.3	**62.8**	69.9	74.5
MoE-Adapter [34]	**84.9**	89.9	89.3	91.4	**86.2**	72.2	33.4	89.4	53.3	61.4	69.9	74.7
GIFT (Ours)	83.2	90.8	**92.6**	**92.8**	85.8	**74.1**	**36.0**	**92.1**	**54.7**	60.0	**70.4**	**75.7**
Last												
Continual Finetune	24.0	67.3	99.1	87.4	44.3	67.0	29.5	92.3	61.3	81.0	88.1	67.4
l_2 baseline	65.1	84.2	96.4	90.2	71.8	74.8	28.8	91.4	70.7	88.2	87.2	77.2
LwF [19]	34.6	69.6	99.3	88.7	61.1	72.5	32.5	88.1	65.6	90.9	87.9	71.9
iCaRL [28]	46.0	81.5	91.3	82.8	66.5	72.2	16.3	91.6	68.1	83.2	87.8	71.6
LwF-VR [6]	27.4	61.2	99.4	86.3	60.6	70.7	23.4	88.0	61.3	84.3	88.1	68.3
WiSE-FT [30]	35.6	76.9	**99.5**	89.1	62.1	71.8	27.8	90.8	67.0	85.6	**87.6**	72.2
ZSCL [35]	78.2	**91.1**	97.6	92.5	87.4	78.2	45.0	92.3	72.7	**96.2**	86.3	83.4
MoE-Adapter [34]	**84.1**	88.5	94.0	91.8	**94.1**	77.8	50.4	93.3	**77.1**	87.7	86.6	84.1
GIFT (Ours)	81.0	90.2	98.6	**94.0**	91.5	**78.6**	**51.7**	**94.6**	75.6	95.4	86.6	**85.3**

D. Ablation Analysis of Image Generation

In this section, we conduct an ablation study on the image generation mechanism, primarily on the hyperparameters of Stable Diffusion inference, i.e., denoising steps and classifier-free guidance scale [13]. Additionally, we examine the impact of eliminating synthetic images of specific downstream task datasets on the distillation performance.

Denoising Steps. For Stable Diffusion, the number of denoising steps is a crucial hyperparameter that balances generation speed and quality. Fewer denoising steps result in faster generation but typically at the cost of lower quality. While our method defaults to 50 denoising steps, it remains effective with lower settings, such as 25 denoising steps, for faster generation. As shown in Tab. 5, reducing the steps to 25 has minimal impact on performance of our method. By fixing the random seed, the images generated with fewer steps correspond to intermediate outputs from more denoising steps. We further visualize and compare the quality of images generated with different denoising steps in our synthetic dataset, as illustrated in Fig. 3.

Table 5. Comparison of synthetic images generated with different denosing steps as data sources for distillation.

Method	Denoising Steps	Transfer	Avg.	Last
GIFT w/ AWC	50 Steps	**69.3**	**77.3**	**86.0**
GIFT w/o AWC	50 Steps	68.9	76.6	85.0
GIFT w/ AWC	25 Steps	69.2	77.2	85.8
GIFT w/o AWC	25 Steps	69.2	76.6	84.8

Classifier-free Guidance Scale. We consider three configurations for the classifier-free guidance scale w: large scale ($w = 10.5$), medium scale ($w = 7.5$) and small scale ($w = 4.5$). Prior studies [29] suggest that when training with large-scale synthetic images, smaller guidance scales are crucial for enhancing the diversity of generated images and boosting performance, because smaller w leads to greater intra-caption variation between generated images. However, as shown in Tab. 6, our method demonstrates consistent performance across different guidance scales, with slightly reduced effectiveness observed for smaller scales. This could be attributed to the fact that when fewer synthetic images are used, **inter-class diversity is more critical than intra-class diversity**. We ensure inter-class diversity by constructing distinct prompts based on different class names. Meanwhile, the increased deviation between the generated images and the text prompts with a small w may negatively affect the memory retention of CLIP, as evidenced by the lower Transfer value.

Eliminating Synthetic Images for Downstream Tasks. We skip the class names of specific downstream dataset so that the classes included in this dataset do not appear in the synthetic images. In doing so, we conduct experiments on skipping the generation of the first datasets of MTIL in order I and order II, i.e., the Aircraft and StanfordCars datasets. These two datasets are selected because they appear first in the sequence and are fine-grained, making them more susceptible to forgetting during continual learning. As can be seen from the accuracy curves in Fig. 4, incorporating synthetic data for a specific downstream task significantly enhances memory retention of that task.

