ANGULAR LDA FOR LEARNING AFTER DEPLOYMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Traditional machine learning models are static and once deployed, they no longer can acquire new knowledge. However, real-world environments are constantly evolving, and deployed AI systems will inevitably encounter classes and situations that were not present during training. To remain effective, these systems must be capable of post-deployment learning: identifying novel classes, acquiring ground-truth supervision, and learning them incrementally on the fly from the new data without relying on retraining with both past and newly collected data. To address this challenge, we propose LAFA, a novel post-deployment learning method that leverages pre-trained models and angular linear discriminant analysis with feature and test-time enhancements. Extensive experiments on eight datasets using three backbones demonstrate that LAFA consistently outperforms strong baselines without saving any past training data. Its performance can match that of offline retraining (or joint training) that has access to all past and present data. We believe the proposed approach is sufficiently mature for practical deployment.

1 INTRODUCTION

Machine learning models are typically trained offline and remain fixed once deployed. Such models are ill-suited for open-world environments, where data distributions evolve and new classes constantly emerge. In contrast, humans can recognize unfamiliar objects and learn them incrementally as they are encountered. Endowing deployed AI systems with a similar capability is the goal of post-deployment learning (Kaymak et al., 2025), and it is the goal of this paper.

This paper considers the following scenario: In the *pre-deployment* stage, a classification model $\mathcal{M}$ is trained offline with labeled training data from a set of *base classes* $\mathcal{C}$ and then $\mathcal{M}$ is deployed. In the *post-deployment* stage, $\mathcal{M}$ iteratively performs the following three tasks: **(1)** classify incoming samples into the *learned classes* and detect *out-of-distribution* (OOD) samples not belonging to any learned class; **(2)** When a sample is identified as OOD, query a human user or a knowledgeable AI agent to *acquire its ground-truth class label*; and **(3)** *incrementally or continually learn* the OOD sample into $\mathcal{M}$. The process continues throughout the model's lifetime. This paper focuses on Tasks (1) and (3); Task (2) (which involves natural-language dialogue) is left to future work.

This setting has two key characteristics: First, *the set of learned classes*—the in-distribution (ID) set—*is dynamic*. It contains the base classes $\mathcal{C}$ plus any class seen after deployment that has since been observed often enough to be considered well learned. Traditional OOD detection assumes a fixed ID set. Second, the model $\mathcal{M}$ *is updated* with every accepted OOD sample.

It faces *three key challenges*. The first is *catastrophic forgetting* (CF) in CL, where learning new tasks degrades performance on earlier ones (McCloskey & Cohen, 1989). The second is that it needs *continual OOD detection*, which has a dynamic ID set while traditional OOD detection has a fixed ID set. The third is *very few samples* per new class for learning—main/frequent classes would have been learned in pre-deployment, and it is also undesirable to ask users to label many samples.

To overcome these challenges, we propose a novel method, called **LAFA** (***L**earning after deployment based on **A**ngular linear discriminant analysis with **F**eature projection and test-time magnitude **A**ugmentation*). LAFA employs a strong *pre-trained model* (PTM) as a fixed feature extractor, a strategy that has become increasingly prevalent in CL (see the next section). The extracted features are then randomly projected into a higher-dimensional space to enhance class separability. Learning is performed using an incremental version of *Angular Linear Discriminant Analysis* (aLDA), which

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.

helps deal with CF, supports learning from limited samples, and facilitates OOD detection. Finally, *test-time magnitude augmentation* (TMA) further improves classification performance.

In summary, this paper makes the following contributions.

1. **Incremental learning with few samples**. aLDA and TMA are particularly suited to continual learning with fewer samples. To our knowledge, both aLDA and TMA have not been used in OOD detection or CL. The work in (Kaymak et al., 2025) based on LDA appears to be the only existing study that performs post-deployment learning (see Sec. 2 for more details). However, our experiments show that LAFA is markedly superior.
2. **Replay-free and Avoiding CF.** With the PTM kept frozen and aLDA updated analytically, LAFA avoids the CF associated with gradient-based training without saving any past data. The incremental updates of aLDA shift decision boundaries and potentially introduce CF, but our results show that such effects are negligible and do not lead to observable CF.
3. **Ready for Practical Use.** Our experiments using eight diverse datasets and three PTMs show that LAFA can match the accuracy of *joint training* (retraining using all past and present data simultaneously)[1], and is sufficiently mature for practical deployment.

2 Related Work

OOD Detection. Many methods have been proposed for OOD detection (Yang et al., 2024). *Confidence-based methods* use logits to estimate classification confidence to compute OOD scores, e.g., directly changing the predicted logits (Hendrycks & Gimpel, 2017), improving prediction (Sun et al., 2021),

or modifying model uncertainty (Liu & Qin, 2024), adding auxiliary mechanisms (Liang et al., 2017; Liu et al., 2020), improving architectures (Hsu et al., 2020; Huang & Li, 2021), and combining features and logits (Wang et al., 2022a). *Distance-based methods* mainly include Mahalanobis distance (Ren et al., 2021) and kNN Yang et al. (2024). Besides these, there are many others, e.g., generating OOD distance measures and pseudo-OOD samples (Wang et al., 2022b), learning better features Hendrycks et al. (2019); Hu et al. (2020), and using contrastive learning (Tack et al., 2020), Our method is very different as it does both OOD detection and online CL.

Continual Learning (CL). Most existing CL focuses on mitigating CF (De Lange et al., 2021; Wang et al., 2023). *Regularization methods* deal with CF by using a regularizer to update parameters critical to previous tasks as little as possible (Kirkpatrick et al., 2017; Xu & Zhu, 2018; Lee et al., 2019). *Replay methods* store some training samples from previous tasks to be used jointly with the new task data (Aljundi et al., 2019; Chaudhry et al., 2019b). Some methods also generate pseudo-replay data to replace the saved replay data (Zhu et al., 2021). *Architectural approaches* make architectural changes. For example, *parameter-isolation* uses masks to prevent the important parameters for previous tasks from being updated, which avoids CF (Serra et al., 2018; Wortsman et al., 2020). *OOD detection* based methods combine OOD detection and parameter-isolation to improve CL accuracy (Kim et al., 2022b). *Prompt-based* methods train a small prompt for each task (Wang et al., 2022c; Smith et al., 2023; Roy et al., 2024). Recent analytic solutions incrementally build a linear model on top of a pre-trained model based on least square or linear discriminant analysis (Hayes & Kanan, 2020; Goswami et al., 2023; McDonnell et al., 2023; Zhuang et al., 2024; Peng et al., 2025; Momeni et al., 2025). Our work is more related to *online CL* (Mai et al., 2022). There are many existing online CL methods (Aljundi et al., 2019; Bang et al., 2022; Koh et al., 2022; Shim et al., 2021; Cha et al., 2021; Zhuang et al., 2024). However, they do not do OOD detection.

OOD Detection + Continual Learning. *Continual OOD detection* is the main direction. Approaches have been proposed for *class-incremental learning* (CIL) (Aguilar et al., 2023; Gummadi et al., 2022; Miao et al., 2025; Momeni et al., 2026), and *task-incremental learning* (TIL) (Kim et al., 2022b; Rios et al., 2022; Liu et al., 2025; Li et al., 2024). Roy et al. (2022) studied new class discovery. A theoretical study was done in (Kim et al., 2025). However, none of them integrate OOD detection and CIL like us to achieve learning after deployment. Additionally, they still have significant CF. Bendale & Boult (2015) has a method for OOD detection and a method for learning new classes given training data. However, they are again not integrated to work after deployment.

