STORAGE-EFFICIENT OUT-OF-DISTRIBUTION DETECTION WITH CLASS-WISE PROTOTYPES

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks can make highly confident errors on out-of-distribution (OOD) inputs, motivating reliable OOD detection. Nearest-neighbor methods incur substantial storage and inference costs by retaining massive training feature banks. We propose CPD, a distance-based post-hoc method that replaces these banks with compact class-wise prototypes without model retraining. CPD applies spherical k-means clustering to ℓ_2-normalized features within each class and computes detection scores by aggregating distances to the nearest prototypes. Across CIFAR-10, CIFAR-100, ImageNet-200, and ImageNet-1K, CPD achieves Near- and Far-OOD AUROC comparable to or better than KNN while requiring substantially less feature storage. On ImageNet-1K with ResNet-50, Swin-T, and ViT-B/16, a single-prototype-per-class configuration reduces feature storage by approximately 99.92% relative to the full KNN feature bank while improving Near- and Far-OOD AUROC by 3.70–4.80 and 1.46–5.00 percentage points, respectively. Further analysis shows that increasing the number of prototypes does not necessarily improve detection performance, as its benefit depends on the aggregation neighborhood size, highlighting the importance of jointly selecting prototype count and scoring strategy.

1 INTRODUCTION

Deep neural networks are widely used for image recognition, yet deployment inputs may belong to categories absent from the training data. For such out-of-distribution (OOD) inputs, models can still produce highly confident but incorrect predictions, making classification confidence alone unreliable for identifying whether an input conforms to the training distribution (Nguyen et al., 2015; Hendrycks & Gimpel, 2017). OOD detection distinguishes in-distribution (ID) from OOD samples. Post-hoc methods construct detection scores from pretrained model outputs or features without retraining, making them convenient to integrate into existing systems. Beyond detection performance, the additional storage and inference costs introduced by these methods are also important considerations for practical deployment.

Feature-distance-based OOD detection methods score inputs by measuring their proximity to ID reference representations, which differ in modeling capability and computational cost. The Mahalanobis method characterizes ID feature distributions using class means and a shared covariance matrix, avoiding the need to retain individual training features (Lee et al., 2018). Mahalanobis++ further shows that feature normalization can substantially affect the performance of such distance-based detectors (Müller & Hein, 2025). In contrast, KNN directly measures proximity between test features and a retained bank of training features without assuming a specific parametric distribution (Sun et al., 2022). Retaining N training features of dimension d requires $\mathcal{O}(Nd)$ storage, while exhaustive query-to-bank distance computation requires $\mathcal{O}(Nd)$ operations. Although actual search cost depends on indexing and implementation, large feature banks remain an important source of storage and inference overhead.

Class prototypes provide an alternative reference representation. For example, POT detects OOD inputs using the optimal transport cost between test inputs and ID prototypes (Ke et al., 2025). We investigate a complementary question: how many prototypes per class are needed to maintain competitive OOD detection performance while reducing storage and distance computation costs? In a normalized feature space, a single prototype summarizes a class with one representative direction,

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.

whereas multiple prototypes can capture different within-class regions. Selecting nearby prototypes further allows scoring to focus on ID representations close to a test sample. However, a more detailed ID representation does not necessarily improve OOD detection, because better approximation of within-class features does not imply greater separability between ID and OOD scores. Our prototype-budget experiments show inconsistent gains from increasing the number of prototypes across datasets: some settings achieve competitive performance with a single prototype, whereas others benefit from multiple prototypes. This motivates treating the prototype budget as a task-dependent representation scale.

Motivated by these observations, we propose CPD, a distance-based post-hoc framework that replaces a large reference feature bank with compact class prototypes. CPD applies ℓ_2 normalization to ID training features from a pretrained model and constructs n unit-norm prototypes per class using within-class spherical k-means. At inference time, CPD computes squared Euclidean distances between a normalized test feature and the prototypes of each class, averages the distances to the k nearest prototypes within each class, and uses the negative minimum class-wise average distance as the ID confidence score. Here, n controls the size of the reference representation, while k controls the number of neighboring prototypes contributing to each class score. The framework therefore accommodates both single-center and multi-prototype settings. For C classes, prototype storage is $\mathcal{O}(Cnd)$. Once the prototypes are constructed, inference no longer requires the original training feature bank, and the pretrained model remains unchanged.

We evaluate CPD on CIFAR-10, CIFAR-100, ImageNet-200, and ImageNet-1K under the OpenOOD v1.5 protocol, comparing Near-OOD and Far-OOD detection performance, detector storage, and computational costs (Zhang et al., 2024). On ImageNet-1K with ResNet-50, CPD reduces detector storage from 10,009.12 MiB to 7.81 MiB relative to KNN, corresponding to a 99.92% reduction, while Near-OOD and Far-OOD AUROC increase by 4.40 and 5.00 percentage points, respectively. Figure 1 compares CPD's performance–storage trade-off with other feature-based detectors. Further analyses examine prototype budgets, aggregation neighborhoods, and clustering versus random sampling under the same reference-vector budget.

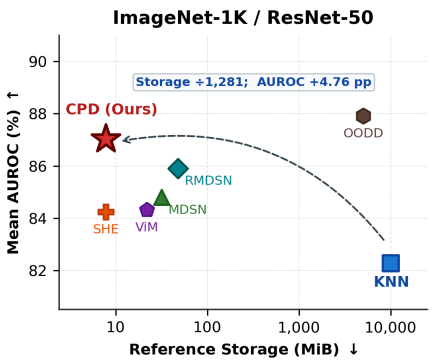


Figure 1: AUROC versus storage (log scale): ImageNet-1K, ResNet-50.

Our main contributions are as follows:

- **We propose CPD, a framework centered on an adjustable class-prototype budget.** CPD performs OOD detection using compact class prototypes and within-class aggregation of distances to nearby prototypes, reducing reliance on large reference feature banks without retraining the classification model.
- **We analyze how the prototype budget and aggregation neighborhood affect OOD detection.** Experiments reveal dataset-dependent benefits of single-center and multi-prototype representations and show that more prototypes do not guarantee continued performance gains.
- **We evaluate detection performance and costs on four image recognition benchmarks.** Through comparisons with existing methods, cost measurements, and ablation studies, we show that CPD maintains competitive detection performance while reducing reference storage relative to KNN and characterize its performance–cost trade-offs against existing approaches.

2 Related Work

Feature-Based Post-Hoc OOD Detection. Feature-based post-hoc methods construct detection scores from the internal representations of pretrained models without retraining the classifier. Beyond directly computing feature distances, some methods modify feature activations or combine

feature representations with classifier outputs. ReAct truncates large activations to suppress excessive feature responses (Sun et al., 2021), while ASH removes a portion of the activations at inference time and rescales those retained to improve ID–OOD separation (Djurisic et al., 2023). ViM constructs a virtual logit from the residual of a feature relative to an ID principal subspace and combines it with the original classifier logits, integrating information from both feature and output spaces (Wang et al., 2022).

Distance-based methods instead assess the proximity of test features to ID reference representations. Mahalanobis detection models class-conditional features using class means and a shared covariance matrix, producing a covariance-aware distance score (Lee et al., 2018). Relative Mahalanobis distance additionally fits a background Gaussian over all ID classes and adjusts the class-conditional distance using the background distance to improve Near-OOD detection (Ren et al., 2021). Mahalanobis++ identifies feature-norm variation as a factor affecting detection performance and demonstrates the benefit of applying ℓ_2 normalization before estimating feature statistics (Müller & Hein, 2025). These methods summarize training features through statistical representations, with the covariance matrix contributing to both the distance metric and detector storage.

Other approaches retain reference samples and construct scores directly from pairwise distances. KNN uses nearest-neighbor distances between test features and retained training features without assuming a specific parametric distribution (Sun et al., 2022). Pairwise OOD detection (POD) averages squared Euclidean distances to sampled reference features within each class and uses the negative minimum class-wise average distance as the ID score, avoiding covariance estimation (Tajwar et al., 2021). These methods differ in how reference features are organized and distances are aggregated, while the number of retained features affects both storage and distance-computation costs.

Prototype-Based OOD Detection. Prototype-based methods summarize ID data using representative vectors against which test inputs are scored. SHE adopts a store-then-compare approach derived from modern Hopfield energy, representing each class with a single stored pattern obtained by averaging its training features, thereby enabling compact and computationally efficient detection (Zhang et al., 2023). POT likewise constructs class-mean prototypes but measures distributional discrepancy using optimal transport costs between test inputs and the prototypes. Its final score additionally incorporates transport costs to virtual outliers generated by linear extrapolation from class prototypes and the average test representation (Ke et al., 2025). POT explicitly uses prototypes to reduce the computational burden of optimal transport, showing that compact representations and computational efficiency are already considered in existing prototype-based detection methods.

Other studies focus on the information captured by prototypes and their reliability. SPROD uses classification-aware prototype construction and group prototype refinement to mitigate the influence of spurious correlations on class representations (Zohrabi et al., 2025). Prototype Fusion constructs normalized class-mean prototypes at multiple network layers and combines layer-wise cosine-similarity scores, exploiting complementary information beyond the penultimate layer (Gul et al., 2026). These methods improve prototype-based detection through prototype refinement, scoring design, and cross-layer information integration.

CPD uses normalized penultimate features and retains POD's within-class distance aggregation, replacing sampled features with spherical k-means prototypes. Prototype count and aggregation neighborhood separately control representation size and scoring locality and are selected on validation data. Random-sampling controls at matched reference-vector budgets isolate the benefit of clustering.

3 Class-wise Prototype Distance

We propose Class-wise Prototype Distance (CPD) for post-hoc semantic OOD detection. As shown in Figure 2, CPD consists of offline prototype construction and online scoring. ID training features are extracted from a frozen pretrained classifier, ℓ_2-normalized, and clustered independently within each class using spherical k-means. At inference, test inputs undergo the same feature pipeline, and distances to nearby class-wise prototypes are aggregated into an ID confidence score. No retraining or modification of the pretrained classifier is required.

