ENERGY RING: STABILIZING FEATURE ENERGY FOR RELIABLE TEST-TIME DISCOVERY

Anonymous authors
Paper under double-blind review

ABSTRACT

Test-Time Discovery (TTD) aims to recognize known classes and detect novel ones while the model is already deployed, operating directly on a continual unlabeled test-time stream. Existing TTD methods are overly conservative, making known and unknown features collapse together and difficult to separate during test-time adaptation. In this paper, we revisit TTD from an energy perspective and find that the natural energy gap between known and unknown classes is unstable and easily collapses. We therefore propose Energy Ring to stabilize this gap by anchoring the two types of classes on separate energy rings. A lightweight adapter enlarges the norm-based gap while preserving known-class directions, and a perturbation consistency module forms compact clusters for unknown classes. During testing, energy-guided confidence weighting improves prediction, and a stability-based rectification mechanism corrects unreliable updates. Experimental results show that Energy Ring significantly improves TTD performance against existing TTD methods. Our code will be available.

1 INTRODUCTION

Novel Class Discovery (NCD) (Pu et al., 2023; Wu et al., 2023) aims to identify new categories from unlabeled data using knowledge from known classes. However, it separates the testing and discovery phases, assuming a fixed unlabeled set available in advance. Such a static setting is unsupported in open environments, where new classes appear continuously after deployment. Recently, Test-Time Discovery (TTD) (Lyu et al., 2025) was proposed to detect and learn novel classes while the model is already deployed. Unlike traditional NCD, TTD operates in a continual, unlabeled stream where both known and unknown samples arrive sequentially under test-time constraints, while discovering and testing in real time. This capability is essential for real-world applications where new concepts may emerge unexpectedly after model deployment. Existing TTD methods usually take a conservative strategy to avoid false discovery of new classes and to protect performance on known ones. As shown in Fig. 1, HM (Lyu et al., 2025) freezes the backbone so that pretrained features stay fixed. This prevents drift on known classes but also blocks useful updates for new classes. DAA (Liu et al., 2025) adds a learnable adapter that allows updates, yet forces all feature vectors to have the same norm. This improves stability but removes a key source of variation, which may lead to crowding and confusion across unknown and known classes. In this paper, we revisit the TTD problem from an energy view. Prior studies (Sodano et al., 2024; Gu et al., 2025; Park et al., 2023) suggest that feature energy, measured by the norm of a feature, reflects class-specific information. In many cases, known classes have higher energy and unknown classes have lower energy. This natural gap can be used as a cue to separate old and new classes at test time, reducing interference and improving discovery. *However, the energy of unknown classes is easy to turn chaotic, and the boundary between high and low energy erodes, yielding the unreliability of the whole system.*

Motivated by this, we aim to *amplify* and *stabilize* the energy difference between known and unknown classes. Features are expected to be distributed at varying heights during TTD just like concentric rings, with each category forming distinct clusters, thereby facilitating easier discrimination and discovery. In this paper, we therefore propose **Energy Ring**, a TTD framework that can anchor known classes on a high-energy ring and unknown classes on a low-energy ring. Specifically, we introduce a lightweight energy ring adapter that enlarges separation while preserving feature directions of known classes to retain prior knowledge. In addition, we enforce consistent perturbations to encourage compact and stable clustering of unknown samples along the energy ring. During testing, the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

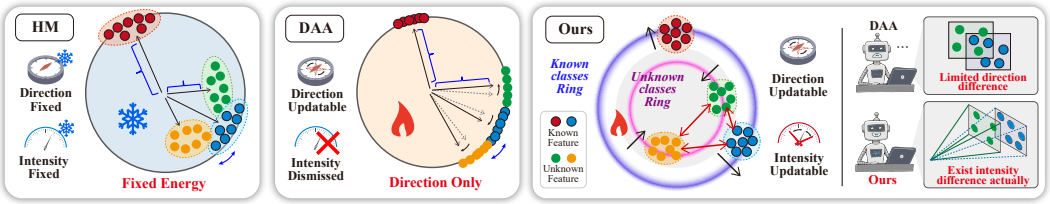


Figure 1: Energy comparisons among existing TTD methods and our method. HM keeps the backbone frozen and classifies based on fixed spatial energy. DAA collapses the original energy intensity onto a single hypersphere. Energy Ring compresses known and unknown classes onto different energy rings, yielding spatial-energy difference and retaining separability.

learned energy-ring structure enables query classification via energy-weighted confidence, while a high-perturbation priority rectification mechanism identifies unstable predictions and corrects them, further improving the accuracy and stability of the energy rings. Our method improves agreement performance by 3.33% compared to DAA (Liu et al., 2025). In summary, our contributions are:

(1) We propose Energy Ring, a TTD framework that stabilizes the inherently fragile energy gap by anchoring known and unknown classes on separate energy rings.
(2) We design a lightweight energy ring adapter and a perturbation-consistency module that jointly enlarge the energy gap, preserve known-class directions, and produce compact, stable clusters for unknown classes.
(3) Extensive experiments demonstrate the effectiveness of Energy Ring, showing clear improvements over prior TTD methods.

2 Related Work

Novel Category Discovery (NCD) (Han et al., 2020; Zhang et al., 2023; Zhong et al., 2021) tackles the problem of recognizing classes that never appear in the training labels. Standard NCD pipelines first learn a transferable representation from the labelled data, then cluster the unlabelled stream to reveal new groups. DTC (Han et al., 2019) exploits semantic anchors from related classes to reduce clustering ambiguity; GCD (Vaze et al., 2022) and PromptCCD (Cendra et al., 2024) replace the traditional k-means stage with contrastive learning prompting for purer prototypes. On-the-fly Category Discovery (OCD) (Du et al., 2023; Zheng et al., 2024) pushes NCD toward real-time operation by hashing incoming images and assigning them to the nearest Hamming bucket. OCD, however, is designed only for discovery whose backbone remains frozen, no fine-tuning on the new classes is performed. NCD methods treat discovery and adaptation as two separate stages.

Test-Time Training (TTT) (Liang et al., 2023; Sun et al., 2020) is a paradigm that adapts a pre-trained model to distribution shifts while the model is already deployed. Instead of relying on a frozen model, TTT injects self-supervised objectives and uses the test stream itself to update the backbone on the fly. Representative methods such as TTT++ (Liu et al., 2021b) and TTT-MAE (Gandelsman et al., 2022) show that this on-line alignment significantly improves accuracy under covariate shift (Lin et al., 2023). Yet the vast majority of TTT work assumes the model is only asked to preserve performance on the known classes, leaving it blind to categories that emerge after deployment.

Test-Time Discovery (TTD) unifies the two desiderata: retain accuracy on old classes while instantly detecting and learning new ones during inference. The first TTD implementation, pioneered by Lyu et al. (Lyu et al., 2025), keeps the network frozen and instead maintains an external memory built with Locality-Sensitive Hashing. By avoiding any gradient update, the method sidesteps catastrophic forgetting, but it also forfeits the possibility of refining the feature space when old and new classes overlap. DAA retains the adaptation spirit of TTT while embracing the discovery mandate of NCD: it updates only a small set of adapters in real time, thereby sharpening the decision boundary between known and emerging classes without erasing previous knowledge.

3 Methods

3.1 TTD Definition and Energy-based Discovery

As shown in Fig. 2, TTD aims to recognize known classes and discover unknown ones in a continuous unlabeled stream after model deployment. Formally, let $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the training set,

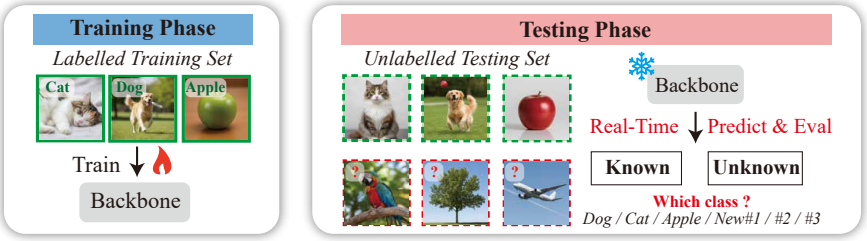


Figure 2: Task setting of TTD. In TTD, a model is pretrained on data containing only known classes. During deployment, the test data may include both known and unknown classes, requiring the model to predict known classes and discover novel ones.

where $x_i \in \mathcal{X}$ is input image and $y_i \in \mathcal{Y}_{\text{kn}}$ is its label from the known class set $\mathcal{Y}_{kn}$. During testing, the model encounters a test stream $\mathcal{D}_{\text{test}} = \mathcal{D}_{\text{kn}} \cup \mathcal{D}_{\text{un}}$, where $\mathcal{D}_{\text{un}}$ may contain unknown classes that were never seen during training. TTD requires model to satisfy two objectives simultaneously. First, *novel class discovery and learning*, where inputs $x \in \mathcal{D}_{\text{test}}^{\text{un}}$ must be identified as belonging to unseen classes and assigned new labels to distinguish them from known-class samples. Second, *unified classification*, where all inputs $x \in \mathcal{D}_{\text{test}}$ are accurately classified into either a known class $y \in \mathcal{Y}_{\text{kn}}$ or a novel class $y \in \mathcal{Y}_{\text{seen}}$, with $\mathcal{Y}_{\text{seen}}$ denoting the set of dynamically discovered novel classes.

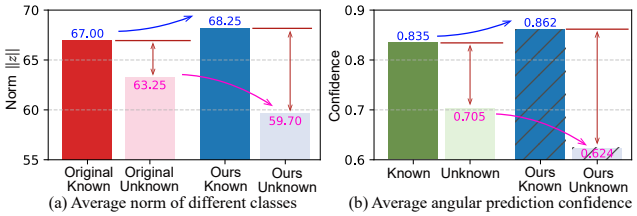


Figure 3: Comparison of norm and angular prediction for test samples. Our method further expands the differences of energy in both intensity and direction.

In this paper, we study TTD from an energy view. During testing, the model receives a mixture of known and unknown samples and must make decisions based on the learned representation $z = F(x)$. For known classes, the supervised objective explicitly optimizes the dependency between visual features and their corresponding class identifiers. In contrast, unknown classes are absent from training and their visual information is not explicitly organized or aligned with the learned class-discriminative space. This distinction often manifests in the magnitude of learned representations. Discriminative training tends to amplify feature responses along directions that are useful for separating known classes, producing more confident and higher-energy representations for in-distribution samples. This motivation is consistent with prior OOD detection and novelty discovery studies showing that known samples often exhibit stronger activation energy, while unknown samples tend to produce lower feature responses (Gu et al., 2025; Sodano et al., 2024; Park et al., 2023). Thus, we use the feature norm $\|z\|_2$ as a soft indicator of representation confidence, rather than as a strict separation criterion between known and unknown classes. This preliminary creates a natural difference in energy, reflected in both the norm and direction of features. As illustrated in Fig. 3, known classes usually appear with higher energy and unknown ones with lower energy. We also report the average feature norms before Energy Ring adaptation with different backbone in the Appendix. Simulated features have lower average norms and therefore provide an appropriate low-energy initialization cue. However, this separation is unstable as their purpose is to expose ERA to ambiguous boundary regions rather than to imitate the semantics of future categories. The norms of different classes overlap, the angular differences are weak, and as more novel classes arrive, these patterns may drift and cause confusion between old and new categories. These observations motivate us to design a method that enlarges and stabilizes the energy difference so that known and unknown classes remain distinguishable during test-time discovery. More validation can be seen in Appendix.

We therefore explicitly construct two non-overlapping energy rings, keeping known classes at higher energy ring and unknown classes at lower energy ring. We introduce a lightweight **Energy Ring Adapter (ERA)** and (i) construct energy rings through warm up after training the backbone model to achieve the effect of expanding the energy difference among known and unknown classes while maintaining the energy direction of the learned old classes; (ii) enhance the stability of the unknown classes energy ring through perturbations and prepare for rectification. During testing, we (iii) utilize the constructed energy rings for energy weighted prediction to improve the accuracy of discovery and classification and (iv) conduct test-time training to correct and optimize the energy rings based on the predicted results and high-perturbation priority rectification mechanism.

