TEST-TIME ERROR MINIMIZATION FOR OPEN SET RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

The success of supervised learning largely stems from explicit error minimization. In contrast, Test-Time Adaptation lacks ground-truth labels, precluding direct optimization. This work enables explicit error minimization for Test-Time Open-Set Recognition (TOSR). Specifically, we identify a previously overlooked fact: under a fixed recall, the challenging objective of Test-Time Error Minimization is equivalent to the tractable objective of Negative Prediction Maximization. Based on that, our algorithm learns a set of discrete boundaries and searches for their ensemble to maximize negative predictions. Experiments demonstrate that our method successfully closes around 50% of the performance gap between TOSR methods and supervised learning on several benchmarks.

1 INTRODUCTION

Open-set recognition (OSR) endows conventional classifiers with the ability to detect unknown classes. Traditional OSR methods follow an offline paradigm, where the model is prohibited from accessing unknown samples. Inspired by the rapid development of Test-Time Adaptation (TTA) (Wang et al., 2020), Test-Time Open-Set Recognition (TOSR) has been introduced (Fan et al., 2024; Gao et al., 2024), allowing models to leverage test-time samples as a free source of information to improve their performance on the fly.

The challenge in TOSR lies in the absence of labels for test-time samples. Existing TOSR approaches typically collect predicted unknowns with high confidence and use them as prototypes. Predictions are then made based on the similarity to the prototypes (Yang et al., 2025). While making progress, TOSR methods still fall substantially short of supervised learning with access to labeled test samples (Fig. 1), suggesting significant room for further improvement. **Can the performance gap between TOSR and supervised learning be further narrowed?**

The success of supervised learning largely stems from explicit error minimization: it uses ground-truth labels to measure prediction errors and explicitly optimizes the model to minimize these errors, such as cross-entropy loss and hinge loss. A key advantage of explicit error optimization is its distribution-agnostic nature, allowing the method to generalize beyond specific distributional forms. This is particularly important for open-set recognition, where the distributions of unknown classes might exhibit diverse irregularities and make it difficult to characterize a prior. Existing TOSR meth-

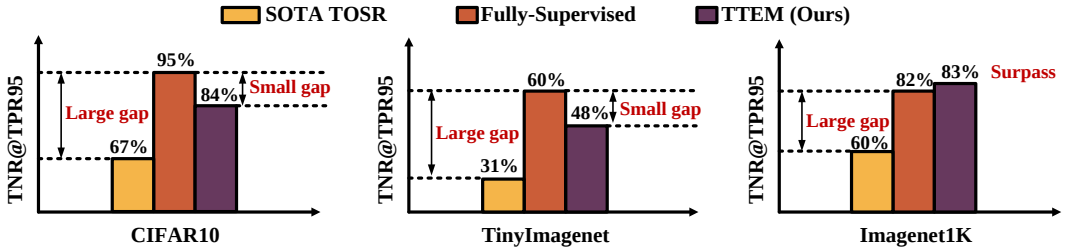


Figure 1: By explicitly minimizing test-time error, our algorithm substantially narrows the performance gap between TOSR methods and fully-supervised methods.

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.

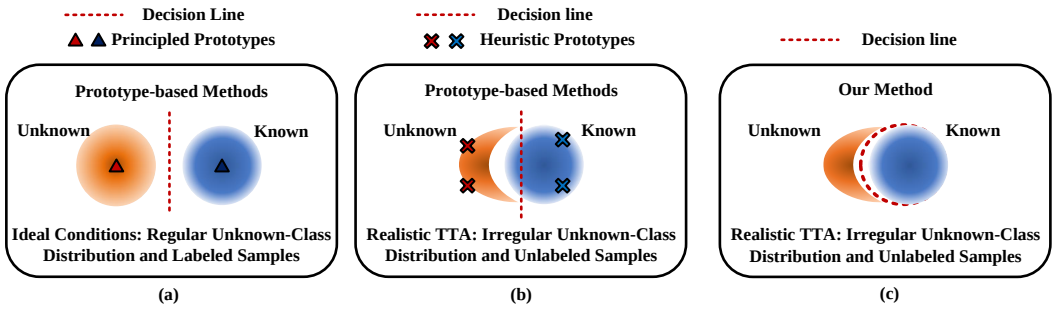


Figure 2: (a) Under ideal conditions, prototype-based methods can achieve error minimization. (b) Under realistic TOSR conditions, prototype-based methods inefficiently approximate test-time error minimization, leading to suboptimal performance. (c) Our method explicitly optimizes test-time error without making distributional assumptions, thereby achieving strong generalization.

ods heuristically identify prototypes for unknown classes, which often fail to adequately represent the distributions of unknown classes, leading to suboptimal performance (Fig. 2). This paper investigates how to explicitly optimize errors for TOSR. The challenge is evident: test samples lack ground-truth labels for error estimation.

Before delving into the algorithm, we present its underlying insight. We view unknowns detection as a problem of identifying the boundaries of the known-class distribution, where samples falling outside these boundaries are recognized as unknowns. Traditional OSR methods estimate the entire boundary of a known-class distribution (Lee et al., 2018), which can incur substantial estimation errors, because the number of samples required to characterize the boundary grows prohibitively with the dimensionality (Lehmann & Casella, 1998; Kay, 1993). Moreover, estimating the entire boundary is also unnecessary, since real-world data distributions are known to lie on low-dimensional manifolds (Bengio et al., 2013). Therefore, we aim to identify a set of discrete boundaries that can capture the major components of the test-time unknown samples (Fig. 3). Based on this insight, we propose a novel objective, termed Negative Prediction Maximization. We show that, under a fixed recall rate, Test-Time Error Minimization is equivalent to Negative Prediction Maximization, which is a technically tractable objective for ensembling discrete boundaries.