[1]The LAFA code is provided in the *Supplementary Materials*.

The mostly related work to LAFA is PLDA (Kaymak et al., 2025). LAFA differs in four aspects. (1) LAFA uses aLDA where means are estimated as directions on the hypersphere, which helps learn and detect OOD samples with limited data. (2) The novelty or OOD score is the aLDA score rather than a separate distance in PLDA. (3) LAFA uses feature projection to improve class separation. (4) Inference reintroduces the feature magnitude to further improve prediction.

3 METHODOLOGY

Problem Statement: In *pre-deployment*, we train an initial model $\mathcal{M}$ using labeled samples from a set of **base classes** $\mathcal{C}$. $\mathcal{M}$ is then deployed to its target application. In *post-deployment*, the system may encounter OOD classes, which are learned incrementally. We track the following class sets:

- $\mathcal{C}$: **base classes** learned in pre-deployment.
- $\mathcal{C}^L$: **well-learned new classes** encountered after deployment that have learned sufficient samples to be regarded as ID classes.
- $\mathcal{C}^E$: **emerging classes**, observed but not yet well-learned.
- $\mathcal{C}^+ = \mathcal{C} \cup \mathcal{C}^L$: the current set of **ID classes**.
- $\mathcal{C}^N = \mathcal{C}^L \cup \mathcal{C}^E$: **new classes** observed after deploying $\mathcal{M}$
- $\mathcal{C}^A = \mathcal{C}^+ \cup \mathcal{C}^E = \mathcal{C} \cup \mathcal{C}^L \cup \mathcal{C}^E$: **all classes** seen so far.

Through incremental learning of the new classes in $\mathcal{C}^N$, $\mathcal{M}$ is updated to cover all classes in $\mathcal{C}^A$.

Optimization Objectives. Post-deployment learning must balance two competing objectives:

1. **Maximize classification accuracy** across all classes, including both the base and new (OOD) classes $\mathcal{C}^N$ encountered after deployment. To learn the OOD classes well, the system needs to detect as many OOD samples as possible to maximize learning opportunities. That is, we want to maximize the recall of OOD detection.
2. **Minimize unnecessary queries:** When a sample is flagged as OOD, the system queries a human or another agent for its ground-truth label. Falsely flagging ID samples as OOD (false positives) wastes human effort and degrades user experience. That is, we also want to maximize the precision of OOD detection.

LAFA consists of four main components: (1) *a frozen PTM* for feature extraction, (2) *a random projection matrix* with non-linear activation, (3) *Angular LDA* for *learning* with a *closed-form incremental solution* (Pang et al., 2005) and using *unit-norm features* (to remove the magnitude information), and (4) a *magnitude-augmented inference system* for classification and OOD detection. The complete pipeline of LAFA is given in Figure 1.

3.1 FEATURE EXTRACTION AND RANDOM PROJECTION

Given an input sample $\mathbf{x} \in \mathbb{R}^{H \times W \times 3}$, we first extract features using a frozen PTM f_θ:

$$\boldsymbol{\phi} = f_\theta(\mathbf{x}) \in \mathbb{R}^d \tag{1}$$

where d is the PTM feature dimension (e.g., $d = 768$). As the PTM is frozen throughout, LAFA avoids CF associated with gradient-based fine-tuning or training.

To improve the linear separability of LDA, we apply a **random projection** with a non-linear activation function for the extracted features to expand the feature dimensions from d to D (Huang et al., 2011). Let $\mathbf{P} \in \mathbb{R}^{d \times D}$ be the fixed random projection matrix with entries drawn from $\mathcal{N}(0, 1)$, where $D > d$ (e.g., $D = 5000$) is the projected dimension. The projected features are computed as:

$$\mathbf{z} = \sigma(\boldsymbol{\phi} \cdot \mathbf{P}) \in \mathbb{R}^D \tag{2}$$

where $\sigma(\cdot)$ is the ReLU activation function applied element-wise. Since the random projection matrix is fixed throughout, it also does not cause any training based CF.

3.2 ANGULAR LDA FOR PRE-DEPLOYMENT LEARNING

Angular LDA learns by computing the **mean direction** (unit-norm) for each class and a **shared covariance** for all classes, which is assumed by all LDA based algorithms. The feature vector

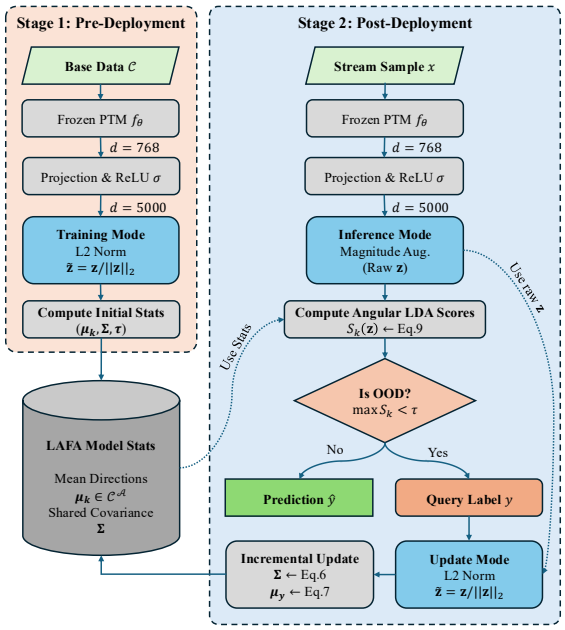

Figure 1: LAFA overview. **Pre-deployment:** statistics built from L2-normalized features. **Post-deployment:** inference uses unnormalized (magnitude-augmented) features; OOD samples trigger a label query and an incremental update.

for each sample is L2-normalized. Specifically, during training, we L2-normalize the **projected features** to constrain them onto the unit hypersphere:

$$\tilde{\mathbf{z}} = \frac{\mathbf{z}}{\|\mathbf{z}\|_2} \in \mathbb{S}^{D-1} \tag{3}$$

This normalization serves two purposes: (1) reduce the variance from the random projection, and (2) provide a principled foundation for OOD detection by ensuring all training samples lie on a low-variance manifold (more details below).

For each base class $k \in \mathcal{C}$, we maintain the following statistics computed from *normalized* features:

Class Mean Direction. Given normalized samples $\{\tilde{\mathbf{z}}_i\}_{i:y_i=k}$ from class k, the class mean direction ($\boldsymbol{\mu}_k$, a unit vector on the hypersphere) is:

$$\boldsymbol{\mu}_k = \frac{\sum_{i:y_i=k} \tilde{\mathbf{z}}_i}{\left\| \sum_{i:y_i=k} \tilde{\mathbf{z}}_i \right\|_2} \tag{4}$$

Shared Covariance Matrix. We estimate a shared within-class covariance matrix across all classes:

$$\boldsymbol{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (\tilde{\mathbf{z}}_i - \boldsymbol{\mu}_{y_i})(\tilde{\mathbf{z}}_i - \boldsymbol{\mu}_{y_i})^\top \tag{5}$$

where N is the total number of training samples. Because $\boldsymbol{\mu}_k$ is a unit direction rather than an arithmetic mean, the residuals $(\tilde{\mathbf{z}}_i - \boldsymbol{\mu}_{y_i})$ do not sum to zero in the Euclidean sense; Eq. 5 is thus a tangent-space proxy for the angular dispersion on $\mathbb{S}^{D-1}$. Especially in high-dimensional spaces, data points occupy a narrow angular region around their respective class means where the hypersphere is locally nearly flat. Consequently, this Euclidean formulation acts as an accurate tangent-space approximation for local angular dispersion (see Appendix A).