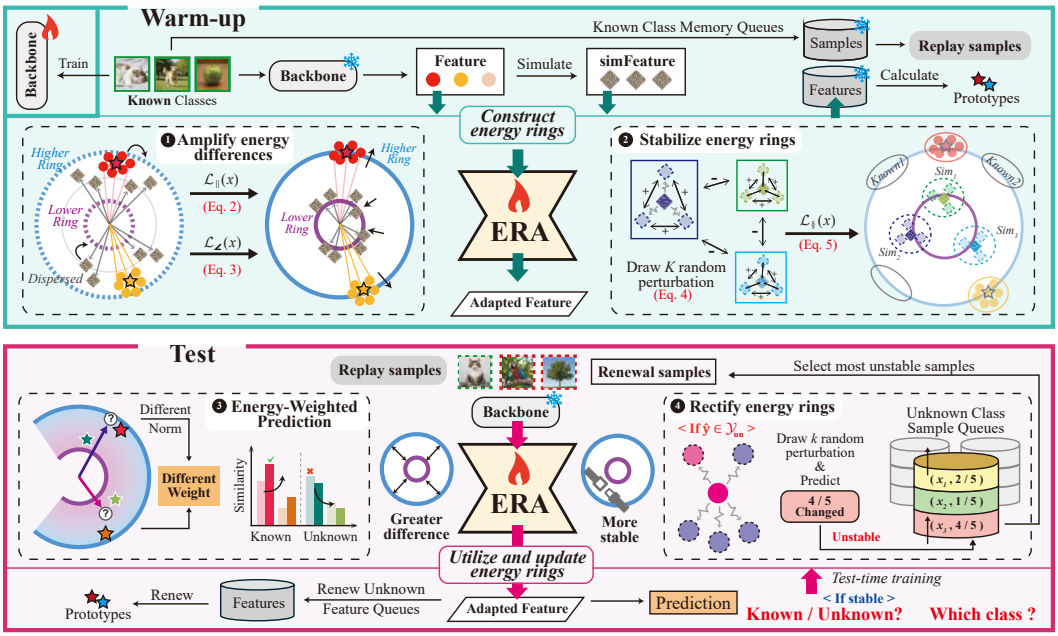


Figure 4: Method Schema. ERA projects features onto concentric energy rings: high-energy ring for known classes and low-energy ring for unknowns, together with perturbation-consistency during training. During testing, we conduct energy-weighted prediction, while selectively rectifying high-perturbation samples to refresh memory queues and update ERA to rectify energy rings.

3.2 Preconstruct Energy Rings through Warm-up

Following DAA (Liu et al., 2025), we also adopt a warm-up strategy to enlarge the gap between known and unknown classes before testing. Unlike DAA, which collapses all features onto a single hypersphere and removes energy differences, our goal is to preserve and further widen the natural gap observed earlier. Since known and unknown classes already show different energy scales, warm-up encourages them to lie on different energy levels, making later discovery more reliable. To achieve this, we introduce an adapter ERA inserted into the pre-trained backbone, which transforms the original feature $\mathbf{r}(x) = f(x)$ into an adapted representation $\mathbf{z} = \mathrm{ERA}(\mathbf{r}(x))$. Specifically, we simulate unknown samples during warm-up by mixing pairs of features with a coefficient λ drawn from a Beta distribution and adding Gaussian noise, forming a synthetic feature $\tilde{\mathbf{r}}(x) = \lambda \mathbf{r}(x) + (1-\lambda)\mathbf{r}(x') + \mathbf{n}$. These mixed features serve as proxies for unknown classes (Liu et al., 2025; Zhang et al., 2018) and help stabilize and enlarge the energy separation before test-time discovery. Below, we construct the energy rings during warm-up through **three aspects**: *1) constraining known and unknown classes onto different energy tracks (norm constraint), 2) maintaining separability for known classes (direction constraint), and 3) enhancing the stability of the energy rings under perturbation.*

First, we enforce a norm constraint that pushes known samples onto an outer ring and unknown samples onto an inner ring, creating a clear spatial gap between them:

$$\mathcal{L}_{||}(x) = \Big(\|\mathrm{ERA}(\mathbf{r}(x))\|_2 - (\mathbf{1}_{\mathrm{kn}}(x)\, r_{\mathrm{kn}} + \mathbf{1}_{\mathrm{un}}(x)\, r_{\mathrm{un}}) \Big)^2, \tag{1}$$

where $\mathbf{1}(x)$ is an indicator function, if x is a known class, $\mathbf{1}_{\mathrm{kn}} = 1, \mathbf{1}_{\mathrm{un}} = 0$ and vice versa. r_{kn} and r_{un} denote known and unknown classes energy ring.

Second, to preserve previously learned knowledge, ERA must not alter the semantic direction of known-class features. We maintain feature direction by maximizing cosine similarity between the original and adapted features:

$$\mathcal{L}_{\angle}(x) = 1 - \cos\big(\mathbf{r}(x),\, \mathrm{ERA}(\mathbf{r}(x))\big). \tag{2}$$

Third, even with spatial separation and direction preservation, the simulated unknown samples generated by mixup and noise remain random and unstable. To enhance the consistency of unknown classes, we introduce a perturbation-based constraint. For each adapted unknown feature $\tilde{\mathbf{z}} =$

ERA($\tilde{\mathbf{r}}(x)$), we create K norm-preserving perturbations by sampling random directions $\boldsymbol{\delta} \sim \mathcal{N}(0, \mathbf{I})$, projecting them onto the tangent space of $\tilde{\mathbf{z}}$, and scaling them to an angular radius ε. The perturbed features therefore remain on the same energy ring:

$$\mathbf{z}_b = \tilde{\mathbf{z}} + \varepsilon \frac{\boldsymbol{\delta} - (\boldsymbol{\delta}^\top \tilde{\mathbf{z}})\tilde{\mathbf{z}}/\|\tilde{\mathbf{z}}\|^2}{\left\|\boldsymbol{\delta} - (\boldsymbol{\delta}^\top \tilde{\mathbf{z}})\tilde{\mathbf{z}}/\|\tilde{\mathbf{z}}\|^2\right\|}. \tag{3}$$

We treat the K perturbations of the same feature as soft positives $\mathcal{P}_b = \{\mathbf{z}_b^{(i)} \mid i = 1, ..., K\}$ and perturbations from different features as soft negatives $\mathcal{N}_b = \{\mathbf{z}_{b'}^{(i)} \mid b' \neq b,\ i = 1, ..., K\}$. This encourages perturbations of the same unknown sample to stay close, while pushing those from different samples apart, improving the stability and separability of unknown classes without altering norm information:

$$\mathcal{L}_{\S} = \frac{1}{K} \sum_b \max\Big(0,\ \|\mathbf{z}_b^{(0)} - \bar{\mathcal{P}}_b\| - \|\mathbf{z}_b^{(0)} - \bar{\mathcal{N}}_b\|\Big). \tag{4}$$

The final warm-up loss combines all three terms:

$$\mathcal{L}_{\text{warmup}} = \mathcal{L}_{||} + \mathcal{L}_{\angle} + \mathcal{L}_{\S}. \tag{5}$$

3.3 Energy Ring Guided Prediction and Discovery

Warm-up establishes the initial energy rings for known and unknown classes. We now move to the testing phase and describe how these energy rings guide prediction and discovery. To provide energy information, similar to DAA, we maintain memory queues for both known and undiscovered classes. For a test image x, its ERA representation $\mathbf{z} = \text{ERA}(\mathbf{r}(x))$. We first compute an energy coefficient ρ based on its position between the two energy rings:

$$\rho = \frac{\|\mathbf{z}\|_2 - r_{\text{un}}}{r_{\text{kn}} - r_{\text{un}}} \in [0, 1]. \tag{6}$$

Let $\boldsymbol{\mu}_c$ be the prototype of class c. Features closer to the outer ring should receive slightly higher similarity toward known classes. Given a small amplitude $\alpha \in (0, 1)$, we adjust cosine similarities within a narrow range $[1-\alpha,\ 1+\alpha]$, so the energy prior nudges but does not dominate the prediction:

$$\tilde{s}_c = \begin{cases} \cos(\mathbf{z}, \boldsymbol{\mu}_c) \cdot [1 + \alpha(2\rho - 1)], & c \in \mathcal{Y}_{\text{kn}} \\ \cos(\mathbf{z}, \boldsymbol{\mu}_c) \cdot [1 - \alpha(2\rho - 1)], & c \in \mathcal{Y}_{\text{seen}}. \end{cases} \tag{7}$$

This weighting increases confidence for known classes on the outer ring and decreases confidence for unknown classes on the inner ring. We then use a confidence threshold γ to decide whether a test sample belongs to an existing class or a new one:

$$\hat{y} = \begin{cases} \arg\max_{c \in (\mathcal{Y}_{\text{kn}} \cup \mathcal{Y}_{\text{seen}})} (\tilde{s}_c), & \text{if } \max_c \tilde{s}_c > \gamma, \\ \text{new discovered class}, & \text{otherwise.} \end{cases} \tag{8}$$

To ensure the reliability of an unknown prediction, we adopt a stability recheck strategy. For a sample predicted as an unknown class, we reuse the perturbation mechanism used in warm-up. We draw k random directions $\boldsymbol{\delta}_k \sim \mathcal{N}(0, \mathbf{I})$, project them at the feature $\mathbf{z}_{\hat{y}}$, and form perturbed features $\mathbf{z}^{(k)} = \mathbf{z} + \boldsymbol{\delta}_k$. For each perturbed feature we recompute the similarity scores $\tilde{s}_c^{(k)}$ and obtain a new prediction. The stability of the original prediction is then measured by

$$S = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}\left[\arg\max_c \tilde{s}_c^{(i)} = \hat{y}\right] \tag{9}$$

where $\mathbf{1}(\cdot)$ is the indicator function. For each unknown-class sample, we store the tuple (x_c^i, S^i) in the queue. Samples with the lowest stability are selected for rectification and added to the test-time training, helping reduce prototype drift and improving the accuracy of class representations. And the model will treat this sample as the first instance of this new seen class. The image feature $\mathbf{z}$ will be used as the initial prototype for this class, denoted as $\boldsymbol{\mu}_{\hat{y}}$. Subsequent samples will be compared to all known and seen prototypes, including this new $\boldsymbol{\mu}_{\hat{y}}$.

3.4 TEST-TIME TRAINING TO STABILIZE ENERGY RINGS

During testing phase, ERA not only produces energy-aware features but also continues to learn from the streaming batch. The process requires real-time updating of the adapter, strengthening the dual ring spatial structure, while preventing the forgetting of old knowledge, and performing memory correction to selectively update the most unstable unknown samples to refine the class margin. For known classes, we continue to use lower energy ring loss and direction-preservation loss, which penalizes deviations between the original and adapted features and constrains known class features on the larger energy ring to ensure that the ERA does not alter the feature representations significantly. Contrary to known classes, stable unknown classes that have been evaluated are processed using larger energy ring loss and contrastive loss, which encourages ERA to better adapt to new knowledge of unknown classes. Specifically, given a test batch data $\mathcal{B}$, the test-time training with ERA can be represented by two kinds of loss functions. The first kind of loss is the self-supervised learning loss via the pseudo labels:

$$\mathcal{L}_{\text{kn}}(x) = \mathbb{E}_{\hat{y}\in\mathcal{Y}_{\text{kn}}}\Big[\mathcal{L}_{||}(x \mid \hat{y}) + \mathcal{L}_{\angle}(x \mid \hat{y})\Big],\ \mathcal{L}_{\text{un}}(x) = \mathbb{E}_{\substack{\hat{y}\in\mathcal{Y}_{\text{un}}\\ S>\kappa}}\Big[\mathcal{L}_{||}(x|\hat{y}, S) + \mathcal{L}_{\text{cont}}(x|\hat{y}, S)\Big], \tag{10}$$

where $\mathcal{L}_{\text{cont}}$ represents Contrastive Loss (Mao et al., 2023). The second kind represents the replay for known and the renewal for seen classes via retrain samples in memory. Specifically, for every discovered class p we retrieve the samples with the lowest S_p^i (highest instability) regularly, and these samples would be reprocessed, re-evaluated, and update ERA again:

$$\mathcal{L}_{\text{kn}}^{\text{replay}}(x) = \mathbb{E}\,\mathcal{L}_{||}(x) + \mathcal{L}_{\angle}(x), \quad \mathcal{L}_{\text{un}}^{\text{renewal}}(x) = \mathbb{E}\,\mathcal{L}_{||}(x) + \mathcal{L}_{\text{cont}}(x). \tag{11}$$

The overall testing phase loss is shown below:

$$\mathcal{L}_{\text{test}}(x) = \lambda_1 \cdot \mathbb{E}_{\hat{y}\in\mathcal{Y}_{\text{kn}}}\Big(\mathcal{L}_{\text{kn}}(x|\hat{y}) + \mathcal{L}_{\text{kn}}^{\text{replay}}(x)\Big) + \lambda_2 \cdot \mathbb{E}_{\hat{y}\in\mathcal{Y}_{\text{seen}}}\left(\mathcal{L}_{\text{un}}(x|\hat{y}) + \mathcal{L}_{\text{un}}^{\text{renewal}}(x|\hat{y})\right), \tag{12}$$

where λ_1 and λ_2 are hyperparameters that balance the contributions of each loss term.

