SAIDO: Generalizable Detection of AI-Generated Images via Scene-Aware and Importance-Guided Dynamic Optimization in Continual Learning

Supplementary Material

6. Theoretical Proof

This section provides a theoretical analysis of the gradient-projection strategy adopted in IDOM, demonstrating its effectiveness in continual learning. Assume that the training of task t has been completed, and task $t+1$ is about to be trained. Let the inputs from the previous task be denoted as x_t, and let the trainable neurons for the new and old tasks be represented as E_{t+1} and E_t, respectively. First, consider the case where a neuron is solely relevant to fake image detection. During the training of the new task, the goal is to prevent forgetting of the old task, which requires that the condition in Eq. (16) should hold.

$$f_{\phi+\phi_n}(E_{t+1}, x_t) = f_{\phi+\phi_n}(E_t, x_t), \tag{16}$$

For E_{t+1}, it is obtained by updating E_t after training on task t. Let ΔE_1 denote the gradient update of the neurons after training on task t; then, Eq. (17) can be derived.

$$E_{t+1} = E_t + \Delta E_1 \tag{17}$$

Then, if the following equation holds, forgetting can be avoided.

$$E_{t+1}x_t = E_t x_t \tag{18}$$

By substituting Eq. (17) into Eq. (18), the following equation can be further obtained.

$$x_t \Delta E_1 = 0 \tag{19}$$

Eq. (19) indicates that if neurons are updated along directions orthogonal to the subspace spanned by the features of previous tasks, forgetting is significantly reduced.

Assuming that the neurons are only related to real image detection, according to our previous analysis, real images exhibit a more compact feature distribution. Therefore, unlike fake images, real images from different tasks are treated as a single class during detection, denoted as $x_{t+1} \approx x_t$. Therefore, during neuron updates, greater emphasis is placed on preserving knowledge from previous tasks by updating neurons along the direction of the projection of the current gradient onto the feature subspace of previous tasks.

$$E_{t+1} = E_t + \Delta E_0^t \tag{20}$$

According to Eq. (20), after completing the t-th training task, $\mathcal{L}\text{oss}(\Delta E_0^t, x_t)$ reaches its minimum. When the neuron is updated along the projection direction of the current gradient onto the feature subspace of the old tasks, and considering $x_{t+1} \approx x_t$ under the gradient projection strategy, it follows that $\Delta E_0^{t+1} = \beta E_0^t$, where β denotes a scaling coefficient. Consequently, $\mathcal{L}_{\text{loss}}(\Delta E_0^{t+1}, xt+1)$ also attains its minimum.

We introduce the Fisher matrix to measure neuron importance, characterizing each neuron's contribution to real-image or fake-image detection. Empirically, it is observed that during network updates, almost no neuron is entirely irrelevant to either real or fake image detection. The current SOTA method RegO [3] attempts to divide neurons using multiple importance thresholds, but such discrete partitioning fails to provide precise control over neuron updates, limiting both plasticity and stability in continual learning. Therefore, aggregated neuron importance is used to determine the proportional relationship between update directions, enabling finer-grained regulation of how each neuron is updated.

$$q_0 = \frac{\tilde{I}_k^{(0)}}{\tilde{I}_k^{(0)} + \tilde{I}_k^{(1)}}, \quad q_1 = 1 - q_0. \tag{21}$$

Based on the theoretical analysis, q_0 and q_1 are used to assign different proportions of the update to the projection of the gradient g_p onto the old-task feature subspace and its orthogonal component g_o. This enables precise control over neuron update directions according to their importance.

$$\widetilde{g} = q_0 \cdot g_p + q_1 \cdot g_o \tag{22}$$

7. Dataset Details

To ensure the rigor of the experiments, this section provides a detailed discussion of the dataset setups in both continual learning tasks and open-world tasks, along with the scene distributions in each dataset.

