The Devil Is in Gradient Entanglement: Energy-Aware Gradient Coordinator for Robust Generalized Category Discovery

Supplementary Material

Appendix Contents

A. Dataset Statistics

We follow the standard Generalized Category Discovery (GCD) [63] protocol to split the known/novel classes and determine the number of samples in the labeled and unlabeled subsets. We evaluate on four fine-grained datasets—CUB [66], Stanford Cars [20], Aircraft [38], and Herbarium19 [61]—and two generic datasets, CIFAR-100 [21] and ImageNet-100 [8]. For most datasets, 50% of the classes are designated as known, with the exception of CIFAR-100, which adopts an 80%/20% known/novel split. Following the standard protocol, the labeled subset is constructed by sampling half of the images from the known classes. The unlabeled subset consists of all remaining images, which includes the other half of the known-class samples and all samples from the novel classes. Detailed statistics for all six datasets are provided in Tab. A.

Table A. Dataset statistics and splits.

Dataset	Labeled		Unlabeled	
	# Images	# Classes	# Images	# Classes
CUB [66]	1,498	100	4,496	200
Stanford Cars [20]	2,000	98	6,144	196
Aircraft [38]	1,666	50	5,001	100
CIFAR-100 [21]	20,000	80	30,000	100
ImageNet-100 [8]	31,860	50	95,255	100
Herbarium19 [61]	8,869	341	25,356	683

B. Extended Experimental Results

B.1. Computational overhead and scalability

In this section, we provide a detailed analysis of the computational overhead and scalability of the proposed Energy-Aware Gradient Coordinator (EAGC). First, it is important to note that EAGC is exclusively a training-time module. It incurs **zero computational cost during inference**. During training, the overhead of EAGC can be evaluated across three main aspects: memory consumption, matrix inversion efficiency, and time overhead. All empirical profiling is conducted and averaged over three independent runs on an NVIDIA A100 GPU, with results summarized in Tab. B.

Table B. Computational overhead analysis on an NVIDIA A100.

Data	Method	All ACC	Peak Mem (MB)	Inv. Cost (ms/ep)	Time Costs (s) Avg. Epoch	Time Costs (s) $\mathcal{E}_r(\cdot)$ Constr.
CUB	SimGCD	60.3	4289.4	-	55.9	-
	+ ***EAGC***	66.5	4927.2	2.9	63.8	467.9
IN-100	SimGCD	83.0	4288.1	-	947.7	-
	+ ***EAGC***	83.5	4921.3	3.5	1262.2	6899.1

Memory Overhead. EAGC requires a slight increase in GPU memory (e.g., from 4289 MB to 4927 MB on the CUB dataset). This increase is primarily due to loading the frozen reference model $\mathcal{E}_r(\cdot)$ to extract reference features. Since $\mathcal{E}_r(\cdot)$ is kept strictly frozen, we do not need to construct or store its computational graph for gradients, ensuring that the memory overhead remains highly manageable.

Scalability of Matrix Inversion. The matrix inversion step in the elastic projection scales efficiently with the feature dimension. Profiling on ImageNet-100 reveals that the inversion cost is 3.5 ms/epoch for a feature dimension of $d = 768$. Increasing the dimension to $d = 1024$ yields an inversion cost of 7.3 ms/epoch. This demonstrates that matrix inversion adds only a small overhead to the training process, confirming the scalability of EAGC to higher-dimensional representations.

Training Time versus Performance Gain. The time overhead introduced by EAGC consists of two components: a one-time construction cost for the reference model $\mathcal{E}_r(\cdot)$ prior to the main training phase, and the per-epoch gradient coordination cost. As shown in Tab. B, on the CUB dataset, the one-time construction takes 467.9 seconds, and the per-epoch overhead is approximately 7.9 seconds (a 14.1% increase relative to the SimGCD baseline). Overall, the method effectively translates this 14.1% increase in

training time into a 6.2% absolute gain in All ACC on the CUB dataset.

B.2. Effectiveness on the Herbarium19 Dataset

To further evaluate the generalizability and robustness of our proposed EAGC on large-scale, long-tailed distributions, we conduct extended experiments on the challenging Herbarium19 dataset. As shown in Tab. C, integrating EAGC consistently improves the performance of both the parametric baseline (SimGCD) and the non-parametric baseline (SelEx). Specifically, EAGC yields an average absolute improvement of 3.4% in All ACC and 4.9% in New ACC across the two baselines. Notably, when applied to SimGCD, EAGC significantly boosts the New ACC by 6.0%. This demonstrates the effectiveness and robustness of our gradient coordination mechanism.

Table C. Extended results on the long-tailed Herbarium19 dataset.