4 EXPERIMENT

4.1 EXPERIMENTAL DETAILS

Datasets. We conduct our experiments based on four benchmark datasets. CIFAR100 (C100) (Krizhevsky et al., 2009) is a commonly used classic dataset. Tiny ImageNet (Le & Yang, 2015) is a subset of ImageNet (Deng et al., 2009). Caltech-UCSD Birds-200-2011 (CUB) (Wah et al., 2011) and FGVC-Aircraft (Maji et al., 2013) are complex fine-grained datasets. All these datasets are split into known and unknown classes (7:3). The model is trained on the known training set and tested on the mixture of known and unknown test sets. Because the dataset is used for discovery, we follow the names of four transformed datasets from HM (Lyu et al., 2025): CIFAR100D, CUB-200D, Tiny-ImageNetD, and AircraftD. More dataset details can be seen in Appendix.

Implementation details. In our implementation, we build our method on the prompt-based method L2P (Wang et al., 2022b), which employs a ViT-B/16 backbone (Khan et al., 2022) following the pretraining procedure of NCD and GMP work. We employed the contrastive loss of the GCD literature when we fine-tune the pretrained model on the known classes, using SGD optimizer and cosine decay learning rate scheduler with an initial learning rate of 0.1 and minimum learning rate of 0.0001, and weight decay of 0.00005. All input images are resized to 224 × 224 and augmented to match the pretrained backbone settings.

Evaluation metrics. Following HM, we use the following metrics. For unknown classes, we use two agreement metrics: (1) *True-label Agreement ratio* (**TA**). This metric measures the maximum proportion of samples from a given true class that are predicted as the same class; and (2) *Cluster Agreement ratio* (**CA**). This metric measures the maximum proportion of samples from a given predicted cluster that have the same true label. We also provide some clustering metrics that traditional NCD methods use, including Hungarian Cluster Accuracy (HCA) (Meilă, 2003), Adjusted Rand Index (ARI) (Rand, 1971), Normalized Mutual Information (NMI) (McDaid et al., 2011) and V-Measure (Rosenberg & Hirschberg, 2007). For known classes, we also employ two key metrics to comprehensively assess the model's performance: *Known Accuracy* (**KA**) and *Known Forgetting* (**KF**). KA measures the

Table 1: Comparisons on four datasets. Real-time evaluation reflects the accumulated performance across all test batches, while post evaluation reassesses all test samples after testing. **Bold** and underlining denote the best and second-best results.

	Method	Real-time Evaluation			Post Evaluation			
		KA↑	TA↑	CA↑	KA↑	TA↑	CA↑	KF↓
CIFAR100D	L2P (Wang et al., 2022b)	59.93±2.15	8.57±2.49	43.10±4.30	50.53±7.25	9.60±1.50	27.39±2.11	27.85±7.21
	DP (Wang et al., 2022a)	66.09±1.01	8.80±1.69	48.34±6.78	56.19±2.00	8.68±2.06	28.93±2.25	29.06±1.99
	GMP (Cendra et al., 2024)	72.77±1.20	7.37±0.88	42.26±4.54	67.21±2.53	13.04±2.59	27.10±1.77	17.69±2.53
	PHE (Zheng et al., 2024)	72.35±1.12	16.74±1.05	37.34±1.31	70.12±1.07	14.55±**1.01**	24.65±1.35	**2.12**±0.92
	HM (Lyu et al., 2025)	79.17±**0.13**	21.13±**0.62**	**56.37**±1.42	**80.73**±1.59	31.03±1.24	34.81±1.22	3.41±1.49
	DAA (Liu et al., 2025)	80.81±0.44	31.94±0.98	45.38±**0.88**	80.27±**0.60**	35.63±1.11	37.31±**1.20**	4.10±0.60
	Ours	**81.79**±0.42	**33.31**±0.77	47.42±1.04	80.40±0.66	**37.23**±1.18	**38.25**±1.34	4.00±**0.50**
CUB200D	L2P (Wang et al., 2022b)	46.22±1.53	9.01±0.87	55.37±7.79	31.97±3.35	4.75±0.73	24.48±1.65	42.29±3.14
	DP (Wang et al., 2022a)	53.69±1.24	43.68±2.20	45.68±5.88	63.37±3.27	45.94±0.91	7.92±1.10	5.85±3.11
	GMP (Cendra et al., 2024)	62.97±1.33	46.44±1.87	47.99±4.34	58.11±3.00	48.02±1.20	10.31±1.45	5.46±2.77
	PHE (Zheng et al., 2024)	57.29±1.23	32.68±1.66	57.22±2.75	54.77±1.75	27.92±**0.43**	21.36±1.36	**2.18**±0.71
	HM (Lyu et al., 2025)	66.20±0.55	**58.30**±2.37	43.33±6.10	64.42±0.65	**65.28**±1.78	37.25±4.90	4.07±0.47
	DAA (Liu et al., 2025)	**68.09±0.33**	54.49±1.03	**58.34±1.35**	66.26±**0.39**	64.13±2.23	44.65±**0.88**	3.60±**0.39**
	Ours	67.10±1.48	57.63±**0.47**	58.19±1.42	**67.38**±1.82	64.23±1.78	**45.53**±1.05	3.00±0.52
T-ImageNetD	L2P (Wang et al., 2022b)	46.25±1.41	7.79±2.92	53.55±6.47	29.50±3.77	10.38±2.42	23.28±**0.79**	47.97±3.77
	DP (Wang et al., 2022a)	46.51±**0.58**	6.41±0.93	58.27±6.10	28.53±3.33	9.63±2.10	26.80±1.38	47.57±3.32
	GMP (Cendra et al., 2024)	62.47±1.40	6.25±1.72	58.02±4.29	63.95±2.04	15.64±2.63	26.31±2.33	16.86±2.04
	PHE (Zheng et al., 2024)	63.28±1.29	13.54±1.20	70.33±3.80	58.39±**1.14**	12.80±**1.05**	19.42±1.45	1.88±**0.35**
	HM (Lyu et al., 2025)	75.31±1.31	16.04±0.76	**73.81**±2.67	74.94±2.20	16.23±1.24	37.43±1.30	**1.15**±2.18
	DAA (Liu et al., 2025)	76.38±0.82	24.10±0.80	71.70±2.02	**75.50**±1.96	23.17±1.35	36.41±1.12	2.39±1.56
	Ours	**76.72**±1.25	**25.23±0.71**	72.60±**1.43**	75.28±1.98	**24.33**±1.38	**37.93**±1.49	1.50±0.95
AircraftD	L2P (Wang et al., 2022b)	43.21±1.08	20.11±1.20	35.52±1.93	40.17±2.34	21.00±1.21	27.21±**1.33**	15.04±2.10
	DP (Wang et al., 2022a)	40.18±1.10	21.88±1.23	34.80±1.79	42.53±**1.46**	20.22±**0.96**	28.15±1.74	12.68±1.74
	GMP (Cendra et al., 2024)	41.26±1.04	21.12±0.67	35.10±2.50	32.56±1.75	21.25±1.02	26.34±1.48	22.65±2.25
	PHE (Zheng et al., 2024)	45.82±**1.02**	22.56±1.05	41.72±2.13	47.28±2.88	25.09±1.76	24.38±1.67	7.93±1.13
	HM (Lyu et al., 2025)	50.35±1.08	28.01±0.85	40.23±2.22	**49.77**±1.94	**29.80**±1.25	28.74±1.53	5.44±1.56
	DAA (Liu et al., 2025)	49.37±1.46	27.79±**0.31**	40.51±**1.05**	48.87±1.73	28.37±1.69	28.99±1.69	5.61±1.04
	Ours	**50.88**±1.42	**28.49**±0.76	**41.90**±1.91	49.42±1.66	29.79±1.50	**30.79**±1.55	**5.20±0.96**

traditional classification accuracy of the model on known classes while KF quantifies the degree of performance degradation on known classes over time. For more details of the metrics, see Appendix.

4.2 Major Comparisons

In this paper, we first compare our methods with naive thresholding-based training-free methods. When exceeding the threshold, the naive method will be considered to have discovered a new class. And we compare with some training-required methods including L2P (Wang et al., 2022b), DP (Wang et al., 2022a), and GMP (Cendra et al., 2024) as these methods update prompts like TTA and CL methods. We also compare with the recent PHE (Zheng et al., 2024) method for OCD and HM (Lyu et al., 2025), DAA (Liu et al., 2025) for TTD.

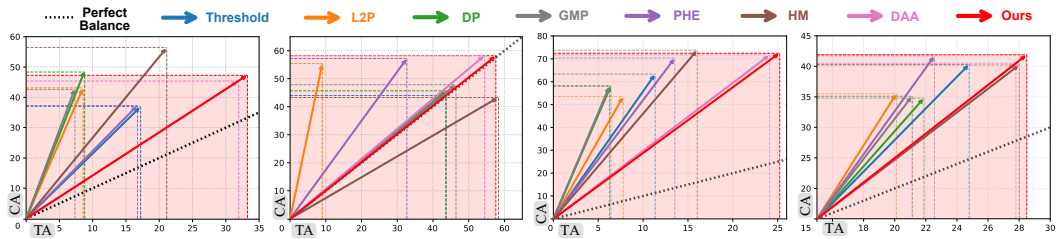
Comparisons on TA & CA. The analysis of our experimental results, as illustrated in Table 1, offers key insights into the performance of various methods in the context of Test-Time Discovery (TTD). Training-based methods, which update model parameters upon encountering new classes, tend to suffer from performance degradation. This decline is largely due to their immediate adaptation to novel classes, resulting in lower TA and CA, as well as an increased risk of catastrophic forgetting. On the other hand, training-free methods, such as HM, avoid parameter updates but has restricted capacity to refine its representations in response to novel data. While DAA incorporates a lightweight module that can be updated which only focuses on energy direction and the amplification of inter-class differences remains limited. Our proposed method strikes a more balanced effectiveness among these challenges, achieving more consistent performance across all four datasets. These results demonstrate that our approach enhances the network's adaptability in discovering novel classes without compromising the retention of previously acquired knowledge.

Comparison of compatibility metrics. TA and CA directly reflect online discovery quality and class consistency as the primary metrics, while the four clustering metrics mainly serve compatibility with post-hoc NCD evaluation. The results in Table 2 demonstrate that our method also has better clustering performance on post evaluation.

Balance of TA & CA. TA and CA are two complementary indicators, and a high individual indicator does not necessarily indicate a good overall classification performance. Therefore, higher values of both metrics, along with greater balance between them, indicate better classification performance. We compared the balance in Fig. 5 and found that some methods have higher values but deviate far from the Perfect Balance line. And our method is more balanced and effective (larger red areas) on different datasets.

Table 2: Comparisons of clustering metrics of post evaluation on four datasets.

Method	CIFAR100D				CUB200D				T-ImageNetD				AircraftD			
	HCA	ARI	NMI	VM	HCA	ARI	NMI	VM	HCA	ARI	NMI	VM	HCA	ARI	NMI	VM
HM (Lyu et al., 2025)	63.69	48.35	73.08	73.08	50.46	27.81	70.25	70.25	56.64	40.35	72.70	72.70	33.30	21.33	46.46	46.46
DAA (Liu et al., 2025)	65.62	53.96	75.20	75.21	58.19	32.43	75.29	75.29	58.91	42.50	73.53	73.53	32.08	21.20	46.44	46.44
Ours	66.38	53.54	75.36	75.36	58.34	33.70	75.53	75.53	58.25	43.26	73.36	73.36	33.21	22.13	47.02	47.02



(a) CIFAR100D (b) CUB200D (c) T-ImageNetD (d) AircraftD

Figure 5: Comparison on balance of Real-time TA and CA. Perfect Balance represents the most balanced TA and CA (TA = CA).

4.3 Analysis on Energy Ring

Comparison on norm and confidence. Fig. 3 shows that there is indeed a certain norm difference between the features of known and unknown classes obtained by the backbone network, as stated in (Sodano et al., 2024; Lyu et al., 2025). The average norm of unknown classes is slightly lower than that of known classes. Our method trains simulated classes as unknown classes and significantly reduce the norm of the simulation classes, which successfully simulate unknown classes to a certain extent, thus reducing the norm of the unknown classes. And our strategy makes their average confidence differences larger, which means that unknown classes are easier to be discovered.