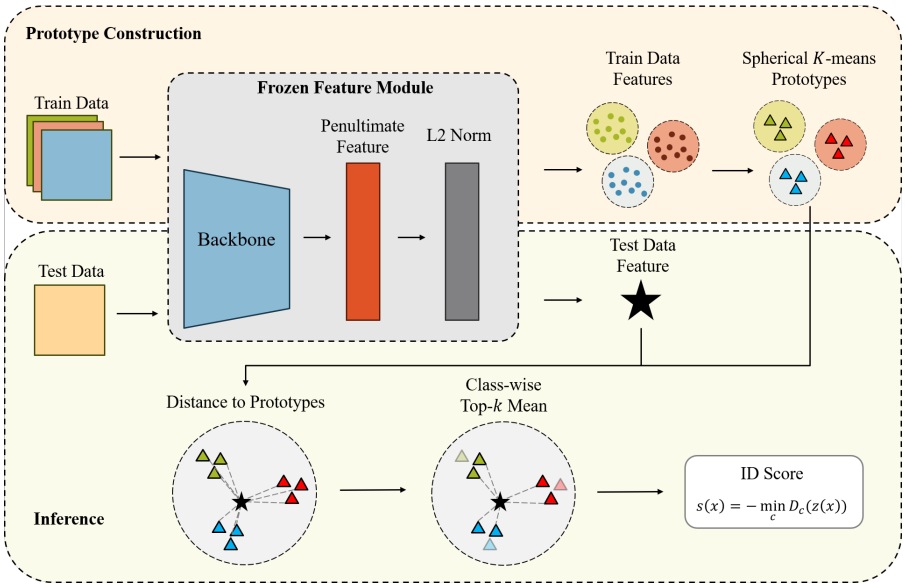


Figure 2: Overview of CPD. Spherical k-means constructs class-wise prototypes from normalized features of a frozen classifier. At inference, CPD averages the k smallest squared Euclidean distances within each class and uses the negative minimum class-wise average as the ID score. Shown: $n = 3$, $k = 2$; faded prototypes are excluded.

3.1 Problem Setup

Let $\mathcal{D}_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the ID training set, where $y_i \in \{1, \ldots, C\}$. Given a fixed feature extractor $f_\theta : \mathcal{X} \to \mathbb{R}^d$ corresponding to the penultimate representation of a pretrained classifier, CPD uses the following normalized feature for $f_\theta(x) \neq \mathbf{0}$:

$$z(x) = \frac{f_\theta(x)}{\|f_\theta(x)\|_2}. \tag{1}$$

Training and test inputs use the same feature extraction and normalization procedure. We write $z_i = z(x_i)$ for normalized training features.

We consider semantic OOD detection, where a test input belongs either to one of the C ID classes or to a semantic class outside the ID label set. CPD assigns an ID confidence score $s(x)$, with higher values indicating greater proximity to the ID reference representation. Given a threshold τ, an input is ID if $s(x) \geq \tau$ and OOD otherwise.

3.2 Class-wise Prototype Construction

For class c, let $\mathcal{I}_c = \{i : y_i = c\}$. CPD constructs n prototypes per class, $\mathcal{P}_c = \{p_{c1}, \ldots, p_{cn}\}$, by independently applying spherical k-means:

$$\min_{\{p_{cj}\}_{j=1}^{n}} \sum_{i \in \mathcal{I}_c} \min_{1 \leq j \leq n} \|z_i - p_{cj}\|_2^2, \qquad \text{s.t. } \|p_{cj}\|_2 = 1. \tag{2}$$

Because features and prototypes have unit norm, nearest-prototype assignment under squared Euclidean distance is equivalent to maximum inner-product assignment.

Throughout this paper, n denotes the number of prototypes per class, while k denotes the number aggregated during scoring. We use $1 \leq k \leq n \leq \min_c |\mathcal{I}_c|$. The budget n controls representation granularity: $n = 1$ yields the unit direction of the class mean when nonzero, whereas $n > 1$ can represent multiple within-class regions. The objective minimizes within-class feature-to-prototype distortion rather than directly optimizing ID–OOD separability.

After construction, CPD retains only the class-wise prototype sets and discards the original training feature bank. All prototypes are weighted equally during scoring, regardless of cluster size.

3.3 Class-wise Prototype Scoring

For a test input x, let $z = z(x)$ and define

$$\delta_{cj}(z) = \|z - p_{cj}\|_2^2. \tag{3}$$

Let $\mathcal{N}_{c,k}(z)$ denote the indices of the k nearest prototypes to z within class c. The class distance and CPD score are

$$D_c(z) = \frac{1}{k} \sum_{j \in \mathcal{N}_{c,k}(z)} \delta_{cj}(z), \qquad s_{\mathrm{CPD}}(x) = -\min_{1 \le c \le C} D_c(z(x)). \tag{4}$$

Prototype selection is performed independently within each class, so scoring is not restricted to the classifier's predicted label. The parameters n and k separately control representation size and aggregation neighborhood size: smaller k emphasizes nearer prototypes, whereas $k = n$ uses the full prototype set of each class.

Unlike standard KNN, which searches an unpartitioned bank of individual training features, CPD uses class-wise clustered prototypes and aggregates distances only within each class. When $k = 1$, Eq. (4) reduces to

$$s_{\mathrm{CPD}}(x) = -\min_{c,j} \|z - p_{cj}\|_2^2.$$

3.4 Geometric Interpretation and Computational Cost

Relationship to Cosine Similarity. For unit vectors,

$$\|z - p_{cj}\|_2^2 = 2 - 2z^\top p_{cj}. \tag{5}$$

Thus, nearest prototypes under squared Euclidean distance are also those with the highest cosine similarities, and

$$s_{\mathrm{CPD}}(x) = -2 + 2 \max_{1 \le c \le C} \left[\frac{1}{k} \sum_{j \in \mathcal{N}_{c,k}(z)} z^\top p_{cj} \right]. \tag{6}$$

CPD therefore measures average directional alignment with nearby prototypes within each class.

Full-Prototype Aggregation ($k = n$). Define

$$\bar{p}_c = \frac{1}{n} \sum_{j=1}^{n} p_{cj}.$$

When $k = n$,

$$s_{\mathrm{CPD}}(x) = -2 + 2 \max_{1 \le c \le C} z^\top \bar{p}_c. \tag{7}$$

Hence, Eq. (7) can be evaluated exactly using one unnormalized mean prototype vector per class; renormalizing it would change the score. In general, $\bar{p}_c$ is not unit norm and need not equal the empirical mean of the original training features because the prototypes are equally weighted.

Its norm reflects directional concentration:

$$\frac{1}{n}\sum_{j=1}^{n}\|p_{cj}-\bar{p}_c\|_2^2 = 1-\|\bar{p}_c\|_2^2. \tag{8}$$

Thus, full aggregation depends on both query alignment and the directional concentration of the equally weighted prototype set.

Storage and Scoring Costs. CPD stores Cn prototypes of dimension d, requiring $\mathcal{O}(Cnd)$ persistent storage. Computing all prototype distances for one query also requires $\mathcal{O}(Cnd)$ operations. If full sorting is used, within-class selection requires $\mathcal{O}(Cn\log n)$ operations; partial selection can avoid full sorting. Distance averaging and cross-class comparison require $\mathcal{O}(Ck)$ and $\mathcal{O}(C)$ operations, respectively.

For a reference bank containing N_{ref} feature vectors with the same dimensionality and precision, the prototype-to-reference feature-vector storage ratio is Cn/N_{ref}, excluding indexing overhead. When $k=n$ and Eq. (7) is implemented using only the mean prototype vector of each class, both persistent storage and per-query scoring cost reduce to $\mathcal{O}(Cd)$.

Prototype construction is a one-time offline cost. If spherical k-means performs T assignment-and-update iterations across all classes, its dominant complexity is approximately $\mathcal{O}(TNnd)$, excluding the one-time feature extraction cost.

4 Experiments

4.1 Experimental Setup

Datasets. Following the OpenOOD v1.5 evaluation protocol (Zhang et al., 2024), we use CIFAR-10, CIFAR-100 (Krizhevsky, 2009), ImageNet-200, and ImageNet-1K (Deng et al., 2009) as ID datasets. For the CIFAR benchmarks, Near-OOD datasets comprise the other CIFAR dataset and Tiny ImageNet (Le & Yang, 2015), while Far-OOD datasets include MNIST (Deng, 2012), SVHN (Netzer et al., 2011), Textures (Cimpoi et al., 2014), and Places365 (Zhou et al., 2018). For the ImageNet benchmarks, Near-OOD datasets comprise SSB-hard (Vaze et al., 2022) and NINCO (Bitterwolf et al., 2023), while Far-OOD datasets include iNaturalist (Van Horn et al., 2018), Textures, and OpenImage-O (Wang et al., 2022).

Models and Evaluation. We use ResNet-18 checkpoints provided by OpenOOD for CIFAR-10, CIFAR-100, and ImageNet-200, and ResNet-50 (He et al., 2016) with the official TorchVision `IMAGENET1K_V1` weights for ImageNet-1K. All models remain frozen throughout evaluation. For CIFAR-10, CIFAR-100, and ImageNet-200, we report mean $\pm$ standard deviation across three independently trained checkpoints with different random seeds. For ImageNet-1K, we report results from a single official checkpoint. Detection performance is evaluated using AUROC and FPR95, with unweighted averages computed separately over Near-OOD and Far-OOD datasets. Supplementary results with Swin-T (Liu et al., 2021) and ViT-B/16 (Dosovitskiy et al., 2021) are provided in Appendix B.2.

Baselines and Hyperparameter Selection. We compare MSP (Hendrycks & Gimpel, 2017), EBO (Liu et al., 2020), ReAct (Sun et al., 2021), MDSN and RMDSN (Müller & Hein, 2025), ViM (Wang et al., 2022), KNN (Sun et al., 2022), POD (Tajwar et al., 2021), SHE (Zhang et al., 2023), and OODD (Yang et al., 2025) in Tables 1 and 2. CPD constructs prototypes exclusively from ID training data. The prototype count n and aggregation neighborhood size k are selected using OpenOOD's hyperparameter search procedure based on AUROC on ID and OOD validation data and are then fixed for testing. Search spaces and detailed settings are provided in Appendix A.

All detection and cost results in Tables 1–4 and Appendices B and C were re-measured in our evaluation environment. Our reimplementation of OODD (Yang et al., 2025) retains its test-time streaming-state updates; detailed settings are provided in Appendix A.3. Tables 1 and 2 include methods with results on all four ID datasets.

Table 1: OOD detection on CIFAR-10 and CIFAR-100. AUROC↑ / FPR95↓ (%). Bold: best; underline: second-best.