Table 6. Comparison of synthetic images generated with different classifier-free guidance scale as data sources for distillation.

Guidance Scale	Image Num	Transfer	Avg.	Last
small	1K	68.2	76.3	85.2
medium		68.9	76.6	85.0
large		68.5	76.3	85.1
small	3K	68.7	76.8	85.0
medium		69.1	76.7	84.9
large		68.8	76.6	85.1

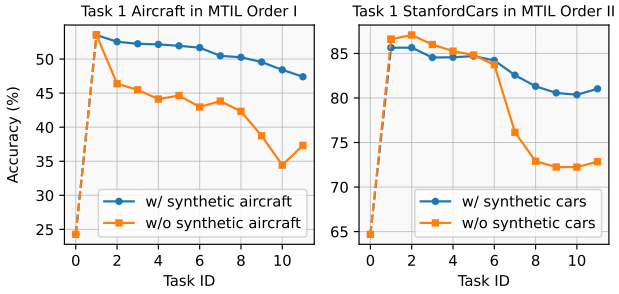


Figure 2. Eliminating synthetic images for specific downstream tasks exacerbates forgetting of these tasks.

E. Visualization: Synthetic Data for Different Datasets

At last, we provide synthetic images for different downstream datasets (i.e., Aircraft, CIFAR100, DTD, EuroSAT and StanfordCars) in Fig. 3. All images are randomly chosen rather than human-picked and are used in our experiments. We observe that for most datasets, synthesized images from the Stable Diffusion model are of high quality, demonstrating its capability to adapt to diverse domains without fine-tuning. But there also exist cases that many unsatisfactory examples are generated, such as the DTD and EuroSAT datasets. Additionally, Stable Diffusion struggles with generating accurate numerical representations, making it unsuitable for datasets like MNIST. It also fails to replicate the low-resolution nature of CIFAR-100 images but successfully captures the classes within the dataset. Despite these limitations, the majority of the generated images are satisfactory, underscoring the significant contribution of Stable Diffusion to overcoming catastrophic forgetting.