Sample Count. For each class k, we maintain the sample count $n_k = |\{i : y_i = k\}|$.

Angular LDA vs. Classical LDA. The classical class mean $\boldsymbol{\mu}_k^{\text{LDA}} = \frac{1}{n_k} \sum_{i:y_i=k} \mathbf{z}_i$ is a point in Euclidean space whose magnitude depends on data scale and class spread. Our $\boldsymbol{\mu}_k$ (Eq. 4) is instead a unit direction on $\mathbb{S}^{D-1}$—the MLE of the von Mises–Fisher mean (Mardia & Jupp, 2009)—encoding only the class's **angular center**. This distinction is critical in the post-deployment regime

with **few OOD samples**: the arithmetic mean and covariance are highly sensitive to outliers and to the magnitude variance introduced by random projection, whereas weighting each sample equally by angular position yields stable estimates of both $\boldsymbol{\mu}_k$ and a better-conditioned $\boldsymbol{\Sigma}$ even with few samples per new class.

3.3 Post-Deployment Incremental Learning

After deployment, LAFA detects OOD samples and queries their labels. When a new labeled sample $(\mathbf{x}, y)$ is obtained, LAFA updates the model statistics **incrementally** (Pang et al., 2005). We still employ Eqs. equation 1, equation 2 and equation 3 to obtain the normalized feature vector $\tilde{\mathbf{z}}$.

Covariance Update. Before updating the class mean, we update the shared covariance using a running average (derivation note is given in Appendix A):

$$\boldsymbol{\Sigma} \leftarrow \frac{N \cdot \boldsymbol{\Sigma} + (\tilde{\mathbf{z}} - \boldsymbol{\mu}_y)(\tilde{\mathbf{z}} - \boldsymbol{\mu}_y)^\top \cdot \frac{N}{N+1}}{N+1} \tag{6}$$

Mean Direction Update. Let $\mathbf{s}_k = \sum_{i:y_i=k} \tilde{\mathbf{z}}_i$ be the running sum of normalized vectors for class k.

$$\mathbf{s}_y \leftarrow \mathbf{s}_y + \tilde{\mathbf{z}}, \quad \boldsymbol{\mu}_y \leftarrow \frac{\mathbf{s}_y}{\|\mathbf{s}_y\|_2} \tag{7}$$

Count Update. $n_y \leftarrow n_y + 1$, $N \leftarrow N + 1$.

Complexity: All updates have $O(D^2)$ complexity and require only $O(|\mathcal{C}^A|D + D^2)$ storage, where $|\mathcal{C}^A|$ is the number of classes seen so far, D is the random projection dimension, $|\mathcal{C}^A|D$ is the storage for saving the class mean directions and D^2 is the storage for the shared covariance. LAFA is thus highly efficient in both time and storage space.

3.4 Test-Time Magnitude Augmentation

This is another novel idea of LAFA, *asymmetric* treatment of feature normalization: while training uses normalized features to build the class statistics, inference uses *raw* (unnormalized) features. This design preserves the discriminative information encoded in the feature magnitude during classification. Given a test sample $\mathbf{x}$, we compute its projected features but do not apply L2-normalization:

$$\mathbf{z} = \sigma(f_\theta(\mathbf{x}) \cdot \mathbf{P}) \in \mathbb{R}^D \tag{8}$$

Angular LDA Score. The classification score for class k is computed using Linear Discriminant Analysis (LDA):

$$h_k(\mathbf{z}) = \mathbf{z}^\top \boldsymbol{\Lambda} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^\top \boldsymbol{\Lambda} \boldsymbol{\mu}_k \tag{9}$$

where $\boldsymbol{\Lambda} = ((1-\epsilon)\boldsymbol{\Sigma} + \epsilon \mathbf{I})^{-1}$ is the regularized precision matrix. The predicted class is:

$$\hat{y} = \arg \max_{k \in \mathcal{C}^+ \cup \mathcal{C}^E} h_k(\mathbf{z}) \tag{10}$$

Magnitude Effect Analysis. Using $\mathbf{z}$ without normalization is equivalent to **augmenting** the normalized feature vector $\hat{\mathbf{z}}$ used in angular LDA with the magnitude r. That is, $\mathbf{z} = r \cdot \hat{\mathbf{z}}$ where $r = \|\mathbf{z}\|_2$ and $\hat{\mathbf{z}} = \mathbf{z}/r$ is the unit direction. Then:

$$h_k(\mathbf{z}) = r \cdot \hat{\mathbf{z}}^\top \boldsymbol{\Lambda} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^\top \boldsymbol{\Lambda} \boldsymbol{\mu}_k \tag{11}$$

The magnitude r scales the angular term $\hat{\mathbf{z}}^\top \boldsymbol{\Lambda} \boldsymbol{\mu}_k$. For well-learned classes this amplifies the discriminative signal; for OOD samples the directional mismatch (low $\hat{\mathbf{z}}^\top \boldsymbol{\Lambda} \boldsymbol{\mu}_k$) is also amplified, aiding OOD detection. The r-independent bias $\frac{1}{2} \boldsymbol{\mu}_k^\top \boldsymbol{\Lambda} \boldsymbol{\mu}_k$ does not overwhelm this: because all $\boldsymbol{\mu}_k$ are unit vectors with shared $\boldsymbol{\Lambda}$, its inter-class spread is bounded by the spectrum of $\boldsymbol{\Lambda}$, and small-r samples are flagged as OOD via Eq. 13 before the bias can drive classification.

Table 1: **Main performance comparison** (F-Score and Accuracy) across datasets in the *Random ID+OOD Data Arrival* setup using DINOv2 as the PTM. **Ablations** are shown at the bottom (-$\mathcal{C}^E$: no Continual OOD Detection -N: no normalization, -M: no magnitude augmentation, -RP: no random projection, [10K]: larger projection size with 10000 dimensions).

	CIFAR100		TinyImageNet		ImageNet-R		CUB		Average
Method	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	Acc.
Joint Fine-Tune		89.74±0.13		85.90±0.10		78.21±0.20		80.92±0.08	83.69
AGEM		72.48±0.43		75.50±0.62		67.02±0.72		49.56±0.89	66.14
GDumb		81.34±3.04		73.56±0.03		63.04±0.80		72.20±0.42	72.53
ER		80.44±0.67		79.94±0.59		69.82±0.36		53.11±0.84	70.83
iCaRL		81.06±0.09		79.23±0.59		67.53±0.57		79.10±0.00	76.73
MIR		81.19±0.29		80.64±0.69		70.20±0.65		54.13±1.39	71.54
GACL		79.44±0.08		78.01±0.13		67.46±0.23		80.19±0.18	76.27
KLDA		82.81±0.04		80.75±0.02		64.13±0.25		80.07±0.17	76.94
LoRanPac		88.52±0.16		83.44±0.19		75.62±0.06		83.32±0.19	82.73
PLDA*		89.18±0.06		84.65±0.01		79.42±0.03		82.54±0.01	83.95
LAFA* (ours)		**91.28**±0.05		**87.09**±0.04		**82.42**±0.12		**83.78**±0.16	**86.14**
PLDA	81.11±0.21	85.80±0.21	82.25±0.16	84.24±0.05	90.89±0.02	79.41±0.04	87.01±0.08	82.15±0.03	82.90
LAFA (ours)	**82.29**±0.24	**89.88**±0.09	**85.13**±0.27	**86.17**±0.34	**92.51**±0.11	**82.38**±0.14	**88.35**±0.42	**83.18**±0.17	**85.40**
Forgetting	-0.07±0.06↓		-0.05±0.09↓		-0.02±0.19↓		-0.10±0.13↓		
LAFA $-\mathcal{C}^E$	71.76±0.53	88.04±0.06	76.41±0.52	84.82±0.17	91.62±0.12	82.32±0.15	87.99±0.17	82.57±0.16	84.44
LAFA −N	70.23±0.27	81.96±0.42	76.47±0.26	82.24±0.19	65.07±0.00	73.64±0.16	66.69±0.03	82.75±0.13	80.15
LAFA −M	78.06±0.28	82.10±0.16	81.28±0.06	80.76±0.06	90.57±0.28	70.85±0.15	87.67±0.07	74.68±0.29	77.10
LAFA −RP	81.18±0.08	87.36±0.06	82.49±0.26	82.86±0.12	90.87±0.14	76.74±0.10	81.69±0.44	76.68±0.18	80.91
LAFA [10K]	82.81±0.40	90.24±0.29	85.53±0.14	86.60±0.10	92.31±0.15	82.09±0.11	87.49±0.97	82.79±0.37	85.43