7.1. Continual Learning Dataset

During the training of continual learning tasks, taking the task-order configuration of Protocol 1 as an example, SAEM first employs multiple rounds of large-model scoring to select the highest-scoring set of base scenes, forming the initial scene configuration $S_{initial}$ = {Activity, Animal, Building, Food, Nature, Object, Person, Vehicle}. Throughout training, the VLLM identifies scene features in the ADM dataset and updates the scene set with $Plant$ and $Clothing$. As subsequent tasks arrive, the scene updates gradually stabilize, resulting in the final scene set: S_{final}

= {Activity, Animal, Building, Clothing, Food, Nature, Object, Person, Plant, Vehicle}

ADM [6]. We randomly select 4K sample pairs from the ADM dataset provided by GenImage [51] as the training set. The test set size for the entire continual learning task is consistently 25% of the training set size. To ensure the diversity of image content, real images are selected from the large-scale ImageNet dataset, while the generative models are trained using ImageNet. The specific scene distribution in the dataset is shown in Fig. 5.

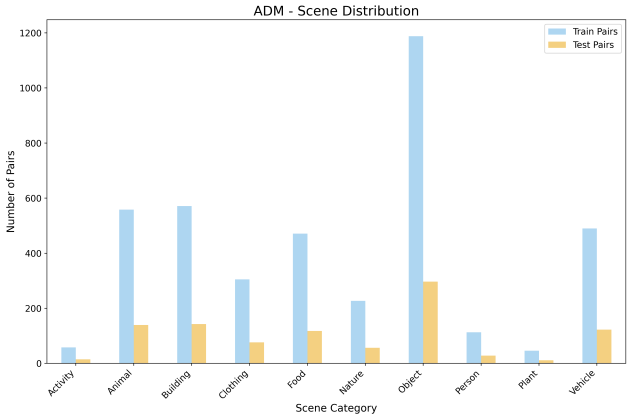

Figure 5. The scene distribution of ADM

GLIDE [25]. 3K sample pairs from the GLIDE subset of GenImage [51] are used as the training data. The generative model is trained on ImageNet. The scene distribution is shown in Fig. 6.

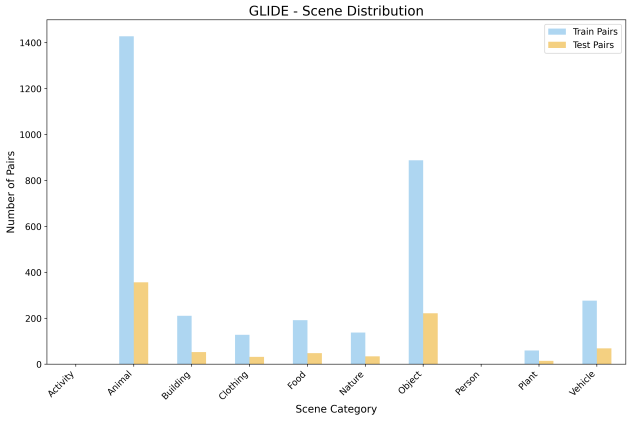

Figure 6. The scene distribution of GLIDE

SAGAN [48]. 3K training pairs are drawn from the SAGAN dataset in OSMA [45], using ImageNet for both real images and generative model training. The scene distribution is in Fig. 7.

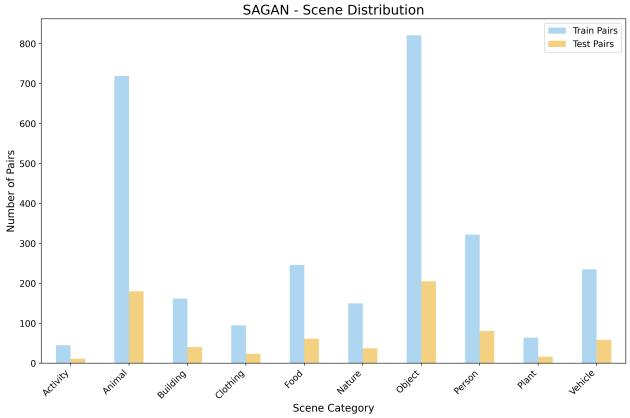

Figure 7. The scene distribution of SAGAN

ProGAN [12]. 6K pairs from the ProGAN dataset in CNNSpot [40] are used. LSUN provides the real images and serves as the training source for the generative model. The scene distribution is shown in Fig. 8.