Table 3: Different energy weight during testing.

Energy weight	Real-time Eval			Post Eval			
α	TA	CA	T+C	TA	CA	T+C	KF
0 (No norm)	31.81	44.59	76.40	35.25	36.99	72.24	4.30
0.01	32.56	45.58	78.14	36.42	37.43	73.85	4.07
0.05	33.52	47.66	81.18	37.74	38.92	76.66	3.65
0.10	29.25	48.23	77.48	34.92	40.95	75.87	3.83
0.50	25.70	49.48	75.18	27.25	36.51	63.76	8.49
(Norm-based)	12.87	27.16	40.03	20.92	24.79	45.71	12.90

Table 4: Ablation study. T: Test-time Training, W: Energy-weighted prediction, P: Perturbation rectification, T+C: TA+CA.

T	W	P	Real-time Eval			Post Eval			
			TA	CA	T+C	TA	CA	T+C	KF
			30.81	42.13	72.94	34.91	38.53	73.44	2.27
✓			24.68	53.74	78.42	24.89	42.68	67.57	6.84
	✓		29.61	45.54	75.15	32.44	36.45	68.89	3.05
✓	✓		31.60	44.99	76.59	34.52	36.52	71.04	4.22
✓	✓	✓	33.52	47.66	81.18	37.74	38.92	76.66	3.65

Energy weight. We exploit the spatial variation in feature responses to re-weight categorical similarity, using a norm-based modulation that remains deliberately light (Sec.3.3). The feature-to-category similarities already lie in [0, 1] and the overlap zone clusters around 0.7. Consequently, as Table 3 shows, only a small modulation factor is required, once the exponent becomes large, the relative reliabilities are overturned and the norm effectively seizes control of the decision; in the limiting case of a purely norm-governed weight, the accuracy collapses.

Ablation study. Table 4 summarizes the ablation results. Test-time Training (T) alone lifts CA from 42.13 to 53.74, but results in larger KF and lowers post-TA, revealing instability. Energy-weighted prediction (W) without T already outperforms the fixed model, pushing real-time TA+CA from 72.94 to 75.15, confirming the effectiveness of energy-weight assistance. Injecting perturbation consistency rectification (P) into the full pipeline helps rectify unstable samples, attaining the best overall trade-off (TA+CA 81.18 real-time and 76.66 post). These results demonstrate that W and P can effectively enable positive test-time update without catastrophic forgetting.

Different Perturbation. We mainly enhance the stability among unknown classes through perturbations, and during testing, we make corrections based on the results of perturbations. In Fig. 6, insufficient perturbations makes the judgment of stability reckless and coincidental changes or no changes can be considered unstable or very stable, so the effect cannot be effectively improved, while excessive perturbations does not significantly improve the effect and introduces greater time cost.

Energy rings during warm-up. We treat the energy ring r_{kn} and r_{un} as a coupled hyper-parameter pair that determines the spatial energy layout of feature. Fig. 7 shows when r_{un} is too small the inner ring is pushed close to the origin with a large gap between r_{kn}, making classification difficult and yielding unstable clusters. The experiment of Fig 3 finds that the norm of backbone features for known classes centers around 70. Therefore, we fix the known class feature norm at 70 and handle unknown classes for better results.

Post visualization using T-SNE. In Fig. 8, we employ t-SNE (Van der Maaten & Hinton, 2008) to

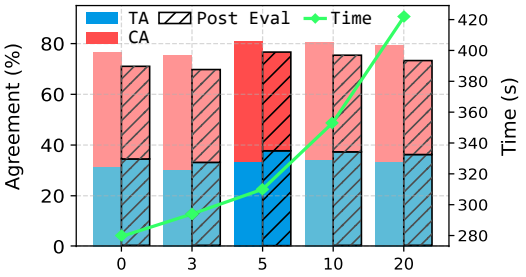


Figure 6: Different perturbations k during testing.

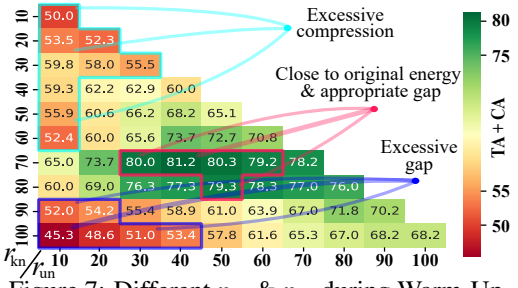
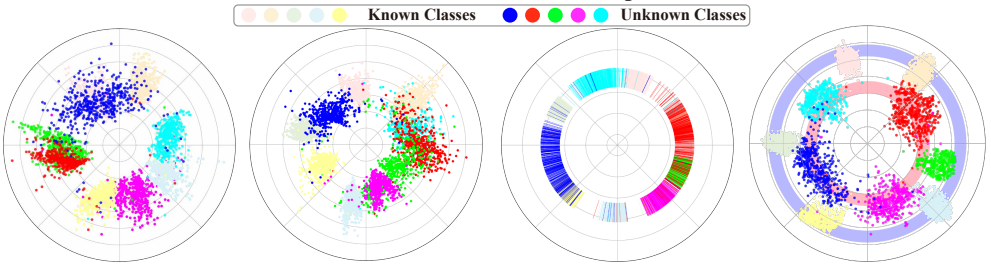


Figure 7: Different r_{kn} & r_{un} during Warm-Up.

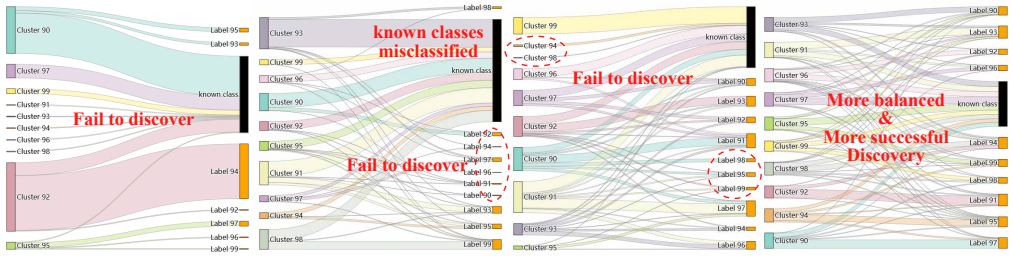
visualize the distribution of test samples. Methods that directly update the base model parameters often risk disrupting the model's existing structure, leading to destructive confusion between classes. In contrast, HM methods cannot update the model, leaving the feature distribution static. Meanwhile, DAA dismisses the spatial difference of feature, limiting its angular amplification effect. In contrast, our method strikes a balance by not only distinguishing features based on angles but also accounting for the energy intensity between the inner and outer energy rings, and by learning updates that separate features across different classes and maintain clearer spatial boundaries than other methods.



(a) L2P (b) HM (c) DAA (d) Ours

Figure 8: T-SNE visualization in angular space on CIFAR100D with 5 known and 5 unknown classes.

Comparison of unknown class label matching. Fig. 9 shows the matching between predictions and ground truth, where the base-model updating method easily leads to chaotic predictions and misclassification. In the fixed-energy method like HM, lots of old samples are classified into unknown clusters. DAA has improved its ability to discover new classes, but there are still some categories that fail to be discovered. Our method can discover a larger number of new classes and make the matching more balanced while classifying fewer known classes as unknown.



(a) L2P (b) HM (c) DAA (d) Ours

Figure 9: Matching comparison between unknown class predictions and ground truth across methods on CIFAR-100D (90+10). Black rectangles indicate old class samples misclassified as unknowns.

5 Conclusion

In this paper, we introduced the Energy Ring framework to address the challenging Test-Time Discovery task. Our method enables instant recognition of unknown classes while safeguarding the accuracy of previously learned ones. The lightweight energy ring adapter compresses feature onto concentric energy rings, preserves known-class directions, enlarges the energy gap and produces compact, stable clusters for unknown classes. Comprehensive experiments on four datasets demonstrate that Energy Ring outperforms in both adaptability and stability. Although energy ring adds marginal computation during warm-up and test-time training, the overhead is negligible compared with the gains in discovery performance. However, the current energy cycle still needs to rely on pre-set energy tracks. Future work will explore more efficient adapter designs, noise-robust update rules, and extensions to multi-modal and continual learning scenarios. And we will explore how to dynamically set energy tracks to achieve wider applicability in the open world.

AI USE STATEMENT

We used generative AI tools only for grammar correction and language polishing during manuscript preparation. Generative AI tools were not used for research ideation, literature review, data collection or generation, code development, model training, experimental design or execution, or analysis of the results. All models evaluated in this work were trained locally without calling generative AI services or external AI APIs. The authors reviewed every AI-assisted edit to ensure technical accuracy and preservation of the intended meaning. We take responsibility for the final content of this work, including all text, claims, and artifacts.

REPRODUCIBILITY STATEMENT

The supplementary materials include the source code for our experiments. The method and experimental protocol are described in Sections 3 and 4 with different hyperparameter analyses. Table 1 reports results from repeated runs as means with standard deviations, and Appendix M provides a 10-run paired analysis.

REFERENCES

Malik Boudiaf, Romain Mueller, Ismail Ben Ayed, and Luca Bertinetto. Parameter-free online test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Chentao Cao, Zhun Zhong, Zhanke Zhou, Tongliang Liu, Yang Liu, Kun Zhang, and Bo Han. Noisy test-time adaptation in vision-language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Fernando Julio Cendra, Bingchen Zhao, and Kai Han. Promptccd: Learning gaussian mixture prompt pool for continual category discovery. In *European Conference on Computer Vision*, 2024.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2009.

Ruoyi Du, Dongliang Chang, Kongming Liang, Timothy Hospedales, Yi-Zhe Song, and Zhanyu Ma. On-the-fly category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Yossi Gandelsman, Yu Sun, Xinlei Chen, and Alexei Efros. Test-time training with masked autoencoders. *Advances in Neural Information Processing Systems*, 2022.

Jiawei Gu, Ziyue Qiao, and Zechao Li. Activation shape matters: Ood detection with norm-entropy fusion. In *Proceedings of ACM International Conference on Multimedia*, 2025.

Kai Han, Andrea Vedaldi, and Andrew Zisserman. Learning to discover novel visual categories via deep transfer clustering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019.

Kai Han, Sylvestre-Alvise Rebuffi, Sébastien Ehrhardt, Andrea Vedaldi, and Andrew Zisserman. Automatically discovering and learning new visual categories with ranking statistics. In *International Conference on Learning Representations*, 2020.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 2022.

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.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015.

Jian Liang, R. He, and Tien-Ping Tan. A comprehensive survey on test-time adaptation under distribution shifts. *International Journal of Computer Vision*, 2023.

Huiwei Lin, Baoquan Zhang, Shanshan Feng, Xutao Li, and Yunming Ye. Pcr: Proxy-based contrastive replay for online class-incremental continual learning. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Jiashuo Liu, Zheyan Shen, Yue He, Xingxuan Zhang, Renzhe Xu, Han Yu, and Peng Cui. Towards out-of-distribution generalization: A survey. *arXiv preprint arXiv:2108.13624*, 2021a.

Tianle Liu, Fan Lyu, Chenggong Ni, Zhang Zhang, Fuyuan Hu, and Liang Wang. Daa: Amplifying unknown discrepancy for test-time discovery. *Advances in Neural Information Processing Systems*, 2025.

Yuejiang Liu, Parth Kothari, Bastien Van Delft, Baptiste Bellot-Gurlet, Taylor Mordan, and Alexandre Alahi. Ttt++: When does self-supervised test-time training fail or thrive? *Advances in Neural Information Processing Systems*, 2021b.

Fan Lyu, Tianle Liu, Zhang Zhang, Fuyuan Hu, and Liang Wang. Test-time discovery via hashing memory. *arXiv preprint arXiv:2503.10699*, 2025.

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

Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In *International Conference on Machine Learning*, 2023.

Aaron F McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms. *arXiv preprint arXiv:1110.2515*, 2011.

Marina Meilă. Comparing clusterings by the variation of information. In *Proceedings of Learning Theory and Kernel Machines*, 2003.

Jaewoo Park, Jacky Chen Long Chai, Jaeho Yoon, and Andrew Beng Jin Teoh. Understanding the feature norm for out-of-distribution detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Nan Pu, Zhun Zhong, and Nicu Sebe. Dynamic conceptional contrastive learning for generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 1971.