Our algorithm consists of three components: (i) discrete-boundary training: we learn a set of discrete boundaries from the test data stream, each serving as a promising candidate for detecting test-time unknown patterns. (ii) discrete-boundary ensembling: we employ a greedy strategy to identify an ensemble of discrete boundaries that maximizes the negative prediction. (iii) adaptive detector: To alleviate the performance degradation at the onset of adaptation, we introduce an adaptive detector that switches between a prior detector and the TTA detector according to their latest performance. Our contributions are summarized as follows:

- **Explicit Error Minimization without Ground-Truth Labels.** We are the first to explicitly minimize test-time error, which is critical for TOSR. Specifically, we identify a previously overlooked fact: under a fixed recall, the challenging objective of Time Error Minimization is equivalent to the tractable objective of Negative Prediction Maximization.
- **Discrete-Boundary Ensemble.** Discrete-Boundary Ensemble provides a practical implementation of Negative Prediction Maximization. Instead of estimating the complete boundary of the known-class distribution, we give a novel perspective of seeking discrete boundaries that can detect the majority of test-time unknown-class samples.
- **Adaptive Detector.** It automatically switches between the prior detector and the TTA detector based on their relative performance, mitigating the performance degradation at the onset of test-time adaptation.
- **Significant Performance Gains.** Our algorithm successfully narrows the performance gap between TOSR and supervised methods by around 50% on several benchmarks, including CIFAR10, TinyImageNet, AwA, aPaY, and ImageNet1K.

2 Related Work

Traditional open-set recognition follows an offline paradigm, where the model is prohibited from accessing test-time unknown classes. They can be broadly categorized into two major groups. (1) Density-based methods, which include explicit probabilistic density estimation methods, such as VAEs (Sun et al., 2020), flow-based density estimators (Zhang et al., 2020) and Extreme-value tail fitting (Oza & Patel, 2019), as well as implicit density estimation methods (Li et al., 2025), such as energy-based approaches (Liu et al., 2020) and prototype-based approaches (Yang et al., 2022; Fu et al., 2024). (2) Adversarial data training, including adversarial data generation (Du et al., 2022; Jiang et al., 2025; Du et al., 2023; Zhou et al., 2021; Yu et al., 2024) and auxiliary data selection (Jiang et al., 2024; Zhu et al., 2023). For more interesting studies, we recommend these surveys (Geng et al., 2021; Yang et al., 2024).

Test-time open-set recognition (TOSR) allows models to adapt to unlabeled test data stream. Existing TOSR methods can be broadly viewed as prototype-based approaches. OODD (Yang et al., 2025) maintains a dynamic OOD dictionary and makes predictions based on the cosine similarity with the stored OOD prototypes. AdaODD (Zhang et al., 2025) identifies a set of high-confidence known and unknown samples and makes predictions based on the Euclidean-distance similarity. DCAC (Wu et al., 2026) dynamically maintains a Class-Aware Cache containing known and unknown samples and uses them to calibrate OOD scores. ANTS (Zhu et al., 2026) and TANL (Zhang et al., 2026) exploit the capabilities of vision-language models and optimize textual prototypes using test-time samples. These prototype-based methods implicitly optimize the test-time error and can achieve optimal performance only under restrictive conditions.

3 Algorithm

We first give the formulation of Test-Time Open-set Recognition and Test-Time Error Minimization. Then, we establish a connection between Test-Time Error Minimization and Negative Prediction Maximization. Building upon that, we present the design of our algorithm.

3.1 Preliminaries

Throughout this paper, we assume a pre-trained closed-set classifier is available and kept fixed, and our focus is on unknown-class detection. We consider two data distributions, denoted by $\mathcal{P}_{pos}$ and $\mathcal{P}_{neg}$, corresponding to the known and unknown distributions, respectively. A test data stream $\mathcal{D}_{test} = [x_1, x_2, \ldots, x_n]$ can be constructed as $x_i \sim (1-\pi)\mathcal{P}_{pos} + \pi\mathcal{P}_{neg}$, where the corresponding labels $y_i \in \{\text{“}positive\text{”}, \text{“}negtive\text{”}\}$ are inaccessible to the model. Test-Time Open-Set Recognition aims to develop an unknown-class detector $h : \mathcal{X} \to \{\text{“}positive\text{”}, \text{“}negative\text{”}\}$ that can simultaneously predict on and adapt to the incoming test data $\mathcal{D}_{test}$. Formally, through adaptation, the detection performance of $h(\cdot)$ on test-time batches can be progressively improved. Performance can be measured by **TNR@TPR95**:

$$\begin{aligned} \text{TNR@TPR95} = & \frac{\sum \mathbb{I}[h(x_i) = \text{“}negative\text{”}]\,\mathbb{I}[y_i = \text{“}negative\text{”}]}{\sum \mathbb{I}[y_i = \text{“}negative\text{”}]} \\ \text{s.t.} \quad & \frac{\sum \mathbb{I}[h(x_i) = \text{“}positive\text{”}]\,\mathbb{I}[y_i = \text{“}positive\text{”}]}{\sum \mathbb{I}[y_i = \text{“}positive\text{”}]} \geq 0.95, \end{aligned} \tag{1}$$

where (x_i, y_i) denotes the samples in the current test-time batch $\mathcal{B}^t$.