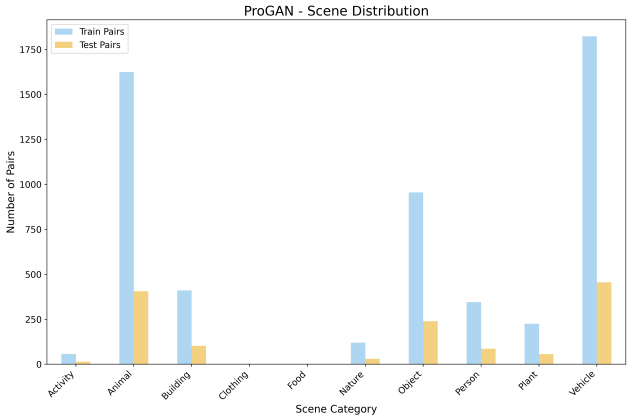

Figure 8. The scene distribution of ProGAN

BigGAN [2]. 3K sample pairs from CNNSpot [40] form the training set, with ImageNet used for both real images and generative model training. The scene distribution is in Fig. 9.

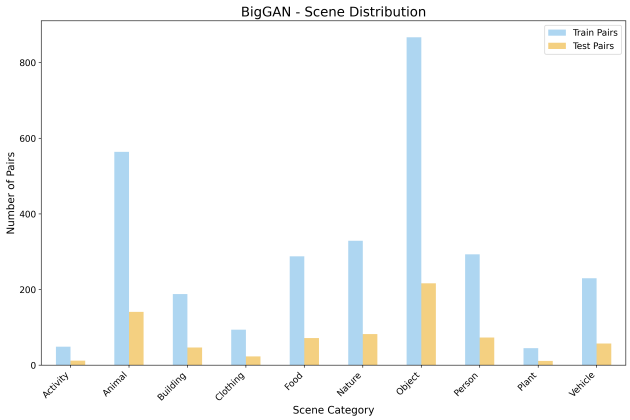

Figure 9. The scene distribution of BigGAN

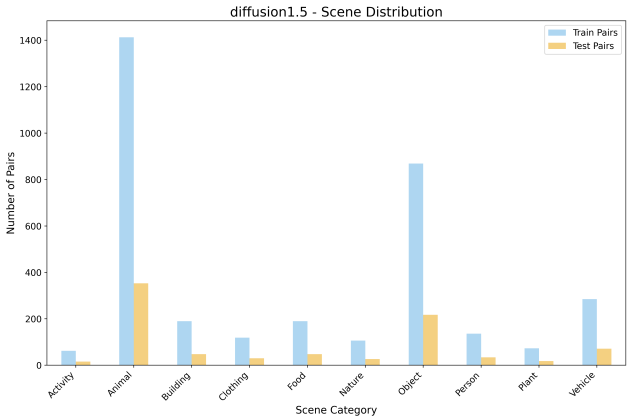

Figure 11. The scene distribution of SD1.5

Wukong [43]. A total of 3.5K sample pairs are randomly selected from the dataset provided by GenImage [51] for the training set. Real images are selected from the ImageNet dataset, while generative models are trained using ImageNet. The scene distribution in the dataset is shown in Fig. 10.

VQDM [10]. A total of 3.5K sample pairs are randomly selected from the dataset provided by GenImage [51] for the training set. Real images are provided by the ImageNet dataset, while generative models are trained using ImageNet. The scene distribution in the dataset is shown in Fig. 12.