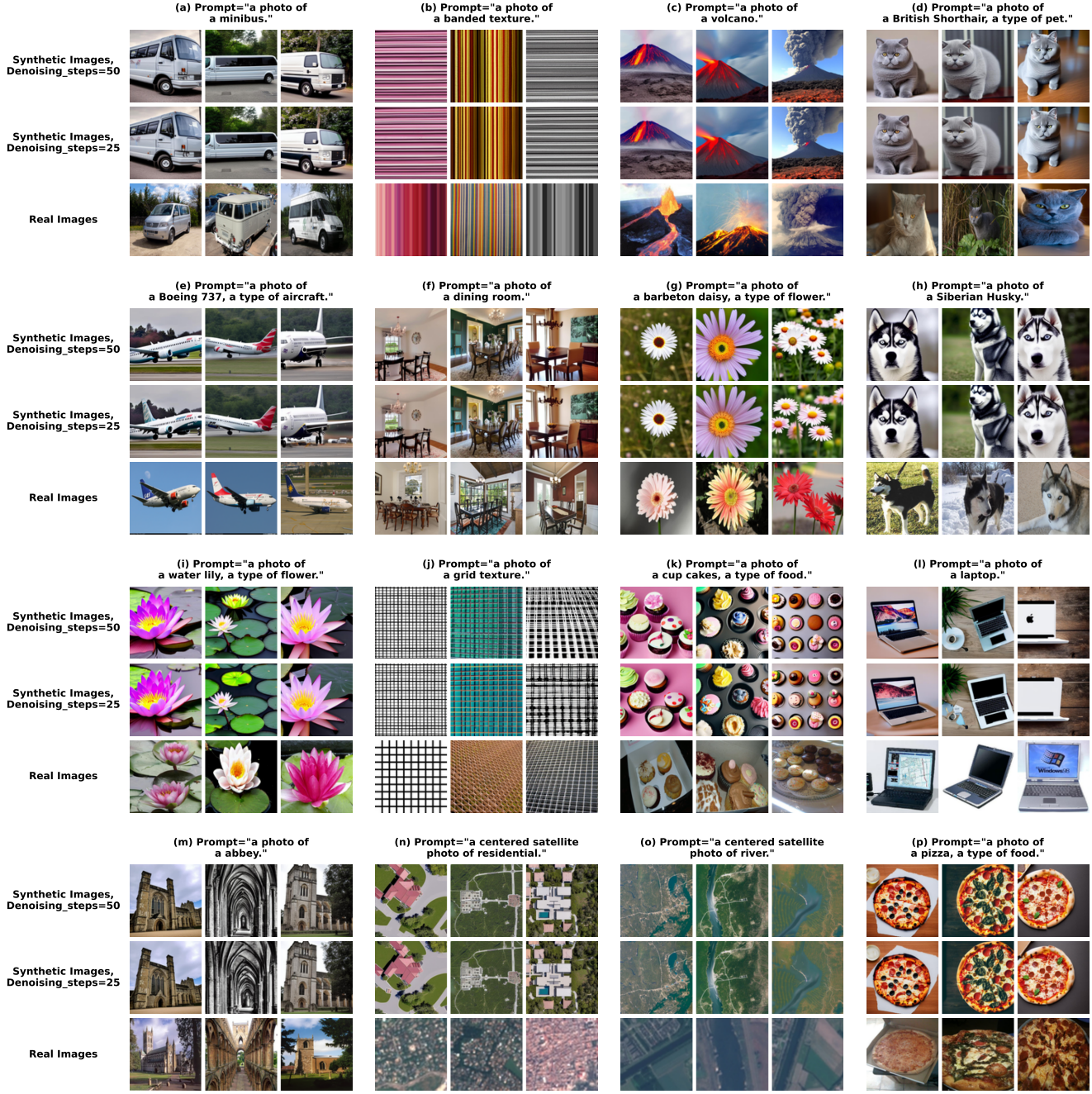


Figure 3. Leveraging the high-quality image generation capabilities of Stable Diffusion, we generate diverse and vivid images across different categories using simple textual prompts. Reducing the number of denoising steps to 25 doesn't bring much visual degradation of the generated images. Our comparative experiments demonstrate that our method remains effective with fewer denoising steps, allowing for faster generation for practical applications.