Table 2: Performance comparison (F-Score and Accuracy) across large or domain-specific datasets in the *Random OOD Data Arrival* setup using DINOv2 as the PTM.

	ImageNet-1K		PathMNIST		CORe50		Food101		Average
Method	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	Acc.
PLDA	65.56±0.11	45.76±0.04	61.19±1.33	78.54±0.32	96.85±0.21	95.69±0.07	75.88±0.49	83.37±0.12	75.84
LAFA (ours)	**70.38**±0.18	**48.92**±0.04	**62.98**±1.65	**80.88**±0.16	**97.36**±0.26	**96.98**±0.14	**81.40**±0.51	**89.65**±0.45	**79.11**

3.5 OOD Detection

LAFA uses the maximum LDA score across all known classes $\mathcal{C}^A$ as the OOD score:

$$\text{OOD-Score}(\mathbf{z}) = \max_{k \in \mathcal{C}^+ \cup \mathcal{C}^E} h_k(\mathbf{z}) \tag{12}$$

A sample is flagged as OOD if it is predicted to be in $\mathcal{C}^E$ or its score falls below a threshold τ:

$$\text{is_OOD}(\mathbf{z}) = \begin{cases} \text{True} & \text{if OOD-Score}(\mathbf{z}) < \tau \\ \text{False} & \text{otherwise} \end{cases} \tag{13}$$

The threshold τ is determined by the system as a low percentile (e.g., 10th percentile) of the OOD scores computed on the base training set. Higher percentiles lead to more skeptical models that unnecessarily query new ID samples, while lower percentiles may cause missed OOD samples.

4 Empirical Evaluation

Datasets. We use four image classification benchmarks and four additional domain-specific datasets. (1) **CIFAR100** (Krizhevsky et al., 2009) contains 100 classes. We use 50 base classes and 50 novel or OOD classes to be learned after deployment. (2) **TinyImageNet** (Le & Yang, 2015) has 200 classes. We use 100 base classes and 100 novel classes. (3) **ImageNet-R** (Hendrycks et al., 2021) contains 200 classes with various artistic renditions (art, cartoons, sculptures, etc.) of ImageNet classes. We use 100 base classes and 100 novel classes. (4) **CUB-200-2011** (Wah et al., 2011) has 200 bird species. We use 100 base and 100 novel classes.

To evaluate scalability and how LAFA performs in specialized domains, we additionally consider four datasets (Table 2): (5) **ImageNet-1K** (Deng et al., 2009) is the full 1,000-class ImageNet at reduced resolution for testing LAFA's scalability. We use 500 base / 500 novel classes. (6)

Table 3: Performance comparison (F-Score and Accuracy) across datasets in the *Random ID+OOD Data Arrival* setup using DeiT-Small-Kim, DINO, and DINOv2 PTMs.

	CIFAR100		TinyImageNet		ImageNet-R		CUB		Average
Method	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	Acc.
PLDA (deit)	65.29±0.35	68.27±0.28	62.21±0.27	58.11±0.13	80.82±0.19	48.09±0.06	62.04±0.75	45.38±1.70	54.96
LAFA (deit)	**70.92**±0.16	**74.64**±0.12	**71.26**±0.15	**64.80**±0.14	**81.71**±0.13	**52.58**±0.30	**78.45**±0.20	**60.46**±0.12	**63.12**
PLDA (dino)	74.89±0.28	77.25±0.20	79.72±0.05	77.20±0.09	85.90±0.19	62.76±0.12	80.00±0.09	60.01±0.07	69.30
LAFA (dino)	**77.67**±0.54	**81.99**±0.18	**82.16**±0.18	**79.45**±0.32	**87.88**±0.10	**65.22**±0.36	**82.76**±0.42	**67.54**±0.27	**73.55**
PLDA (dinov2)	81.11±0.21	85.80±0.21	82.25±0.16	84.24±0.05	90.89±0.02	79.41±0.04	87.01±0.08	82.15±0.03	82.90
LAFA (dinov2)	**82.29**±0.24	**89.88**±0.09	**85.13**±0.27	**86.17**±0.34	**92.51**±0.11	**82.38**±0.14	**88.35**±0.42	**83.18**±0.17	**85.40**

Table 4: Performance comparison (F-Score and Accuracy) across datasets in the *Class-Incremental OOD Data Arrival* setup using DINOv2 as the PTM.

	CIFAR100		TinyImageNet		ImageNet-R		CUB		Average
Method	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	Acc.
AGEM		68.32±0.62		73.83±0.54		63.51±0.89		50.71±1.11	64.09
GDumb		81.45±2.23		71.76±0.41		62.63±0.45		72.01±1.99	71.96
ER		69.53±1.75		73.73±0.55		66.85±0.53		55.60±0.45	66.43
iCaRL		78.38±0.17		74.70±0.27		68.85±1.01		79.03±0.00	75.24
MIR		71.16±0.62		74.56±1.27		67.29±0.41		56.36±0.72	67.34
GACL		79.27±0.11		77.96±0.15		67.69±0.04		80.09±0.22	76.25
KLDA		82.84±0.05		80.77±0.03		64.11±0.25		80.08±0.18	76.95
LoRanPac		88.37±0.13		83.49±0.13		75.66±0.05		83.58±0.23	82.77
PLDA*		89.18±0.04		84.67±0.07		79.44±0.04		82.58±0.04	83.97
LAFA* (ours)		**91.42**±0.09		**87.01**±0.08		**82.49**±0.18		**83.73**±0.05	**86.16**
PLDA	79.47±0.94	85.70±0.43	80.49±0.17	83.62±0.16	89.80±0.03	79.43±0.03	87.66±1.29	82.09±0.01	82.71
LAFA (ours)	**79.54**±0.65	**88.76**±0.57	**80.96**±0.01	**84.63**±0.30	**93.67**±0.18	**82.29**±0.04	**89.02**±1.03	**82.68**±0.23	**84.59**

PathMNIST (Yang et al., 2023) is a histopathology image datasets with 9 tissue types (classes), representing a medical imaging domain. We use 5 base / 4 novel classes. (7) **CORe50** (Lomonaco & Maltoni, 2017) consists images of 50 objects (classes) captured from sequential video frames at multiple angles, a highly realistic scenario dataset. We use 25 base / 25 novel classes. (8) **Food101** (Bossard et al., 2014) has images of 101 food categories. We use 51 base / 50 novel classes.

More details about the datasets are in Appendix B. For **ease of presentation**, we refer to the base classes as ID classes and all novel classes as OOD classes from now on.

4.1 Compared Methods

Joint Fine-Tune. It trains on all data (base + OOD) jointly.