Method	SimGCD			SelEx		
	All	Old	New	All	Old	New
Baseline	44.0	58.0	36.4	39.6	54.9	31.3
+ EAGC	**48.4**	**59.8**	**42.4**	**42.0**	**55.0**	**35.0**

B.3. Results with Unknown K

Our EAGC is a plug-and-play module applied during GCD model training and can also be used when the number of novel classes is unknown. Using the class-number estimation method from GCD [63], we evaluate the effect of integrating EAGC into the baselines on CUB, Stanford Cars, and ImageNet-100. The results are shown in Tab. D. Specifically, on CUB (estimated 231 vs. ground-truth 200 classes), Stanford Cars (estimated 230 vs. 196), and ImageNet-100 (estimated 108 vs. 100), our EAGC consistently improves the baselines—boosting SimGCD by an average of 6.6% and SelEx by 6.7% in All ACC. Notably, on the two fine-grained datasets, even under large estimation errors in the number of classes, EAGC significantly enhances the novel-class discovery performance, yielding an average improvement of 10.9% in New ACC across both baselines.

Table D. Comparison without Number of Categories K

Method	CUB			Stanford Cars			ImageNet-100		
	All	Old	New	All	Old	New	All	Old	New
SimGCD [74]	61.5	66.4	59.1	49.1	65.1	41.3	81.7	91.2	76.8
+EAGC	66.5	69.4	65.0	63.3	74.8	**57.8**	82.2	94.1	76.2
SelEx [50]	72.0	72.3	71.9	58.7	75.3	50.8	85.4	94.0	81.0
+EAGC	**83.1**	**75.4**	**86.9**	**64.2**	**79.1**	56.9	**88.9**	**95.5**	**85.5**
Avg. △	+8.1	+3.1	+10.5	+9.9	+6.8	+11.3	+2.0	+2.2	+2.0

B.4. Exploration of the Reference Model

Our AGA relies on a reference model trained in a supervised manner on the labeled subset to perform gradient alignment. Here, we conduct an extensive analysis of the reference model.

1) Training-related hyperparameters. In EAGC, we use a batch size of 32 for fine-grained datasets and 128 for generic datasets, and we train for 30 epochs on all datasets. *i) Batch size.* GCD training typically uses mixed labeled and unlabeled data within a mini-batch with a batch size of 128. However, for supervised training on the labeled subset, we find that using a smaller batch size yields better results for fine-grained datasets. As shown in Tab. E, using a batch size of 32 reduces $\mathcal{L}_{\text{cls}}$ by an average of 0.18 across the three datasets and improves All ACC by an average of 3.3% compared to a batch size of 128. *ii) Training epoch.* We evaluate training for 20, 30, and 100 epochs, with results shown in Tab. F. Overall, too few epochs may lead to underfitting on known classes, while too many epochs can cause overfitting. We therefore choose a balanced setting and fix the training schedule to 30 epochs unless otherwise specified.

Table E. Effect of batch size for training the reference model.

Batch	CUB				Stanford Cars				Aircraft			
Size	$\mathcal{L}_{\text{cls}}$	All	Old	New	$\mathcal{L}_{\text{cls}}$	All	Old	New	$\mathcal{L}_{\text{cls}}$	All	Old	New
32	0.021	66.5	71.0	64.3	0.034	**62.9**	**76.0**	**56.6**	0.042	**57.7**	**60.4**	**56.3**
128	0.035	**68.6**	**71.3**	**67.3**	0.052	58.2	71.8	51.7	0.554	50.3	58.3	46.3

Table F. Effect of the number of training epochs for the reference model.

Training	CUB				Stanford Cars				Aircraft			
Epoch	$\mathcal{L}_{\text{cls}}$	All	Old	New	$\mathcal{L}_{\text{cls}}$	All	Old	New	$\mathcal{L}_{\text{cls}}$	All	Old	New
20	0.046	**67.8**	**71.8**	**65.9**	0.130	55.1	71.8	47.0	0.232	55.6	**60.9**	52.9
30	0.021	66.5	71.0	64.3	0.034	**62.9**	**76.0**	**56.6**	0.042	57.7	60.4	56.3
100	0.014	65.7	67.5	64.8	0.018	54.9	69.0	48.1	0.011	**57.9**	58.3	**57.7**

2) Exploring trainable parameters. In EAGC, we **follow the baselines' configuration for the reference model's trainable parameters**, e.g., setting the number of unfrozen blocks to 1 for SimGCD and 2 for SelEx. We further explore the effect of trainable parameters using SimGCD as the baseline. As shown in Tab. G, using more trainable parameters helps the model better learn known classes, resulting in lower $\mathcal{L}_{\text{cls}}$.

Table G. Effect of the number of unfrozen ViT blocks in the reference model.