Test-Time Error Minimization is defined as:

$$\arg\min_{h(\cdot)} \quad \sum \mathbb{I}[h(x_i) \neq y_i]] \tag{2}$$

where (x_i, y_i) denotes all encountered samples in the test data stream. Under a fixed 95% known-class recall, Test-Time Error Minimization is equivalent to TNR@TPR95 maximization. Test-Time Error Minimization possesses two desirable properties: (i) Similar to Empirical Risk Minimization, the performance of $h(\cdot)$ is guaranteed by the generalization bound. (ii) As test-time adaptation progresses and more samples are encountered, the generalization bound becomes tighter.

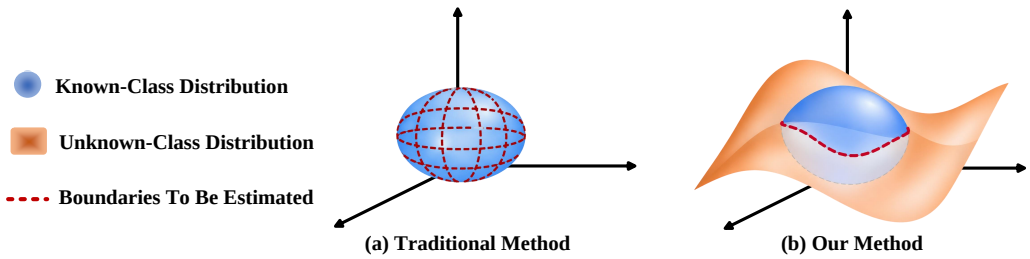


Figure 3: Typical OSR methods estimate the entire boundary of the known classes, whose performance is limited by the explosive growth in estimated parameters as the dimensionality increases. Our method instead estimates a set of discrete boundaries that can capture the major portions of the unknown-class distribution, substantially reducing the complexity of boundary estimation.

In this paper, we aim to explicitly minimize the test-time error without access to ground-truth labels. In the following, we will show that this challenging objective can be transformed into a tractable objective of Negative Prediction Maximization.

Proposition 1. *For an unknown-class detector operating on a test-time data stream, under a fixed α known-class recall, e.g., $\alpha = 95\%$, Test-Time Error Minimization is equivalent to Negative Prediction Maximization.*

Proof. Consider a test-time data stream containing P_{all} known-class samples and N_{all} unknown-class samples. Let h_a and h_b be two unknown-class detectors with the same recall rate α over the data stream. Their corresponding test-time errors are denoted by ϵ_a and ϵ_b. Further, assume their numbers of negative predictions satisfy $N_a > N_b$. Given a negative prediction consists of either a false positive (FP) or a true negative (TN), we have $N_a = \text{FP}_a + \text{TN}_a, N_b = \text{FP}_b + \text{TN}_b$. Since $\text{FP}_a = \text{FP}_b = P_{\text{all}} * (1 - \alpha)$, $N_a > N_b$ implies $\text{TN}_a > \text{TN}_b$. Furthermore, since N_{all} consist of true negatives (TN) and false negatives (FN), we have $\text{FN}_a = N_{\text{all}} - \text{TN}_a$ and $\text{FN}_b = N_{\text{all}} - \text{TN}_b$. Therefore, we have $\text{FN}_a < \text{FN}_b$. Together with $\text{FP}_a = \text{FP}_b$, it follows that $\text{FN}_a + \text{FP}_a < \text{FN}_b + \text{FP}_b$. We conclude that $\epsilon_a < \epsilon_b$. □

The above result provides two key insights for TOSR algorithm design. First, when pursuing Test-Time Error Minimization under a fixed known-class recall—a setting aligned with the objective of TOSR, Negative Prediction Maximization can be used as a tractable surrogate. Second, when two unknown-class detectors achieve the same known-class recall, we select the one that produces more negative predictions. In practice, one may notice that the ground-truth labels of the test-time data are unavailable, and thus the recall rate cannot be directly controlled. To remedy this issue, we use a held-out validation set containing known-class samples to select the threshold such that two detectors achieve the same recall rate on the validation set. Note that it is also a standard threshold calibration strategy in OSR. The rationale is as follows: Suppose that two unknown-class detectors, h_a and h_b, achieve the same recall rate on a validation set. Then, we have $\Pr(\text{FP}_a > \text{FP}_b) = \Pr(\text{FP}_a < \text{FP}_b)$. Consequently, if we observe $N_a > N_b$ on the test-time data stream, it follows that $\Pr(\epsilon_a < \epsilon_b) > \Pr(\epsilon_b < \epsilon_a)$. If the validation set size increases or the gap between N_a and N_b becomes larger, $\Pr(\epsilon_a < \epsilon_b)$ increases accordingly. The exact calculation of the confidence level of $\Pr(\epsilon_a < \epsilon_b)$ requires sophisticated probabilistic computation methods, which is beyond the scope of this work. Nevertheless, the above qualitative result is sufficient to suggest Negative Prediction Maximization as a grounded probabilistic criterion for adaptive selection.