Method	CIFAR-10		CIFAR-100	
	Near-OOD	Far-OOD	Near-OOD	Far-OOD
MSP	88.04$_{\pm 0.31}$ / 48.15$_{\pm 4.94}$	90.75$_{\pm 0.52}$ / 31.69$_{\pm 2.17}$	80.21$_{\pm 0.16}$ / <u>54.84</u>$_{\pm 0.71}$	77.83$_{\pm 0.55}$ / 58.52$_{\pm 1.37}$
EBO	87.61$_{\pm 0.57}$ / 61.63$_{\pm 6.00}$	91.25$_{\pm 1.10}$ / 41.46$_{\pm 6.24}$	<u>80.82</u>$_{\pm 0.11}$ / 55.77$_{\pm 0.78}$	79.83$_{\pm 0.76}$ / 56.47$_{\pm 1.74}$
ReAct	86.97$_{\pm 0.87}$ / 63.94$_{\pm 8.93}$	90.24$_{\pm 2.10}$ / 45.51$_{\pm 11.60}$	80.69$_{\pm 0.08}$ / 56.47$_{\pm 0.51}$	80.45$_{\pm 0.61}$ / 54.07$_{\pm 1.92}$
MDSN	88.19$_{\pm 0.94}$ / 46.62$_{\pm 1.91}$	93.49$_{\pm 0.24}$ / 26.06$_{\pm 1.05}$	74.18$_{\pm 0.28}$ / 67.48$_{\pm 0.70}$	<u>83.22</u>$_{\pm 1.06}$ / 51.59$_{\pm 2.19}$
RMDSN	89.79$_{\pm 0.34}$ / 36.21$_{\pm 2.80}$	92.05$_{\pm 0.15}$ / 24.61$_{\pm 0.57}$	80.42$_{\pm 0.11}$ / 55.14$_{\pm 0.56}$	82.27$_{\pm 0.15}$ / 53.23$_{\pm 0.31}$
ViM	88.73$_{\pm 0.35}$ / 44.72$_{\pm 2.64}$	<u>93.50</u>$_{\pm 0.29}$ / 24.98$_{\pm 0.60}$	75.04$_{\pm 0.17}$ / 62.61$_{\pm 0.30}$	81.64$_{\pm 0.77}$ / <u>50.75</u>$_{\pm 1.22}$
KNN	<u>90.67</u>$_{\pm 0.25}$ / **33.91**$_{\pm 0.47}$	92.99$_{\pm 0.17}$ / 24.21$_{\pm 0.56}$	80.16$_{\pm 0.19}$ / 61.17$_{\pm 0.16}$	82.43$_{\pm 0.21}$ / 53.61$_{\pm 0.31}$
POD	88.76$_{\pm 0.61}$ / 40.98$_{\pm 2.46}$	91.30$_{\pm 0.22}$ / 30.13$_{\pm 1.37}$	74.37$_{\pm 0.18}$ / 73.45$_{\pm 0.50}$	78.04$_{\pm 1.83}$ / 60.22$_{\pm 1.96}$
SHE	81.58$_{\pm 0.61}$ / 79.46$_{\pm 4.26}$	85.37$_{\pm 1.73}$ / 66.34$_{\pm 7.21}$	78.83$_{\pm 0.20}$ / 59.38$_{\pm 0.38}$	76.98$_{\pm 1.44}$ / 64.00$_{\pm 3.33}$
OODD	**90.95**$_{\pm 0.17}$ / 35.03$_{\pm 0.90}$	**95.85**$_{\pm 0.19}$ / **16.86**$_{\pm 0.56}$	**83.36**$_{\pm 0.24}$ / **53.89**$_{\pm 0.14}$	**93.97**$_{\pm 0.09}$ / **26.05**$_{\pm 0.34}$
CPD	90.59$_{\pm 0.26}$ / <u>34.89</u>$_{\pm 0.46}$	93.03$_{\pm 0.18}$ / <u>23.97</u>$_{\pm 0.64}$	80.69$_{\pm 0.13}$ / 58.14$_{\pm 0.76}$	82.12$_{\pm 0.31}$ / 52.02$_{\pm 0.19}$

Table 2: OOD detection on ImageNet-200 and ImageNet-1K. AUROC↑ / FPR95↓ (%). Bold: best; underline: second-best.

Method	ImageNet-200		ImageNet-1K	
	Near-OOD	Far-OOD	Near-OOD	Far-OOD
MSP	83.29$_{\pm 0.07}$ / 54.93$_{\pm 0.44}$	90.13$_{\pm 0.11}$ / 35.39$_{\pm 0.46}$	76.07 / **65.72**	85.30 / 51.44
EBO	82.41$_{\pm 0.09}$ / 60.31$_{\pm 0.69}$	90.84$_{\pm 0.27}$ / 34.83$_{\pm 1.58}$	76.03 / 68.50	89.50 / 38.39
ReAct	81.77$_{\pm 1.18}$ / 62.64$_{\pm 2.58}$	92.28$_{\pm 0.69}$ / 28.48$_{\pm 1.16}$	<u>77.40</u> / 66.92	93.67 / 26.36
MDSN	81.68$_{\pm 0.37}$ / 60.51$_{\pm 1.30}$	94.49$_{\pm 0.31}$ / <u>21.65</u>$_{\pm 0.90}$	72.02 / 72.10	93.32 / 25.17
RMDSN	<u>83.45</u>$_{\pm 0.23}$ / <u>54.41</u>$_{\pm 0.65}$	90.22$_{\pm 0.39}$ / 31.75$_{\pm 0.92}$	**78.54** / <u>66.18</u>	90.82 / 38.13
ViM	78.59$_{\pm 0.30}$ / 59.35$_{\pm 0.92}$	91.26$_{\pm 0.22}$ / 27.22$_{\pm 0.43}$	71.82 / 71.70	92.63 / 24.76
KNN	81.46$_{\pm 0.22}$ / 60.31$_{\pm 0.61}$	93.15$_{\pm 0.28}$ / 27.30$_{\pm 0.90}$	70.60 / 71.27	90.05 / 34.41
POD	71.07$_{\pm 0.87}$ / 70.96$_{\pm 0.62}$	82.33$_{\pm 0.34}$ / 44.07$_{\pm 1.14}$	55.29 / 82.77	71.93 / 60.02
SHE	80.10$_{\pm 0.34}$ / 66.87$_{\pm 1.08}$	89.79$_{\pm 0.75}$ / 42.13$_{\pm 1.53}$	74.13 / 72.62	90.97 / 41.45
OODD	**86.34**$_{\pm 0.10}$ / **52.27**$_{\pm 0.42}$	**95.51**$_{\pm 0.03}$ / **21.33**$_{\pm 0.26}$	76.35 / 68.33	**95.63** / **19.25**
CPD	81.88$_{\pm 0.40}$ / 63.14$_{\pm 0.44}$	<u>94.56</u>$_{\pm 0.28}$ / 22.49$_{\pm 0.80}$	75.00 / 68.87	<u>95.05</u> / <u>22.35</u>

4.2 OOD Detection Performance

CIFAR Results. Table 1 shows that CPD closely matches KNN on CIFAR-10, with Near-/Far-OOD AUROCs of 90.59%/93.03% versus 90.67%/92.99%. On CIFAR-100, CPD improves Near-OOD AUROC by 0.53 percentage points, while Far-OOD AUROC decreases by 0.31 points; Near-/Far-OOD FPR95 decrease by 3.03/1.59 points. These results indicate that class-wise prototypes preserve performance comparable to KNN's full feature bank. CPD also exceeds MDSN and RMDSN in CIFAR-10 Near-OOD AUROC, whereas OODD achieves higher Near- and Far-OOD AUROC on both CIFAR benchmarks under our evaluation settings.

ImageNet Results. On ImageNet-200 (Table 2), CPD achieves 94.56% Far-OOD AUROC, exceeding KNN by 1.41 percentage points and reducing FPR95 from 27.30% to 22.49%. Near-OOD AUROC improves by 0.42 points, but FPR95 increases by 2.83 points, showing that better overall ranking need not improve performance at 95% ID acceptance. On ImageNet-1K, validation selects one prototype per class. CPD achieves Near-/Far-OOD AUROCs of 75.00%/95.05%, exceeding KNN by 4.40/5.00 points and reducing FPR95 by 2.40/12.06 points. CPD surpasses MDSN and RMDSN in Far-OOD AUROC but remains below RMDSN in Near-OOD performance. Section 4.3 examines the associated storage and computational costs.

Complete comparisons with Swin-T and ViT-B/16 are provided in Tables B5 and B6 in Appendix B.2. On both backbones, CPD achieves higher Near-OOD and Far-OOD AUROC than KNN but remains below MDSN and RMDSN, indicating that its relative advantage varies across backbone architectures.

4.3 Computational and Storage Costs

Evaluation Protocol. Table 3 compares storage and inference costs on CIFAR-100 and ImageNet-1K, where CPD uses a larger prototype set and one prototype per class, respectively. All methods use validation-selected hyperparameters. Storage measures the retained numerical state after offline setup, excluding the backbone and runtime memory. Inference time is measured at batch size 200 and includes data loading, host-to-device transfer, backbone inference, and postprocessing. KNN uses CPU-based FAISS retrieval, whereas CPD performs scoring on the GPU.

Table 3: Offline state size and inference time per sample at batch size 200. Lower is better; best values are bold. CPD is shaded.

Method	CIFAR-100		ImageNet-1K	
	State↓ (MiB)	Time↓ (ms/sample)	State↓ (MiB)	Time↓ (ms/sample)
MDSN	1.39	**0.526**	31.63	3.932
RMDSN	2.39	0.527	47.64	3.939
SHE	**0.20**	0.529	**7.81**	**3.888**
KNN	97.66	0.591	10009.12	14.508
OODD	48.83	0.544	5004.55	5.392
CPD	52.08	0.534	**7.81**	3.902

Storage Costs. On ImageNet-1K, CPD requires 7.81 MiB, reducing storage by 99.92% relative to KNN's 10,009.12 MiB and remaining below MDSN and RMDSN. SHE has the same footprint. On CIFAR-100, CPD requires 52.08 MiB, a 46.67% reduction relative to KNN but more than MDSN, RMDSN, and SHE. Thus, CPD reduces full feature-bank storage, while its advantage over compact methods depends on the prototype budget.

