Generate, Refine, and Encode: Leveraging Synthesized Novel Samples for On-the-Fly Fine-Grained Category Discovery

Supplementary Material

Appendix

A. Implementation Details

A.1. Datasets Details

Dataset Details. As shown in Table A.1, we evaluate our method across multiple benchmarks, including the introduction of the iNaturalist 2017 [8] dataset to the On-the-Fly Category Discovery (OCD) task. This demonstrates the robustness of our approach in addressing challenging fine-grained datasets. The iNaturalist 2017 dataset, collected from the citizen science platform iNaturalist, comprises 675,170 training and validation images spanning 5,089 fine-grained natural categories, including Plantae (plants), Insecta (insects), Aves (birds), and Mammalia (mammals), distributed across 13 super-categories. These super-categories exhibit substantial intra-category variation, posing significant challenges for fine-grained classification. Following the setup in OCD [2], the categories of each dataset are split into subsets of known and unknown categories. Specifically, 50% of the samples from the seen categories are used to form the support set, $\mathcal{D}_S$ for training, while the remainder forms the query set $\mathcal{D}_Q$ for on-the-fly testing.

Table A.1. Statistics of datasets used in our experiments.

	CUB	Scars	Pets	Arachnida	Animalia	Mollusca
$\lvert Y_S\rvert$	100	98	19	28	39	47
$\lvert Y_Q\rvert$	200	196	38	56	77	93
$\lvert \mathcal{D}_S\rvert$	1.5K	2.0K	0.9K	1.7K	1.5K	2.4K
$\lvert \mathcal{D}_Q\rvert$	4.5K	6.1K	2.7K	4.3K	5.1K	7.0K

A.2. Evaluation Metric Details

For our evaluation, we focus on three super-categories: Arachnida, Animalia, and Mollusca. Following the OCD protocol [2], the categories within each dataset are split into seen and unseen subsets. Specifically, 50% of the samples from seen categories form the labeled training set $\mathcal{D}_S$, while the remaining samples are included in the unlabeled set $\mathcal{D}_Q$ for on-the-fly testing.

A.3. Algorithm Pipeline for SLE-based Inference

In the testing stage, we first establish a dynamic leader memory, which is initialized by the category-specific leader features in Sec. 3.2.

Then, we calculate the maximal intra-category distance, Δ_{max}, as an adaptive threshold to determine if this instance belongs to unknown categories following an online clustering fashion. Given a test instance, $\mathbf{x}_j \in \mathcal{D}_Q$, if it is predicted to belong to known categories, we assign its estimated category label $\hat{y}_j$ to the category label according to the nearest leader. Otherwise, we create a new leader with the instance feature and append it to the dynamic leader memory. During the testing process, we update both known- and unknown-category leaders by momentum averaging of corresponding instance features. The algorithm is deliberated in Algorithm 1.

A.4. Details of the Compared Methods.

Since the OCD task requires real-time inference and is relatively new, traditional baselines from NCD and GCD are inappropriate for this scenario. Thus, we selected the SMILE model [2] as a comparative baseline

Algorithm 1: On-the-Fly Inference based on SLE.

Input: Test data $\mathcal{D}_Q$, trained backbone network $f(\cdot)$, maximal intra-category distance Δ_{max}, and a set of known-category leader features $\mathcal{C}_S = \{\mathbf{l}_i\}_{i=1}^{|\mathcal{Y}_S|}$

for $\mathbf{x}_i \in \mathcal{D}_Q$ **do**
  Extract instance feature $f(\mathbf{x}_i)$;
  for *leader feature* $\mathbf{l}_j \in \mathcal{C}_S$ **do**
    Compute l_2 distance $||\mathbf{l}_j - f(\mathbf{x}_i)||_2^2$;
    if $||\mathbf{l}_j - f(\mathbf{x}_i)||_2^2 \geq \Delta_{max}$ **then**
      Add $f(\mathbf{x}_i)$ to $\mathcal{C}_S$;
      Return $\hat{y}_j = |\mathcal{C}_S| + 1$; `// Create a new category`
  $\hat{y}_i = argmin_j ||\mathbf{l}_j - f(\mathbf{x}_i)||_2^2$;
  $\mathcal{C}[\hat{y}_i] = \eta \cdot \mathcal{C}[\hat{y}_n] + (1-\eta) \cdot f(\mathbf{x}_i)$;
  Return $\hat{y}_i$; `// belong to a known category`

Algorithm 2: Iterative Training

Input: Feature Extractor $f(\cdot)$, Projection Head $\mathcal{H}(\cdot)$, Labeled data $\mathcal{D}_S$ and synthesized data $\mathcal{D}^G$.