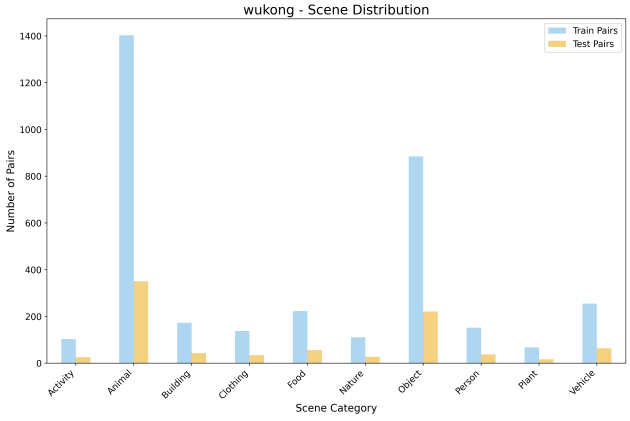

Figure 10. The scene distribution of Wukong

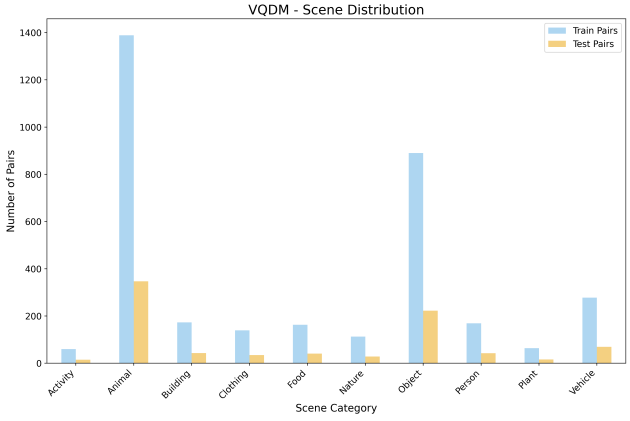

Figure 12. The scene distribution of VQDM

SD1.5 [30]. A total of 3.5K sample pairs are randomly selected from the dataset provided by GenImage [51] for the training set. Real images are selected from the ImageNet dataset, while generative models are trained using ImageNet. The scene distribution in the dataset is shown in Fig. 11.

Midjourney-V5 [23]. A total of 3.5K sample pairs are randomly selected from the dataset provided by GenImage [51] for the training set. Real images are selected from the ImageNet dataset, while generative models are trained using ImageNet. The scene distribution in the dataset is shown in Fig. 13.

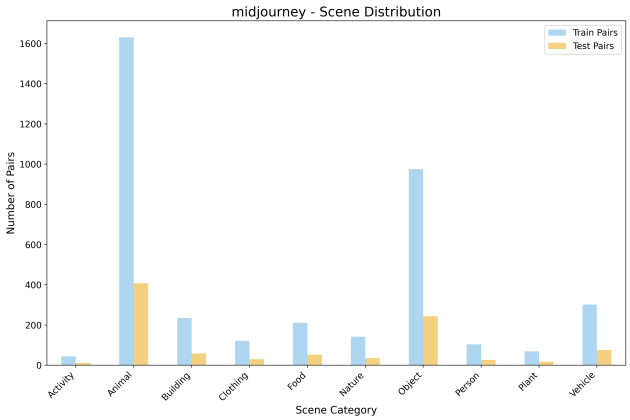

Figure 13. The scene distribution of Midjourney-V5

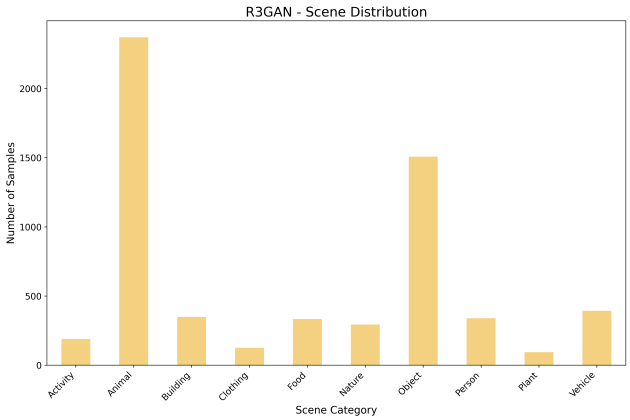

Figure 15. The scene distribution of R3GAN

7.2. Open-World Dataset

To evaluate the generalization ability of different methods when facing open-world datasets, this section considers generative models that are more advanced than those in the continual learning dataset. This setup also reflects the models' adaptability when encountering more sophisticated generative models.

StyleGAN-xl [33]. We collect 3K sample pairs as the open-world test set. To ensure rigorous evaluation of generalization, the distribution of this test set is kept as distinct as possible from that of the continual learning tasks. Therefore, real images are randomly selected from Open Images V7, while fake images are produced by generative models pretrained on ImageNet, as provided by StyleGAN-xl [33].