Replay-based CL. *A-GEM* (Chaudhry et al., 2019a) constrains gradients via episodic memory; *GDumb* (Prabhu et al., 2020) stores balanced samples and retrains; *ER* (Rolnick et al., 2019) replays stored samples; *iCaRL* (Rebuffi et al., 2017) combines nearest-mean classification with distillation; *MIR* (Aljundi et al., 2019) selects maximally interfered samples. See Appendix C for PTM adaptation to suit these models.

Analytic CL. *GACL* (Zhuang et al., 2024) uses analytic Gaussian updates; *LoRanPac* (Peng et al., 2025) combines random projection with low-rank adaptation; *KLDA* (Momeni et al., 2025) applies a kernel to LDA for class separability. Note that using LDA to replace aLDA in our system is done in ablation experiments.

Post-Deployment Learning. Most related method to ours, *PLDA* (Kaymak et al., 2025).

Since the CL methods above do not perform OOD detection, we assume the availability of a perfect OOD detector to help them. This represents an ***idealized setting*** for them. We also report **PLDA*** and **LAFA***, which are the idealized versions of PLDA and LAFA.

4.2 Environmental Setups

As in (Kaymak et al., 2025), for all datasets, we use 90% of the training data for base (ID) class training in pre-deployment and reserve 10% of the training data from both the base ID and OOD classes in the post-deployment phase. We evaluate in two setups: (1) **Random ID+OOD Data Arrival**, where base ID and OOD samples arrive in a random stream after deployment. This is the

most realistic setting for post-deployment learning, as in practice, data from any class (old or new) may arrive at any time in an unpredictable order. (2) **Class-Incremental (CIL) OOD Data Arrival**, where OOD classes are divided equally into five sequential tasks mixed with ID data. This is the *most challenging setting* of continual learning Wang et al. (2023). We do not consider the blurry task boundary setting Kang et al. (2022), as it lies between settings (1) and (2).

4.3 Implementation Details

Pre-trained Models Our main PTM is **DINOv2 base** (Oquab et al., 2023), a self-supervised ViT trained on the large curated LVD-142M dataset without class labels, producing 768-dim features with strong transfer across vision tasks, and crucially avoiding ***class-label leakage*** to the end task. We also evaluate with **DINO base** (Caron et al., 2021) and **DeiT-Small-Kim** (Kim et al., 2022a)—a DeiT-Small (Touvron et al., 2021) pre-trained on 611 ImageNet-1k classes after removing the 389 that overlap with CIFAR100 and TinyImageNet, preventing ***image leakage*** between pre-training and pre/post-deployment learning. The backbone is kept frozen throughout; a frozen strong PTM is the standard practice in recent analytic CL (e.g., LoRanPac, GACL, KLDA, PLDA).

First-Session Adaptation (FSA). Following (Peng et al., 2025), we apply first-session adaptation, where the PTM is fine-tuned using the base class data during pre-deployment, which give better results. FSA is applied to LoRanPac, PLDA, and LAFA in our main experiments. Those traditional replay-based baselines already adapt the PTM via adapters on all classes. We report results *without* FSA in Appendix I, showing that LAFA already outperforms the baselines without FSA.

Random Projection. We project the 768-dimensional DINOv2 features into a $D = 5000$-dimensional space using a fixed random projection matrix $\mathbf{P} \in \mathbb{R}^{768\times 5000}$ with entries randomly drawn from $\mathcal{N}(0, 1)$. ReLU activation is applied after projection.

Covariance Regularization. We use shrinkage regularization when computing the precision matrix: $\mathbf{\Lambda} = ((1-\epsilon)\mathbf{\Sigma} + \epsilon\mathbf{I})^{-1}$. Following Ledoit-Wolf style heuristics (Ledoit & Wolf, 2004), we adapt ϵ based on sample size: $\epsilon = 10^{-5}$ for CIFAR100 and TinyImageNet, and $\epsilon = 10^{-4}$ for ImageNet-R and CUB. This adaptive shrinkage provides stronger regularization when covariance estimates are less reliable. Note that with $D = 5000$, $\mathbf{\Sigma}$ has $\sim 1.25 \times 10^7$ entries while the post-deployment sample count N is often comparable to (or below) D; then the sample covariance is severely rank-deficient, and the shrinkage term $\epsilon\mathbf{I}$ is the main mechanism keeping $\mathbf{\Lambda}$ well-conditioned.

OOD Threshold τ. It is set at the 10th percentile of OOD scores computed on the base training set. It flags about 10% of base samples as borderline, a conservative threshold.

Well-learned Class. A novel/OOD class is considered "well-learned" after observing 30 samples, at which point it is added to the set of ID classes $\mathcal{C}^+$.

Evaluation Metrics. We evaluate LAFA using two metrics. For classification, we use **Total Accuracy** across all classes (both ID and OOD, detected after deployment) on the test set after all incremental learning is completed. For OOD detection (post-deployment), we use **F-Score**. All results are mean $\pm$ std over 3 runs. In Appendix D, we explain why we cannot use threshold-free OOD detection metrics like AUROC and AUPRC.

4.4 Results

Random ID+OOD Data Arrival (Table 1). LAFA achieves the best performance across all datasets in both F-Score and accuracy, consistently outperforming both the prior post-deployment method, PLDA, and strong CL baselines, which work in the **idealized setting** with perfect OOD detection (thus no F-score reported). Notably, LAFA's accuracy even exceeds that of joint fine-tuning (or offline retraining) using all the data together, which demonstrates that the proposed approach is highly effective in learning with limited data, even better than joint fine-tuning. This highlights its strong potential for real-world applications. For OOD detection, LAFA obtains better F-Scores across all datasets, demonstrating its ability to balance precision (avoiding unnecessary queries) and recall (capturing learning opportunities).

Forgetting Analysis. It is clear that LAFA avoids CF caused by gradient-based training. However, updating the shared covariance may also result in forgetting due to class decision boundary shifts. We measure it as the accuracy change of base classes before and after learning all OOD classes

(eliminating the effect of learning significantly more classes). As shown in Table 1, LAFA exhibits *negative forgetting* (slight improvement) across all datasets. This shows that the incremental covariance update (Eq. 6) improves rather than corrupts the shared covariance estimate, which is highly desirable as in our setting the data is generally limited.

Domain-Specific and Large-Scale Evaluation (Table 2). We test LAFA on four additional datasets that are either large-scale (ImageNet-1K with 1,000 classes) or belong to specialized domains (PathMNIST for medical imaging, CORe50 for robotic object recognition, Food101 for food classification). These datasets are important because real-world post-deployment learning often involves domain-specialized agents. LAFA consistently outperforms PLDA across all four datasets. Notably, on **CORe50**, which consists of sequential video frames of objects captured from multiple angles LAFA achieves an F-Score of 97.36% and accuracy of 96.98%, demonstrating that the method is particularly well-suited to deployment scenarios involving sequential multi-view observations.

Generalization Across PTMs (Table 3). To validate that LAFA's improvements are not specific to DINOv2, we evaluate with two additional PTMs. LAFA consistently outperforms PLDA with both backbones, with even larger relative improvements on the weaker DINO features.

Class-Incremental OOD Data Arrival (Table 4). In this less likely but more challenging setup, where OOD classes arrive in sequential tasks, LAFA maintains its advantage with consistent improvements over PLDA across all datasets in both accuracy and F-Score.

Runtime and Memory Efficiency. Due to space limits, we report the results in Appendix E.

4.5 Ablation Studies

We conduct ablation studies to understand the contribution of each component in LAFA. Results are shown at the bottom of Table 1. Appendix F gives additional analysis of the results.