Unfrozen	CUB				Stanford Cars			
Blocks	$\mathcal{L}_{\text{cls}}$	All	Old	New	$\mathcal{L}_{\text{cls}}$	All	Old	New
1	0.021	66.5	71.0	64.3	0.034	62.9	**76.0**	56.6
2	0.015	65.5	**73.3**	61.5	0.028	56.9	74.9	48.2
3	0.013	66.2	71.2	63.7	0.025	60.6	72.1	55.1
4	0.012	**67.3**	69.6	**66.1**	0.016	**66.4**	74.7	**62.4**
5	0.012	66.6	69.7	65.1	0.019	57.7	71.4	51.1

C. Theory

C.1. Motivation from Proximal Regularization

In GCD, the joint objective $\mathcal{L}_{\text{GCD}} = \alpha\mathcal{L}_{\text{sup}} + \beta\mathcal{L}_{\text{unsup}}$ combines a relatively reliable supervised signal from labeled data with a much noisier unsupervised signal driven by self-supervision or pseudo-labels. This intrinsic asymmetry means that the overall gradient can easily deviate from the desirable supervised direction, especially in the early training stage where $\mathcal{L}_{\text{unsup}}$ is highly unstable. *Proximal regularization* [40, 81] is a classical tool for stabilizing optimization under noisy or conflicting gradients, by penalizing deviations from a trusted reference point and thereby damping harmful updates. Viewed through this lens, it is natural to interpret our Anchor-based Gradient Alignment (AGA) as introducing a proximal force that keeps the optimization of labeled samples close to a supervised anchor, while still allowing the model to benefit from unlabeled objectives.

C.2. Proximal Interpretation of AGA

Given the reference feature $\hat{z}^l$ obtained from the reference model $\mathcal{E}_r(\cdot)$, AGA introduces the following alignment term for labeled samples:

$$\nabla_{z^l} g_{\text{align}} = \lambda_a(z^l - \hat{z}^l). \tag{13}$$

This term corresponds exactly to the gradient of a proximal regularizer that constrains the labeled representation z^l to remain within a neighborhood of the reliable supervised optimum $\hat{z}^l$. Under this view, the effective objective for labeled data becomes:

$$\mathcal{L}'_{\text{sup}} = \mathcal{L}_{\text{sup}} + \frac{\lambda_a}{2}\|z^l - \hat{z}^l\|_2^2, \tag{14}$$

which is a classical proximal step widely used in numerical optimization and robust learning. Therefore, AGA can be rigorously interpreted as a *feature-space proximal point update*, where the reference feature $\hat{z}^l$ serves as the proximal center and λ_a governs the radius of the trust region. This viewpoint clarifies that AGA does not override unsupervised learning; instead, it stabilizes supervised learning by regularizing labeled features toward a reliable optimum, thereby suppressing gradient distortion induced by $\mathcal{L}_{\text{unsup}}$. In our EAGC framework, AGA serves as a principled proximal regularizer for labeled-sample gradients under a unified gradient-optimization perspective.

C.3. Theoretical Properties

We next show that viewing AGA as a proximal regularizer implies two beneficial properties under mild local assumptions.

Lemma 1 (Gradient variance reduction). Consider the labeled feature z^l in a local neighborhood of the supervised optimum $\hat{z}^l$, and assume that the supervised loss *admits a second–order approximation*

$$\mathcal{L}_{\text{sup}}(z^l) \approx \mathcal{L}_{\text{sup}}(\hat{z}^l) + \tfrac{1}{2}(z^l - \hat{z}^l)^\top H(z^l - \hat{z}^l), \tag{15}$$

where $H \succeq 0$ denotes the local Hessian. We model the interference from the unsupervised objective as a stochastic disturbance ξ acting on the shared backbone's gradient pathway. Following standard analyses of noise-driven gradient perturbations, we assume

$$\mathbb{E}[\xi] = 0, \quad \text{Cov}[\xi] = \Sigma. \tag{16}$$

Then, under a step size that ensures stable local updates around the supervised optimum and the quadratic approximation above, the steady–state covariance of the labeled–feature gradient with AGA is no larger (in the PSD sense) than that without AGA:

$$\text{Cov}\big[\nabla_{z^l}\mathcal{L}'_{\text{sup}}(z^l)\big] \preceq \text{Cov}\big[\nabla_{z^l}\mathcal{L}_{\text{sup}}(z^l)\big], \tag{17}$$

where $\mathcal{L}'_{\text{sup}}(z^l) = \mathcal{L}_{\text{sup}}(z^l) + \frac{\lambda_a}{2}\|z^l - \hat{z}^l\|_2^2$ denotes the proximal-regularized objective.

Proof. Under the quadratic approximation, the supervised gradient is

$$\nabla_{z^l}\mathcal{L}_{\text{sup}}(z^l) = H(z^l - \hat{z}^l). \tag{18}$$

A single feature–level gradient step *without* AGA (ignoring higher–order terms) can be written as

$$z^l_{t+1} = z^l_t - \eta\big(H(z^l_t - \hat{z}^l) + \xi_t\big), \tag{19}$$

where $t \in \{0, 1, 2, \dots\}$ denotes the iteration index of gradient descent, $\eta > 0$ is the learning rate and ξ_t is the stochastic perturbation from the unlabeled objective. Defining $\delta_t = z^l_t - \hat{z}^l$, this becomes a linear stochastic system:

$$\delta_{t+1} = (I - \eta H)\delta_t - \eta\xi_t. \tag{20}$$

Assuming the Markov chain is stable, the stationary covariance $\Sigma_\delta = \text{Cov}[\delta_t]$ satisfies (up to $\mathcal{O}(\eta^2)$ terms):

$$H\,\Sigma_\delta + \Sigma_\delta H \;\approx\; \eta\,\Sigma. \tag{21}$$

Consequently, the covariance of the supervised gradient is:

$$\text{Cov}\big[\nabla_{z^l}\mathcal{L}_{\text{sup}}(z^l)\big] = \text{Cov}[H\delta_t] \approx H\Sigma_\delta H. \tag{22}$$