Output: f and $\mathcal{H}(\cdot)$.

for $n = 1$ ***in*** $[1, 200]$ **do**
  Extract features and execute clustering algorithm to assign category labels $\mathcal{Y}_A$;
  Generate a set of category-specific leader features $\mathcal{C}_A$;
  for $i = 1$ ***in*** $[1, max\_iteration]$ **do**
    Sample mini-batches from $\mathcal{D}_S \cup \mathcal{D}_G$;
    Calculate overall optimization objective by Eq. (A.1);
    Update f and $\mathcal{H}(\cdot)$ by SGD [7];
  end
end

and included three hash-like competitive methods, SMILE baseline **(BaseHash)** [2], Prototypical Hash Encoding **(PHE)** [13], Ranking Statistics **(RankStat)** [3] and Winner-take-all **(WTA)** [5], and one online clustering method **Sequential Leader Clustering (SLC)** [4] for evaluation. The elaborated introductions are follows:

- **Sequential Leader Clustering (SLC)** [4]: This traditional clustering method is tailored for sequential data analysis.
- **Ranking Statistics (RankStat)** [3]: RankStat identifies the top-3 indices in feature embeddings to serve as category descriptors.
- **Winner-take-all (WTA)** [5]: WTA utilizes the indices of the highest values within groups of features as the basis for category description. These three robust baselines are established in accordance with the SMILE.
- **Prototypical Hash Encoding (PHE)** [13]: PHE leverages Category-aware Prototype Generation (CPG) to capture intra-category diversity and Discriminative Category Encoding (DCE) to enhance hash code discrimination, ensuring effective category discovery in streaming data.

A.5. SMILE Approach and Training Details

We use the pre-trained Stable Diffusion v1.4 to fuse semantic latent embeddings without any fine-tuning. The OCD models (*i.e.*, based projector $\mathcal{H}(\cdot)$, hash projector $\mathcal{H}_h(\cdot)$ [2] and ViT-B-16 [1] backbone) are optimized by SGD [7] optimizer during training process. We set the initial learning rate to be $1e-2$ with a batch size of 128. A weight decay of $5e-5$ is applied as a regularization term during training. We use the Cosine Annealing for learning rate scheduling, which gradually decreases the learning rate to $1e-5$. Our model is trained for 100 epochs on a single NVIDIA A100-SXM GPU. Our model is implemented in PyTorch 2.0.0.

Table A.2. Results with Kmeans evaluation.

Method	Mollusca			CUB		
	All	Old	New	All	Old	New
Upbound	39.4	44.6	36.7	59.2	56.7	60.4
Ours	**35.2**	**43.3**	**30.8**	**54.1**	**53.0**	**54.7**
SMILE	29.5	34.1	27.1	49.8	46.2	51.6

As discussed in Section 3.4, the total loss is:

$$\mathcal{L} = \mathcal{L}_{sup} + \mathcal{L}_{reg} + \alpha * \mathcal{L}_{sle} + \beta * \mathcal{L}_c, \tag{A.1}$$

where L_{sup} and L_{reg} are basic losses from SMILE [2]. Following our definition, the L_{sup} formulated as:

$$\mathcal{L}_{sup} = -\frac{1}{|P_i|} \sum_{p \in P_i} \log \frac{\exp\left(\mathcal{H}(f(\mathbf{x}_i)) \cdot \mathcal{H}(f(\mathbf{x}_p))\right)}{\sum_{j=1}^{|B|} \mathbb{1}_{[j \neq i]} \exp\left(\mathcal{H}(f(\mathbf{x}_i)) \cdot \mathcal{H}(f(\mathbf{x}_j))\right)}, \tag{A.2}$$

where the P_i is the positive set in a mini-batch and $|B|$ is batch size. Similarly, L_{reg} is formulated as:

$$\hat{\boldsymbol{h}}_i = \text{hash}\left(\mathcal{H}_h\left(f\left(\mathbf{x}_i\right)\right)\right), \tag{A.3}$$

$$\mathcal{L}_{reg} = -\left|\hat{\boldsymbol{h}}_i\right|. \tag{A.4}$$

Pseudo-code for Iterative Training. We iteratively perform leader feature generation and leader-based representation learning to dynamically update the leader features. The pseudo-code for the iterative training is elaborated below.

Gradually Increase α. As the OCD backbone network is unsupervised and pre-trained by the DINO [1] approach that acquires a strong representation ability, model fine-tuning benefits from preserving the representation ability while adapting to target data. Inspired by the warm-up strategy [12], we design a linear growth paradigm for α. From

Table B.1. Comparison with Diff-Mix on CUB dataset.