Effect of Continual OOD Detection ($-\mathcal{C}^E$). Removing the emerging class set $\mathcal{C}^E$ from OOD detection significantly degrades F-Score, especially on larger datasets. This demonstrates the importance of leveraging emerging class statistics for improved OOD detection.

Effect of Angular Normalization ($-$N). Using standard LDA instead of Angular LDA causes the most substantial performance drops across all metrics.

Effect of Magnitude Augmentation ($-$M). Removing test-time magnitude augmentation (i.e., using normalized features at inference, like during training) causes drops in both accuracy and F-score.

Effect of Random Projection ($-$RP). Removing the random projection layer and applying Angular LDA directly on PTM features results in modest accuracy decreases.

Effect of Projection Dimension ([10K]). Increasing the projection dimension to 10,000 yields improved results on CIFAR100 and TinyImageNet, which have larger numbers of samples per class, but slightly weaker results on the datasets where fewer samples are available per class. This is because larger projection dimensions increase the sparsity and thus requires more data for robustness.

OOD Detection and Well-learned Thresholds Sensitivity. Due to space limits, we report the results in Appendix G and Appendix H.

5 Conclusion

As AI systems move toward more autonomous operations, it is essential to continuously learn after deployment so that the model can evolve in a manner analogous to human learning. To this end, we propose a novel method for the task. By combining feature projection with angular LDA, and test-time magnitude augmentation, our approach supports effective discovery of new tasks and incremental learning without observing catastrophic forgetting. Extensive experimental results demonstrate that LAFA not only outperforms strong baselines but also matches or even exceeds—the performance of offline retraining (joint training) methods that have access to all old and new data.

Limitations: The main limitation of this work is that we did not study AI–human interaction for acquiring class labels for newly detected OOD samples. This process can be complex, involving natural language dialogue, human error, and even adversarial scenarios. We also did not study possible labeling errors in the process. We leave these to our future work.

AI USE STATEMENT

During the preparation of this work, the authors utilized an AI assistant to support the development and debugging of the implementation of the methods, refine the clarity of the written text. The AI assistant was not used to review literature, draft the related work section, or generate references. The authors independently designed the study, conducted all research, verified all factual claims, and take complete responsibility for the content of this paper.

REFERENCES

Eduardo Aguilar, Bogdan Raducanu, Petia Radeva, and Joost Van de Weijer. Continual evidential deep learning for out-of-distribution detection. In *CVPR*, 2023.

Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. *arXiv preprint arXiv:1908.04742*, 2019.

Jihwan Bang, Hyunseo Koh, Seulki Park, Hwanjun Song, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on a contaminated data stream with blurry task boundaries. In *CVPR*, pp. 9275–9284, 2022.

Abhijit Bendale and Terrance Boult. Towards open world recognition. In *CVPR*, pp. 1893–1902, 2015.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *ECCV*, pp. 446–461, 2014.

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

Hyuntak Cha, Jaeho Lee, and Jinwoo Shin. Co2l: Contrastive continual learning. In *CVPR*, pp. 9516–9525, 2021.

Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *ICLR*, 2019a.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. *ICML*, 2019b.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE TPAMI*, 44(7):3366–3385, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255. Ieee, 2009.

Dipam Goswami, Yuyang Liu, Bartłomiej Twardowski, and Joost Van De Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. *Advances in Neural Information Processing Systems*, 36:6582–6595, 2023.

Meghna Gummadi, David Kent, Jorge A Mendez, and Eric Eaton. Shels: Exclusive feature sets for novelty detection and continual learning without class boundaries. In *CoLLaS*, 2022.

Tyler L Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. In *CVPR workshops*, 2020.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.

Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. *NeurIPS*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021.

Yen-Chang Hsu, Yilin Shen, Hongxia Jin, and Zsolt Kira. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *CVPR*, 2020.

Wenpeng Hu, Mengyu Wang, Qi Qin, Jinwen Ma, and Bing Liu. Hrn: A holistic approach to one class learning. *NeurIPS*, 2020.

Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *IEEE Trans. on SMC, Part B (Cybernetics)*, 42(2), 2011.

Rui Huang and Yixuan Li. Mos: Towards scaling out-of-distribution detection for large semantic space. In *CVPR*, 2021.

Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *CVPR*, 2022.

Derda Kaymak, Gyuhak Kim, Tomoya Kaichi, Tatsuya Konishi, and Bing Liu. Learning after model deployment. *ECAI-2025*, 2025.

Gyuhak Kim, Bing Liu, and Zixuan Ke. A multi-head model for continual learning via out-of-distribution replay. In *CoLLaS*, 2022a.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. A theoretical study on solving continual learning. *NeurIPS*, 2022b.

Gyuhak Kim, Changnan Xiao, Tatsuya Konishi, Zixuan Ke, and Bing Liu. Open-world continual learning: Unifying novelty detection and continual learning. *Artificial Intelligence*, 338:104237, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13), 2017.

Hyunseo Koh, Dahyun Kim, Jung-Woo Ha, and Jonghyun Choi. Online continual learning on class incremental blurry task configuration with anytime inference. In *ICLR*, 2022.

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

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7:7, 2015.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004. ISSN 0047-259X. doi: https://doi.org/10.1016/S0047-259X(03)00096-4.

Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *CVPR*, 2019.

Yujie Li, Xin Yang, Hao Wang, Xiangkun Wang, and Tianrui Li. Learning to prompt knowledge transfer for open-world continual learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13700–13708, 2024.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Litian Liu and Yao Qin. Fast decision boundary based out-of-distribution detector. In *ICML*, 2024.

Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. *NeurIPS*, 33:21464–21475, 2020.

Yuhang Liu, Wenjie Zhao, and Yunhui Guo. H2st: Hierarchical two-sample tests for continual out-of-distribution detection. In *CVPR*, 2025.

Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *CoRL*, 2017.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Kanti V Mardia and Peter E Jupp. *Directional statistics*. John Wiley & Sons, 2009.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

Mark D McDonnell, Dong Gong, Amin Parvaneh, Ehsan Abbasnejad, and Anton van den Hengel. Ranpac: Random projections and pre-trained models for continual learning. *NeurIPS*, 2023.

Wenjun Miao, Guansong Pang, Trong-Tung Nguyen, Ruohuan Fang, Jin Zheng, and Xiao Bai. Opencil: Benchmarking out-of-distribution detection in class incremental learning. *Pattern Recognition*, pp. 112163, 2025.

Saleh Momeni, Sahisnu Mazumder, and Bing Liu. Continual learning using a kernel-based method over foundation models. In *AAAI-2025*, volume 39, pp. 19528–19536, 2025.

Saleh Momeni, Changnan Xiao, and Bing Liu. Continual out-of-distribution detection with analytic neural collapse. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 15564–15572, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Shaoning Pang, Seiichi Ozawa, and Nikola Kasabov. Incremental linear discriminant analysis for classification of data streams. *IEEE transactions on Systems, Man, and Cybernetics, part B (Cybernetics)*, 35(5):905–914, 2005.

Liangzu Peng, Juan Elenter, Joshua Agterberg, Alejandro Ribeiro, and Rene Vidal. Loranpac: Low-rank random features and pre-trained models for bridging theory and practice in continual learning. In *ICLR*, 2025.

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *EECV*, 2020.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, 2017.

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.

Amanda Rios, Nilesh Ahuja, Ibrahima Ndiour, Utku Genc, Laurent Itti, and Omesh Tickoo. incdfm: Incremental deep feature modeling for continual novelty detection. In *ECCV 2022*, 2022.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. Experience replay for continual learning. In *NeurIPS*, 2019.