With AGA, the effective objective becomes

$$\mathcal{L}'_{\text{sup}}(z^l) = \mathcal{L}_{\text{sup}}(z^l) + \tfrac{\lambda_a}{2}\|z^l - \hat{z}^l\|_2^2, \tag{23}$$

whose gradient is

$$\nabla_{z^l}\mathcal{L}'_{\text{sup}}(z^l) = (H + \lambda_a I)(z^l - \hat{z}^l). \tag{24}$$

The corresponding feature dynamics (again in the local quadratic regime) are

$$\delta_{t+1} = (I - \eta(H + \lambda_a I))\delta_t - \eta\xi_t. \tag{25}$$

Table H. Evaluation on loss-based proximal regularization (replacing gradient-level AGA with the explicit loss $\mathcal{L}'_{\text{sup}}$)

Method	CUB			Stanford Cars		
	All	Old	New	All	Old	New
SelEx [50]	73.6	75.3	72.8	58.5	75.6	50.3
+EAGC	**83.2**	73.9	**87.9**	**65.7**	83.7	**57.0**
+EAGC$_{\text{loss}}$	82.1	**74.5**	85.9	65.1	**83.9**	56.1

By the same argument, the stationary covariance Σ'_δ now satisfies

$$(H + \lambda_a I)\Sigma'_\delta + \Sigma'_\delta(H + \lambda_a I) \;\approx\; \eta\,\Sigma, \tag{26}$$

and thus

$$\mathrm{Cov}\big[\nabla_{z^l}\mathcal{L}'_{\text{sup}}(z^l)\big] \approx (H + \lambda_a I)\,\Sigma'_\delta\,(H + \lambda_a I). \tag{27}$$

Solving the Lyapunov equations in both cases yields the well–known "ridge" effect:

$$\Sigma'_\delta \;\preceq\; \Sigma_\delta, \quad (H+\lambda_a I)^{-1}\Sigma(H+\lambda_a I)^{-1} \;\preceq\; H^{-1}\Sigma H^{-1}, \tag{28}$$

whenever $\lambda_a > 0$ and $H \succeq 0$. Equivalently,

$$\mathrm{Cov}\big[\nabla_{z^l}\mathcal{L}'_{\text{sup}}(z^l)\big] \;\preceq\; \mathrm{Cov}\big[\nabla_{z^l}\mathcal{L}_{\text{sup}}(z^l)\big]. \tag{29}$$

□

C.4. Empirical Exploration and Validation

Building on SelEx [50], which achieves the best performance, we design two sets of experiments to examine the role of proximal regularization: (i) we directly adopt $\mathcal{L}'_{\text{sup}}$ in Eq. (14) as an explicit optimization objective for labeled data and disable the AGA module within EAGC; (ii) we replace AGA with relation-based distillation losses.

(i) Loss-based proximal regularization. As shown in Tab. H, when we replace AGA with $\mathcal{L}'_{\text{sup}}$ as the optimization objective for labeled data, the overall performance is almost identical to that of the original AGA: compared with the SelEx baseline, we obtain an average improvement of about 7.6% in All ACC on the two datasets. This result validates our design from two perspectives: first, AGA essentially implements, at the gradient level, the proximal regularizer induced by $\mathcal{L}'_{\text{sup}}$; second, proximal regularization with respect to a reference model can substantially boost overall GCD performance.

(ii) Relation-based distillation in place of proximal regularization. This experiment has two aims: (i) to verify whether a reference model trained solely on labeled data can provide effective structural guidance for known classes; and (ii) to assess to what extent different forms of constraints can substitute for AGA. As illustrated in Fig. A, we construct two relation-based distillation variants on top of the reference model: (a) Gram-matrix distillation, which aligns the global relations among labeled samples (denoted as `+RelDistill`$_{\texttt{Gram}}$); and (b) sample-level relation distillation based on neighbor distributions (denoted as `+RelDistill`$_{\texttt{sample}}$). The results are summarized in Tab. I. We observe that Gram-matrix distillation can enforce that the overall pairwise relations among labeled samples remain consistent with those of the reference model, but it cannot prevent global drift in the feature space; as a result, it still trails our AGA by an average of 4.4% in All ACC across the two datasets. In contrast, the sample-level relation distillation more effectively constrains the labeled samples and better preserves the model's discriminative ability on known classes: across the two datasets, its All ACC is on average 3.0% lower than AGA, while its Old ACC is 1.9% higher. Overall, both the gradient-level proximal regularization implemented by AGA and the relation-based distillation derived from the reference model can substantially strengthen the original SelEx baseline, albeit to different extents.