Method	CUB		
	All	Old	New
SMILE [2]	32.2	50.9	22.9
SMILE [2] + Diff-Mix [9]	32.9	54.4	22.1
SMILE [2] + DiffGRE (Ours)	35.4	58.2	23.8

the 0^{th} to the 50^{th} epoch, the α increase from $0.1 \times \alpha$ to α. Then, the α is kept for the remaining 50 epochs.

Sampling Strategy of Leader-based Contrastive Learning. To facilitate the leader-based contrastive learning in a mini-bach, we follow [12] to randomly sample $N^C \times N^I$ images to form a mini-batch, where N^C is the number of categories in a mini-batch and N^I is the number of images for each category. In our experiments, we set $N^C = 8$ and $N^I = 16$ for 128 batch size.

B. Exploration Experiments

B.1. Gap in Data Synthesis.

Although our DiffGRE can generate virtual images to help OCD model training, it is difficult to reasonably evaluate the quality of the generation. We conduct the exploration experiment in Tab. A.2. We first use the $\mathcal{D}_Q$ as the generated data to execute our SLE training, and then directly use the Kmeans [6] with ground-truth K to generate category labels. We report the experimental results as "Upbound" in Tab. A.2. Based on the results, we find that our virtual images contribute almost the same as real data on unknown categories. Moreover, compared with baseline accuracy based on SMILE, the gap between Upbound and ours is significantly narrowed by our DiffGRE framework, *e.g.*, ALL-ACC is from 9.9% to 4.2% on the Mollusca dataset.

B.2. Comparison with Diff-Mix

Quantitative Comparison. In this section, we compare our method with the Diff-Mix data augmentation approach, which only augments known categories. We implement experiments on the CUB dataset. We followed the original settings of Diff-Mix and fed it with source images and targeted classes names to generate augmented images. Results are shown in Table B.1. "SMILE + Diff-Mix" refers to replacing the Attribute Composition Generation (ACG) module in the DiffGRE framework with Diff-Mix, in order to make a fair comparison. Diff-Mix performs well on known categories, but there is no significant improvement on unknown categories. This proves that Diff-Mix can only synthesize images within known categories, leading to its inefficiency in the OCD task. On the other hand, it is observed that our method achieves higher accuracy than Diff-Mix, which indicates the effectiveness of our implementation to synthesize novel samples belonging to unknown categories.

Qualitative Comparison. In addition to the quantitative comparison, we also conducted a qualitative comparison between our method and the Diff-Mix. Results are provided in Figure B.1. The Diff-Mix used an image from category A as the source image and converted it to the targeted class B. The output is located in the top-right corner of Figure B.1. Compared with Diff-Mix, our method takes two images from different categories as inputs and synthesizes a novel sample. It can be observed that our synthesized image is more likely to belong to an unknown category.

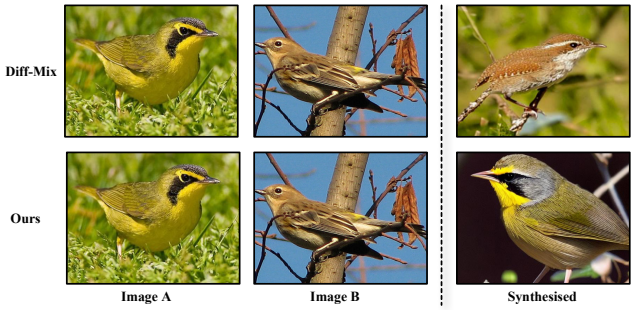

Figure B.1. Our image generation compared with Diff-Mix [9] on the CUB dataset.

Table B.2. Analysis on the hyperparameter γ.

Dataset	γ	Number of Remained Images	Number of Generated Images	Number of Labeled Images
CUB	0.40	1.2K	5.0K	1.5K
Scars	0.65	1.1K	4.9K	2.0K
Pets	0.25	0.3K	1.4K	0.9K
Arachnida	0.40	1.2K	2.8K	1.7K
Animalia	0.20	0.6K	3.8K	1.5K
Mollusca	0.30	1.6K	3.5K	2.4K

Table B.3. Statistics on SLE-based Clusters.