3.2 DISCRETE-BOUNDARY TRAINING

The boundary estimation of high-dimensional distributions requires a prohibitively large number of samples to achieve a given estimation accuracy (Lehmann & Casella, 1998; Kay, 1993). For example, a full-covariance Gaussian in a 512-dimensional feature space requires estimating 131,840 parameters, making it difficult to reliably characterize from only a few hundred samples per class. Our key insight is that estimating the complete boundary is redundant, and it is sufficient to identify a set of discrete boundaries that can capture the majority of test-time unknown patterns (Fig. 2). In this subsection, we project the deep feature space into a specific one-dimensional subspace and

then locate the boundary of known classes in the subspace, which is more reliable. These learned boundaries in the subspaces essentially correspond to a series of discrete boundaries in the original deep-feature space.

For a test-time batch $\mathcal{B}^t$, we design a two-step procedure to obtain a discrete boundary, with each step leveraging a small-sized dataset, namely, $\mathcal{D}^a_{tr}$ and $\mathcal{D}^b_{tr}$. Specifically, $\mathcal{D}^a_{tr} = \mathcal{D}^+ \cup \mathcal{D}^-$, where $\mathcal{D}^+ = \{(z_i, 1) \mid z_i = f(x_i), x_i \sim \mathcal{P}_{pos}\}$, and $\mathcal{D}^- = \{(z_i, -1) \mid z_i = f(x_i), x_i \in \mathcal{B}^t\}$. $\mathcal{D}^b_{tr} = \{z_i \mid z_i = f(x_i), x_i \sim \mathcal{P}_{pos}\}$. For $\mathcal{D}^+$ and $\mathcal{D}^b_{tr}$, we employ two independent small-sized known-class replay sets, respectively. Since we only need to store deep features $z_i = f(x_i)$ instead of raw images x_i, their total storage consumption is less than 10 MB in all our experiments.

At the first stage, we use $\mathcal{D}^a_{tr}$ to train a subspace projection $g(z)$:

$$g(z) = \sum_i \alpha_i y_i K(z_i, z), \tag{3}$$

where α_i is obtained by solve the kernel SVM problem:

$$\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j K(z_i, z_j) - \sum_i \alpha_i. \tag{4}$$

The subspace projection $g(z)$ essentially learns a kernel-induced space in which the known-class replay data and online test-time data can be well-separated. Because the online test-time data may contain unknown samples, the learned subspace has the potential to be discriminative. We note that our method is agnostic to the specific choice of subspace projection. Linear projection is also applicable, but we empirically find that kernel projection achieves better generalization.

At the second stage, we locate the known-class boundary in the subspace. If the test-time batch D^- consists purely of unknown samples, $b = 0$ naturally provides a desirable boundary. However, a test-time batch may contain both known and unknown samples. Therefore, we recalibrate the boundary b by fitting a Gaussian distribution on $\mathcal{D}^b_{tr}$:

$$b = \mu + \sigma \Phi^{-1}(1 - \gamma), \tag{5}$$

where $\mu = \frac{1}{|\mathcal{D}^b_{tr}|} \sum z_i, \sigma^2 = \frac{1}{|\mathcal{D}^b_{tr}|} \sum (z_i - \mu)^2$, and $\Phi(\cdot)$ denotes the cumulative distribution function (CDF) of the standard Gaussian. γ is set to 99.9% to ensure that the boundary b includes all known-class samples with a high probability. We use Gaussian fitting because we observe that the projected known-class samples exhibit a unimodal distribution in the subspace, which generally holds for valid backbones (Lee et al., 2018). We emphasize that we make no assumption about the distribution of unknown classes.

Consequently, for a test-time batch $\mathcal{B}^t$, we obtain an unknown-class detector $h^{(t)}$ based on the learned discrete boundary:

$$h^{(t)}(x) = \begin{cases} \text{“}potitive\text{”}, & g^{(t)}(f(x)) - b^{(t)} > 0, \\ \text{“}negative\text{”}, & g^{(t)}(f(x)) - b^{(t)} \leq 0. \end{cases} \tag{6}$$

The unknowns generalization of h depends on the discriminability of the subspace $g(\cdot)$, which is further determined by the proportion of unknown-class samples in $\mathcal{D}^-$. To further improve the discriminability, we leverage a prior unknown-class detector h_0 (*e.g.*, a softmax-based detector) as a free lunch to filter $\mathcal{D}^-$ as $\mathcal{D}^-_{filtered} = \{(z_i, -1) \mid h_0(x_i) = \text{“}negative\text{”}, \{z_i, -1\} \in \mathcal{D}^-\}$.

3.3 DISCRETE-BOUNDARY ENSEMBLING

A single discrete boundary has limited unknown-class detection capability. In this subsection, we ensemble multiple discrete boundaries to form a powerful unknown-class detector. The preceding Prop. 1 provides a grounded and feasible objective: searching for a set of discrete boundaries that maximizes the negative predictions.