Andrew Rosenberg and Julia Hirschberg. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2007.

Matteo Sodano, Federico Magistri, Lucas Nunes, Jens Behley, and Cyrill Stachniss. Open-world semantic segmentation including class similarity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International Conference on Machine Learning*, 2020.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Generalized category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Technical report, California Institute of Technology, 2011.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, 2022a.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022b.

Yanan Wu, Zhixiang Chi, Yang Wang, and Songhe Feng. Metagcd: Learning to continually learn in generalized category discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, pp. 5635–5662, 2024.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. 2018.

Sheng Zhang, Salman Khan, Zhiqiang Shen, Muzammal Naseer, Guangyi Chen, and Fahad Shahbaz Khan. Promptcal: Contrastive affinity learning via auxiliary prompts for generalized novel category discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

Haiyang Zheng, Nan Pu, Wenjing Li, Nicu Sebe, and Zhun Zhong. Prototypical hash encoding for on-the-fly fine-grained category discovery. *Advances in Neural Information Processing Systems*, 2024.

Zhun Zhong, Enrico Fini, Subhankar Roy, Zhiming Luo, Elisa Ricci, and Nicu Sebe. Neighborhood contrastive learning for novel class discovery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

Lei Zhu, Yanyu Xu, Yong Liu, Rick Siow Mong Goh, and Xinxing Xu. Ad 2 mix: Adversarial and adaptive mixup for unsupervised domain adaptation. In *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025.

APPENDIX

A COMPARISON OF TTD TASK AND OTHER CATEGORY DISCOVERY TASK

We compare TTD with related settings. Out-of-Distribution (OOD) (Yang et al., 2024; Liu et al., 2021a) detection neither discovers novel classes nor adapts during inference. TTA (Boudiaf et al., 2022) handles distribution shifts via self-supervised learning but assumes all test samples belong to known classes. NCD (Han et al., 2019) and GCD (Vaze et al., 2022) both aim to identify unknown classes at test time under an offline inference paradigm through clustering the entire test set. OCD (Du et al., 2023) resembles TTD in identifying known and unknown classes, but does not incorporate test-time learning from novel classes.

Test-Time Discovery (TTD) is a challenging task that focuses on class shifts rather than domain shifts during test time. It requires the model to not only discover new classes but also classify them accurately while maintaining robust performance on previously seen classes. This dual requirement is particularly demanding due to several intrinsic complexities: the intricate nature of class discovery, the scarcity of labeled data for new classes, and the often ambiguous boundaries between classes. The key challenges in TTD can be summarized as follows:

(1) Distinguishing between the discovery of new classes and the identification of already discovered ones.
(2) Learning and adapting to new classes with limited sample sizes during testing phase.
(3) Avoiding catastrophic forgetting, where the process of learning new classes can inadvertently degrade the model's performance on previously learned classes.

Table 5: Comparison of evaluation protocols across related tasks. K: known classes, U: unknown classes.

Task	Train	Test	Discovery	Evaluation
NCD	K	U	Post-hoc	Post-hoc
GCD	K	K + U	Post-hoc	Post-hoc
CGCD	K	K + U_1 + U_2 + $\cdots$	Post-hoc	Post-hoc
OCD	K	K + U	Real-time	Post-hoc
TTA	K	Shifted K	N/A	Real-time
TTD	K	K + U	Real-time	Real-time

Setting Comparison. In Table 5, we compare TTD with related settings, focusing on their evaluation protocols. In contrast, TTD requires models to classify known classes and initiate novel ones under sequential test-time constraints, while being evaluated in real time rather than post-hoc. *Existing NCD, GCD, CGCD, and OCD rely on post-hoc reclustering where the data for discovery and evaluation are inconsistent, which obscures the difficulties of real-time deployment.* TTD addresses this gap by jointly measuring discovery and classification during testing, thereby motivating our hash-based memory method.

Challenges. TTD requires real-time evaluation, making it more demanding than conventional NCD. The main challenges are threefold: *(1) distinguishing novel class discovery from the identification of already discovered ones*, *(2) coping with the scarcity of samples when a novel class first appears*, and *(3) maintaining performance on known classes while incorporating new ones to avoid forgetting (Kirkpatrick et al., 2017).*

B DATASET DETAILS

We conduct our experiments using three widely recognized benchmark datasets: CIFAR100 (C100) (Krizhevsky et al., 2009), Caltech-UCSD Birds-200-2011 (CUB) (Wah et al., 2011), Tiny ImageNet (Le & Yang, 2015) and FGVC-Aircraft (Maji et al., 2013). Each of these datasets is systematically partitioned into known and unknown classes. The model undergoes training on the known training set and is subsequently evaluated on a mixed set containing both known and unknown

classes. Since the primary objective of these datasets is to facilitate new class discovery, we denote the transformed versions as CIFAR100D, CUB-200D, Tiny-ImageNetD, and AircraftD to reflect this adaptation.

Table 6: Statistic of the used datasets.

		CIFAR100D			**CUB-200D**		
Dataset	Labeled	Known	Unknown	No. of samples	Known	Unknown	No. of samples
TrainSet	✓	70	0	35000	140	0	4195
TestSet		70	30	10000	140	60	5794
		Tiny-ImagenetD			**AircraftD**		
Dataset	Labeled	Known	Unknown	No. of samples	Known	Unknown	No. of samples
TrainSet	✓	140	0	70000	70	0	2331
TestSet		140	60	10000	70	30	3333

The dataset partitioning follows the scheme outlined in Table 6. Specifically, during the training phase, we divide the training set into known and unknown classes based on their class index order. For instance, in CIFAR100, the first 70 classes are designated as known, while the remaining 30 classes are treated as unknown. The supervised training process is then conducted using only the known classes within the labeled training set. More precisely, CIFAR100D consists of classes 0–69 (70 known classes in total), CUB-200D includes classes 0–139 (140 known classes in total), Tiny-ImageNetD comprises classes 0–139 (140 known classes in total), and AircraftD includes classes 0-69 (70 known classes in total), all of which are utilized for training.

During the test phase, the model is evaluated on the entire unlabeled test set, which includes samples from all categories, enabling new class discovery and classification. While the category labels remain structured according to the original known-unknown splits (e.g., 70+30 for CIFAR100D and AircraftD and 140+60 for CUB-200D and Tiny-ImageNetD), these labels are only used for metric evaluation and are not provided to the model during inference. This setup ensures a realistic scenario for open-world learning, where the model must autonomously identify and categorize previously unseen classes.

C Metric Definition

The evaluation process is structured into two distinct parts: one focusing on known classes and the other on unknown classes. To ensure a comprehensive assessment, we follow HM (Lyu et al., 2025) and employ both real-time evaluation and post-evaluation strategies. For test-time evaluation, real-time performance is a critical factor. Thus, we compute and report real-time scores for all evaluation metrics as the model processes each test sample. This approach provides immediate insights into the model's performance and enables dynamic tracking of classification accuracy and discovery efficiency. Alongside these real-time scores, we present the final accumulated values, which represent the overall average performance across the entire test set. In addition, recognizing that traditional novel class discovery (NCD) methods typically rely on post-evaluation, we also incorporate this approach for comparative analysis. In post-evaluation, all test samples are revalidated collectively after the entire test phase is complete. This post-hoc evaluation allows for a more refined assessment by leveraging the full distribution of test samples, potentially improving class assignment and clustering accuracy. By providing both real-time and post-evaluation scores, we ensure a thorough and balanced evaluation of the model's effectiveness in handling both known and unknown classes.

C.1 Metrics for known classes

For the evaluation of known classes, we employ two key metrics to comprehensively assess the model's performance: Known Accuracy (KA) and Known Forgetting (KF).

(1) *Known Accuracy* (**KA**). KA measures the traditional classification accuracy of the model on known classes, reflecting its ability to correctly recognize and classify samples that were part of the training set. This metric serves as a standard benchmark for evaluating the retention of previously learned knowledge:

$$\text{KA} = \mathbb{E}_{c \in \mathcal{Y}_{\text{known}}} \frac{1}{|\mathcal{D}_c^{\text{test}}|} \sum_{x \in \mathcal{D}_c^{\text{test}}} \mathbf{1}(\hat{y}(x) = c), \tag{13}$$

where $\mathcal{Y}_{\text{known}}$ is set of predefined known classes, $\mathcal{D}_c^{\text{test}}$ is test samples with ground-truth class c, $\hat{y}(x)$ is the predicted label for sample x, $\mathbf{1}(\cdot)$ is the indicator function (1 if prediction matches true class c, 0 otherwise)

(2) *Known Forgetting* (**KF**). KF, on the other hand, quantifies the degree of performance degradation on known classes over time. It captures the extent to which the model forgets previously learned information as it encounters new data, particularly when adapting to novel classes. A lower KF score indicates better knowledge retention, while a higher score suggests significant forgetting.

$$\text{KF} = \text{KA}_{\text{pre}} - \text{KA}_{\text{post}}, \tag{14}$$

where KA_{post} and KA_{pre} are the KA computed on all test data with known classes, before and after Testing Phase.

C.2 Metrics for unknown classes

For unknown classes, since the predicted label space $\mathcal{Y}_{\text{seen}}^{\text{GT}}$ does not match the cluster label space $\mathcal{Y}_{\text{seen}}$, we propose agreement metrics to assess effectiveness. In the test set $\mathcal{D}^{\text{test}}$, a sample x has a true label $y \in \mathcal{Y}_{\text{seen}}^{\text{GT}}$ and a predicted cluster label $\hat{y}(x) \in \mathcal{Y}_{\text{seen}}$. We define the subset of $\mathcal{D}^{\text{test}}$ with true label c as $\mathcal{D}_c^{\text{test}}$, and the cluster with predicted label p as $\mathcal{C}_p^{\text{test}}$.

(1) *True-label Agreement ratio* (**TA**). This metric measures the maximum proportion of samples from a given true class that are predicted as the same class:

$$\text{TA} = \mathbb{E}_{c \in \mathcal{Y}_{\text{seen}}^{\text{GT}}} \frac{1}{|\mathcal{D}_c^{\text{test}}|} \max_{p \in \mathcal{Y}_{\text{seen}}} \left(\sum\nolimits_{x \in \mathcal{D}_c^{\text{test}}} \mathbf{1}[\hat{y}(x) = p] \right), \tag{15}$$

where $\mathbf{1}(\cdot)$ is the indicator function (1 if true, 0 otherwise).

(2) *Cluster Agreement ratio* (**CA**). This metric measures the maximum proportion of samples from a given predicted cluster that are with the same true label:

$$\text{CA} = \mathbb{E}_{p \in \mathcal{Y}_{\text{seen}}} \frac{1}{|\mathcal{C}_p^{\text{test}}|} \max_{c \in \mathcal{Y}_{\text{seen}}^{\text{GT}}} \left(\sum\nolimits_{(x,y) \in \mathcal{C}_p^{\text{test}}} \mathbf{1}(y = c) \right). \tag{16}$$

C.3 Clustering metrics

Traditional novel class discovery (NCD) methods typically rely on post-cluster evaluation, where the quality of the discovered clusters is assessed after the entire test set has been processed. To ensure a comprehensive comparison with existing approaches, we also report several widely used clustering evaluation metrics, including Hungarian Cluster Accuracy (HCA) (Meilă, 2003), Adjusted Rand Index (ARI) (Rand, 1971), Normalized Mutual Information (NMI) (McDaid et al., 2011), and V-Measure (Rosenberg & Hirschberg, 2007). Note that these metrics are only evaluated after TTD, say post evaluation.

(1) *Hungarian Cluster Accuracy* (**HCA**). This metric measures the clustering accuracy by computing an optimal one-to-one mapping between predicted clusters and ground-truth labels using the Hungarian algorithm. It provides an intuitive evaluation of how well the discovered clusters align with the actual class distributions. HCA can be computed as

$$\text{HCA} = \mathbb{E}_{(x,y) \in \mathcal{D}^{\text{test}}}(y = \text{map}(\hat{y}(x))), \tag{17}$$

where $\text{map}(\cdot)$is the optimal mapping from clustering to true labels obtained based on the Hungarian algorithm

(2) *Adjusted Rand Index* (**ARI**). ARI quantifies the similarity between the predicted clustering assignments and the ground-truth labels while adjusting for chance. It accounts for both correct pairwise clustering and misclustered pairs, offering a robust measure of clustering consistency.

$$\text{ARI} = \frac{\text{RI} - \mathbb{E}[\text{RI}]}{\max(\text{RI}) - \mathbb{E}[\text{RI}]}, \tag{18}$$

where Rand Index $(\text{RI}) = \frac{a+b}{C_n^2}$, a is the logarithm of samples of the same class assigned to the same cluster, and b is the logarithm of samples of different classes assigned to different clusters. n is the total number of samples, combination $C_n^2 = \frac{n(n-1)}{2}$ and $\mathbb{E}[\text{RI}]$ is the expected value of RI.