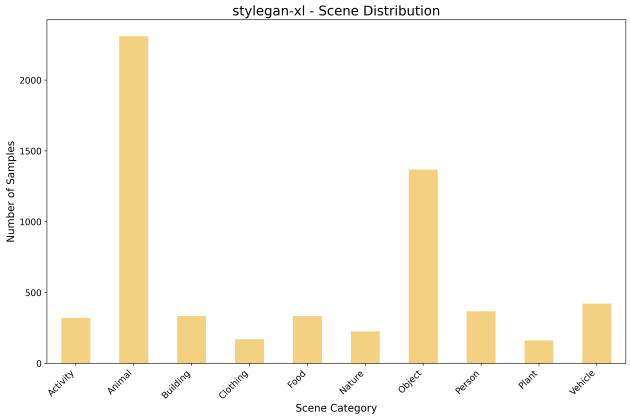

Figure 14. The scene distribution of StyleGAN-xl

R3GAN [11]. A total of 3K sample pairs are used, with real images randomly drawn from Open Images V7 and fake images generated by ImageNet-pretrained models provided by R3GAN [11].

FLUX1-dev [1]. A total of 4.5K sample pairs are randomly selected from AIGIBench [17]. Following AIGIBench settings, real images are drawn from Open Images V7, while fake images are generated using the FLUX1-dev pretrained diffusion model, which incorporates high-quality text prompts via the Gemini API.

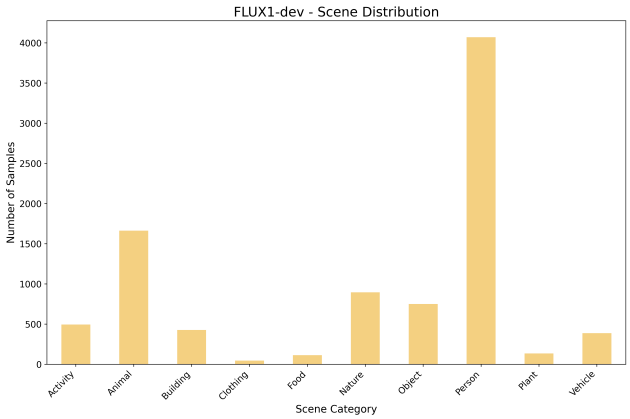

Figure 16. The scene distribution of FLUX1-dev

Midjourney-V6 [24]. A total of 3K sample pairs are randomly selected from AIGIBench [17], with real images from Open Images V7 and fake images generated by the Midjourney-V6 pretrained diffusion model.

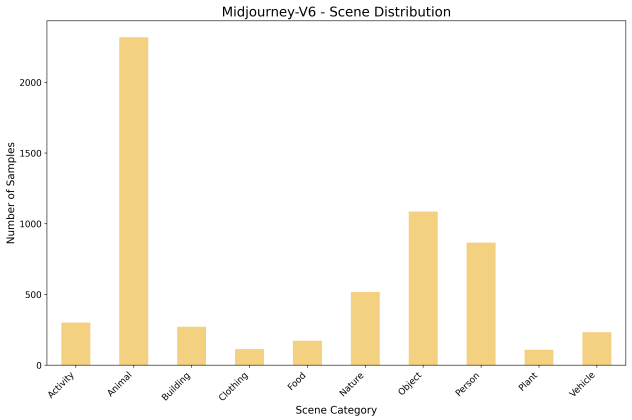

Figure 17. The scene distribution of Midjourney-V6

SD3 [7]. From AIGIBench [17], 4.5K sample pairs are randomly selected, where real images come from Open Images V7 and fake images are produced by the Midjourney-V6 pretrained diffusion model.