To achieve this objective, we need to record the prediction of each boundary on every test-time sample. Each prediction has three possible states: positive, negative, and N/A, where N/A indicates that either the sample appeared before the boundary was obtained or the sample has not been encountered. Based on the prediction records, we can then employ a greedy algorithm to identify an

ensemble of discrete boundaries that achieves the maximum negative predictions: First, we select the boundary that produces the largest negative predictions among all existing boundaries, and add it to an ordered boundary list:

$$h^1_{sorted}(\cdot) = \arg\max_t \quad \frac{|\{x_i|h^{(t)}(x_i) = \text{“}negative\text{”}\}|}{|\{x_i|h^{(t)}(x_i) \neq \text{“}N\backslash A\text{”}\}|}. \tag{7}$$

where $x_i \in \mathcal{D}_{test}$. Here, the number of negative predictions is normalized by the number of encountered test samples. In the subsequent iterative process, we select one discrete boundary at each iteration k that provides the largest gain in negative predictions for the ordered boundary list, and add it to the ordered boundary list again:

$$h^k_{sorted}(\cdot) = \arg\max_t \quad \frac{|\{x_i|h^{(t)}(x_i) = \text{“}negative\text{”}, h^j_{sorted}(x_i) \neq \text{“}negative\text{”}, j < k\}|}{|\{x_i|h^{(t)}(x_i) \neq \text{“}N\backslash A\text{”}\}|}. \tag{8}$$

where $x_i \in \mathcal{D}_{test}$. The generated $\{h^k_{sorted}\}^T_{k=1}$ is essentially a ranking of all obtained boundaries. We select the top-K boundaries to form the ensemble, where K is determined according to the known-class recall. During inference, the unknown-class detector $H^{(t)}$ induced by the ensemble operates as follows:

$$H^{(t)}(x) = \begin{cases} \text{“}negative\text{”}, & if \quad \exists k < K, \quad h^k_{\text{sorted}}(x) = \text{“negative”}. \\ \text{“}positive\text{”}, & otherwise. \end{cases} \tag{9}$$

The greedy algorithm is executed for every test-time batch $\mathcal{B}^t$. As more test-time batches are fed into the model, an increasing number of boundaries are learned. To satisfy the memory budget, we can remove the bottom-M boundaries in $\{h^k_{sorted}\}^T_{k=1}$.

3.4 ADAPTIVE DETECTOR

TOSR methods typically require several test-time batches to reach peak performance. To overcome the cold-start performance issue, we leverage a prior unknown-class detector $h_0(\cdot)$ as a free lunch (*e.g.*, a softmax-based detector). Specifically, we design an adaptive detector $H^{*(t)}(\cdot)$ that dynamically switches between the prior detector $h_0(\cdot)$ and TTA detector $H^{(t)}(\cdot)$ based on their performance on the previous batch $\mathcal{B}^{t-1}$. According to Prop. 1, negative prediction maximization provides a grounded criterion for switching:

$$H^{*(t)}(\cdot) = \begin{cases} H^{(t)}(\cdot), & if \quad |\{x_i|H^{(t)}(x_i) = \text{“}negative\text{”}\}| > |\{x_i|h_0(x_i) = \text{“}negative\text{”}\}|. \\ h_0(\cdot), & otherwise. \end{cases} \tag{10}$$

where $x_i \in \mathcal{B}^{t-1}$. The prior detector $h_0(\cdot)$ has an advantage in detecting unknown unknowns, while the test-time detector $H^{(t)}(\cdot)$ is more effective in identifying known unknowns. Our designed adaptive detector effectively combines the strengths of both types, thereby further improving the test-time performance.

4 EXPERIMENT

4.1 SETUP

Datasets. We conduct experiments on eight intra-dataset benchmarks and two cross-dataset benchmarks. The benchmarks cover digit datasets, natural image datasets, fine-grained datasets, few-shot datasets, and large-scale datasets. For the intra-dataset benchmarks, each dataset is randomly partitioned into known and unknown classes, with the known-to-unknown class ratio ranging from 1:1 to 1:10. Specifically, we use splits of 6:4 for CIFAR-10 and SVHN, 10:90 for CIFAR-100, 20:180 for TinyImageNet and CUB, 10:40 for AwA, 10:22 for aPY, and 500:500 for ImageNet-1K. For the cross-dataset benchmarks, *i.e.*, CIFAR10-SVHN and SVHN-CIFAR10, the entire CIFAR10 or SVHN dataset is treated as either the known or unknown classes. Based on the partitioning of known and unknown classes, the dataset is divided into a training set and a test set. The training set contains only known classes and is used to train a closed-set classifier and a prior unknown-class detector. The test set consists of an even mixture of known and unknown samples, with their ratio controlled at 1:1. No labels are provided during test-time adaptation; the ground-truth labels are revealed only for performance evaluation.

Table 1: Comparison experiments on three popular OSR datasets using ***CLIP*** as the backbone. The results are reported in terms of *TNR@TPR95*.

CLIP	**CIFAR10**		**TinyImagenet**		**Imagenet1K**	
	Last	**Avg**	**Last**	**Avg**	**Last**	**Avg**
Softmax	41.0	41.0	25.9	25.9	31.6	31.6
RTL *(CVPR'2024)*	48.5	44.6	31.5	30.6	35.4	34.5
TANL *(CVPR'2026)*	67.6	65.8	28.5	28.0	43.2	43.0
ANTS *(CVPR'2026)*	65.8	64.6	27.8	27.5	44.9	44.5
AdaODD *(TPAMI'2026)*	54.8	52.8	19.1	21.4	62.9	59.8
DCAC *(AAAI'2026)*	41.5	41.2	27.4	27.6	43.2	43.1
TTEM	84.3	81.8	48.5	43.5	82.8	81.5
Adaptive TTEM	**84.3** (+16.7)	**82.0** (+16.2)	**48.5** (+17.0)	**43.7** (+13.1)	**82.8** (+19.9)	**81.6** (+21.8)
Fully-Supervised	95.3		60.7		82.1	

Table 2: Comparison experiments on three datasets using ***ResNet-101*** as the backbone. The results are reported in terms of *TNR@TPR95*.