(3) *Normalized Mutual Information* (**NMI**). NMI assesses the mutual dependence between predicted and true labels by measuring the shared information between the two distributions. A higher NMI value indicates better alignment between the discovered clusters and the actual categories. The value interval of NMI is [0,1], and a larger value indicates a higher degree of information sharing between the clustering results and the real labels.

$$\text{NMI}(\mathcal{U}, \mathcal{V}) = \frac{2 \cdot I(\mathcal{U}; \mathcal{V})}{H(\mathcal{U}) + H(\mathcal{V})}, \tag{19}$$

where $\mathcal{U}$ is collection of true labels and $\mathcal{V}$ is collection of predictions. $I(\mathcal{U}; \mathcal{V})$ is Mutual Information where $I(\mathcal{U}; \mathcal{V}) = H(\mathcal{U}) - H(\mathcal{U}|\mathcal{V})$. $H(\mathcal{U})$ is the entropy of true label, $H(\mathcal{U}) = -\sum_{c=1}^{C} p(c) \log p(c)$, and $H(\mathcal{V})$ is the entropy of prediction, $H(\mathcal{V}) = -\sum_{k=1}^{K} p(k) \log p(k)$.

(4) *V-Measure* (**VM**). The VM is taken in the interval [0,1], which simultaneously constrains the purity and coverage of the clusters through the harmonic mean. Both VM and NMI are symmetric metrics that support the comparison of clusters and categories at different scales.

$$\text{V-Measure} = \frac{2 \cdot h \cdot c}{h + c}, \tag{20}$$

where homogeneity $h = 1 - \frac{H(\mathcal{U}|\mathcal{V})}{H(\mathcal{U})}$, and completeness $c = 1 - \frac{H(\mathcal{V}|\mathcal{U})}{H(\mathcal{V})}$. $H(\mathcal{U}|\mathcal{V}) = -\sum_{k=1}^{K}\sum_{t=1}^{T} p(k,t) \log \frac{p(k,t)}{p(k)}$ and $H(\mathcal{V}|\mathcal{U}) = -\sum_{t=1}^{T}\sum_{k=1}^{K} p(t,k) \log \frac{p(t,k)}{p(t)}$, where $p(t) = \frac{N_t}{N}$ is the sample proportion of class t, $p(k) = \frac{N_k}{N}$ is the sample proportion of cluster k, and $p(t,k) = \frac{count_k(t)}{N}$ is the joint distribution probability.

D Additional Validation of the Energy Prior

D.1 Energy intensity and Energy Ring geometry

We use *energy intensity* to denote the scalar feature norm $\|\mathbf{z}\|_2$. It provides a radial coordinate and is treated as a soft confidence cue, consistent with prior analyses of feature norms and out-of-distribution behavior (Park et al., 2023; Gu et al., 2025). The term *Energy Ring* refers to the complete geometry rather than to the norm alone: radial constraints create separate layers, angular preservation anchors known-class semantics, perturbation consistency stabilizes unknown-like neighborhoods, and energy-weighted routing with selective rectification uses this structure throughout the online stream. This distinction is empirically important.

The energy prior is motivated by an average tendency rather than a universal sample-level ordering. A frozen discriminative representation often assigns larger norms to known samples than to unknown samples because the former are aligned with directions optimized during supervised training. The two sample-level distributions can nevertheless overlap, so a fixed norm threshold is insufficient for either class recognition or novel-class discovery. Energy Ring uses the radial signal only as a complement to angular semantics and local stability.

D.2 Generalization across datasets and backbones

Table 7 reports the average feature norms before Energy Ring adaptation. The known–unknown ordering holds across all listed same-domain combinations of ViT-B/16, VGG16, and ResNet18. Simulated features have even lower average norms and therefore provide an appropriate low-energy

Table 7: Average feature norms before Energy Ring adaptation across datasets and backbones. The asterisk marks corrupted-domain settings, which are outside the standard same-domain TTD protocol.

Setting	Backbone	Known	Unknown	Simulated
CIFAR100D	ViT-B/16	67.00	63.25	52.64
CUB200D	ViT-B/16	71.73	68.21	58.27
Tiny-ImageNetD	ViT-B/16	72.09	66.52	56.48
AircraftD	ViT-B/16	66.13	64.01	59.02
CIFAR100D	VGG16	92.17	82.02	73.91
CIFAR100D	ResNet18	28.51	17.44	15.72
CUB200D	VGG16	85.39	78.18	74.17
CUB200D	ResNet18	32.58	27.95	25.22
CIFAR10-C*	VGG16	83.52	82.12	81.06
CIFAR10-C*	ResNet18	30.14	28.80	27.37
CIFAR100-C*	VGG16	84.85	84.31	84.05
CIFAR100-C*	ResNet18	29.67	29.56	28.35

initialization cue; their purpose is to expose ERA to ambiguous boundary regions rather than to imitate the semantics of future categories.

The corrupted-domain controls also identify the boundary of this prior. On CIFAR10-C and CIFAR100-C, the known–unknown gaps become much smaller. Simultaneous severe domain shift and novel-category discovery is broader than the same-domain TTD setting considered in this work; under such shifts, radial evidence may weaken or disappear. Angular preservation and perturbation stability remain complementary signals, but they cannot guarantee separation when both radial and semantic cues deteriorate. Similar limitations arise for semantically adjacent or highly fine-grained unknown classes.

D.3 Direction of the radial ordering

The two-ring construction is effective because it preserves and amplifies the initial ordering, not merely because it imposes two arbitrary radii. Reversing the assignment forces initially lower-norm novel features toward the outer ring and across the known-feature layer, increasing interference with known semantics. Table 8 confirms this interpretation: the reversed assignment underperforms both the observed-order construction and DAA in real-time and post-evaluation results.

Table 8: Control experiment with the assignment of known and unknown target radii reversed.

Assignment	Real-time TA	Real-time CA	Real-time $TA + CA$	Post $TA + CA$	KF
DAA	32.40	46.02	78.42	73.52	3.54
Observed order: $r_{\mathrm{kn}} > r_{\mathrm{un}}$ $[70, 40]$	33.52	47.66	81.18	76.66	3.65
Reversed order: $r_{\mathrm{kn}} < r_{\mathrm{un}}$ $[40, 70]$	30.09	44.17	74.26	70.54	3.82

D.4 Complementary roles of the components

Energy Ring is organized around five operations. First, the radial loss in Eq. 1 maps real known features toward r_{kn} and synthetic unknown-like features toward r_{un}. Second, Eq. 2 preserves the directions of known and replay-known features so that modifying their radii does not erase learned semantics. Third, the perturbation objective in Eq. 4 encourages local consistency among perturbations of the same synthetic feature and separates perturbations of different features. Fourth, energy-weighted prediction uses the radial position as a bounded modulation of cosine similarity. Finally, test-time training and stability-based rectification update ERA from the reliable portion of the stream.

These roles are complementary rather than interchangeable. Table 4 shows that unconstrained test-time training can improve cluster purity while increasing forgetting and reducing post-evaluation performance. Energy weighting supplies the radial prior, whereas perturbation rectification prevents unstable pseudo-labels from dominating online updates. Their complete $T + W + P$ configuration produces the strongest joint real-time and post-evaluation trade-off.

Table 9: Detailed comparisons of clustering metrics on Post Evaluation.

	Method	HCA↑	ARI↑	NMI↑	VM↑
CIFAR100D	L2P (Wang et al., 2022b)	50.33±2.08	37.29±1.68	69.94±5.04	69.94±5.04
	DP (Wang et al., 2022a)	55.87±1.14	44.39±1.57	72.44±0.89	72.44±0.89
	GMP (Cendra et al., 2024)	58.06±2.25	46.42±1.82	71.48±0.66	71.48±0.66
	PHE (Zheng et al., 2024)	58.53±1.04	42.73±**0.99**	70.74±1.44	70.74±1.44
	HM (Lyu et al., 2025)	63.69±**0.18**	48.35±1.76	73.08±**0.54**	73.08±**0.54**
	DAA (Liu et al., 2025)	65.62±1.29	**53.96**±1.73	75.21±1.41	75.21±1.41
	Ours	**66.38**±1.16	53.54±1.50	**75.36**±1.26	**75.36**±1.26
CUB200D	L2P (Wang et al., 2022b)	41.58±1.33	27.18±1.65	71.15±1.82	71.15±1.82
	DP (Wang et al., 2022a)	48.36±2.38	24.19±1.00	48.60±**0.14**	48.60±**0.14**
	GMP (Cendra et al., 2024)	48.06±2.25	26.42±1.82	71.48±0.21	71.48±0.21
	PHE (Zheng et al., 2024)	49.60±1.73	24.46±**0.17**	65.58±1.20	65.58±1.20
	HM (Lyu et al., 2025)	50.46±1.20	27.81±1.30	70.25±0.38	70.25±0.38
	DAA (Liu et al., 2025)	58.19±1.23	32.43±1.05	75.29±0.71	75.29±0.71
	Ours	**58.34±1.13**	**33.70**±1.07	**75.53**±0.88	**75.53**±0.88
T-ImageNetD	L2P (Wang et al., 2022b)	43.63±1.38	33.37±1.36	69.81±0.79	69.81±0.79
	DP (Wang et al., 2022a)	42.54±1.38	32.98±1.04	68.59±1.21	68.59±1.21
	GMP (Cendra et al., 2024)	56.18±**1.03**	43.00±1.88	72.79±1.94	72.79±1.94
	PHE (Zheng et al., 2024)	45.24±1.62	34.37±1.33	64.89±1.48	64.89±1.48
	HM (Lyu et al., 2025)	56.64±2.93	40.35±**0.74**	72.70±0.99	72.70±0.99
	DAA (Liu et al., 2025)	**58.91**±1.85	42.50±1.45	**73.53±0.12**	**73.53±0.12**
	Ours	58.25±1.74	**43.26**±1.24	73.36±1.31	73.36±1.31
AircraftD	L2P (Wang et al., 2022b)	30.27±0.87	16.33±0.64	43.05±1.68	43.05±1.68
	DP (Wang et al., 2022a)	31.66±1.63	17.05±0.27	44.02±**0.35**	44.02±**0.35**
	GMP (Cendra et al., 2024)	27.74±0.82	17.61±**0.17**	42.22±1.20	42.22±1.20
	PHE (Zheng et al., 2024)	31.76±**0.29**	19.76±1.36	44.70±1.13	44.70±1.13
	HM (Lyu et al., 2025)	**33.30**±0.73	21.33±1.76	46.46±0.94	46.46±0.94
	DAA (Liu et al., 2025)	32.08±1.87	21.20±0.96	46.44±0.85	46.44±0.85
	Ours	33.21±1.45	**22.13**±1.09	**47.02**±1.03	**47.02**±1.03

The synthetic features defined in Sec. 3 are used only for robust initialization. They populate transition regions near known-class boundaries; they neither define nor semantically represent the unknown classes encountered later. Actual unknown prototypes are always initialized from arriving test features.

E More TTD Comparisons

We also provide a detailed post-evaluation comparison on clustering metrics of our method with several approaches on four benchmark datasets: CIFAR100D, CUB-200D, Tiny-ImageNetD, and AircraftD. Our method achieves the best performance on HCA, ARI, NMI, and VM metrics. This indicates that our method excels in hierarchical clustering, clustering quality, and matching accuracy. The results are summarized in Table 9 and Table 10 demonstrating that our method also has better clustering performance on both real-time and post evaluation.

Table 10: Comparisons of clustering metrics on four datasets.