Data	$\lvert\mathcal{Y}_A\rvert$	Img/ Cls	Img/ Cls (L)
CUB	116	23	15
Scars	171	27	20
Pets	37	19	49
Arachnida	91	19	59
Animalia	76	15	38
Mollusca	103	28	51

B.3. Comparison with MixUp and CutMix

Quantitative Comparison. To better understand our method, we conducted experiments on three datasets CUB, Stanfors Cars and Ocford Pets to compare our method with two different mixing methods CutMix and MixUp. Quantitative results are summarized in Table B.4. Similarly, "SMILE + CutMix" and "SMILE + MixUp" involve replacing the Attribute Composition Generation (ACG) module in the DiffGRE framework with CutMix and MixUp, respectively, to ensure a fair comparison. We can observe that our method significantly outperforms other methods. Especially in the unknown categories, our method achieves higher accuracy than CutMix and MixUp across all the three datasets. The results further demonstrate the effectiveness

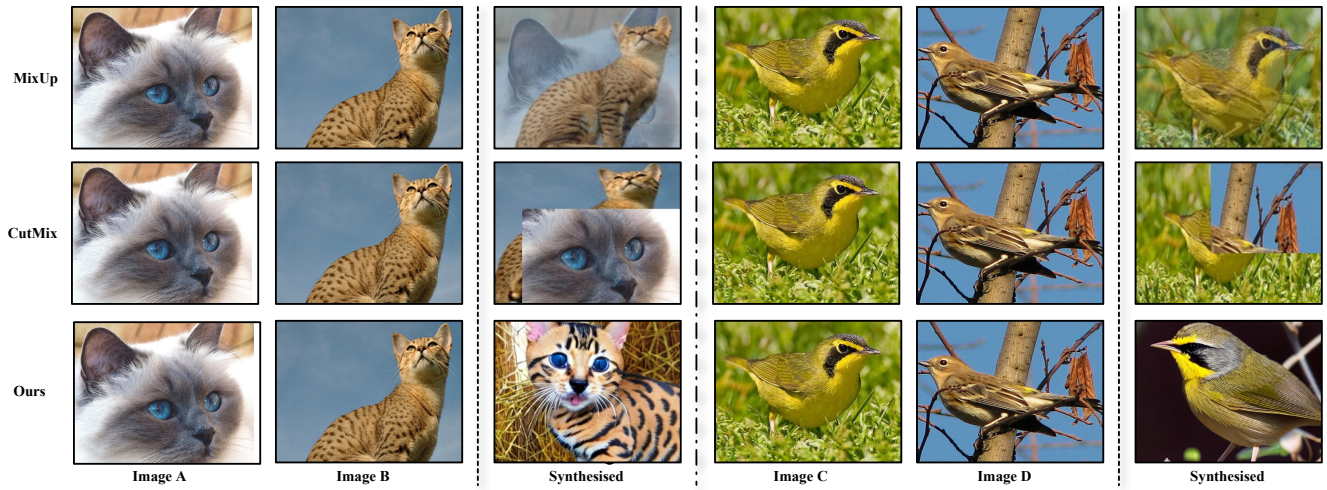

Figure B.2. Our image generation compared with MixUp and CutMix on the CUB and the Pets dataset.

Table B.4. Comparison with CutMix and MixUp on CUB, Stanfors Cars and Oxford Pets.

Method	CUB			SCars			Pets		
	All	Old	New	All	Old	New	All	Old	New
SMILE [2]	32.2	50.9	22.9	26.2	46.7	16.3	41.2	42.1	40.7
SMILE [2] + CutMix [10]	32.8	54.6	21.9	28.2	55.0	15.3	40.9	46.3	38.0
SMILE [2] + MixUp [11]	33.3	55.6	22.1	26.9	51.8	14.9	37.9	41.6	35.9
SMILE [2] + DiffGRE (Ours)	35.4	58.2	23.8	30.5	59.3	16.5	42.4	42.1	42.5

of dual interpolation in the latent space of the Diversity-Driven Refinement (DDR) module, compared to directly performing pixel-level interpolation.

Qualitative Comparison. We also visualize the generated images on different datasets (CUB and Oxford Pets) from our method, MixUp and CutMix. Examples are illustrated in Figure B.2. First, we notice that for MixUp and CutMix, the results look like two images are randomly stitched together. But our synthesized images are more like real images. Second, our synthesized images simultaneously incorporate attributes from both input images. For example, in the case on the left, the synthesized cat has blue eyes, which is the same as the input image A. This visualization intuitively demonstrates the effectiveness of semantic latent interpolation in the Diversity-Driven Refinement (DDR) module.

B.4. Details of Diversity-Driven Refinement

In Section 4.4, we discussed the impact of γ in the Diversity-Driven Refinement (DDR) module on the Arachnida dataset, and it is observed that the model achieves the best performance on unknown categories when the number of the remaining images is comparable to that of labeled data. In this section, further details about the chosen value of γ, number of remained images and synthesized images are shown in Table B.2. Following the similar strategy, we set different values for γ on six datasets in our experiments to ensure the number of remained images matches the size of the labeled training set.