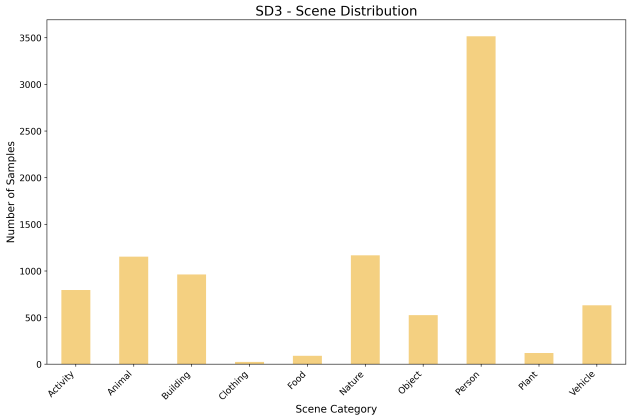

Figure 18. The scene distribution of SD3

Imagen3 [9]. A total of 4.5K sample pairs are randomly drawn from AIGIBench [17], with real images sourced from Open Images V7 and fake images generated by the pretrained Midjourney-V6 diffusion model.

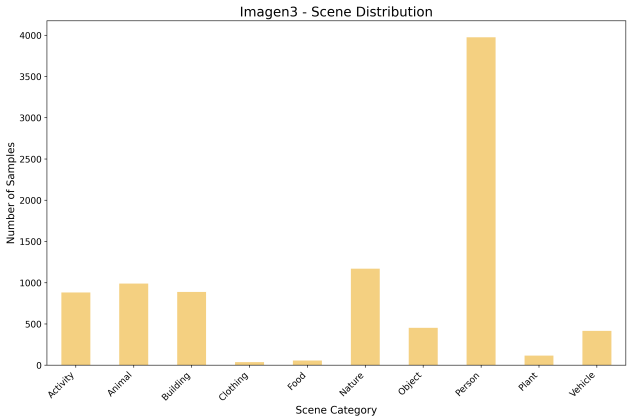

Figure 19. The scene distribution of Imagen3

8. Additional Order Configurations

To ensure comprehensive experimental settings, nine generative image models are considered in the continual learning tasks, and long-sequence continual learning experiments with multiple task orders are conducted. This section presents and analyzes the performance comparisons for the alternative task sequences.

- **Extra Order 1.** $\mathcal{D}_1$ = {ADM, GLIDE, SAGAN, ProGAN, BigGAN, Wukong, Midjourney-V5, SD1.5, VQDM}.
- **Extra Order 2.** $\mathcal{D}_2$ = {ADM, ProGAN, Wukong, SAGAN, BigGAN, SD1.5, GLIDE, VQDM, Midjourney-V5}.

As shown in Tab. 6 and Tab. 7, our method achieves superior performance compared with SOTA methods under various task order settings. This indicates that our method is less susceptible to variations in the order of continual learning tasks, demonstrating its stability. Notably, our method exhibits substantially less degradation in overall performance across various task sequences compared to current SOTA methods. Under the task order of Protocol 1, our method not only achieves the highest average accuracy on most tasks, but also maintains an average accuracy of 99.81% on the first task (ADM) and 95.61% on the final task (Midjourney-V5). In contrast, the current SOTA method drops from 99.21% to 92.13% over the same progression. Similar trends are observed under other task orders, highlighting the superior balance between plasticity and stability provided by our method.

9. Cost Analysis

A more detailed analysis of the computational cost of the proposed method is provided in this section, with emphasis on both inference and training efficiency, to better assess its practicality and deployability in real-world scenarios.

Table 6. Performance evaluation and comparison with other methods (%), with the best results highlighted in bold, the second-best results underlined, and the results arranged according to Extra Order 1.

Method	Venue	Continual Learning	Replay Set	1-ADM	2-GLIDE		3-SAGAN		4-ProGAN	
				AA	AA	AF	AA	AF	AA	AF
CLIP+LoRA	-	×	–	**99.99**	96.87	6.26	92.78	10.83	97.79	2.09
Universe	CVPR'23	×	–	93.70	76.27	19.53	89.80	8.79	85.04	17.38
NPR	CVPR'24	×	–	92.86	92.56	**-1.76**	80.40	19.76	72.89	23.55
RegO	AAAI'25	✓	×	99.49	98.58	1.16	94.86	6.77	**97.84**	**1.10**
Tang et al.	TIFS'25	✓	✓	99.35	99.06	0.07	97.51	**1.18**	95.09	2.82
SAIDO (Ours)	-	✓	×	99.81	**99.61**	0.32	**98.40**	2.17	97.03	2.90