Method	**Real-time Evaluation**															
	CIFAR100D				CUB200D				T-ImageNetD				AircraftD			
	HCA	ARI	NMI	VM	HCA	ARI	NMI	VM	HCA	ARI	NMI	VM	HCA	ARI	NMI	VM
HM (Lyu et al., 2025)	61.49	43.04	82.78	82.78	52.84	34.90	83.51	83.51	61.76	38.18	87.24	87.24	30.02	20.16	44.25	44.25
DAA (Liu et al., 2025)	66.14	52.29	85.06	85.06	63.36	40.16	87.77	87.77	63.41	41.35	87.60	87.60	30.44	20.54	44.52	44.52
Ours	67.23	53.48	85.52	85.52	64.28	41.20	88.11	88.11	64.15	41.83	88.04	88.04	31.38	22.16	45.18	45.18

F Running time

We evaluate the running time of the compared methods. The results are shown in Fig. 10, which shows that training-required methods require significantly more time. The traditional method of updating the backbone achieves the effect of adapting to new knowledge by updating the parameters

of the basic backbone. This not only brings huge time and memory consumption, but also makes the model easy to destroy. HM (Lyu et al., 2025) takes less time than methods that update the model due to the lack of training, but it still takes longer due to its approach of finding neighbours and building hash buckets and graphs. DAA (Liu et al., 2025) uses lightweight modules, updating this module takes less time than HM searching for neighbours. Our method, due to the application of perturbation and other judgment strategies, is slightly slower than DAA, but still faster than HM.

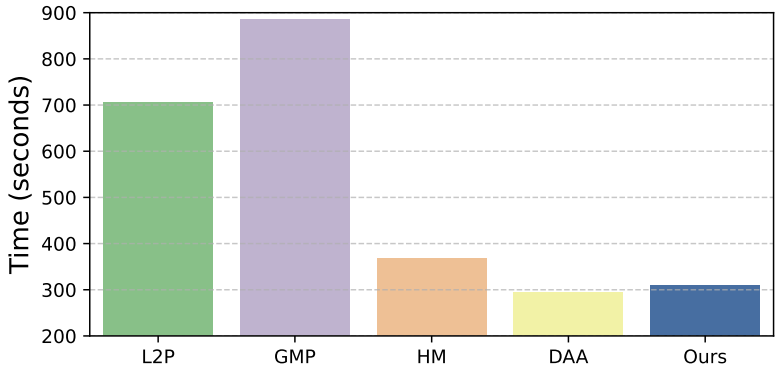


Figure 10: Comparisons of running time.

G DIFFERENT DISCOVERABLE CLASS NUMBERS

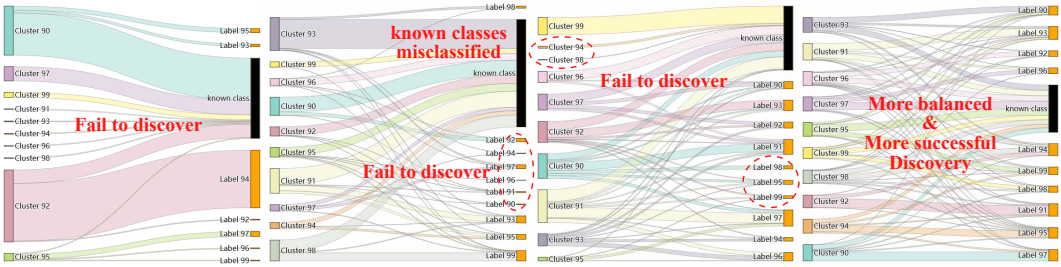
Due to the varying proportions of known and unknown classes across scenarios, selecting an appropriate number of discoverable classes becomes critical for optimal TTD performance (Table 11). Our method consistently surpasses both HM and DAA under all investigated settings. When the budget of discoverable classes is enlarged, TA and CA basically improve, reflecting stronger adaptation to emerging classes and more coherent clustering. Nevertheless, once this budget greatly exceeds the actual number of novel categories, online evaluation still exhibits rising TA and CA, yet post-hoc analysis reveals a clear TA drop and heightens knowledge forgetting (higher KF). An overly generous discoverable-class allowance boosts short-term adaptability but simultaneously injects noise that erodes previously acquired knowledge.

Table 11: Comparisons of different discoverable class numbers.

	Known +	**Real-time Eval**			**Post Eval**			
	Unknown	TA	CA	T+C	TA	CA	T+C	KF
HM	70+30	21.11	56.87	77.98	31.03	34.81	65.84	3.47
	80+20	11.57	64.15	75.72	20.90	31.09	51.99	0.69
	90+10	14.92	52.98	67.90	21.50	30.61	52.11	0.44
	70+∞	20.37	92.94	113.31	22.63	47.09	69.72	10.69
DAA	70+30	32.40	46.02	78.42	35.97	37.55	73.52	3.54
	80+20	36.21	48.13	84.34	32.56	31.44	64.00	1.73
	90+10	41.84	30.55	72.39	31.00	27.96	58.96	1.57
	70+∞	31.78	89.58	121.36	22.35	49.32	71.67	11.07
Ours	70+30	33.52	47.66	81.18	37.74	38.92	76.66	3.65
	80+20	38.93	48.11	87.04	35.91	33.37	69.29	1.77
	90+10	38.50	37.51	76.01	32.63	30.26	62.89	1.71
	70+∞	33.35	86.55	119.90	23.65	51.38	75.03	10.23

H MORE DETAILS OF LABEL MATCHING

We conduct experiments in the main text on matching comparison between unknown class predictions and ground truth. As shown in Fig. 11, we use the Sankey diagram for illustration. Each Sankey diagram is structured with clusters on the left and true labels on the right, with the width of each flow indicating the number of samples. In the diagrams, the samples assigned to each unknown cluster point to the category of the real label through a line. The flow lines between clusters and labels represent the matching or misclassification of samples. The ideal perfect situation is that all samples in each cluster belong to the same category, meaning that a cluster contains many samples and all lines point to the same real label. The diagrams reveal several phenomena: in the L2P method, there is a clear failure to discover new classes, as indicated by the absence of connections to the unknown class labels. The HM method shows some misclassification of known classes into unknown clusters which has a larger black rectangle representing known classes. The DAA method also fails to discover new classes, similar to L2P, with no connections to the unknown class labels, as highlighted by the red dashed lines around clusters 98 and 99. In contrast, our method demonstrates a more balanced and successful discovery of new classes, with a greater number of connections between the predicted clusters and the unknown class labels, indicating a more accurate classification of samples into their respective classes. The black rectangles in the diagrams indicate instances where old class samples are misclassified as unknowns, which is a phenomenon observed in the L2P and DAA methods but not very significant in our method, suggesting an improvement in classification accuracy and the ability to distinguish between known and unknown classes.

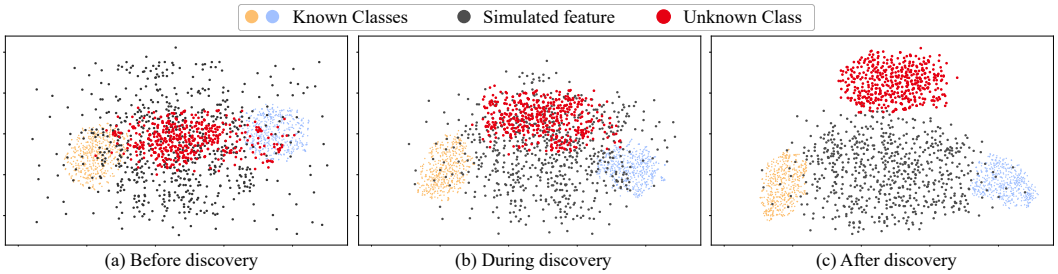


(a) L2P (b) HM (c) DAA (d) Ours

Figure 11: Matching comparison between unknown class predictions and ground truth across methods on CIFAR-100D (90+10). Black rectangles indicate old class samples misclassified as unknowns.

I EFFECTIVENESS OF SIMULATED SAMPLES

In the absence of sufficient information, the emergence of unknown classes and the most difficult part to discover are often in areas of confusion. Therefore, discovery in TTD starts from ambiguous regions where known and unknown samples overlap due to missing semantic information. The goal is not to model unknown semantics, but to expose the adapter to these regions so it learns robust separation. Using boundary-like samples as surrogates for unknowns is common practice (Cao et al., 2025; Zhu et al., 2025). As is shown in Fig. 12, unknown samples initially lie near known-class confusion regions and progressively separate during adaptation. And early discovery is prone to misclassification, which Energy Ring addresses via perturbation-based stability and rectification.



(a) Before discovery (b) During discovery (c) After discovery

Figure 12: Effectiveness of Simulated Samples.

J RELATION TO DAA

Our method builds upon DAA, which is a strong baseline for TTD. However, Energy Ring differs from DAA at a conceptual and functional level. **(1) Core assumption.** DAA collapses features onto a single hypersphere and discards norm information, while Energy Ring explicitly preserves two separated energy manifolds for known and unknown classes and treats feature norm as a first-class signal. **(2) Mechanism.** Our Eq. 1 constructs dual energy rings whose relative positions are used for known/unknown routing and confidence weighting at test time. Eq. 2 - Eq. 4 operate under this dual-manifold structure to preserve known classes and stabilize unknowns, rather than performing contrastive augmentation as in DAA. **(3) Empirical effect.** These differences lead to consistent improvements over DAA across datasets, as shown in our main results and ablations.

K HYPER-PARAMETER ANALYSIS

K.1 DIFFERENT PERTURBATIONS K DURING WARM-UP

During warm up, we create K norm-preserving perturbations by sampling random directions and encourage perturbations of the same unknown sample to stay close, while pushing those from different samples apart, improving the stability and separability of unknown classes. Unlike training during testing, warm-up is to make the framework more adaptable before testing.

Therefore, we evaluate ERA after warm-up at different K values by Hungarian algorithm (Meilă, 2003) which is widely used in NCD, GCD, CCD works (Cendra et al., 2024; Du et al., 2023; Zheng et al., 2024; Wu et al., 2023), as shown in Fig. 13. Due to the large number of samples in the training set, it is not appropriate to generate too many perturbation samples for each simulated sample. Adding a certain number of perturbation samples is beneficial for ERA to adapt to unknown classes. However, when too many perturbations are applied, it increases the computation time and prolongs the entire warm-up phase.

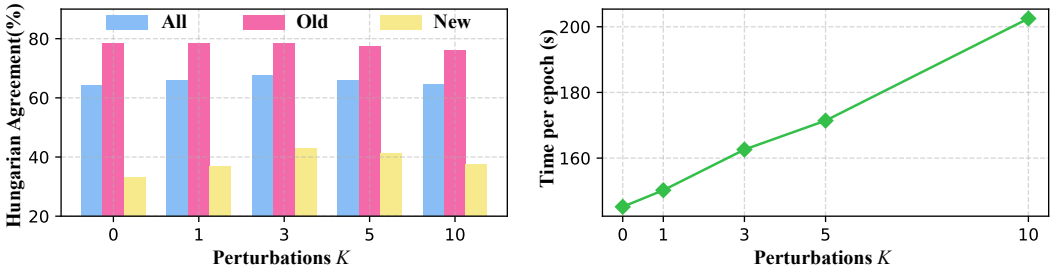


Figure 13: Comparison on different perturbations during warm-up.

K.2 FREQUENCY OF PERTURBATION RECTIFICATION

The frequency of perturbation rectification plays a crucial role in balancing model performance and computational efficiency. As shown in Fig. 14, we conduct experiments to investigate the impact of changing the perturbation rectification frequency on the overall performance of our method. The results indicate that as the perturbation rectification frequency decreases, both TA and CA significantly deteriorate. This indicates that a lower perturbation rectification frequency can lead to insufficient updates and improvements to the model during the testing phase, resulting in poor recognition performance for both known and new categories.

On the other hand, although increasing the frequency of perturbation rectification can improve the model's ability to adapt to new data, it can also lead to a significant increase in computational overhead. Frequent perturbation rectification operations require more time and resources to process each batch, which is impractical for real-time or resource limited applications. In addition, as the frequency increases, the performance improvement brought by each additional perturbation rectification operation decreases, indicating the existence of a point of diminishing returns.

Therefore, selecting the optimal frequency for perturbation rectification is crucial for achieving a balance between performance and efficiency. Moderate frequency ensures that the model can effectively utilize disturbance correction for continuous adaptation while avoiding excessive computational costs.