Anurag Roy, Riddhiman Moulick, Vinay K Verma, Saptarshi Ghosh, and Abir Das. Convolutional prompting meets language models for continual learning. In *CVPR*, 2024.

Subhankar Roy, Mingxuan Liu, Zhun Zhong, Nicu Sebe, and Elisa Ricci. Class-incremental novel class discovery. In *ECCV*, pp. 317–333, 2022.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *ICML*, 2018.

Dongsub Shim, Zheda Mai, Jihwan Jeong, Scott Sanner, Hyunwoo Kim, and Jongseong Jang. Online class-incremental continual learning with adversarial shapley value. In *AAAI*, 2021.

James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free continual learning. In *CVPR*, 2023.

Yiyou Sun, Chuan Guo, and Yixuan Li. React: Out-of-distribution detection with rectified activations. *NeurIPS*, 34:144–157, 2021.

Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *NeurIPS*, 33:11839–11852, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. Jul 2011.

Haoqi Wang, Zhizhong Li, Litong Feng, and Wayne Zhang. Vim: Out-of-distribution with virtual-logit matching. In *CVPR*, 2022a.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application, 2023.

Mengyu Wang, Yijia Shao, Haowei Lin, Wenpeng Hu, and Bing Liu. Cmg: A class-mixed generation approach to out-of-distribution detection. *Proceedings of ECML/PKDD-2022*, 2022b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *CVPR*, 2022c.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. *NeurIPS*, 2020.

Ju Xu and Zhanxing Zhu. Reinforced continual learning. In *NIPS*, pp. 899–908, 2018.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.

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

Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *CVPR*, 2021.

Huiping Zhuang, Yizhu Chen, Di Fang, Run He, Kai Tong, Hongxin Wei, Ziqian Zeng, and Cen Chen. GACL: Exemplar-free generalized analytic continual learning. In *NeurIPS*, 2024.

APPENDIX

A NOTE ABOUT EQ. 6

Eq. 6 is the standard incremental covariance update, restructured as a recursive running-average for $O(1)$ streaming updates. The justification in the angular setting as follows: Since all features are L2-normalized to lie on the unit hypersphere $\mathbb{S}^{D-1}$, the covariance captures the within-class angular spread around the mean direction. Importantly, normalization and the use of Euclidean covariance play complementary roles: normalization removes the magnitude variance introduced by random projection, while the Euclidean covariance accurately captures the local angular dispersion because, in high-dimensional spaces, the data occupies a small angular region around each class mean where the hypersphere is locally nearly flat (much like how the Earth's surface appears flat at ground level despite being curved globally). This tangent-space approximation is a well-known property and our negative forgetting results (Table 1) empirically confirm it works well in practice.

Note that when a new class is first observed, the covariance is updated using the feature itself, as the class mean is treated as a zero vector at that point. However, this has a negligible effect considering the massive base initialization and the subsequent post-deployment covariance updates that quickly dilute it. Since we use a shared covariance matrix, the first sample immediately benefits from the robust covariance already estimated from all base classes, mitigating the few-shot variance problem.

B MORE DETAILS ABOUT THE EXPERIMENTAL DATASETS

We conduct experiments on four image classification benchmarks and four additional domain-specific datasets. (1) **CIFAR100** (Krizhevsky et al., 2009) contains 60,000 32×32 images across 100 classes, with 500/100 train/test per class. We use 50 classes as base classes and 50 classes as novel classes to be learned after deployment. (2) **TinyImageNet** (Le & Yang, 2015) 200 classes of 64×64 images with train/test split of 500/50. We use 100 classes as base classes and 100 as novel classes. (3) **ImageNet-R** (Hendrycks et al., 2021) contains 23,918 training images across 200 classes with various artistic renditions (art, cartoons, sculptures, etc.) of ImageNet classes. We use 100 base classes and 100 novel classes. (4) **CUB-200-2011** (Wah et al., 2011) is a bird classification dataset with 5,994 images across 200 bird species. We use 100 base classes and 100 novel classes.

To evaluate scalability and domain specialization, we additionally consider four datasets (Table 2): (5) **ImageNet-1K** (32×32) (Deng et al., 2009) is the full 1,000-class ImageNet at reduced resolution, testing LAFA's ability to scale to large class vocabularies (500 base / 500 novel). (6) **PathMNIST** (Yang et al., 2023) contains 107,180 histopathology images across 9 tissue types, representing a medical imaging domain where specialized autonomous agents are highly desirable (5 base / 4 novel). (7) **CORe50** (Lomonaco & Maltoni, 2017) consists of 164,866 images of 50 objects captured from sequential video frames at multiple angles, making it a highly realistic scenario: an agent could capture images of an object from different viewpoints over time (25 base / 25 novel). (8) **Food101** (Bossard et al., 2014) includes 101,000 images across 101 food categories, a domain-specific benchmark relevant to specialized recognition agents (51 base / 50 novel). These datasets are chosen because they are either large-scale (ImageNet-1K), or they belong to specialized domains (medical, object manipulation, food), demonstrating LAFA's applicability to real-world specialized agents.

C ADAPTING CNN-BASED CL METHODS TO VIT-BASED FEATURES

For adapting CNN-designed CL baselines (ER, A-GEM, iCaRL, MIR) to the frozen ViT feature setting, we used adapters to train these models on top of the frozen ViT features. This is the standard practice adopted in recent CL works.

D CAN WE USE THRESHOLD-FREE OOD DETECTION METRICS?

For OOD detection, there are also threshold-free metrics like AUROC and AUPR. However, we cannot use them as these metrics are designed for static OOD detection settings where the ID (in-

distribution) class set is fixed. Our post-deployment system is a dynamic system where OOD classes transition into ID classes as they become well-learned. This fundamentally violates the assumptions of standard AUROC/AUPR computation — at different points in the deployment, the same class can be OOD or ID. F-Score captures both precision and recall simultaneously, providing a comprehensive evaluation of dynamic OOD detection quality.

E Runtime and Memory Efficiency

All experiments are conducted on a single NVIDIA A6000 GPU, requiring only $\sim$3.5 GB of GPU memory.

Per-sample latency. The average combined per-sample *detect-and-update* cost (OOD detection and incremental statistic update) is $\sim$16.6 ms for PLDA and $\sim$17.4 ms for LAFA on CIFAR100 (100 classes), and $\sim$20.0 ms for LAFA on ImageNet-1K (1000 classes). LAFA thus adds only $\sim$0.8 ms over PLDA, and the cost scales well with the number of classes. Crucially, all of these latencies are **negligible for practical open-world deployments**, where samples arrive randomly over time. This per-sample cost on the order of tens of milliseconds is effectively instantaneous from the user's perspective.

Memory footprint. As discussed in Sec. 3, the storage is bounded by $O(|\mathcal{C}^A|D + D^2)$: the class mean directions ($|\mathcal{C}^A| \times D$) and the shared covariance matrix (D^2), with no replay buffer, which is particularly efficient in scenarios where the number of classes continues to increase.

End-to-end wall-clock time. A complete experiment on TinyImageNet (200 classes), including both the pre-deployment base training and the full post-deployment streaming phase, finishes in approximately 8 minutes on a single A6000 GPU.

F More Analysis of Ablation Experimental Results

We conduct ablation studies to understand the contribution of each component in LAFA. Results are shown at the bottom of Table 1.

Effect of Continual OOD Detection ($-\mathcal{C}^E$). Removing the emerging class set $\mathcal{C}^E$ from OOD detection significantly degrades F-Score across all datasets. This demonstrates the importance of leveraging emerging class statistics for improved OOD detection.