Method	5-BigGAN		6-Wukong		7-Midjourney-V5		8-SD1.5		9-VQDM		New. ACC (↑)
	AA	AF	AA	AF	AA	AF	AA	AF	AA	AF	
CLIP+LoRA	94.34	6.43	76.46	27.57	84.63	17.08	82.62	19.02	71.46	31.37	**99.34**
Universe	70.60	27.20	77.36	21.76	79.09	20.60	61.01	28.69	78.24	24.15	89.65
NPR	75.48	23.15	67.85	31.49	72.89	73.64	75.30	21.24	66.40	31.53	99.12
RegO	96.75	2.65	92.82	6.27	88.31	10.25	89.25	8.75	89.92	7.63	96.70
Tang et al.	94.93	2.48	94.45	2.98	92.86	**4.62**	92.32	5.09	92.84	4.61	97.91
SAIDO (Ours)	**98.45**	**1.12**	**98.00**	**1.52**	**94.67**	5.16	**96.38**	**3.16**	**96.22**	**3.27**	99.13

Table 7. Performance evaluation and comparison with other methods under (%), with the best results highlighted in bold, the second-best results underlined, and the results arranged according to Extra Order 2.

Method	Venue	Continual Learning	Replay Set	1-ADM	2-ProGAN		3-Wukong		4-SAGAN	
				AA	AA	AF	AA	AF	AA	AF
CLIP+LoRA	-	×	–	**99.86**	98.09	1.21	85.69	19.66	85.43	18.21
Universe	CVPR'23	×	–	93.70	90.06	17.95	83.17	30.19	81.98	30.1
NPR	CVPR'24	×	–	92.86	86.18	14.10	72.89	29.06	63.24	40.37
RegO	AAAI'25	✓	×	99.49	97.91	0.65	90.76	8.74	92.92	5.77
Tang et al.	TIFS'25	✓	✓	99.21	96.76	1.76	93.73	5.48	90.71	9.34
SAIDO (Ours)	-	✓	×	99.69	**98.27**	**0.56**	**97.74**	**1.48**	**96.84**	**2.79**

Method	5-BigGAN		6-SD1.5		7-GLIDE		8-VQDM		9-Midjourney-V5		New. ACC (↑)
	AA	AF	AA	AF	AA	AF	AA	AF	AA	AF	
CLIP+LoRA	87.05	15.26	94.10	6.26	89.73	11.29	97.53	2.17	90.32	10.16	**99.36**
Universe	85.93	26.03	81.85	30.81	54.41	33.82	80.54	35.58	77.93	30.15	90.68
NPR	69.05	31.65	80.43	17.42	62.21	35.78	64.66	33.08	72.46	23.12	93.01
RegO	94.25	4.10	92.90	4.97	92.89	5.04	92.68	5.05	88.19	9.30	96.45
Tang et al.	90.17	9.84	94.08	4.81	91.79	7.56	93.14	5.70	92.58	5.92	97.84
SAIDO (Ours)	**96.75**	**2.94**	**98.25**	**1.09**	**97.47**	**2.22**	**98.27**	**1.16**	**94.26**	**5.55**	99.19

9.1. Inference Cost

SAIDO employs a lightweight Qwen3-VL-4B (running on 2×RTX 4090) only for scene recognition and routing, rather than for backbone feature extraction or full-image reasoning. As a result, the introduced overhead is fixed and moderate, with an average routing latency of 0.201 s and a computational cost of 1.01 TFLOPs per image. By restricting the role of the VLLM to scene classification and prompt

generation, the proposed method incurs substantially lower overhead than previous VLLM-based detectors [46], while still benefiting from semantic guidance for routing.

9.2. Training Cost

During training, both the CLIP backbone and the VLLM are frozen, and only lightweight LoRA modules are updated, which effectively controls the optimization cost. On 2×RTX 4090, SAIDO requires 245 s per epoch, compared with 227 s for Tang et al. [38]. Although the per-epoch cost is slightly higher, SAIDO converges much faster, reaching convergence in 10 epochs, whereas replay-based training typically requires 40 epochs and introduces additional computational and storage overhead due to historical data replay. In addition, experiments show that the scene set stabilizes quickly during continual learning, and the number of LoRA adapters can be effectively controlled by adjusting the granularity of scene partitioning. These properties further support the efficiency and practicality of the proposed framework.