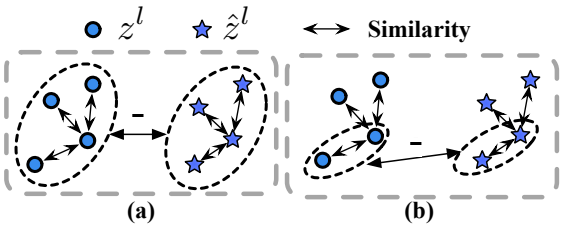

Figure A. Comparison of two relation-based distillation variants: (a) Gram-matrix distillation; (b) sample-level relation distillation.

Table I. Comparison of relation-based distillation variants and gradient-level AGA.

Method	CUB			Stanford Cars		
	All	Old	New	All	Old	New
SelEx [50]	73.6	75.3	72.8	58.5	75.6	50.3
+EAGC	**83.2**	73.9	**87.9**	**65.7**	**83.7**	**57.0**
+RelDistill$_{\text{Gram}}$	80.4	68.5	86.3	59.8	79.7	50.1
+RelDistill$_{\text{sample}}$	82.9	**79.5**	84.6	60.0	81.9	49.5

C.5. Empirical Validation of Gradient Entanglement Hypotheses

To provide deeper insights into the optimization dynamics of Generalized Category Discovery (GCD) and empirically validate our core hypotheses, we conduct a detailed gradient analysis using the SimGCD baseline on the CUB dataset. Specifically, we aim to examine two key assumptions: first, that novel-class samples can induce gradient deviation; and second, that known-class gradients may dominate the optimization process, contributing to representation collapse.

Impact of Novel Classes on Gradient Deviation. We first analyze the gradient directions at the same parameter checkpoint under four distinct data settings: labeled data only (g_L), labeled combined with known-class unlabeled

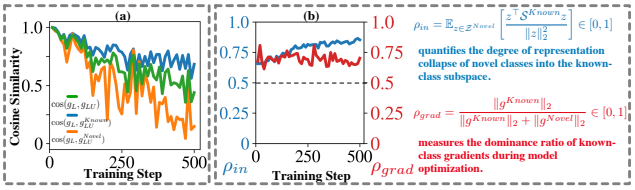

Figure B. Empirical validation of gradient entanglement. (a) Cosine similarity between the reference supervised gradient (g_L) and various mixed gradients. (b) The evolution of gradient dominance (ρ_{grad}) and representation collapse coefficient (ρ_{in}) during the joint training process.

data (g_{LU}^{Known}), labeled combined with novel-class unlabeled data (g_{LU}^{Novel}), and the standard GCD setting using all data (g_{LU}). As illustrated in Fig. B (a), the gradient g_{LU}^{Known} maintains a high cosine similarity with the reference supervised gradient g_L. In contrast, the introduction of novel-class unlabeled samples (yielding g_{LU}^{Novel} and g_{LU}) noticeably reduces this similarity, resulting in a clear angular deviation. This observation empirically supports our hypothesis that novel-class samples can directly distort the optimization direction of the supervised objective.

Gradient Dominance and Representation Collapse. Furthermore, we hypothesize that the optimization process is largely dominated by known-class gradients, which inadvertently pull novel-class representations into the known-class subspace. To validate this, we track two metrics throughout the training process: gradient dominance (ρ_{grad}), defined as the proportion of the total gradient norm contributed by known classes, and the representation collapse coefficient (ρ_{in}), which quantifies the extent to which novel features are projected into the known-class subspace. As shown in Fig. B (b), ρ_{grad} consistently remains above 0.5, confirming the dominance of known-class gradients. Concurrently, ρ_{in} steadily increases as training progresses. This suggests that dominant known-class gradients progressively attract novel-class representations, thereby increasing subspace overlap and reducing the separability of novel categories.

D. Broader Impact and Limitations Discussion

Broader Impact. This work improves category discovery in mixed known–novel settings from an optimization perspective, which may enhance the reliability of open-world visual systems when encountering unseen categories. In real-world applications, such capability is particularly relevant for unknown object discovery and dynamic environment understanding, where models are required to distinguish between known and previously unseen objects [31, 75]. In addition, more stable category discovery can facilitate the organization of large-scale unlabeled data, potentially reducing manual annotation efforts in applications such as industrial anomaly detection and novel pattern discovery [18]. Beyond the GCD setting, the proposed optimization perspective on mitigating interference between known and novel representations may also provide insights for broader open-world visual understanding tasks, including open-vocabulary and domain-generalized segmentation [23–25, 90, 94].

Limitations. This method still has several limitations. First, EAGC relies to some extent on the quality of known-class representations, including the reference model introduced by AGA and the known-class subspace constructed by EEP. If these representations are not sufficiently accurate or stable, the effectiveness of gradient coordination may be affected. Second, although EAGC can be integrated into existing frameworks in a plug-and-play manner, it still introduces additional computational overhead, which may become a burden in large-scale training or resource-constrained scenarios. Finally, the current evaluation is mainly conducted on standard GCD benchmarks, and its performance in more complex open-world settings, such as larger category spaces, significant domain shifts, or dynamically changing environments, remains to be further studied.