ResNet-101	**CUB**		**AwA**		**aPaY**	
	Last	**Avg**	**Last**	**Avg**	**Last**	**Avg**
Softmax	26.2	26.2	56.3	56.3	35.5	35.5
RTL *(CVPR'2024)*	16.9	17.2	58.1	57.3	40.1	39.7
OODD *(CVPR'2025)*	21.1	20.9	45.6	41.3	29.6	28.3
AdaODD *(TPAMI'2026)*	10.0	9.8	64.5	70.2	44.1	40.3
DCAC *(AAAI'2026)*	27.8	27.8	71.5	71.4	40.2	40.1
TTEM	28.1	21.5	81.6	78.1	62.1	62.0
Adaptive TTEM	**28.1**(+0.3)	**27.8**(+0.0)	**81.6**(+10.1)	**78.4**(+7.0)	**62.1**(+18.0)	**62.1**(+21.8)
Fully-Supervised	41.3		90.1		86.2	

Implementation Details. Our method optimizes only the final decision function while keeping the feature extractor frozen. The test-time batch size is set to 128, and the boundary hyperparameter γ is set to 0.001. For most datasets, we retain 100 samples per known class for replay, except for CUB, where we retain 30 samples per class, and ImageNet-1K, where we retain 10 samples per class. Since we store only the deep features of the replay samples, the total memory overhead is less than 10 MB, making our method suitable for deployment on edge devices. The storage budget for discrete boundaries is capped at 300. A softmax-based prior detector is used for adaptive selection. Additionally, we use the labeled test set to train a kernel SVM offline as a fully-supervised method for comparison, because kernel SVM remains highly competitive in terms of last-layer prediction-making capability. The code will be released upon acceptance.

Algorithm 1 Adaptive TTEM

Input: A trained backbone $f(\cdot)$, a prior detector h_0, $\mathcal{D}_{test}$, $\mathcal{D}_{tr}^{a}$, and $\mathcal{D}_{tr}^{b}$.

1: for $\mathcal{B}^t$ in $\mathcal{D}_{test}$:
2: Update prediction records.
3: Obtain a new boundary $h^{(t)}(\cdot)$
4: Greedy Rank $\{h_{sorted}^{k}\}_{k=1}^{T}$.
5: Update $H^{(t)}(\cdot)$.
6: Update $H^{*(t)}(\cdot)$.
7: Perform prediction using $H^{*(t)}(\cdot)$.

Evaluation Metrics. Since our method is designed to achieve a high recall for known classes, we use *TNR@TPR95* as the primary evaluation metric. Compared with threshold-agnostic metrics such as *AUROC* and *AUPR*, *TNR@TPR95* better reflects the requirement of open-set recognition to maintain high accuracy on known classes. Specifically, we report both the *Average TNR@TPR95* and the *Last-Batch TNR@TPR95*. The former reflects the method's immediate performance, while the latter characterizes its peak performance after sufficient adaptation. Since we employ a dual-branch decision architecture for closed-set classification and unknown-class detection, respectively, our method does not compromise the performance of closed-set classification.

Table 3: Comparison experiments on three popular OSR datasets using ***VGG-9*** as the backbone. The results are reported in terms of *TNR@TPR95*.

VGG-9	**CIFAR10**		**SVHN**		**CIFAR100**	
	Last	**Avg**	**Last**	**Avg**	**Last**	**Avg**
Softmax	24.9	24.9	52.4	52.4	35.3	35.3
RTL *(CVPR'2024)*	30.6	27.1	44.8	44.5	34.8	34.4
OODD *(CVPR'2025)*	25.5	24.3	60.1	60.2	36.9	25.6
AdaODD *(TPAMI'2026)*	37.2	36.5	63.5	62.9	47.0	43.6
DCAC *(AAAI'2026)*	34.4	33.8	58.5	58.6	52.9	52.4
TTEM	45.1	40.7	68.4	63.4	60.1	55.5
Adaptive TTEM	**45.1**$_{(+7.9)}$	**42.4**$_{(+5.9)}$	**68.4**$_{(+4.9)}$	**64.4**$_{(+1.5)}$	**60.1**$_{(+7.2)}$	**56.7**$_{(+4.3)}$
Fully-Supervised	57.0		81.0		63.8	

Table 4: Comparison experiments on cross-datasets using ***VGG-9*** as the backbone. The results are reported in terms of *TNR@TPR95*.