Inference Costs. On ImageNet-1K, CPD requires 3.902 ms per sample versus KNN's 14.508 ms, a 73.10% reduction and 3.72× speedup. On CIFAR-100, inference time decreases from 0.591 to 0.534 ms and remains comparable to MDSN, RMDSN, and SHE. CPD's clearest efficiency gains therefore arise when replacing a large full feature bank.

4.4 Detection Performance under Storage Budgets

Budget-Constrained Evaluation. We impose a per-class prototype budget $B \in \{1, 5, 10, 50, 100, 200, 300\}$. For each budget and checkpoint, (n, k) is selected by validation AUROC from configurations satisfying $n \leq B$ and then fixed for testing. Figure 3 shows average realized storage and Near- and Far-OOD AUROC on CIFAR-100 and ImageNet-200. Budget settings yielding identical configurations across all checkpoints are merged. This evaluates the complete budget-constrained selection procedure rather than the effect of prototype count alone.

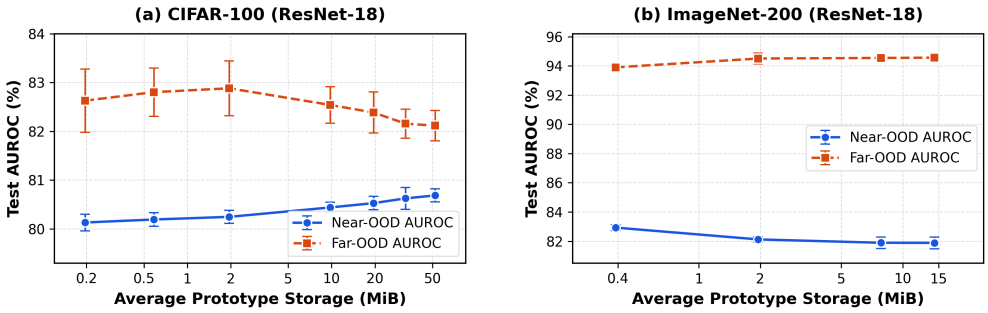


Figure 3: Prototype storage–detection performance trade-offs on CIFAR-100 and ImageNet-200. The x-axis shows average realized prototype storage on a logarithmic scale; the y-axis shows average test AUROC. Error bars indicate ±1 standard deviation across three checkpoints.

At $B = 5$, CIFAR-100 requires 0.59 MiB and achieves Near-/Far-OOD AUROCs of 80.19%/82.80%. Compared with $B = 300$ (52.08 MiB), storage decreases by 98.87%, with AUROC

changes of $-0.50/+0.68$ percentage points. On ImageNet-200, all checkpoints select $(n, k) = (5, 5)$ at $B = 5$, requiring 1.95 MiB and achieving 82.12%/94.51% AUROC. Relative to $B = 300$ (14.32 MiB; 81.88%/94.56%), storage decreases by 86.38%, while Near-OOD AUROC increases by 0.24 points and Far-OOD AUROC decreases by 0.05 points. Thus, small budgets substantially reduce storage while preserving comparable test performance on both datasets.

4.5 Ablation Studies

Table 4 evaluates input normalization, prototype construction, and aggregation on ImageNet-1K with $n = 50$, distinct from the validation-selected $n = 1$ in the main evaluation. Here, normalization refers to training features before clustering, distinct from centroid normalization within spherical k-means.

Table 4: ImageNet-1K ablations ($n = 50$). AUROC (%); best values are bold. ✓/✗: feature normalization on/off. Full CPD is shaded.

ℓ_2 Norm	Prototype	Aggregation	Near-OOD	Far-OOD
✗	Random	All-mean	55.24	71.39
✗	Random	Top-5	58.87	70.86
✗	k-means	All-mean	58.60	78.04
✗	k-means	Top-5	62.54	76.22
✓	Random	All-mean	68.60	91.78
✓	Random	Top-5	72.37	90.57
✓	k-means	All-mean	71.22	**94.16**
✓	k-means	Top-5	**75.33**	92.81

Normalization improves both Near- and Far-OOD AUROC in all comparisons. With normalized features, k-means outperforms random sampling under Top-5 aggregation by 2.96 and 2.24 percentage points, respectively. Top-5 also improves Near-OOD AUROC over All-mean by 4.11 percentage points but reduces Far-OOD AUROC by 1.35 percentage points, motivating validation-based selection of the aggregation neighborhood. Additional representation-error and aggregation analyses are provided in Appendix D.

5 Conclusion

In this paper, we proposed CPD, a post-hoc OOD detector that replaces the full training feature bank with class-wise prototypes and aggregates distances to nearby prototypes. Across four ID benchmarks, CPD achieves detection performance comparable to or better than KNN with substantially lower feature storage. On ImageNet-1K, one prototype per class reduces storage by 99.92% and yields a $3.72\times$ inference speedup over KNN under our evaluation setup. Budget-constrained experiments show that smaller prototype budgets can preserve comparable performance, while ablations highlight the benefits of feature normalization and clustered prototypes. Although CPD does not consistently outperform compact statistical methods, it offers an adjustable performance–storage trade-off. Future work includes alternative prototype construction criteria, class-adaptive prototype budgets, and query-adaptive distance aggregation.

AI use statement

We used ChatGPT and Gemini to assist with literature review, experimental design, mathematical derivations, method implementation, and manuscript structuring, drafting, translation, and editing. No synthetic datasets were generated using AI. The authors take responsibility for the final content of this work.

References

Julian Bitterwolf, Maximilian Müller, and Matthias Hein. In or out? fixing ImageNet out-of-distribution detection evaluation. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 2471–2506, 2023. URL https://proceedings.mlr.press/v202/bitterwolf23a.html.

Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pat-*

tern Recognition, 2014. URL https://openaccess.thecvf.com/content_cvpr_2014/html/Cimpoi_Describing_Textures_in_2014_CVPR_paper.html.

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 Conference on Computer Vision and Pattern Recognition*, 2009.

Li Deng. The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. URL https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/MNIST-SPM2012.pdf.

Andrija Djurisic, Nebojsa Bozanic, Arjun Ashok, and Rosanne Liu. Extremely simple activation shaping for out-of-distribution detection. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ndYXTEL6cZz.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.11929.

Shreen Gul, Mohamed Elmahallawy, Ardhendu Tripathy, and Sanjay Madria. Prototype fusion: A training-free multi-layer approach to OOD detection. *arXiv preprint arXiv:2603.23677*, 2026. URL https://arxiv.org/abs/2603.23677.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1610.02136.

Ao Ke, Wenlong Chen, Chuanwen Feng, Yukun Cao, Xike Xie, S. Kevin Zhou, and Lei Feng. Prototype-based optimal transport for out-of-distribution detection. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence*, pp. 3018–3026, 2025. doi: 10.24963/ijcai.2025/336. URL https://www.ijcai.org/proceedings/2025/336.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. Cs231n project report, Stanford University, 2015. URL https://cs231n.stanford.edu/reports/2015/pdfs/yle_project.pdf.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/abdeb6f575ac5c6676b747bca8d09cc2-Abstract.html.

Weitang Liu, Xiaoyun Wang, John D. Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://papers.neurips.cc/paper/2020/hash/f5496252609c43eb8a3d147ab9b9c006-Abstract.html.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021. URL https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper.html.

Maximilian Müller and Matthias Hein. Mahalanobis++: Improving OOD detection via feature normalization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 45151–45184. PMLR, 2025. URL https://proceedings.mlr.press/v267/muller25a.html.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011. URL https://ai.stanford.edu/~twangcat/papers/nips2011_housenumbers.pdf.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 427–436, 2015. URL https://openaccess.thecvf.com/content_cvpr_2015/html/Nguyen_Deep_Neural_Networks_2015_CVPR_paper.html.

Jie Ren, Stanislav Fort, Jeremiah Liu, Abhijit Guha Roy, Shreyas Padhy, and Balaji Lakshminarayanan. A simple fix to Mahalanobis distance for improving near-OOD detection. *arXiv preprint arXiv:2106.09022*, 2021. URL https://arxiv.org/abs/2106.09022.

Yiyou Sun, Chuan Guo, and Yixuan Li. ReAct: Out-of-distribution detection with rectified activations. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper/2021/hash/01894d6f048493d2cacde3c579c315a3-Abstract.html.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 20827–20840. PMLR, 2022. URL https://proceedings.mlr.press/v162/sun22d.html.

Fahim Tajwar, Ananya Kumar, Sang Michael Xie, and Percy Liang. No true state-of-the-art? OOD detection methods are inconsistent across datasets. In *ICML Workshop on Uncertainty and Robustness in Deep Learning*, 2021. URL https://arxiv.org/abs/2109.05554.

Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The iNaturalist species classification and detection dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Van_Horn_The_INaturalist_Species_CVPR_2018_paper.html.

Sagar Vaze, Kai Han, Andrea Vedaldi, and Andrew Zisserman. Open-set recognition: A good closed-set classifier is all you need? In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2110.06207.

Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. ViM: Out-of-distribution with virtual-logit matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4921–4930, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Wang_ViM_Out-of-Distribution_With_Virtual-Logit_Matching_CVPR_2022_paper.html.

Yifeng Yang, Lin Zhu, Zewen Sun, Hengyu Liu, Qinying Gu, and Nanyang Ye. OODD: Test-time out-of-distribution detection with dynamic dictionary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 30630–30639, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yang_OODD_Test-time_Out-of-Distribution_Detection_with_Dynamic_Dictionary_CVPR_2025_paper.html.

Jingyang Zhang, Jingkang Yang, Pengyun Wang, Haoqi Wang, Yueqian Lin, Haoran Zhang, Yiyou Sun, Xuefeng Du, Yixuan Li, Ziwei Liu, Yiran Chen, and Hai Li. OpenOOD v1.5: Enhanced benchmark for out-of-distribution detection. *Journal of Data-centric Machine Learning Research*, 2024. URL https://openreview.net/forum?id=cnnTnJQigs. Dataset Certification.

Jinsong Zhang, Qiang Fu, Xu Chen, Lun Du, Zelin Li, Gang Wang, Xiaoguang Liu, Shi Han, and Dongmei Zhang. Out-of-distribution detection based on in-distribution data patterns memorization with modern Hopfield energy. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=KkazG4lgKL.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. URL https://places2.csail.mit.edu/.

Reihaneh Zohrabi, Hosein Hasani, Mahdieh Soleymani Baghshah, Anna Rohrbach, Marcus Rohrbach, and Mohammad Hossein Rohban. Spurious-aware prototype refinement for reliable out-of-distribution detection. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.23881.