On the other hand, we summarize the clustering results of SLE in Tab. B.3, where the first column shows the number of categories to which samples from both known and virtual categories have been assigned. Moreover, we compared the average number of images per category (the 3rd Column) from SLE with the labeled training set (the 4th Column) and find they share the same scale. These results confirm sufficiency of the generated samples.

Table B.5. Virtual Category Assignment Strategy

Index	Method	CUB		
		All	Old	New
a)	w/o class centers	33.5	55.5	22.5
b)	w/o category assignment	31.3	51.9	20.9
c)	SMILE [2] + DiffGRE (Ours)	35.4	58.2	23.8

B.5. Virtual Category Assignment Strategy

We verify the effectiveness of our design for the virtual category assignment in Tab. B.5. Specifically, **a)** by replacing class center distances with distances to all training samples, the results show a decrease in performance compared to our proposed design. **b)** When virtual category assignment is removed, performance further drops. **c)** These results are compared to the reference performance of our full method, which highlights the effectiveness of our approach.

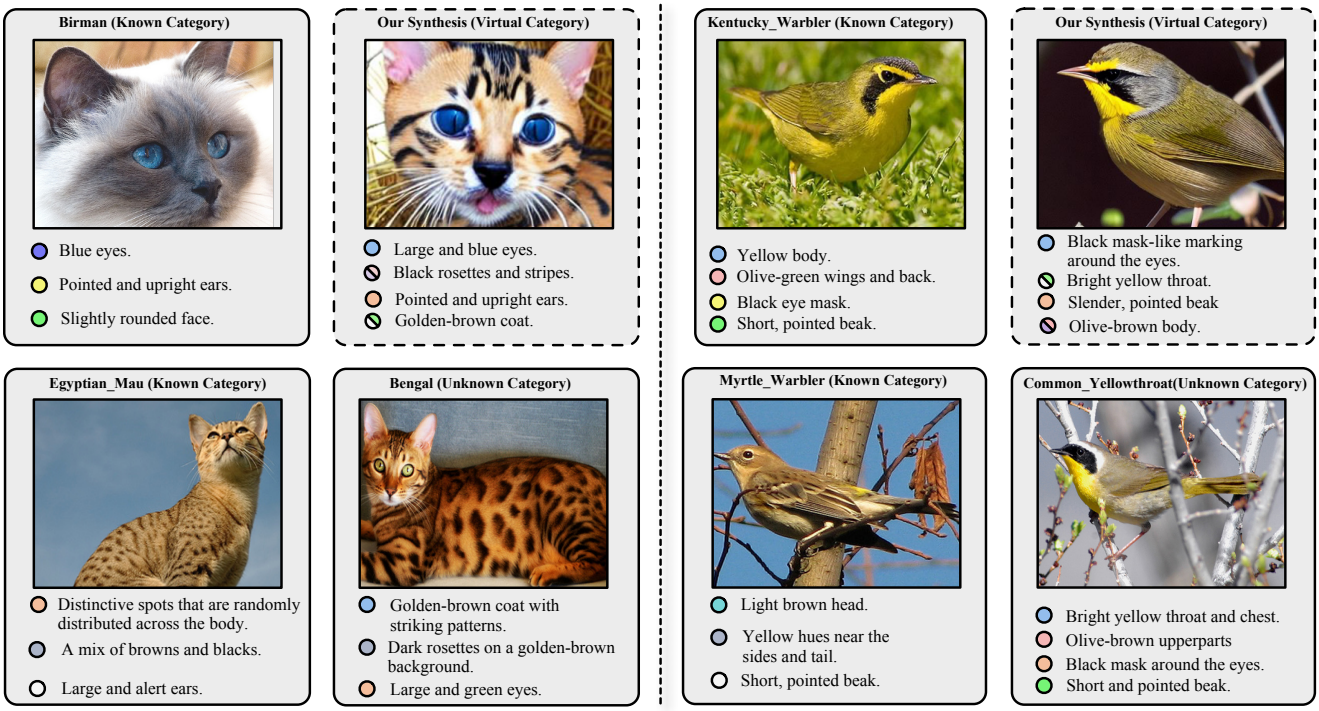

Figure B.3. Additional Examples for Attribute Composition Generation (ACG).

Table B.6. Comparison of baseline methods on training time, inference time, model size, and ALL-ACC for the CUB dataset. "*" denotes results obtained with 4 GPUs in parallel.