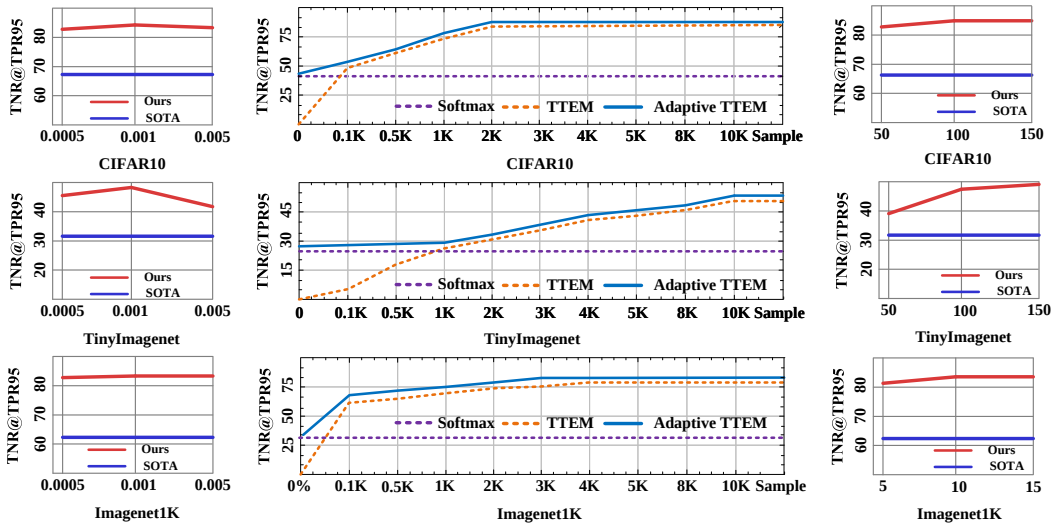
VGG-9	**CIFAR10-SVHN**		**SVHN-CIFAR10**	
	Last	**Avg**	**Last**	**Avg**
Softmax	62.2	62.2	70.4	70.4
RTL *(CVPR'2024)*	95.8	96.5	78.5	77.9
OODD *(CVPR'2025)*	97.6	93.0	83.3	80.4
AdaODD *(TPAMI'2026)*	98.4	98.2	83.9	82.8
DCAC *(AAAI'2026)*	80.5	79.5	81.5	80.5
TTEM	99.4	99.2	85.7	84.8
Adaptive TTEM	**99.4**$_{(+1.0)}$	**99.3**$_{(+1.1)}$	**85.7**$_{(+2.4)}$	**85.1**$_{(+2.3)}$
Fully-Supervised	100.0		93.5	

4.2 Main Results

CLIP-based Evaluation (Table 1). We evaluate our method with CLIP backbone on three representative datasets—CIFAR-10, TinyImageNet, and ImageNet-1K, covering small-, medium-, and large-scale settings, respectively. Across all three benchmarks, our method closes more than 50% of the performance gap between TOSR methods and the fully-supervised method. Remarkably, on ImageNet-1K, our method even surpasses the supervised Kernel SVM. We attribute this result to two factors. First, our greedy optimization prioritizes known classes, whereas the Kernel SVM treats known and unknown samples symmetrically during supervised training. Second, the strong feature extractor of CLIP, together with the clarity of ImageNet-1K, leads to well-separated known and unknown distributions in the feature space. This favorable representation geometry provides an ideal setting for our method to fully release its potential.

Resnet101-based Evaluation (Table 2). We evaluate our method using ImageNet-pretrained ResNet-101 backbone on three datasets: CUB, AWA, and aPaY. On AWA and aPaY, our method closes approximately 50% of the performance gap, consistent with the gains observed with the CLIP-based Evaluation. This demonstrates that the advantage of our approach can also generalize to conventional representations. On CUB, however, the improvement is relatively limited. We argue that the few-shot setting and fine-grained discrepancy in CUB make boundary estimation more challenging. This difficulty is also reflected in the limited performance of the fully-supervised method.

VGG9-based Evaluation (Table 3, Table 4)). We evaluate our method with a VGG-9 backbone trained from scratch using known-class samples. On CIFAR-10, our method closes approximately 40% of the performance gap, while on CIFAR-100, it achieves comparable performance to that of the fully-supervised method. The improvement on SVHN is relatively limited. We attribute this to the unique challenges of SVHN, where an image may contain multiple digits, introducing visual interference for recognition. Moreover, we further conduct cross-dataset evaluation by alternating between CIFAR10 and SVHN. Due to the substantial distribution shift between the two datasets, our method approaches a performance plateau.



(a) γ (b) Learning Process (c) Replay Size

Figure 4: (a) Sensitivity analysis of γ. (b) Analysis of the adaptive decision function. The figure shows the performance at each time step during adaptation. (c) Sensitivity analysis of replay size.

4.3 Analyses and Discussions

Analyses on γ (Fig. 4a). We observe that γ is relatively robust around 0.001 across different benchmarks and backbones. Larger values of γ will rapidly increase the error rate on known classes, whereas smaller values of γ will gradually increase the error rate on unknown classes.

Analyses on replay dataset (Fig. 4c). We observe that larger replay sets lead to more accurate boundary estimation, but at the cost of slower adaptation. A reasonable choice is to select a replay set size when the marginal benefit of doing so becomes small. Among the benchmarks, TinyImageNet exhibits the slowest convergence, which we attribute to its low-resolution images and label noise, resulting in a less compact distribution.

Analyses on adaptive detector (Fig. 4b). We argue that the prior unknown-class detector and the TTA unknown-class detector are complementary: The prior detector provides superior performance during the initial stage of test-time adaptation. As test-time adaptation progresses, the TTA detector gradually improves and eventually surpasses the prior detector, achieving peak performance after sufficient adaptation to the data stream. Our proposed adaptive detector combines the strengths of both detectors by automatically switching between them. It helps the TTA detector navigate unstable periods.

Limitations. Our method adapts relatively slowly compared with existing approaches. This is primarily because our method involves test-time error optimization process, whereas existing methods perform direct predictions based on statistics or similarity measures. Developing a more efficient optimization strategy for the proposed method is therefore a potential direction for future work.