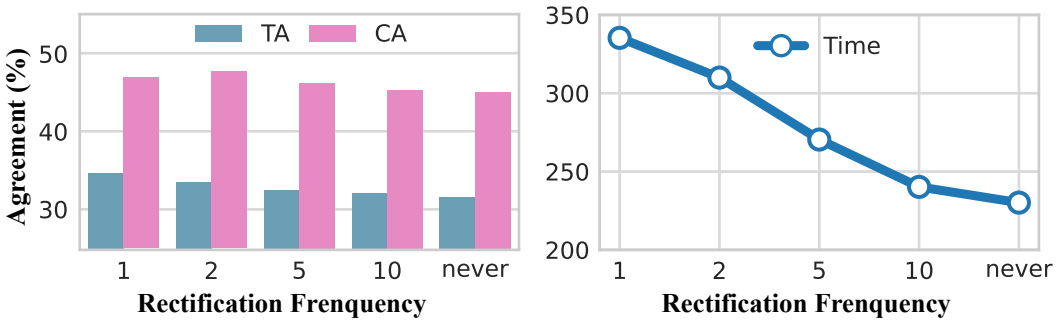


Figure 14: Trends of TA, CA and time with different perturbation rectification frenquency.

K.3 Different replay samples

The size of the replay buffer quietly influences the trade-off between plasticity and stability in our testing phase. The result in Fig. 15 reveals that every extra handful of previously-seen examples carried into the current batch improves knowledge representation of the network. When there is no replay sample, the model's connection to earlier classes largely decreases: the KF metric spikes to its maximum, a clear signature of catastrophic forgetting in which newly arriving knowledge wash away everything that was learned before.

As we gradually enlarge the pool of replayed examples, the KF trace descends in a steady, almost monotonic fashion. The ERA begins to anchor its decision boundaries to both the fresh distribution and the cached snapshots of the past, so earlier concepts remain accessible even after the optimizer has cycled through several subsequent tasks. Beyond a certain buffer size the accuracy curve starts to flatten; the marginal reduction in forgetting becomes vanishingly small while the computational cost continues to climb linearly.

Consequently, the replay-buffer cardinality must be calibrated as carefully as the learning rate or weight decay. A value chosen too low leaves the model vulnerable to abrupt knowledge erosion; a value chosen too high yields only incremental robustness while silently inflating training time and storage comsumption. The best balance lies in a moderate region where the buffer is large enough to let the optimizer rehearse a representative slice of past data, yet small enough to keep each iteration within practical latency limits, thereby sustaining a stable balance between remembering what matters and learning what is new.

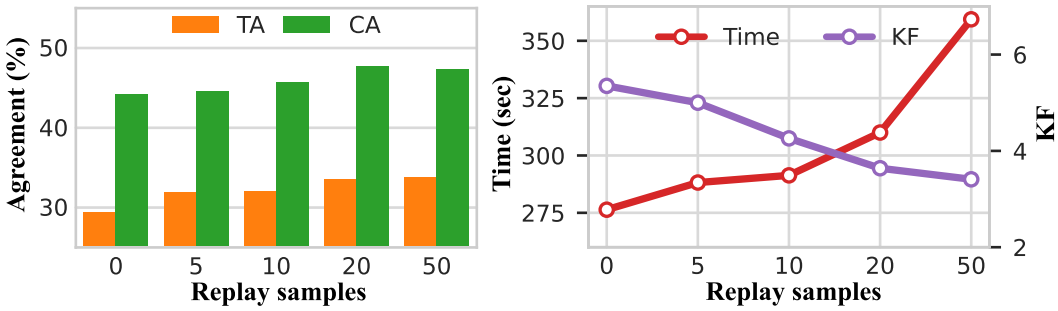


Figure 15: Trends of TA, CA and KF with different replay samples.

K.4 λ_1 and λ_2 during TTT

During test-time training (TTT), the overall testing phase loss for the ERA is shown below:

Table 12: Comparisons of ratio between λ_1 and λ_2 during TTT.

Ratio	**Real-time Eval**			**Post Eval**			
$\lambda_1 : \lambda_2$	TA	CA	T+C	TA	CA	T+C	KF
100 : 1	30.33	43.78	74.11	33.54	34.29	67.84	2.21
20 : 1	32.63	44.40	77.03	35.63	37.71	73.34	3.34
10 : 1	33.52	47.66	81.18	37.74	38.92	76.66	3.65
2 : 1	31.85	49.75	81.60	35.31	40.11	75.42	5.66
1 : 1	27.15	55.37	82.52	28.11	37.20	65.31	8.25
1 : 5(crashed)	79.04	11.60	—	99.41	0.85	—	—
1 : 10(crashed)	83.57	10.07	—	100	0.00	—	—

$$\mathcal{L}_{\text{test}}(x) = \lambda_1 \cdot \mathbb{E}_{\hat{y}\in\mathcal{Y}_{\text{kn}}}\left(\mathcal{L}_{\text{kn}}(x|\hat{y}) + \mathcal{L}_{\text{kn}}^{\text{replay}}(x)\right) + \lambda_2 \cdot \mathbb{E}_{\hat{y}\in\mathcal{Y}_{\text{seen}}}\left(\mathcal{L}_{\text{un}}(x|\hat{y}) + \mathcal{L}_{\text{un}}^{\text{renewal}}(x|\hat{y})\right), \tag{21}$$

where λ_1 and λ_2 are hyperparameters that balance the contributions of each loss term.

We explore the impact of different ratios between λ_1 and λ_2 on the overall performance of our method. The results are presented in Table 12. Known class loss enables the model to better avoid forgetting old knowledge, while unknown class loss enables the model to better adapt to new knowledge of unknown classes. We conduct experiments with different ratios of known classes loss to unknown classes loss to determine the optimal balance between these two objectives.

When the model places more emphasis on predicting known categories, it focuses on old knowledge of known categories, resulting in relatively high TA and CA. However, the model's ability to distinguish between known and unknown categories is limited to some extent. As the ratio decreases, the model begins to pay more attention to the loss of unknown classes and adapts more aggressively to new knowledge, which helps to improve the separation between unknown classes. This leads to a slight increase in CA, indicating better class separation. However, the overall TA value was affected, indicating a slight impairment in the model's ability to accurately classify known categories. However, excessive emphasis on positional loss may undermine old knowledge. When the ratio further decreases, the model crashes, as shown by extremely high TA and near-zero CA values. This indicates that overemphasizing the loss of unknown classes can make the model unstable and lead to poor performance.

These results emphasize the importance of carefully balancing known class loss and unknown class loss in the TTT process. The optimal ratio ensures that the model can effectively improve its prediction of known classes while maintaining clear boundaries between known and unknown classes. This balance is crucial for achieving high accuracy and robustness in identifying known and new categories.

K.5 Comparison of different threshold γ

During testing phase, we base on the max similarity scores and apply a confidence threshold γ to determine whether a test sample x belongs to a seen class or an unseen class.

$$\hat{y} = \begin{cases} \arg\max_{c\in(\mathcal{Y}_{\text{kn}}\cup\mathcal{Y}_{\text{seen}})}(\tilde{s}_c), & \text{if } \max_c \tilde{s}_c > \gamma, \\ \text{new discovered class}, & \text{otherwise.} \end{cases} \tag{22}$$

γ was selected through grid search in our experiment, and we analyze the impact of γ. The typical threshold of γ is 0.7. If the threshold is too high, unknown classes are difficult to discover and will be mistakenly classified as known classes. On the contrary, if the threshold is too low, incoming samples will be erroneously classified as unknown classes. And our method is not sensitive to selection from one dataset to other datasets.

Table 13: Comparisons of different γ during Testing phase.

Threshold	**Real-time Eval**			**Post Eval**			
γ	TA	CA	T+C	TA	CA	T+C	KF
0.5	12.67	47.70	60.37	29.34	34.74	64.08	5.01
0.6	22.99	46.28	69.27	30.07	35.41	65.48	4.96
0.7	33.52	47.66	81.18	37.74	38.92	76.66	3.65
0.8	28.96	44.25	73.21	31.98	30.67	62.65	6.29
0.9	13.92	45.88	59.80	32.39	27.44	59.83	10.16

L Parameter Selection and Computational Cost

L.1 Target radii

The quantities r_{kn} and r_{un} are global scalar target radii rather than sample-specific features. Figure 7 shows that excessive separation and excessive compression both degrade performance. For ViT-B/16, we set $r_{\text{kn}} = 70$ because the known backbone-feature norms are centered near 70; the same pair is then used across all four same-domain datasets.

We also evaluate an automatic calibration rule that uses only warm-up data:

$$r_{\text{kn}} = \text{median}_{x \in \mathcal{D}_{\text{kn}}} \|f(x)\|_2, \qquad r_{\text{un}} = \text{median}_{\tilde{\mathbf{r}} \in \mathcal{D}_{\text{sim}}} \|\tilde{\mathbf{r}}\|_2. \tag{23}$$

The resulting radii are approximately 67 and 52. This rule preserves the observed ordering but produces less separation than the manually selected pair, leading to the lower performance reported in Table 14. The result suggests a viable data-driven initialization that does not use unknown-class validation labels, while leaving adaptive radius selection as future work.

Table 14: Manual and median-calibrated target radii.

Radius choice	Real-time TA	Real-time CA	Post TA	Post CA	KF
Manual $[70, 40]$	33.52	47.66	37.74	38.92	3.65
Median-calibrated $[67, 52]$	30.65	41.28	33.79	34.21	3.93

L.2 Stability-rechecking overhead

Feature-level perturbations add modest latency and little memory relative to DAA, as shown in Table 15. Increasing k beyond the default makes the stability estimate more expensive without a commensurate gain, consistent with Fig. 6. The default Energy Ring configuration remains faster than HM, and its peak memory is close to DAA.

Table 15: Latency and memory cost of stability rechecking.

Method / setting	Latency/sample (ms)	Latency/perturbation (ms)	Peak memory (MB)	Stream time (s)
HM	271	–	13919	368
DAA	142	–	9256	294
Energy Ring, $k = 3$	156	3.2	9335	294
Energy Ring, default $k = 5$	156	3.2	9335	310
Energy Ring, $k = 10$	156	3.2	9335	353

M Statistical Robustness against DAA

TA and CA characterize complementary failure modes. TA measures whether samples from a true novel class consistently receive one discovered label, whereas CA measures whether each discovered cluster is dominated by one true class. Considering either metric alone can favor fragmentation or

Table 16: Paired full-stream comparison between DAA and Energy Ring on $TA + CA$ over 10 runs.

Dataset	Protocol	Runs	DAA	Energy Ring	Δ [95% CI]	$t(9)$	Raw p	Holm p	d_z	Sign-flip p
CIFAR100D	Real-time	10	76.69 ± 1.27	81.17 ± 1.25	4.48 [3.42, 5.54]	9.55	0.000005	0.000011	3.02	0.001953
CIFAR100D	Post	10	72.90 ± 1.62	76.01 ± 1.85	3.11 [1.53, 4.69]	4.45	0.001602	0.001602	1.41	0.003906
CUB200D	Real-time	10	112.84 ± 1.93	115.92 ± 1.43	3.07 [1.29, 4.86]	3.90	0.003625	0.007251	1.23	0.005859
CUB200D	Post	10	108.86 ± 2.49	110.67 ± 1.48	1.81 [−0.12, 3.73]	2.12	0.062775	0.062775	0.67	0.066406
Tiny-ImageNetD	Real-time	10	95.36 ± 1.85	98.16 ± 1.31	2.79 [1.28, 4.31]	4.17	0.002416	0.002416	1.32	0.005859
Tiny-ImageNetD	Post	10	59.59 ± 1.51	62.63 ± 1.29	3.04 [1.62, 4.46]	4.84	0.000918	0.001836	1.53	0.001953
AircraftD	Real-time	10	68.60 ± 0.62	69.78 ± 1.59	1.18 [−0.13, 2.49]	2.03	0.072771	0.072771	0.64	0.087891
AircraftD	Post	10	57.34 ± 1.69	59.87 ± 1.87	2.53 [0.69, 4.36]	3.11	0.012584	0.025169	0.98	0.015625

over-merging; we therefore use $TA + CA$ as the pre-specified primary discovery metric and interpret the method as improving their joint trade-off rather than uniformly maximizing every individual metric.

We perform 10 paired full-stream runs for every dataset and evaluation protocol. Each Energy Ring run is matched to a DAA run with the same known/unknown split, pretrained checkpoint, stream order, and batch partition. Table 16 reports the paired mean difference, its 95% confidence interval, a two-sided paired t-test, Cohen's d_z, the raw and Holm-corrected p-values, and an exact paired sign-flip test. Pairing controls stream-level variation and directly tests the difference between methods rather than drawing conclusions from overlapping marginal ranges.

Six of the eight comparisons remain significant after Holm correction. The post-evaluation result on CUB200D and the real-time result on AircraftD are positive but do not cross the corrected significance threshold. These two cases are therefore interpreted as modest gains, consistent with the greater difficulty of fine-grained separation when radial and semantic differences are small.