10. Sensitivity and Robustness Analysis

In this section, we provide a more detailed analysis of the robustness of the proposed method from two aspects: sensitivity to VLLM-based scene classification and sensitivity to key hyperparameters. These experiments are designed to further examine the stability of SAIDO under scene ambiguity and different parameter settings, and to verify that the proposed configuration achieves a favorable balance between stability and plasticity.

10.1. VLLM Sensitivity

Although images containing multiple semantics may introduce scene ambiguity, the VLLM-based scene classification remains highly stable in practice. On 1,000 randomly sampled images, the misclassification rate is below 2%, indicating that the predicted scene labels are generally reliable. Moreover, even when occasional misrouting occurs, the resulting variation in SAIDO's detection performance remains within 1.4% on average. These results suggest that the proposed routing strategy is robust to scene ambiguity and does not materially affect the overall detection performance.

10.2. Hyperparameter Sensitivity

We further analyze the sensitivity of SAIDO to key hyperparameters by varying $\alpha \in \{0.5, 0.6, 0.75, 0.9\}$ and $e \in \{0.5, 1, 1.5\}$. For α, AF reaches its minimum value of 3.94% at $\alpha = 0.75$, compared with 4.86% at $\alpha = 0.5$, and then slightly increases to 4.17% at $\alpha = 0.9$. This trend indicates that an appropriate value of α is important for maintaining memory stability throughout continual learning. For e, increasing it to 1.5 further reduces AF from 4.23% (at $e = 0.5$) to 3.92%, but also lowers the detection accuracy from 95.78% to 93.27%. This result suggests that excessively increasing e may improve stability at the cost of plasticity. Overall, these observations verify that our default setting achieves a favorable balance between stability and plasticity.

11. Further Discussion on Feature Distribution Assumptions

We further discuss the assumption in our paper that fake image features exhibit larger distributional variations, whereas real image features are relatively more compact. Specifically, although generated images may approximate the real data manifold at the semantic level, different generative architectures still introduce distinguishable artifacts, which lead to observable differences in feature distributions. In contrast, real images are constrained by physical imaging processes and therefore tend to follow more stable feature distributions. Although this assumption is to some extent empirical, its validity is further supported by experiments on realistic benchmarks such as AIGIBench [17].

12. Role of Scene-Based Partitioning

In this section, the role of scene-based partitioning in continual learning is further analyzed. Although the performance gain of SAEM is limited when the content differences across tasks are small, this behavior is closely related to the task distribution itself. To further verify that scene-domain shifts can undermine model stability and lead to performance degradation, a setting with pronounced scene variation is constructed by selecting three representative scene categories, namely Person, Building, and Vehicle, and the performance of single-LoRA and SAEM is compared in long-sequence continual learning tasks. The results show that when scene-domain shifts become significant, single-LoRA suffers from severe interference and a clear drop in performance, whereas SAEM can isolate domain-specific updates and thereby effectively alleviate such interference. As a result, SAEM improves the average continual learning accuracy by 2.13% and the generalization performance by 1.28%. Moreover, in open-world settings, scene-based partitioning maintains a relatively stable number of LoRA adapters, offering better practicality and scalability than method-specific expert expansion.

13. Future Directions

SAEM introduces a scene-based expert partitioning paradigm for continual AI-generated image detection, which can be further extended to finer-grained partitioning strategies according to the characteristics of specific tasks. For example, in face forgery detection, expert partitioning

could be further refined based on attributes such as gender and age, enabling more precise modeling of domain-specific distributional variations.

In addition, more optimization strategies can be explored, such as soft assignment or multi-expert ensembles. Compared with the current assignment strategy, these approaches may provide greater flexibility and robustness in handling complex multi-semantic scenes. Subsequent work can focus on more effective assignment and domain optimization strategies to further improve detection performance.