5 Conclution

This paper realizes explicit test-time error minimization for open-set recognition. Specifically, we uncover that the challenging objective of Time Error Minimization can be transformed to the tractable objective of Negative Prediction Maximization. Based on that, we provide a practical implementation of Negative Prediction Maximization termed Discrete-Boundary Ensemble. Besides, we design an adaptive detector that combines the complementary strengths of a prior detector and a TTA detector, enabling stable performance throughout the adaptation process. Extensive experiments demonstrate that our method successfully closes the performance gap between TOSR and supervised methods.

AI Use Statement. No generative AI tools were used in the preparation of this manuscript.

Reproducibility Statement. We provide the experimental settings, evaluation protocols, and implementation configurations and hyperparameters in the main paper Section 4.1. We will publicly release the source code upon acceptance.

REFERENCES

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Xuefeng Du, Zhaoning Wang, Mu Cai, and Yixuan Li. Vos: Learning what you don't know by virtual outlier synthesis. *arXiv preprint arXiv:2202.01197*, 2022.

Xuefeng Du, Yiyou Sun, Jerry Zhu, and Yixuan Li. Dream the impossible: Outlier imagination with diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

Ke Fan, Tong Liu, Xingyu Qiu, Yikai Wang, Lian Huai, Zeyu Shangguan, Shuang Gou, Fengjian Liu, Yuqian Fu, Yanwei Fu, et al. Test-time linear out-of-distribution detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23752–23761. IEEE, 2024.

Yimin Fu, Zhunga Liu, and Zicheng Wang. Logit prototype learning with active multimodal representation for robust open-set recognition. *Science China Information Sciences*, 67(6):162204, 2024.

Zhengqing Gao, Xu-Yao Zhang, and Cheng-Lin Liu. Unified entropy optimization for open-set test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23975–23984, 2024.

Chuanxing Geng, Sheng-Jun Huang, and Songcan Chen. Recent advances in open set recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3614–3631, 2021.

Guosong Jiang, Pengfei Zhu, Bing Cao, Dongyue Chen, and Qinghua Hu. Unknown support prototype set for open set recognition. *International Journal of Computer Vision*, pp. 1–18, 2025.

Wenyu Jiang, Hao Cheng, MingCai Chen, Chongjun Wang, and Hongxin Wei. Dos: Diverse outlier sampling for out-of-distribution detection. In *International Conference on Learning Representations*, 2024.

Steven M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*, volume I. Prentice Hall, 1993.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Advances in Neural Information Processing Systems*, 31, 2018.

Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer, New York, 2nd edition, 1998.

Chaohua Li, Enhao Zhang, Chuanxing Geng, and Songcan Chen. Unlocking better closed-set alignment based on neural collapse for open-set recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 18227–18235, 2025.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *Advances in neural information processing systems*, 33:21464–21475, 2020.

Poojan Oza and Vishal M Patel. C2ae: Class conditioned auto-encoder for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2307–2316, 2019.

Xin Sun, Zhenning Yang, Chi Zhang, Keck-Voon Ling, and Guohao Peng. Conditional gaussian distribution learning for open set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13480–13489, 2020.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Yanqi Wu, Qichao Chen, Runhe Lai, Xinhua Lu, Jia-Xin Zhuang, Zhilin Zhao, Weishi Zheng, and Ruixuan Wang. Dcac: Dynamic class-aware cache creates stronger out-of-distribution detectors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 10817–10825, 2026.

Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, Qing Yang, and Cheng-Lin Liu. Convolutional prototype network for open set recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 44(05):2358–2370, 2022.

Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *International Journal of Computer Vision*, 132(12):5635–5662, 2024.

Yifeng Yang, Lin Zhu, Zewen Sun, Hengyu Liu, Qinying Gu, and Nanyang Ye. Oodd: Test-time out-of-distribution detection with dynamic dictionary. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 30630–30639. IEEE, 2025.

Wenlong Yu, Ruonan Liu, Dongyue Chen, and Qinghua Hu. Explainability enhanced object detection transformer with feature disentanglement. *IEEE Transactions on Image Processing*, 33: 6439–6454, 2024.

Hongjie Zhang, Ang Li, Jie Guo, and Yanwen Guo. Hybrid models for open set recognition. In *European Conference on Computer Vision*, pp. 102–117, 2020.

Yabin Zhang, Maya Varma, Yunhe Gao, Jean-Benoit Delbrouck, Jiaming Liu, Chong Wang, and Curtis Langlotz. Activation matters: Test-time activated negative labels for ood detection with vision-language models. *arXiv preprint arXiv:2603.25250*, 2026.

YiFan Zhang, Xue Wang, Tian Zhou, Kun Yuan, Zhang Zhang, Liang Wang, and Rong Jin. Model-free test time adaptation for out-of-distribution detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Da-Wei Zhou, Han-Jia Ye, and De-Chuan Zhan. Learning placeholders for open-set recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401–4410, 2021.

Jianing Zhu, Geng Yu, Jiangchao Yao, Tongliang Liu, Gang Niu, Masashi Sugiyama, and Bo Han. Diversified outlier exposure for out-of-distribution detection via informative extrapolation. *Advances in Neural Information Processing Systems*, 2023.

Wenjie Zhu, Yabin Zhang, Xin Jin, Wenjun Zeng, and Lei Zhang. Ants: Adaptive negative textual space shaping for ood detection via test-time mllm understanding and reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20–30, 2026.