Future Work. There are several promising directions for future work. First, while the current method is evaluated on standard GCD benchmarks, it would be valuable to study its applicability in more complex open-world settings, such as scenarios with significant domain shifts, noisy pseudo-labels, or more dynamic and non-stationary environments [22, 56, 89]. Second, as EAGC is largely decoupled from specific model architectures, it can be combined with stronger representation learning and pre-training frameworks to further improve the quality of known-class structures and enhance gradient coordination [51, 52]. Finally, it is also of interest to extend the proposed method to broader tasks, including multimodal understanding [27, 28], 3D point cloud perception [47, 48], and robotics-related applications [54, 76].

E. Qualitative Analysis

We present t-SNE visualizations of feature distributions across the CUB, Stanford Cars, and Aircraft datasets in Fig. C, comparing DINO, two baselines (SimGCD and SelEx), and these baselines integrated with our EAGC. To ensure visual clarity, we display only the last 50 classes (all belonging to novel categories) based on class indices. As illustrated, while the baselines demonstrate improved clustering capabilities over the raw DINO features, the incorporation of EAGC yields sharper boundaries and clearer inter-cluster separation. Furthermore, we also visualize the attention maps for DINO, baselines, and EAGC-integrated baselines in Figs. D to G. Benefiting from stabilized gradient updates, our EAGC-integrated baselines demonstrate **more concentrated and precise attention**. The attention focuses on critical fine-grained details, such as bird eyes and beaks,

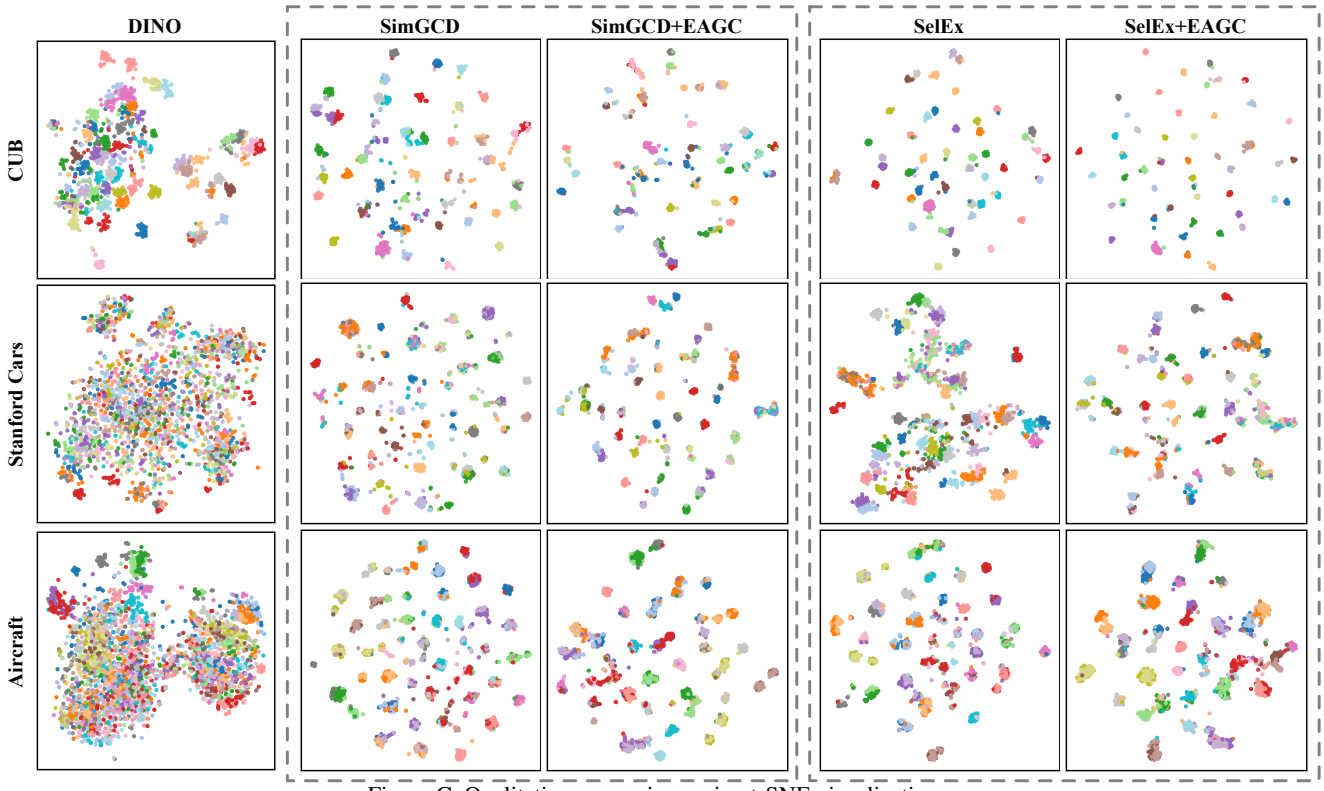

Figure C. Qualitative comparison using t-SNE visualizations.