Method	Inference Strategy	Finetune Time ≈(h)	Synthesis Time ≈(h)	Training Time ≈(s)	Model Size ≈(M)	Inference Time ≈(s)	ALL-ACC (%)
SMILE [2]	Hash-like	0	0	960.6	13.4	11.2	32.2
SMILE [2] + Diff-Mix [9]	SLE-based	16	2*	8799.9	13.4	17.5	32.9
SMILE [2] + DiffGRE (Ours)	SLE-based	0	12	8799.9	13.4	17.5	35.4
SMILE [2] + DiffGRE (w/o DDR)	SLE-based	0	12	9160.5	13.4	17.5	30.4

Table B.7. Evaluation on the hyperparameter β.

Value	CUB			SCars		
	All	Old	New	All	Old	New
0.5	34.6	57.6	23.0	28.4	54.7	15.7
0.75	33.9	56.2	22.8	28.1	54.0	15.5
1.0	35.4	58.2	23.8	30.5	59.3	16.5
1.25	34.8	57.8	23.2	26.9	49.9	15.7
1.5	33.2	55.1	22.3	27.4	51.6	15.7

C. Additional Visualization Results

C.1. TSNE comparison between our DiffGRE and Diff-Mix

We visualize the features generated by our DiffGRE and Diff-Mix [9] by TSNE in Fig. B.4. Through comparing these two sub-figures, we find that the images generated by Diff-Mix [9] are surrounded by the known-category and unknown-category samples, which indicates that the diversity of the generated images is limited. In contrast, our DiffGRE can generate diverse images that even belong to synthesized purely new categories. Thus, our methods are more effective in improving OCD performance by generating additional and virtual category information.

C.2. Additional successful examples

We discussed visualization results in Section 4.5 in our main submission. In this section, we provide additional examples for Attribute Composition Generation (ACG). These examples are presented in Figure B.3. Positive examples can support that our method is able to synthesize novel samples that include additional category knowledge. An example is shown on the right of Figure B.3. We find that the synthesized sample has attributes such as a black mask-like marking around the eyes, an olive-brown body, and a short, pointed beak, closely resembling a sample from the unknown category Common Yellowthroat.

D. Computational Consumption

We also compare computational costs of our framework, SMILE and Diff-Mix in Table B.6, analyzing Training Time, Inference Time, ALL-ACC, and other metrics. Ex-

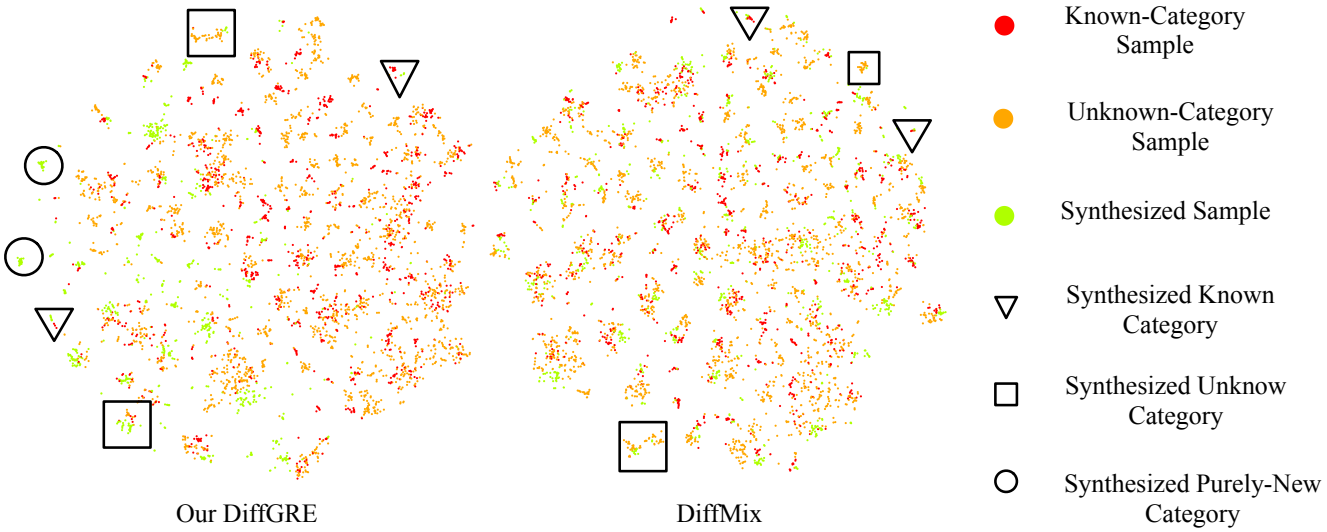

Figure B.4. TSNE visualization of the feature of the images generated by our DiffGRE and Diff-Mix [9]. Through comparing these two sub-figures, we find that the images generated by Diff-Mix [9] are surrounded by the known-category and unknown-category samples, which indicates that the diversity of the generated images is limited. In contrast, our DiffGRE can generate diverse images that even belong to synthesized purely new categories. Thus, our methods are more effective in improving OCD performance by generating additional and virtual category information.