Effect of Angular Normalization ($-$N). Removing L2-normalization during training (using standard LDA instead of Angular LDA) causes the most substantial performance drops across all metrics. This confirms that normalization is critical for handling the high-variance features from random projection and for robust estimation of class statistics with limited OOD samples. The unit hypersphere constraint provides a stable geometric structure essential for both classification and OOD detection.

Effect of Magnitude Augmentation ($-$M). Removing test-time magnitude augmentation (i.e., using normalized features at inference, like during training) causes notable drops in both accuracy and F-score. This confirms that retaining feature magnitude at test time provides valuable discriminative information that complements the angular class structure learned during training.

Effect of Random Projection ($-$RP). Removing the random projection layer and applying Angular LDA directly on PTM features results in modest accuracy decreases. While the PTM features are already highly discriminative, the random projection provides additional separability through dimensionality expansion and non-linear activation.

Effect of Projection Dimension ([10K]). Increasing the projection dimension from 5,000 to 10,000 yields improved results on CIFAR100 and TinyImageNet, which have larger numbers of training samples per class, but slightly lower accuracy and F-Score on ImageNet-R and CUB, where fewer samples are available per class. This reveals an important trade-off: larger projection dimensions increase the representational capacity but also increase sparsity in the higher-dimensional space, requiring more samples to reliably estimate class statistics. When sufficient data is available, the richer representation improves separability; when data is scarce, the curse of dimensionality degrades covariance estimation.

G OOD Detection Threshold Sensitivity

We evaluated LAFA across OOD percentile thresholds (τ) of 5, 10, 20, 30, 40, and 50 across all four datasets.

Table 5: Sensitivity analysis for OOD threshold percentile.

	CIFAR100		TinyImageNet		ImageNet-R		CUB		Average
OOD Perc.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	Acc.
5	83.58±0.57	89.58±0.18	87.06±0.03	85.87±0.09	93.31±0.18	82.38±0.08	89.20±0.12	82.69±0.35	85.13
10	82.25±0.56	89.70±0.20	84.95±0.19	86.28±0.30	92.17±0.28	82.32±0.10	88.21±0.31	83.06±0.33	85.34
20	77.67±0.36	90.02±0.31	80.19±0.51	86.32±0.23	88.03±0.09	82.24±0.08	84.73±0.31	83.35±0.10	85.48
30	74.39±0.61	90.56±0.10	75.66±0.18	86.52±0.10	84.76±0.03	82.58±0.10	82.44±0.23	83.47±0.05	85.78
40	70.49±0.42	90.63±0.18	71.13±0.09	86.57±0.10	80.74±0.05	82.48±0.05	80.05±0.31	83.47±0.21	85.79
50	66.98±0.20	90.61±0.08	66.77±0.07	86.61±0.14	77.36±0.08	82.55±0.14	76.86±0.11	83.67±0.15	85.86

While the F-Score varies more due to the inherent precision–recall trade-off, the overall classification accuracy remains robust across all settings. This strong stability confirms that our projected and normalized angular space effectively learns even with less number of detected OOD samples in lower percentiles, making LAFA insensitive to the exact percentile choice.

H Well-Learned Threshold Sensitivity

We have run an ablation varying well-learned (WL) threshold of 5, 10, 20, 30, 40, and 50 samples.

Table 6: Sensitivity analysis for well-learned class threshold.

	CIFAR100		TinyImageNet		ImageNet-R		CUB		Average
WL Thresh.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	Acc.
5	37.32±0.50	84.61±0.43	41.20±0.63	82.04±0.35	72.27±0.69	79.11±0.40	88.14±0.57	82.66±0.31	82.11
10	57.61±0.51	87.29±0.11	61.56±0.27	84.07±0.23	86.51±0.21	81.51±0.09	88.57±0.25	82.93±0.23	83.95
20	74.96±0.35	89.25±0.32	78.39±0.33	85.62±0.28	92.06±0.26	82.30±0.14	88.06±0.12	82.86±0.19	85.01
30	82.25±0.56	89.70±0.20	84.95±0.19	86.28±0.30	92.17±0.28	82.32±0.10	88.21±0.31	83.06±0.33	85.34
40	86.49±0.31	90.10±0.16	89.00±0.20	86.60±0.22	92.42±0.07	82.50±0.07	88.55±0.39	83.13±0.21	85.58
50	88.60±0.19	90.33±0.44	91.44±0.12	86.62±0.13	92.62±0.08	82.49±0.06	88.49±0.49	83.12±0.17	85.64

Thanks to Angular Normalization drastically reducing the variance, the class mean direction stabilizes rapidly. In the 20–50 range, total accuracy varies by less than $\pm 0.6\%$ on all datasets. The F-Score increases with the threshold, reflecting a precision–recall trade-off: a higher threshold delays classifying a new class until more samples are observed, yielding higher precision but at the cost of more "waste queries" during the learning phase. The value 30 provides a reliable empirical sweet spot, balancing detection precision and learning speed.

I Effect of First-Session Adaptation

Table 7 and Table 8 shows the performance comparison of PLDA and LAFA accross different datasets and PTMs without FSA.

Table 7: Performance comparison (F-Score and Accuracy) across large or domain-specific datasets in the *Random OOD Data Arrival* setup using DINOv2 as the PTM without FSA.

	ImageNet-1K		PathMNIST		CORe50		Food101		Average
Method	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	F-Score	Acc.	Acc.
PLDA	63.81±0.15	40.50±0.06	72.21±0.44	83.32±0.66	89.96±0.34	75.23±0.20	75.32±0.51	80.25±0.16	69.83
LAFA (ours)	**65.56**±0.25	**42.59**±0.04	**77.76**±0.77	**86.81**±0.77	**92.03**±0.70	**84.89**±0.42	**79.92**±0.71	**87.72**±0.33	**75.50**

Table 8: Performance comparison (F-Score and Accuracy) across datasets in the *Random ID+OOD Data Arrival* setup using DeiT-Small-Kim, DINO, and DINOv2 PTMs without FSA.

	CIFAR100		TINYIMAGENET		IMAGENET-R		CUB		AVERAGE
METHOD	F-SCORE	ACC.	F-SCORE	ACC.	F-SCORE	ACC.	F-SCORE	ACC.	ACC.
PLDA (DEIT)	65.89±0.49	62.55±0.18	65.41±0.10	55.72±0.18	73.96±0.02	38.49±0.07	67.19±0.06	42.84±0.00	49.90
LAFA (DEIT)	**68.01**±0.45	**64.38**±0.34	**68.70**±0.36	**56.48**±0.19	**77.22**±0.22	**40.72**±0.13	**74.99**±0.73	**50.24**±0.16	**52.95**
PLDA (DINO)	67.23±0.06	68.98±0.11	74.25±0.25	71.94±0.22	77.51±0.16	51.41±0.10	76.63±0.08	54.36±0.01	61.67
LAFA (DINO)	**73.03**±0.48	**74.25**±0.47	**79.95**±0.24	**75.11**±0.15	**82.36**±0.23	**53.94**±0.16	**80.88**±0.49	**64.78**±0.46	**67.02**
PLDA (DINOV2)	79.69±0.02	85.28±0.05	80.63±0.24	81.70±0.28	87.89±0.15	74.02±0.05	85.39±0.14	81.63±0.01	80.66
LAFA (DINOV2)	**83.02**±0.53	**87.26**±0.25	**84.70**±0.27	**84.32**±0.17	**90.44**±0.26	**77.94**±0.03	**88.34**±0.28	**82.51**±0.10	**83.01**