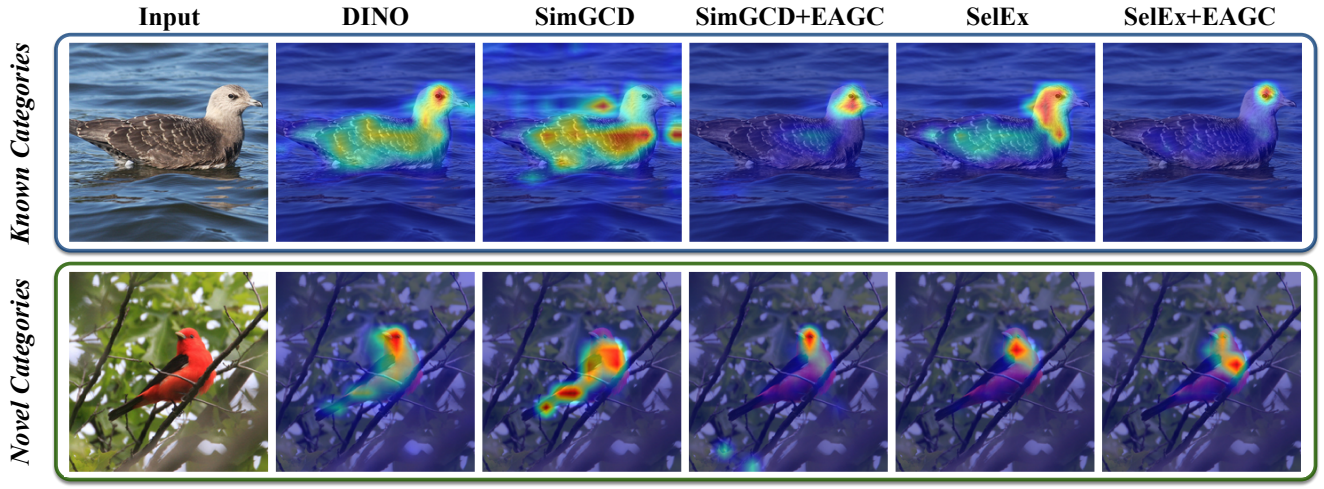

Figure D. Visualization of attention maps of baselines and EAGC-integrated baselines on CUB.

car logos, and aircraft portholes.

Figure E. Additional Visualization of attention maps of baselines and EAGC-integrated baselines on CUB.