Table B.8. Comparison with **training-free hash-like inference** methods, with the best results in **bold** and the second best underlined.

Method	Arachnida			Mollusca			Oxford Pets			Average		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
RankStat [3]	26.6	51.0	10.0	29.3	55.2	15.5	33.2	42.3	28.4	29.7	49.5	18.0
WTA [5]	28.1	55.5	10.9	30.3	**55.4**	17.0	35.2	**46.3**	29.3	31.2	**52.4**	19.1
SMILE [2] + DiffGRE (Ours)	**35.4**	**66.8**	**15.6**	**36.5**	44.2	**32.5**	**42.4**	42.1	**42.5**	**38.2**	51.0	**30.2**
Method	CUB			Stanford Cars			Animalia			Average		
	All	Old	New	All	Old	New	All	Old	New	All	Old	New
RankStat [3]	21.2	26.9	18.4	14.8	19.9	12.3	31.4	54.9	21.6	22.5	33.9	17.4
WTA [5]	21.9	26.9	19.4	17.1	24.4	13.6	33.4	59.8	22.4	24.1	37.0	18.5
SMILE [2] + DiffGRE (Ours)	**35.4**	**58.2**	23.8	**30.5**	**59.3**	16.5	**37.4**	**69.3**	24.3	**34.4**	**62.3**	21.5

periments are conducted on the CUB dataset. Diff-Mix is applied to generate 5000 images, matching the number of images generated by our framework. "SMILE + Diff-Mix" refers to substituting the Attribute Composition Generation (ACG) module in the DiffGRE framework with Diff-Mix, in order to provide a balanced comparison. Notably, Diff-Mix involves two steps for generating new images: first, finetuning its diffusion model, and second, using the pre-trained model to synthesize samples. Our framework achieves better performance with slightly higher inference time than SMILE. On the other hand, the total generation time for "SMILE + Diff-Mix" is **16**+**2** = **18** hours, while our framework completes the process in just **12** hours, as it does not require finetuning. Additionally, when the DDR module is removed from our framework, we observe longer training time and lower accuracy, further highlighting the importance of the Diversity-Driven Refinement (DDR) module.

E. Additional Hyper-Parameter Analyses

Interpolation Parameters. As shown in Fig. E.1, we analyzed the impact of λ_v and λ_l, for visual embedding interpolation and latent embedding interpolation. Theoretically, if λ_v or λ_l are close to 0/1, the generated image resembles one of the original images, resulting in low diversity. At 0.5, there is higher diversity but also increased ambiguity. The results show that the parameters are not sensitive, thus we empirically set these interpolation parameters to $\lambda_v = 0.7$ and $\lambda_l = 0.8$, respectively.

Loss Weight. We analyzed the impact of hyperparameter β in Section 4.4. Additional results of the analysis on β are summarized in Table B.7. Experiments are implemented on two datasets CUB and Stanford Cars. Our model attains the highest accuracy when β increases from 0.5 to 1.5 on the two datasets. Based on the results, we set $\beta = 1.0$ for all datasets during training.

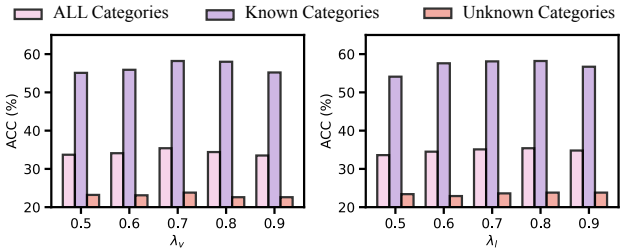

Figure E.1. Illustration for the impact of the varying interpolation parameters, λ_v and λ_l, for visual embedding and latent embedding, respectively.

F. Additional Evaluation of our DiffGRE compared with training-free hash-like methods

We further conduct experiments to evaluate our method compared with training-free hash-like approaches. Results are provided in Tab. B.8. The combination of "SMILE + DiffGRE" consistently outperforms all competitors, achieving the highest ACC-ALL averages, which surpasses WTA by 8.7% and RankStat by 10.2% across the six datasets.