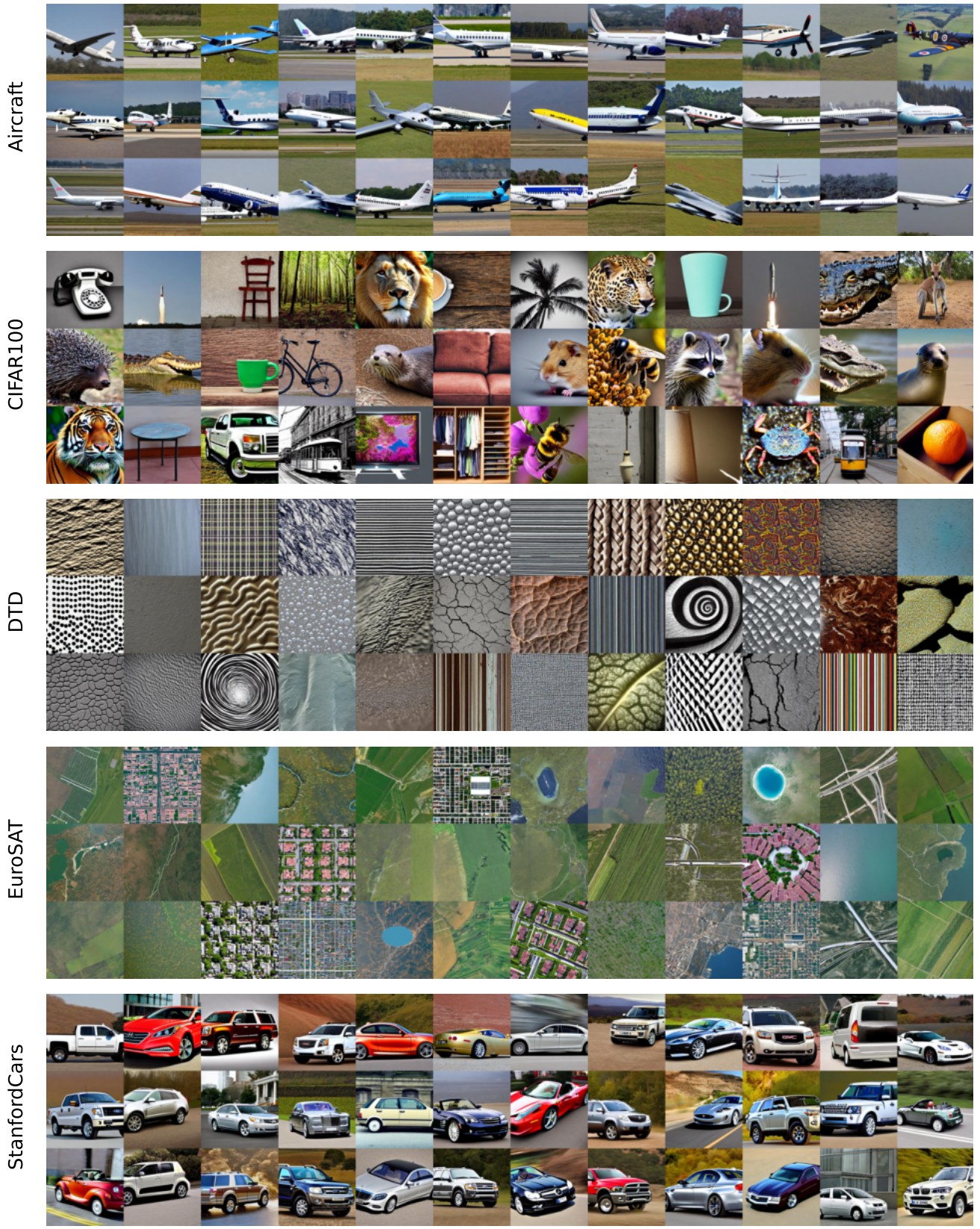


Figure 4. Visualization of synthetic data for different downstream datasets.

References

[1] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, 2014. 1, 3, 4

[2] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *ECCV*, 2018. 1

[3] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *CVPR*, 2014. 1, 3, 4

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1

[5] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 2012. 1, 3, 4

[6] Yuxuan Ding, Lingqiao Liu, Chunna Tian, Jingyuan Yang, and Haoxuan Ding. Don't stop learning: Towards continual learning for the clip modeltoward. *arXiv preprint arXiv:2207.09248*, 2022. 1, 3, 4

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[8] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020. 1

[9] Arthur Douillard, Alexandre Ramé, Guillaume Couairon, and Matthieu Cord. Dytox: Transformers for continual learning with dynamic token expansion. In *CVPR*, 2022. 1

[10] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, 2004. 1, 3, 4

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[12] Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019. 1, 3, 4

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

[14] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *CVPR*, 2019. 1

[15] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 2017. 1

[16] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, 2013. 1, 3, 4

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 3, 4

[18] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 1

[19] Zhizhong Li and Derek Hoiem. Learning without forgetting. *TPAMI*, 2017. 1, 3, 4

[20] Yu Liu, Sarah Parisot, Gregory Slabaugh, Xu Jia, Ales Leonardis, and Tinne Tuytelaars. More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning. In *ECCV*, 2020. 1

[21] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[23] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 1, 3, 4

[24] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *NeurIPS*, 2019. 1

[25] 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*, 2008. 1, 3, 4

[26] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *CVPR*, 2012. 1, 3, 4

[27] Jathushan Rajasegaran, Munawar Hayat, Salman H Khan, Fahad Shahbaz Khan, and Ling Shao. Random path selection for continual learning. *NeurIPS*, 2019. 1

[28] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017. 1, 3, 4

[29] Yonglong Tian, Lijie Fan, Phillip Isola, Huiwen Chang, and Dilip Krishnan. Stablerep: Synthetic images from text-to-image models make strong visual representation learners. *NeurIPS*, 2024. 5

[30] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *CVPR*, 2022. 3, 4

[31] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *CVPR*, 2019. 1

[32] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010. 1, 3, 4

[33] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017. 1

[34] Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *CVPR*, 2024. 3, 4

[35] Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *ICCV*, 2023. 1, 2, 3, 4

[36] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *CVPR*, 2021. 1