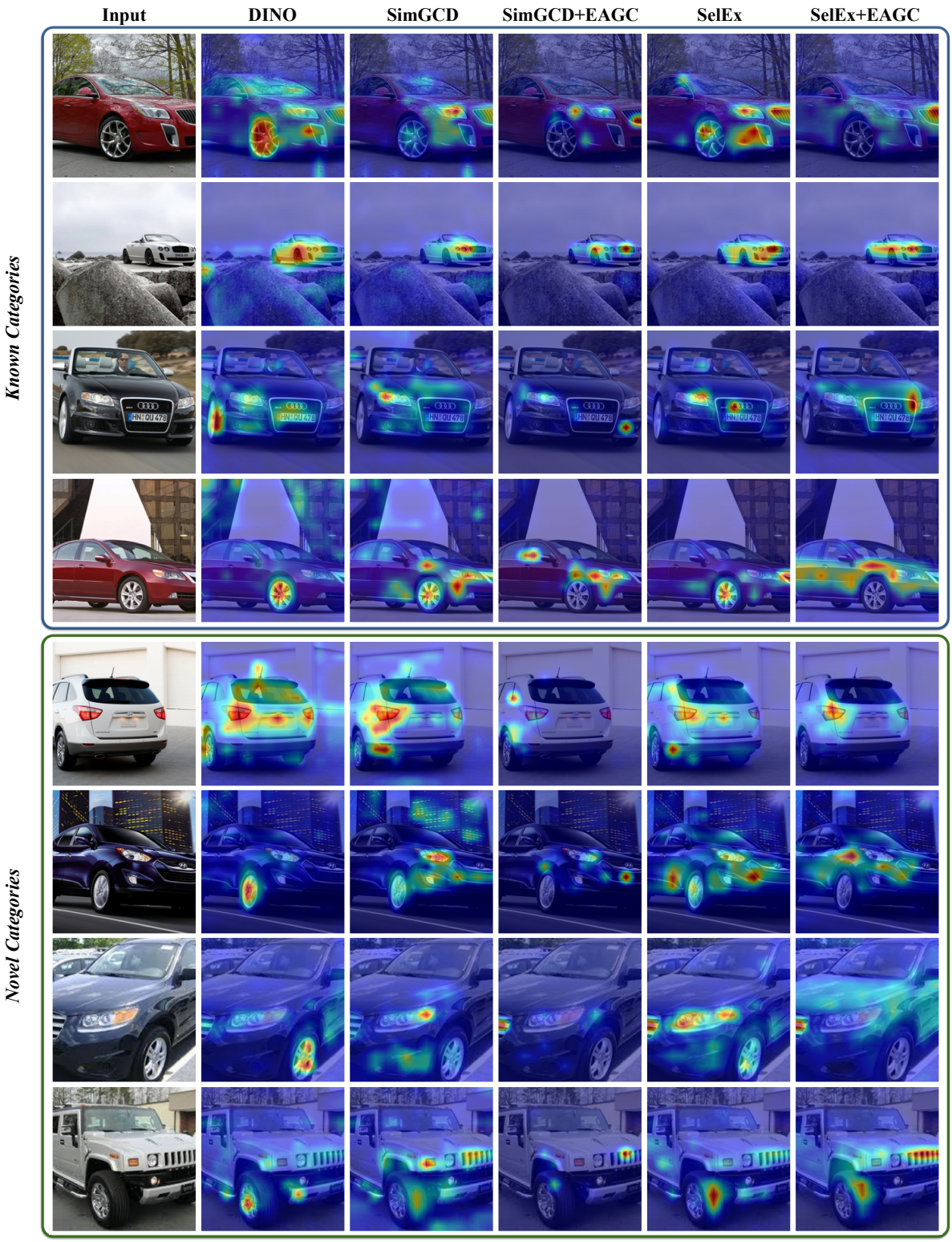

Figure F. Visualization of attention maps of baselines and EAGC-integrated baselines on Stanford Cars.

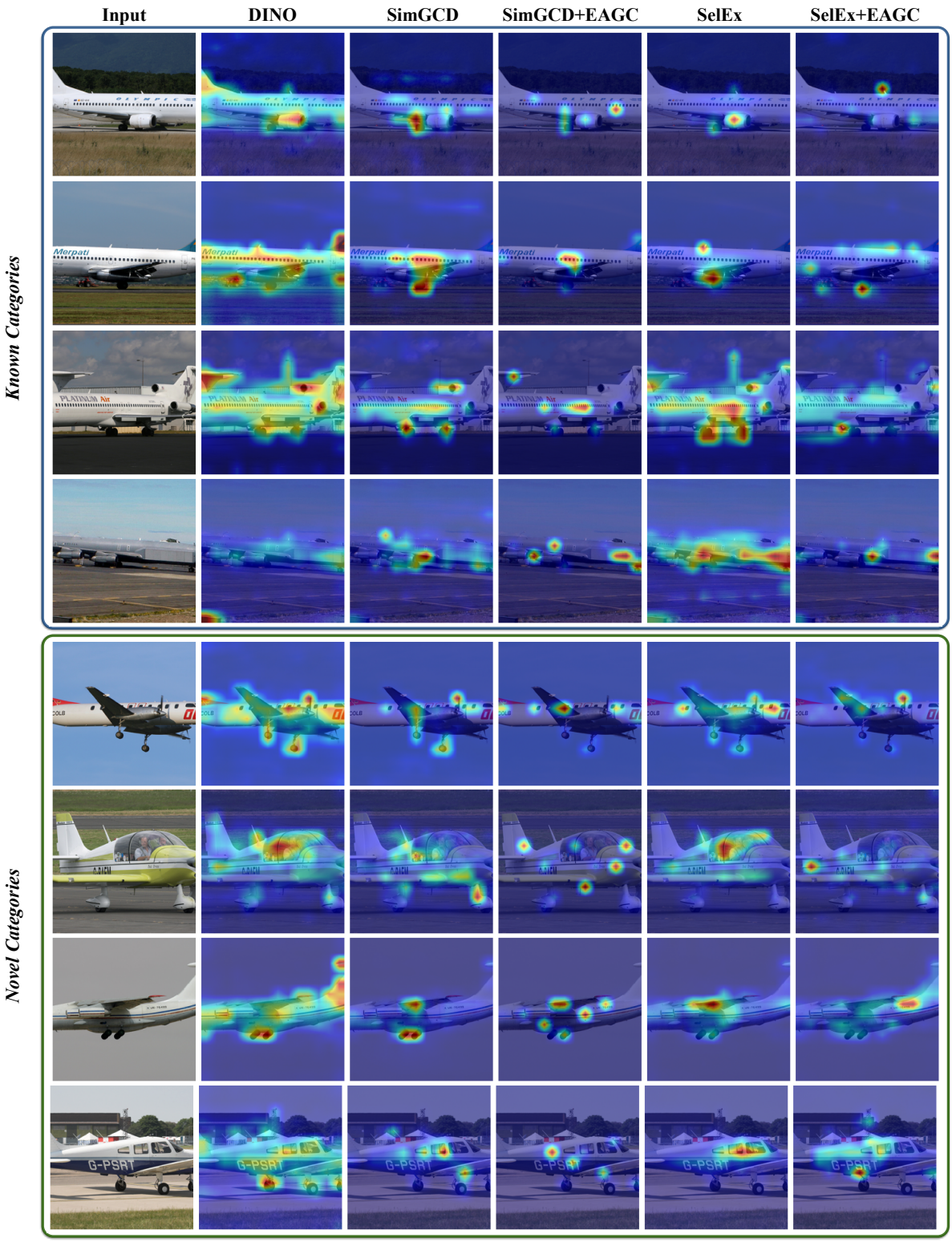

Figure G. Visualization of attention maps of baselines and EAGC-integrated baselines on Aircraft.