G. Quality Evaluation and Failure Rate

To quantitatively assess the visual quality and semantic diversity of the synthesized samples, we conduct evaluations on the CUB dataset. Specifically, we use the Fréchet Inception Distance (FID) to evaluate image quality, where our method achieves a lower score (20.7) compared to DiffMix (23.4), indicating improved visual fidelity (lower is better). This demonstrates the effectiveness of our approach in producing high-quality samples suitable for the OCD setting. Additionally, to measure the semantic diversity of synthesized images, we adopt the CLIP Blending Similarity (CLIP-BS) metric [14], which quantifies the distance between synthesized blends and their source concepts. Our method achieves a higher CLIP-BS score (9.13 vs. 8.32), suggesting that our re-composition strategy yields more novel and semantically distinct images. Although the DDR module effectively filters out low-quality generations, a small proportion of failure cases still occurs due to the inherent randomness of diffusion models, which is a common limitation in generative synthesis. On the CUB dataset, we observe that the failure rate remains below 4%, indicating the overall reliability of our generation process.

H. Zero-shot and Cross-domain OCD

To explore the generalization ability of OCD models in the absence of labeled data, we conduct additional zero-shot experiments based on the SMILE baseline. As shown in Table H.1 (first row), for each dataset, the model is trained solely on synthesized samples and tested on the corresponding test set. This setting allows us to examine whether synthetic data alone can support effective category discovery without any real supervision. The performance drops significantly, indicating that the lack of labeled data hinders the model's ability to learn robust and discriminative features. These results highlight the limitations of relying purely on generative data and suggest that zero-shot OCD remains a challenging and open research problem.

To further investigate model transferability across domains, we train the model on the combined synthetic samples from the Arachnida (A), Mollusca (B), and Animalia (C) datasets and test it separately on each domain. This setting aims to evaluate whether training on mixed-domain synthetic data can improve generalization in a cross-domain OCD scenario. As shown in the second row of Table H.1, performance remains low, likely due to the absence of domain-specific supervision, which is particularly important in fine-grained scenarios. Although building a universal OCD model is an appealing goal, our results suggest that simply mixing synthetic data from multiple domains is insufficient for robust and transferable category discovery.

Table H.1. Zero-shot and cross-domain OCD results under hash-like inference. We report performance when training on synthetic data from either the same or combined domains.

Training Source	**Arachnida (A)**			**Mollusca (B)**			**Animalia (C)**		
	All	Old	New	All	Old	New	All	Old	New
A→A / B→B / C→C	14.8	14.3	15.1	16.2	12.4	18.3	18.8	10.5	22.3
A+B+C → A / B / C	13.2	12.2	13.9	13.9	11.2	15.3	15.4	9.5	17.8

I. Limitation

Our framework includes the offline generation. While this reduces the computational cost, the diffusion model lacks end-to-end optimization, thereby leading to unsatisfactory generative results. We will explore finetune-based approaches in the future.

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2

[2] Ruoyi Du, Dongliang Chang, Kongming Liang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. On-the-fly category discovery. In *CVPR*, 2023. 1, 2, 3, 4, 5, 6

[3] Kai Han, Sylvestre-Alvise Rebuffi, Sebastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Autonovel: Automatically discovering and learning novel visual categories. *IEEE TPAMI*, 2021. 2, 6

[4] John A Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975. 2

[5] Xuhui Jia, Kai Han, Yukun Zhu, and Bradley Green. Joint representation learning and novel category discovery on single-and multi-modal data. In *ICCV*, 2021. 2, 6

[6] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967. 3

[7] Ning Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 1999. 2

[8] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *CVPR*, 2018. 1

[9] 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 *CVPR*, 2024. 3, 5, 6

[10] 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 *ICCV*, 2019. 4

[11] Hongyi Zhang, YND Moustapha Cisse, and David Lopez-Paz. mixup: Beyond 903 empirical risk minimization. In *ICLR*, 2018. 4

[12] Feng Zheng, Cheng Deng, Xing Sun, Xinyang Jiang, Xiaowei Guo, Zongqiao Yu, Feiyue Huang, and Rongrong Ji. Pyramidal person re-identification via multi-loss dynamic training. In *CVPR*, 2019. 2, 3

[13] Haiyang Zheng, Nan Pu, Wenjing Li, Nicu Sebe, and Zhun Zhong. Prototypical hash encoding for on-the-fly fine-grained category discovery. *NeurIPS*, 2025. 2

[14] Yufan Zhou, Haoyu Shen, and Huan Wang. Freeblend: Advancing concept blending with staged feedback-driven interpolation diffusion. *arXiv preprint arXiv:2502.05606*, 2025. 7