A Additional Experimental Details

A.1 Hardware and Software Environment

All experiments were conducted in a Linux environment equipped with an NVIDIA GeForce RTX 3090 GPU with 24 GB of device memory, an Intel Core i7-10700K CPU with 8 cores and 16 threads, and 32 GB of host memory. The software stack comprised Python 3.10.14, PyTorch 1.13.1, TorchVision 0.14.1, and FAISS 1.9.0, with CUDA Runtime 11.7, cuDNN 8.5.0, and NVIDIA driver 535.230.02. Test data loading and backbone forward passes used a batch size of 200. Before timed inference, the backbone was warmed up with 15 dummy forward passes. The scope and definitions of the cost measurements follow Section 4.3.

A.2 Hyperparameter Selection

We followed OpenOOD's validation-set search procedure, selecting hyperparameters independently for each checkpoint based on validation AUROC and then using the selected configurations for test-set evaluation. Table A1 lists the parameters and candidate values actually searched, and Table A2 reports the final configurations. Methods without hyperparameter search are omitted.

Table A1: Hyperparameter search spaces. CPD configurations satisfy $k \leq n$.

Method	Search parameter	Candidate values
ReAct	Clipping percentile p	$\{85, 90, 95, 99\}$
ViM	Dimension parameter d	$\{256, 512, 1000\}$
KNN	Number of neighbors K	$\{50, 100, 200, 500, 1000\}$
POD	Exemplars per class m	$\{20, 50, 100\}$
OODD	ID neighbors K_1; queue capacity q	$K_1 \in \{5, 10, 100\}$; $q \in \{128, 512, 2048\}$
CPD	Prototypes per class n; aggregation size k	$n \in \{1, 3, 5, 10, 50, 100, 200, 300\}$; $k \in \{1, 3, 5, 10, 20, 50, 100, 200, 300\}$

For CIFAR-10, CIFAR-100, and ImageNet-200, multiple configurations are listed in checkpoint order `s0`, `s1`, and `s2`. A single configuration indicates that all three checkpoints selected the same setting. Each ImageNet-1K backbone uses a single pretrained checkpoint.

A.3 Implementation and Evaluation Details of Compared Methods

Our evaluation framework builds on OpenOOD and integrates evaluation of multiple methods, feature caching, and cost logging. All methods requiring feature vectors use cached features. For a given dataset and checkpoint, these methods share the same backbone feature cache, except OODD, which uses a separate feature extraction pipeline. POD is reimplemented in this work. To accommodate memory constraints, some methods use chunked processing, accumulation of statistics, and vectorized scoring. The reported costs reflect our implementations and device placement: KNN uses CPU-based FAISS retrieval, whereas CPD stores its prototypes and performs scoring on the GPU.

Table A2: Selected hyperparameters by dataset and backbone. Checkpoint-specific settings are explicitly labeled `s0`, `s1`, and `s2`; unlabeled entries are shared by all three checkpoints. Each ImageNet-1K backbone uses one checkpoint.

Method: parameter	CIFAR-10 ResNet-18	CIFAR-100 ResNet-18	ImageNet-200 ResNet-18
ReAct: p	`s0`: 99 `s1`: 90 `s2`: 85	99	`s0`: 95 `s1`: 99 `s2`: 99
ViM: d	256	256	256
KNN: K	50	50	500
POD: m	50	100	50
OODD: (K_1, q)	`s0`: (100,128) `s1`: (10,128) `s2`: (100,128)	(100,512)	(100,2048)
CPD: (n, k)	`s0`: (200,3) `s1`: (300,5) `s2`: (300,3)	`s0`: (300,5) `s1`: (300,1) `s2`: (200,3)	`s0`: (5,5) `s1`: (100,100) `s2`: (5,5)
Method: parameter	ImageNet-1K ResNet-50	ImageNet-1K Swin-T	ImageNet-1K ViT-B/16
ReAct: p	95	90	90
ViM: d	256	256	256
KNN: K	200	500	500
POD: m	100	50	100
OODD: (K_1, q)	(100,512)	(100,2048)	(100,2048)
CPD: (n, k)	(1,1)	(1,1)	(1,1)

OODD retains its test-time streaming updates. The queue is reset independently for each ID–OOD pair. All ID test samples and the test samples from the paired OOD dataset are randomly interleaved using NumPy random seed 100. The internal streaming scoring batch size is 512, specified separately from the validation-selected queue capacity q in Table A2. For each batch, ID reference scores are computed first and the queue is updated; final scores are then computed using the updated queue. Evaluation metrics are calculated from the ID and OOD scores generated by each paired stream.

B Complete OOD Detection Results

B.1 Results on Individual OOD Test Datasets

Tables B1–B4 report complete results on the individual OOD test datasets for the four ID benchmarks, complementing the mean Near-OOD and Far-OOD performance in the main text. Each cell reports AUROC and FPR95, both in percent; higher AUROC and lower FPR95 are better. CIFAR-10, CIFAR-100, and ImageNet-200 results are means and standard deviations across three checkpoints, whereas ImageNet-1K uses a single pretrained checkpoint. TIN denotes Tiny ImageNet.

CPD's improvements over KNN depend on the OOD source. On ImageNet-1K, CPD outperforms KNN in both AUROC and FPR95 on all five OOD test datasets; on CIFAR-100, the gains do not extend to every source. ImageNet-200 also illustrates differences between metrics: CPD achieves slightly higher AUROC than KNN on NINCO and SSB-hard, but higher FPR95, showing that improved overall ranking need not improve performance at a specific operating point.

Table B1: Per-dataset results on CIFAR-10 / ResNet-18. AUROC↑ / FPR95↓ (%).

Method	CIFAR-100 Near-OOD	TIN Near-OOD	MNIST Far-OOD
MSP	$87.19_{\pm 0.40}$ / $53.11_{\pm 6.03}$	$88.89_{\pm 0.24}$ / $43.19_{\pm 3.91}$	$92.63_{\pm 1.92}$ / $23.64_{\pm 7.14}$
EBO	$86.36_{\pm 0.71}$ / $66.57_{\pm 5.48}$	$88.85_{\pm 0.45}$ / $56.69_{\pm 6.57}$	$94.32_{\pm 3.10}$ / $24.98_{\pm 15.83}$
ReAct	$85.69_{\pm 1.25}$ / $67.87_{\pm 9.14}$	$88.25_{\pm 0.53}$ / $60.01_{\pm 8.74}$	$92.59_{\pm 4.10}$ / $34.18_{\pm 22.77}$
MDSN	$87.31_{\pm 0.78}$ / $50.83_{\pm 2.28}$	$89.07_{\pm 1.10}$ / $42.41_{\pm 2.07}$	$95.45_{\pm 0.33}$ / $17.80_{\pm 2.06}$
RMDSN	$88.83_{\pm 0.37}$ / $41.14_{\pm 4.10}$	$90.75_{\pm 0.38}$ / $31.27_{\pm 1.64}$	$92.99_{\pm 0.69}$ / $21.99_{\pm 1.78}$
ViM	$87.75_{\pm 0.34}$ / $49.18_{\pm 3.84}$	$89.72_{\pm 0.41}$ / $40.25_{\pm 1.59}$	$94.76_{\pm 0.46}$ / $18.34_{\pm 1.74}$
SHE	$80.31_{\pm 0.84}$ / $80.99_{\pm 4.19}$	$82.85_{\pm 0.49}$ / $77.92_{\pm 4.34}$	$90.43_{\pm 5.83}$ / $42.23_{\pm 25.23}$
KNN	$89.73_{\pm 0.17}$ / $37.62_{\pm 0.37}$	$91.62_{\pm 0.33}$ / $30.19_{\pm 0.86}$	$94.26_{\pm 0.47}$ / $20.04_{\pm 1.65}$
POD	$87.88_{\pm 0.62}$ / $44.45_{\pm 2.65}$	$89.64_{\pm 0.60}$ / $37.51_{\pm 2.45}$	$92.66_{\pm 0.67}$ / $24.24_{\pm 2.83}$
OODD	$89.81_{\pm 0.15}$ / $39.51_{\pm 1.09}$	$92.08_{\pm 0.21}$ / $30.56_{\pm 0.97}$	$98.00_{\pm 0.36}$ / $9.05_{\pm 1.69}$
CPD	$89.60_{\pm 0.17}$ / $38.93_{\pm 0.31}$	$91.57_{\pm 0.35}$ / $30.84_{\pm 0.65}$	$94.37_{\pm 0.32}$ / $19.90_{\pm 1.40}$

Method	SVHN Far-OOD	Textures Far-OOD	Places365 Far-OOD
MSP	$91.56_{\pm 0.46}$ / $25.42_{\pm 1.92}$	$89.89_{\pm 0.87}$ / $35.21_{\pm 5.57}$	$88.92_{\pm 0.58}$ / $42.47_{\pm 4.79}$
EBO	$91.98_{\pm 1.11}$ / $34.20_{\pm 6.79}$	$89.47_{\pm 0.85}$ / $52.00_{\pm 7.63}$	$89.25_{\pm 0.96}$ / $54.68_{\pm 8.11}$
ReAct	$88.73_{\pm 4.84}$ / $49.68_{\pm 20.25}$	$89.20_{\pm 1.71}$ / $53.91_{\pm 13.33}$	$90.45_{\pm 0.94}$ / $44.26_{\pm 3.93}$
MDSN	$94.80_{\pm 0.26}$ / $19.26_{\pm 1.87}$	$94.75_{\pm 0.49}$ / $22.99_{\pm 2.83}$	$88.97_{\pm 0.83}$ / $44.18_{\pm 2.49}$
RMDSN	$91.71_{\pm 0.23}$ / $23.20_{\pm 1.33}$	$92.15_{\pm 0.33}$ / $24.55_{\pm 0.56}$	$91.36_{\pm 0.10}$ / $28.70_{\pm 0.16}$
ViM	$94.58_{\pm 0.57}$ / $19.01_{\pm 0.66}$	$95.16_{\pm 0.41}$ / $21.14_{\pm 1.95}$	$89.50_{\pm 0.47}$ / $41.44_{\pm 2.66}$
SHE	$86.59_{\pm 1.64}$ / $62.26_{\pm 5.11}$	$81.57_{\pm 1.49}$ / $84.57_{\pm 6.50}$	$82.88_{\pm 1.50}$ / $76.30_{\pm 6.49}$
KNN	$92.77_{\pm 0.36}$ / $22.39_{\pm 1.71}$	$93.16_{\pm 0.29}$ / $24.06_{\pm 0.56}$	$91.77_{\pm 0.29}$ / $30.36_{\pm 0.81}$
POD	$91.88_{\pm 0.13}$ / $25.33_{\pm 1.13}$	$91.55_{\pm 0.51}$ / $30.58_{\pm 3.05}$	$89.12_{\pm 1.05}$ / $40.37_{\pm 4.79}$
OODD	$96.94_{\pm 0.54}$ / $12.73_{\pm 2.31}$	$94.87_{\pm 0.32}$ / $20.21_{\pm 1.04}$	$93.58_{\pm 0.34}$ / $25.46_{\pm 1.08}$
CPD	$92.87_{\pm 0.39}$ / $21.41_{\pm 1.76}$	$93.27_{\pm 0.29}$ / $23.34_{\pm 0.58}$	$91.62_{\pm 0.36}$ / $31.21_{\pm 1.26}$

Table B2: Per-dataset results on CIFAR-100 / ResNet-18. AUROC↑ / FPR95↓ (%).

Method	CIFAR-10 Near-OOD	TIN Near-OOD	MNIST Far-OOD
MSP	$78.47_{\pm 0.08}$ / $58.90_{\pm 1.14}$	$81.96_{\pm 0.24}$ / $50.77_{\pm 0.70}$	$76.08_{\pm 2.28}$ / $57.24_{\pm 5.72}$
EBO	$79.05_{\pm 0.13}$ / $59.18_{\pm 0.91}$	$82.58_{\pm 0.10}$ / $52.35_{\pm 0.71}$	$79.18_{\pm 1.67}$ / $52.61_{\pm 4.70}$
ReAct	$78.65_{\pm 0.06}$ / $61.30_{\pm 0.53}$	$82.72_{\pm 0.10}$ / $51.64_{\pm 0.49}$	$78.37_{\pm 1.95}$ / $56.03_{\pm 6.94}$
MDSN	$71.76_{\pm 0.51}$ / $75.23_{\pm 0.80}$	$76.60_{\pm 0.46}$ / $59.73_{\pm 1.96}$	$82.73_{\pm 0.77}$ / $49.13_{\pm 3.20}$
RMDSN	$78.08_{\pm 0.23}$ / $61.43_{\pm 0.49}$	$82.76_{\pm 0.06}$ / $48.84_{\pm 0.71}$	$79.37_{\pm 2.65}$ / $52.49_{\pm 6.33}$
ViM	$72.21_{\pm 0.51}$ / $70.59_{\pm 0.51}$	$77.87_{\pm 0.16}$ / $54.62_{\pm 0.51}$	$81.89_{\pm 1.25}$ / $48.30_{\pm 1.29}$
SHE	$78.15_{\pm 0.03}$ / $60.41_{\pm 0.62}$	$79.52_{\pm 0.41}$ / $58.34_{\pm 1.02}$	$76.76_{\pm 1.30}$ / $58.78_{\pm 3.32}$
KNN	$77.02_{\pm 0.31}$ / $72.81_{\pm 0.55}$	$83.31_{\pm 0.20}$ / $49.53_{\pm 0.68}$	$82.36_{\pm 1.87}$ / $48.58_{\pm 5.73}$
POD	$70.88_{\pm 0.25}$ / $85.65_{\pm 0.93}$	$77.86_{\pm 0.26}$ / $61.24_{\pm 0.07}$	$81.94_{\pm 1.41}$ / $46.25_{\pm 2.83}$
OODD	$81.41_{\pm 0.36}$ / $60.47_{\pm 0.85}$	$85.31_{\pm 0.27}$ / $47.32_{\pm 0.62}$	$98.81_{\pm 0.56}$ / $7.47_{\pm 4.25}$
CPD	$77.80_{\pm 0.19}$ / $68.17_{\pm 1.60}$	$83.57_{\pm 0.26}$ / $48.12_{\pm 0.24}$	$81.59_{\pm 1.62}$ / $49.00_{\pm 5.33}$

Method	SVHN Far-OOD	Textures Far-OOD	Places365 Far-OOD
MSP	$78.68_{\pm 1.16}$ / $58.43_{\pm 3.21}$	$77.32_{\pm 0.87}$ / $61.79_{\pm 1.60}$	$79.22_{\pm 0.35}$ / $56.65_{\pm 1.07}$
EBO	$82.28_{\pm 2.19}$ / $53.19_{\pm 3.98}$	$78.35_{\pm 1.03}$ / $62.38_{\pm 2.54}$	$79.50_{\pm 0.29}$ / $57.69_{\pm 1.05}$
ReAct	$83.25_{\pm 1.23}$ / $49.89_{\pm 2.38}$	$80.15_{\pm 0.56}$ / $55.01_{\pm 0.99}$	$80.01_{\pm 0.13}$ / $55.34_{\pm 0.60}$
MDSN	$86.53_{\pm 3.34}$ / $44.46_{\pm 7.83}$	$86.95_{\pm 0.96}$ / $48.13_{\pm 3.31}$	$76.68_{\pm 0.87}$ / $64.63_{\pm 0.62}$
RMDSN	$84.16_{\pm 1.59}$ / $51.88_{\pm 5.28}$	$82.56_{\pm 1.02}$ / $54.77_{\pm 1.83}$	$83.01_{\pm 0.59}$ / $53.76_{\pm 0.78}$
ViM	$82.91_{\pm 4.61}$ / $46.29_{\pm 6.70}$	$85.91_{\pm 0.96}$ / $46.85_{\pm 2.78}$	$75.86_{\pm 0.45}$ / $61.57_{\pm 0.91}$
SHE	$81.24_{\pm 4.96}$ / $58.57_{\pm 9.37}$	$73.64_{\pm 1.58}$ / $73.34_{\pm 4.08}$	$76.28_{\pm 0.63}$ / $65.29_{\pm 1.11}$
KNN	$84.27_{\pm 1.35}$ / $51.48_{\pm 3.93}$	$83.66_{\pm 1.02}$ / $53.61_{\pm 2.90}$	$79.42_{\pm 0.58}$ / $60.76_{\pm 1.11}$
POD	$75.65_{\pm 6.07}$ / $62.17_{\pm 8.40}$	$81.30_{\pm 0.27}$ / $59.86_{\pm 2.46}$	$73.27_{\pm 0.53}$ / $72.59_{\pm 0.53}$
OODD	$98.46_{\pm 0.44}$ / $10.23_{\pm 3.49}$	$92.39_{\pm 0.47}$ / $35.65_{\pm 2.85}$	$86.21_{\pm 0.57}$ / $50.86_{\pm 0.96}$
CPD	$84.59_{\pm 1.05}$ / $47.67_{\pm 2.84}$	$82.12_{\pm 1.39}$ / $54.28_{\pm 1.65}$	$80.17_{\pm 0.47}$ / $57.13_{\pm 0.75}$

Table B3: Per-dataset results on ImageNet-200 / ResNet-18. AUROC↑ / FPR95↓ (%).

Method	SSB-hard Near-OOD	NINCO Near-OOD
MSP	$80.32_{\pm 0.06}$ / $66.09_{\pm 0.06}$	$86.27_{\pm 0.13}$ / $43.77_{\pm 0.88}$
EBO	$79.72_{\pm 0.03}$ / $69.96_{\pm 0.31}$	$85.10_{\pm 0.15}$ / $50.66_{\pm 1.15}$
ReAct	$78.83_{\pm 1.58}$ / $71.94_{\pm 2.18}$	$84.70_{\pm 0.79}$ / $53.34_{\pm 2.97}$
MDSN	$77.43_{\pm 0.45}$ / $73.97_{\pm 1.17}$	$85.94_{\pm 0.38}$ / $47.04_{\pm 1.56}$
RMDSN	$80.87_{\pm 0.20}$ / $65.97_{\pm 0.19}$	$86.03_{\pm 0.28}$ / $42.86_{\pm 1.21}$
ViM	$73.90_{\pm 0.38}$ / $71.51_{\pm 0.60}$	$83.29_{\pm 0.22}$ / $47.20_{\pm 1.41}$
SHE	$78.18_{\pm 0.29}$ / $72.78_{\pm 0.39}$	$82.01_{\pm 0.43}$ / $60.96_{\pm 2.02}$
KNN	$76.88_{\pm 0.30}$ / $73.88_{\pm 0.31}$	$86.05_{\pm 0.15}$ / $46.74_{\pm 0.96}$
POD	$65.95_{\pm 1.02}$ / $83.06_{\pm 0.83}$	$76.19_{\pm 0.73}$ / $58.87_{\pm 0.67}$
OODD	$82.32_{\pm 0.21}$ / $63.42_{\pm 0.32}$	$90.35_{\pm 0.03}$ / $41.13_{\pm 0.62}$
CPD	$77.36_{\pm 0.72}$ / $77.69_{\pm 1.25}$	$86.40_{\pm 0.10}$ / $48.59_{\pm 0.54}$

Method	iNaturalist Far-OOD	Textures Far-OOD	OpenImage-O Far-OOD
MSP	$92.77_{\pm 0.31}$ / $26.53_{\pm 0.85}$	$88.37_{\pm 0.16}$ / $44.43_{\pm 0.89}$	$89.24_{\pm 0.02}$ / $35.21_{\pm 0.34}$
EBO	$92.52_{\pm 0.62}$ / $26.40_{\pm 2.81}$	$90.80_{\pm 0.19}$ / $41.31_{\pm 2.28}$	$89.20_{\pm 0.32}$ / $36.79_{\pm 1.38}$
ReAct	$93.62_{\pm 1.08}$ / $22.99_{\pm 2.75}$	$92.86_{\pm 0.58}$ / $29.49_{\pm 1.53}$	$90.36_{\pm 0.43}$ / $32.97_{\pm 0.83}$
MDSN	$94.66_{\pm 0.51}$ / $20.48_{\pm 1.69}$	$97.44_{\pm 0.19}$ / $12.58_{\pm 0.56}$	$91.35_{\pm 0.34}$ / $31.89_{\pm 1.29}$
RMDSN	$92.30_{\pm 0.60}$ / $24.39_{\pm 1.15}$	$89.93_{\pm 0.35}$ / $36.51_{\pm 1.53}$	$88.44_{\pm 0.28}$ / $34.35_{\pm 0.44}$
ViM	$90.97_{\pm 0.44}$ / $27.37_{\pm 0.50}$	$94.61_{\pm 0.14}$ / $20.39_{\pm 0.24}$	$88.20_{\pm 0.21}$ / $33.90_{\pm 0.88}$
SHE	$91.40_{\pm 1.59}$ / $34.38_{\pm 4.28}$	$90.52_{\pm 0.23}$ / $45.53_{\pm 3.09}$	$87.46_{\pm 0.86}$ / $46.49_{\pm 1.60}$
KNN	$93.97_{\pm 0.44}$ / $24.43_{\pm 1.34}$	$95.30_{\pm 0.02}$ / $24.53_{\pm 0.27}$	$90.17_{\pm 0.39}$ / $32.94_{\pm 1.37}$
POD	$83.76_{\pm 0.46}$ / $40.90_{\pm 1.72}$	$84.59_{\pm 0.33}$ / $40.91_{\pm 0.38}$	$78.64_{\pm 0.79}$ / $50.40_{\pm 1.85}$
OODD	$98.38_{\pm 0.05}$ / $9.92_{\pm 0.41}$	$95.18_{\pm 0.08}$ / $25.83_{\pm 0.46}$	$92.99_{\pm 0.03}$ / $28.23_{\pm 0.19}$
CPD	$96.40_{\pm 0.27}$ / $15.71_{\pm 1.24}$	$95.04_{\pm 0.11}$ / $23.31_{\pm 0.88}$	$92.25_{\pm 0.46}$ / $28.44_{\pm 0.64}$

Table B4: Per-dataset results on ImageNet-1K / ResNet-50. AUROC↑ / FPR95↓ (%).

Method	SSB-hard Near-OOD	NINCO Near-OOD	iNaturalist Far-OOD	Textures Far-OOD	OpenImage-O Far-OOD
MSP	72.16 / 74.62	79.98 / 56.83	88.43 / 43.38	82.48 / 60.95	84.99 / 49.98
EBO	72.35 / 76.43	79.70 / 60.56	90.61 / 31.34	88.74 / 45.74	89.15 / 38.09
ReAct	73.09 / 78.04	81.71 / 55.80	96.32 / 16.74	92.80 / 29.67	91.88 / 32.67
MDSN	64.91 / 82.20	79.13 / 62.00	90.61 / 32.38	98.95 / 5.11	90.39 / 38.02
RMDSN	72.86 / 79.77	84.21 / 52.60	91.92 / 30.19	91.14 / 45.14	89.40 / 39.06
ViM	65.10 / 81.13	78.54 / 62.28	89.54 / 30.67	97.96 / 10.56	90.40 / 33.05
SHE	71.83 / 75.61	76.42 / 69.63	92.58 / 34.04	93.63 / 35.63	86.70 / 54.66
KNN	61.78 / 84.16	79.41 / 58.39	86.31 / 40.93	97.06 / 17.51	86.78 / 44.79
POD	47.49 / 91.91	63.09 / 73.63	66.78 / 63.64	81.49 / 49.15	67.54 / 67.26
OODD	70.45 / 78.36	82.25 / 58.30	98.82 / 5.90	96.99 / 14.91	91.07 / 36.95
CPD	67.33 / 82.01	82.67 / 55.72	95.74 / 19.75	97.68 / 12.75	91.73 / 34.57

B.2 Cross-Backbone Results

Tables B5 and B6 provide additional ImageNet-1K results with Swin-T and ViT-B/16. On both backbones, CPD achieves higher AUROC and lower FPR95 than KNN on all five OOD test datasets. However, its mean Near-OOD and Far-OOD AUROCs remain below those of MDSN and RMDSN, indicating that its performance advantage over compact statistical methods does not consistently extend across backbones. Corresponding storage and timing results are reported in Appendix C to characterize these performance–cost trade-offs.

C Additional Cost Results

This section reports additional costs of feature-based methods across the four ID datasets and the ImageNet-1K backbones. Table C1 reports the time required to construct the training feature cache.

Table B5: Per-dataset results on ImageNet-1K / Swin-T. AUROC↑ / FPR95↓ (%).

Method	SSB-hard Near-OOD	NINCO Near-OOD	iNaturalist Far-OOD	Textures Far-OOD	OpenImage-O Far-OOD
MSP	71.75 / 81.02	81.69 / 60.36	89.84 / 37.50	83.27 / 61.49	85.81 / 48.97
EBO	68.28 / 87.51	78.42 / 79.23	85.17 / 61.18	79.00 / 84.29	80.24 / 80.20
ReAct	69.36 / 85.01	82.12 / 60.08	90.08 / 31.34	87.04 / 53.98	87.85 / 42.54
MDSN	73.85 / 83.25	87.24 / 49.35	97.28 / 10.84	92.69 / 30.05	93.83 / 24.33
RMDSN	75.19 / 81.21	87.81 / 48.41	97.32 / 10.97	89.32 / 37.85	93.16 / 25.79
ViM	68.94 / 88.55	81.85 / 60.79	94.62 / 17.98	92.69 / 29.46	92.29 / 26.62
SHE	70.75 / 85.03	82.74 / 67.53	92.78 / 33.25	88.86 / 51.32	88.04 / 52.88
KNN	64.21 / 85.08	79.16 / 58.14	88.91 / 31.15	90.56 / 35.79	88.62 / 36.30
POD	56.90 / 89.39	70.20 / 69.29	80.70 / 49.47	84.95 / 45.53	82.40 / 51.83
OODD	73.78 / 75.85	86.29 / 51.34	98.68 / 7.50	95.54 / 24.22	94.73 / 24.81
CPD	69.72 / 81.95	83.27 / 52.86	93.49 / 22.14	90.89 / 34.69	91.55 / 29.36

Table B6: Per-dataset results on ImageNet-1K / ViT-B/16. AUROC↑ / FPR95↓ (%).

Method	SSB-hard Near-OOD	NINCO Near-OOD	iNaturalist Far-OOD	Textures Far-OOD	OpenImage-O Far-OOD
MSP	69.03 / 86.14	78.09 / 77.10	88.16 / 42.72	85.09 / 56.28	84.85 / 56.29
EBO	59.16 / 91.98	66.05 / 94.19	79.26 / 83.53	81.22 / 83.51	76.49 / 88.95
ReAct	63.24 / 90.22	75.45 / 78.54	86.09 / 48.34	86.69 / 55.91	84.26 / 57.58
MDSN	74.09 / 83.94	88.61 / 44.51	97.20 / 11.79	90.63 / 31.22	93.13 / 25.26
RMDSN	74.77 / 83.36	88.65 / 43.81	96.96 / 13.32	89.33 / 34.67	92.68 / 26.48
ViM	69.34 / 90.08	84.63 / 57.39	95.71 / 17.65	90.61 / 40.13	92.15 / 29.78
SHE	68.11 / 85.73	84.16 / 56.06	93.56 / 22.14	92.67 / 25.46	91.03 / 33.70
KNN	65.93 / 86.23	82.22 / 54.91	91.45 / 27.88	91.12 / 33.20	89.83 / 34.86
POD	61.05 / 88.14	76.50 / 63.68	83.52 / 42.78	86.30 / 43.85	84.54 / 47.18
OODD	76.17 / 75.37	88.86 / 46.10	98.94 / 5.74	96.09 / 21.44	95.20 / 22.89
CPD	70.35 / 83.38	85.18 / 48.76	93.89 / 21.31	91.21 / 32.09	91.68 / 28.86

All relevant methods share the same backbone feature cache except OODD, which uses a separate extraction pipeline. Tables C2–C4 report persistent storage, detector setup time, and average inference time per sample for validation-selected configurations. Setup time excludes feature extraction and hyperparameter search.

CPD uses less storage than KNN on all four ID datasets, although the reduction depends on the selected prototype count. On CIFAR-100, CPD retains more prototypes and requires more storage than compact statistical methods; on ImageNet-1K with ResNet-50, one prototype per class reduces storage to 7.81 MiB. Offline setup costs exhibit different trade-offs: CPD takes longer to set up than KNN on the CIFAR benchmarks and ImageNet-200, but less time under the single-prototype configuration on ImageNet-1K with ResNet-50.

On Swin-T and ViT-B/16, CPD also selects one prototype per class and requires 2.93 MiB, compared with 3,753.42 MiB for KNN and 8.11 and 10.37 MiB for MDSN and RMDSN. CPD's average inference time is lower than KNN's but comparable to those of compact statistical methods, with no consistent speed advantage over the latter. Together with Appendix B, these results show that CPD offers lower storage requirements on these backbones while remaining below MDSN and RMDSN in detection performance. Timing comparisons primarily concern methods using the same backbone, since end-to-end time across backbones is also affected by backbone inference.

D Representation Error and Distance Aggregation

D.1 ID Representation Error and OOD Detection Performance

Figure D1 compares ID representation error Q_{ID} and validation AUROC across prototype counts n, with $k = 1$ fixed. Here, Q_{ID} is the mean squared distance between each ID sample and its nearest prototype within its own class. As n increases, representation error generally decreases on all four

Table C1: Training feature extraction time (s) by dataset and backbone.

Dataset / backbone	Shared backbone feature cache	OODD extraction pipeline
CIFAR-10 / ResNet-18	$12.834_{\pm 0.471}$	$53.027_{\pm 0.032}$
CIFAR-100 / ResNet-18	$12.963_{\pm 0.442}$	$53.411_{\pm 0.232}$
ImageNet-200 / ResNet-18	$1188.196_{\pm 2.824}$	$2550.517_{\pm 7.318}$
ImageNet-1K / ResNet-50	6153.760	10182.227
ImageNet-1K / Swin-T	6933.828	10153.705
ImageNet-1K / ViT-B/16	6171.149	14762.776

Table C2: Cost comparison for CIFAR-10 / ResNet-18 and CIFAR-100 / ResNet-18. Setup time excludes feature extraction and hyperparameter search.

Method	CIFAR-10 / ResNet-18			CIFAR-100 / ResNet-18		
	Storage↓ (MiB)	Setup↓ (s)	Inference↓ (ms/sample)	Storage↓ (MiB)	Setup↓ (s)	Inference↓ (ms/sample)
MDSN	$1.04_{\pm 0.00}$	$0.290_{\pm 0.319}$	$0.524_{\pm 0.001}$	$1.39_{\pm 0.00}$	$0.142_{\pm 0.032}$	$0.526_{\pm 0.001}$
RMDSN	$2.04_{\pm 0.00}$	$0.123_{\pm 0.000}$	$0.525_{\pm 0.001}$	$2.39_{\pm 0.00}$	$0.151_{\pm 0.006}$	$0.527_{\pm 0.002}$
ViM	$0.52_{\pm 0.00}$	$0.111_{\pm 0.000}$	$0.524_{\pm 0.001}$	$0.70_{\pm 0.00}$	$0.148_{\pm 0.002}$	$0.525_{\pm 0.000}$
SHE	$0.02_{\pm 0.00}$	$0.015_{\pm 0.001}$	$0.527_{\pm 0.001}$	$0.20_{\pm 0.00}$	$0.021_{\pm 0.002}$	$0.529_{\pm 0.001}$
KNN	$97.66_{\pm 0.00}$	$0.065_{\pm 0.002}$	$0.589_{\pm 0.001}$	$97.66_{\pm 0.00}$	$0.066_{\pm 0.005}$	$0.591_{\pm 0.000}$
POD	$0.98_{\pm 0.00}$	$0.013_{\pm 0.000}$	$0.526_{\pm 0.001}$	$19.53_{\pm 0.00}$	$0.023_{\pm 0.000}$	$0.528_{\pm 0.001}$
OODD	$48.83_{\pm 0.00}$	$0.032_{\pm 0.001}$	$0.549_{\pm 0.008}$	$48.83_{\pm 0.00}$	$0.042_{\pm 0.003}$	$0.544_{\pm 0.002}$
CPD	$5.21_{\pm 1.13}$	$1.097_{\pm 0.119}$	$0.530_{\pm 0.001}$	$52.08_{\pm 11.28}$	$1.759_{\pm 0.168}$	$0.534_{\pm 0.001}$

Table C3: Cost comparison for ImageNet-200 / ResNet-18 and ImageNet-1K / ResNet-50. Setup time excludes feature extraction and hyperparameter search.

Method	ImageNet-200 / ResNet-18			ImageNet-1K / ResNet-50		
	Storage↓ (MiB)	Setup↓ (s)	Inference↓ (ms/sample)	Storage↓ (MiB)	Setup↓ (s)	Inference↓ (ms/sample)
MDSN	$1.78_{\pm 0.00}$	$0.517_{\pm 0.034}$	$3.549_{\pm 0.005}$	31.63	22.763	3.932
RMDSN	$2.79_{\pm 0.00}$	$0.526_{\pm 0.006}$	$3.556_{\pm 0.008}$	47.64	23.723	3.939
ViM	$0.89_{\pm 0.00}$	$0.506_{\pm 0.001}$	$3.556_{\pm 0.002}$	21.82	24.117	3.876
SHE	$0.39_{\pm 0.00}$	$0.082_{\pm 0.007}$	$3.584_{\pm 0.009}$	7.81	2.937	3.888
KNN	$505.76_{\pm 0.00}$	$0.657_{\pm 0.001}$	$3.770_{\pm 0.004}$	10009.12	12.073	14.508
POD	$19.53_{\pm 0.00}$	$0.069_{\pm 0.002}$	$3.583_{\pm 0.003}$	781.25	2.321	3.915
OODD	$252.88_{\pm 0.00}$	$0.175_{\pm 0.009}$	$3.705_{\pm 0.009}$	5004.55	3.176	5.392
CPD	$14.32_{\pm 21.43}$	$2.434_{\pm 1.264}$	$3.602_{\pm 0.004}$	7.81	9.379	3.902

Table C4: Cost comparison for ImageNet-1K / Swin-T and ImageNet-1K / ViT-B/16. Setup time excludes feature extraction and hyperparameter search.

Method	ImageNet-1K / Swin-T			ImageNet-1K / ViT-B/16		
	Storage↓ (MiB)	Setup↓ (s)	Inference↓ (ms/sample)	Storage↓ (MiB)	Setup↓ (s)	Inference↓ (ms/sample)
MDSN	8.11	4.947	4.761	8.11	5.906	4.448
RMDSN	10.37	4.925	4.737	10.37	5.014	4.423
ViM	4.44	4.725	4.762	4.44	4.593	4.414
SHE	2.93	1.659	4.744	2.93	2.224	4.423
KNN	3753.42	4.679	7.540	3753.42	5.399	8.122
POD	146.48	1.045	4.732	292.97	1.187	4.426
OODD	1876.71	1.300	5.930	1876.71	1.303	5.628
CPD	2.93	1.875	4.778	2.93	1.931	4.425

datasets, but AUROC follows different trends: the CIFAR benchmarks generally improve with local fluctuations, whereas the ImageNet benchmarks deteriorate. In particular, ImageNet-1K AUROC decreases from 91.90% to 85.51%.

These results show that lower ID representation error does not guarantee better OOD detection. Q_{ID} measures only how well the prototypes approximate ID features and does not directly characterize the ranking of ID and OOD scores. Since data distributions, backbone features, and OOD sources also differ, the differences in trends across datasets cannot be attributed solely to the number of classes.

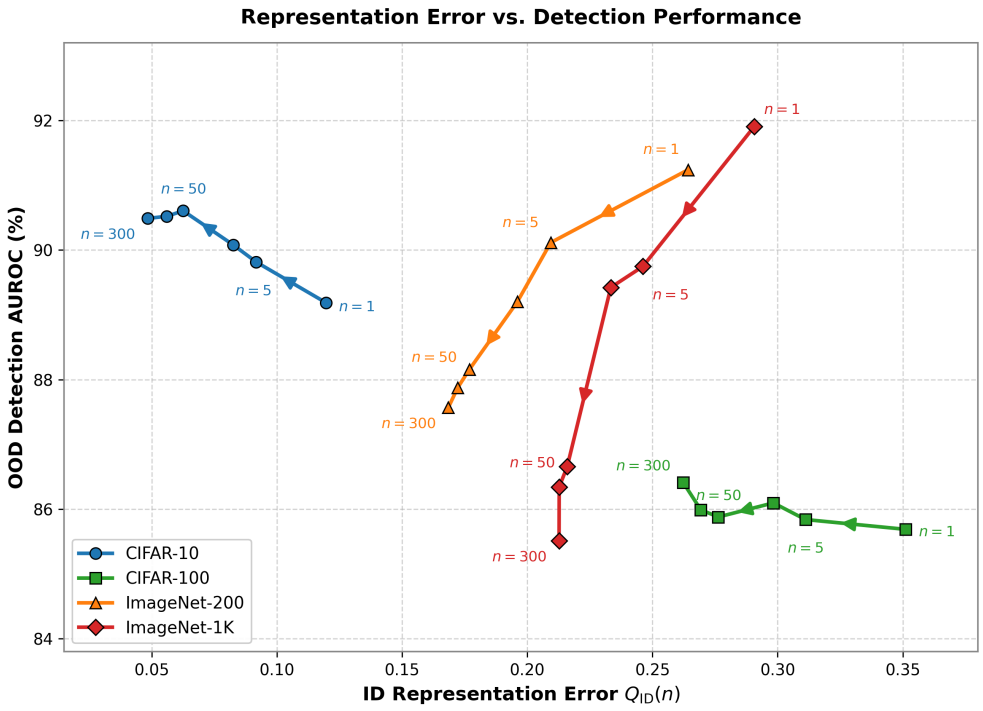


Figure D1: Relationship between ID representation error and validation AUROC. Each point corresponds to a prototype count n with $k = 1$ fixed; arrows indicate increasing n. The CIFAR benchmarks and ImageNet-200 use checkpoint `s0`; ImageNet-1K uses a single pretrained ResNet-50 checkpoint.

D.2 Distance Aggregation with Fixed Prototype Sets

Figure D2 holds the prototype set fixed within each configuration and varies only the aggregation size k. The CIFAR benchmarks perform better with small k, with AUROC decreasing as the aggregation neighborhood expands. In contrast, the ImageNet benchmarks benefit substantially from aggregating multiple prototypes. For example, on ImageNet-1K with $n = 100$, increasing k from 1 to 50 improves AUROC from 85.95% to 90.68%.

However, full aggregation is not always optimal: ImageNet-1K AUROC decreases to 90.09% at $n = 100, k = 100$, and $k = 5$ also outperforms $k = 10$ when $n = 10$. On ImageNet-200, $k = n$ achieves the highest AUROC among the tested candidates for both prototype counts. These results support jointly selecting the prototype count and aggregation neighborhood rather than assuming that more prototypes or broader aggregation necessarily improve detection.

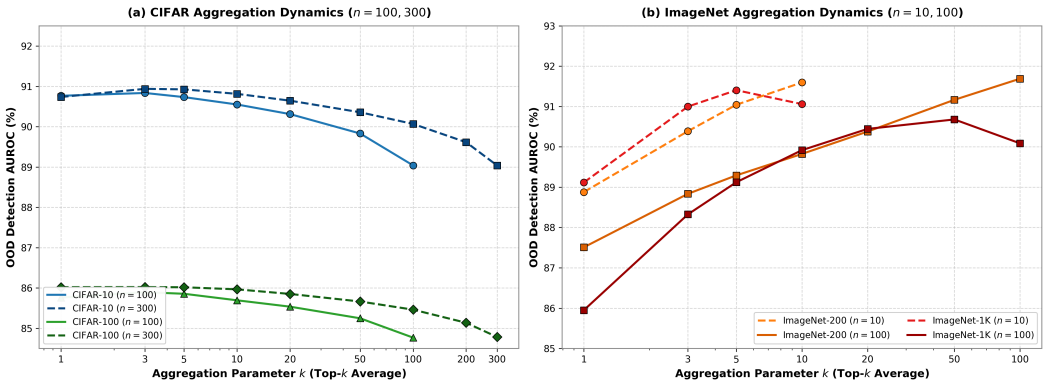


Figure D2: Effect of aggregation neighborhood size on validation AUROC with fixed prototype sets. (a) CIFAR benchmarks with $n \in \{100, 300\}$. (b) ImageNet benchmarks with $n \in \{10, 100\}$. The CIFAR benchmarks and ImageNet-200 report means across three checkpoints; ImageNet-1K uses a single pretrained ResNet-50 checkpoint. The prototypes in Figures D1 and D2 were constructed by the representation-error analysis and hyperparameter-search pipelines, respectively; identical (n, k) values therefore need not correspond to identical prototype sets.