PATCHBELL: INCREMENTAL LEARNING VIA STATISTICAL MODELING OF LEARNED REPRESENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Class-incremental learning (CIL) aims to continuously incorporate new classes while preserving previously learned knowledge. However, updating model parameters on new-class examples often causes *catastrophic forgetting* on previously learned classes. While recent approaches increasingly leverage pretrained encoders, they mainly rely on image-level embeddings and overlook the statistical structure of patch-level representations. We introduce a probabilistic approach to CIL directly in the frozen representation space, based on class-conditional distributions of *patch-level* embeddings. We show that these patch populations exhibit an approximately Gaussian structure and formalize a class-conditional likelihood model. Patch-level modeling provides orders of magnitude more samples, enabling reliable class estimation from limited data. However, class-conditional patch modeling introduces two challenges: independently modeled class likelihoods are not directly comparable, and many patches capture non-discriminative content. To address these challenges, we propose PATCHBELL (Patch-Based Entropy-weighted Log-Likelihood), which represents each class with fixed-size statistics and classifies images using comparable likelihoods and confidence-weighted patch evidence. The same confidence scores naturally provide spatial explanations. New classes are incorporated in a single pass, without replay or modification of stored class models, eliminating forgetting by design and remaining robust to incremental task composition. Because PATCHBELL requires neither parameter adaptation nor language supervision, it scales naturally to large foundation models and works with both vision-language and vision-only encoders. Experiments on standard CIL benchmarks and multiple foundation models demonstrate state-of-the-art performance with substantially lower incremental update cost.

1 INTRODUCTION

Class-incremental learning (CIL) considers the problem of continuously incorporating new classes while maintaining recognition of previously learned categories (Rebuffi et al., 2017). In the standard setting, training data from previously observed classes may no longer be available, while prediction must be performed over the complete set of classes encountered so far. Conventional approaches address this challenge through replay (Rebuffi et al., 2017; Chaudhry et al., 2019; Buzzega et al., 2020), regularization of parameter updates (Kirkpatrick et al., 2017; Zenke et al., 2017), or task-specific model components (Rusu et al., 2016; Mallya & Lazebnik, 2018; Serra et al., 2018). Although effective, these approaches generally require repeated optimization throughout the incremental learning process, adapting the model to newly introduced classes while degrading performance on previously seen categories, a phenomenon known as *catastrophic forgetting*. This effect becomes more pronounced as the number of incremental tasks, and consequently the number of parameter updates, increases.

Foundation models (Radford et al., 2021; Siméoni et al., 2025) offer a different opportunity. These models are pretrained at massive scale to produce general-purpose representations that transfer effectively to a wide range of downstream recognition tasks. Prior work has leveraged foundation models for CIL in several ways. One line of work focuses on optimization, through adapting prompts, training auxiliary modules, or tuning model parameters when incorporating new classes (Huang et al., 2024; Zheng et al., 2023; Liu et al., 2025; Huang et al., 2025; Wang et al., 2025; Zhou et al., 2025a). These methods still require repeated optimization, remain susceptible to catastrophic forgetting, and

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

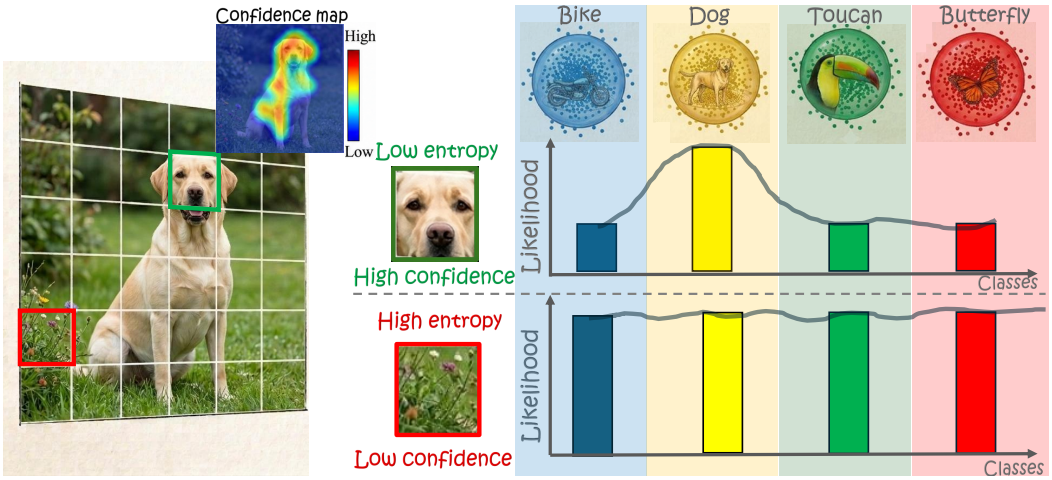


Figure 1: **Illustration of patch confidence and class-wise likelihoods.** Given the class-conditional likelihood distribution of each patch, we identify distinctive patches using entropy-based patch confidence. Generic background patches exhibit high likelihoods across multiple classes, resulting in low confidence (high entropy), whereas discriminative class-specific patches yield high confidence.

become increasingly costly as foundation-model size grows; prompt-based methods are additionally restricted to vision-language models only. Another line of work keeps the foundation model frozen and performs incremental learning directly in its representation space, typically using nearest-class-mean prototypes or covariance-based classifiers applied to full image representations (Petit et al., 2023a;b; Goswami et al., 2023). However, these methods so far rely on relatively simple statistical models of global image features. This leaves open the opportunity to leverage richer statistical structure in frozen foundation-model representations for class-incremental learning.

In parallel, recent work suggests that foundation-models exhibit a surprisingly simple probabilistic structure, enabling likelihood-based modeling directly in the representation space (Balestriero & LeCun, 2025; Betser et al., 2026). These observations, however, have primarily concerned global image representations. We show that approximately Gaussian structure also emerges in class-conditional populations of *patch-level* features. Extending this probabilistic view along these two dimensions introduces distinct challenges. Class-conditional modeling requires likelihoods from independently normalized class distributions to be made comparable, while patch modeling must account for the mixture of discriminative regions and shared background content within each image.

We address the class imbalance challenge by incorporating the log-determinant Jacobian correction associated with each class-specific transformation, yielding comparable class-conditional likelihoods. We address the discriminative regions isolation challenge through cross-class likelihood entropy: shared or non-discriminative patches tend to exhibit high entropy across classes, whereas class-specific patches produce sharper, low-entropy distributions (see Figure 1). Building on these two principles, we introduce PATCHBELL (Patch-Based Entropy-weighted Log-Likelihood) approach which models each class through its patch-feature distribution and aggregates class-conditional patch evidence according to its cross-class uncertainty. This formulation is particularly well suited to CIL: new classes are incorporated without modifying previously learned ones, each image contributes many patches for estimating class statistics, storage remains fixed per class, and the method applies equally to vision-language and vision-only encoders.

We evaluate PATCHBELL across standard CIL benchmarks and multiple foundation-model encoders. The results show that our probabilistic formulation achieves state-of-the-art performance while requiring substantially lower incremental update cost. We examine a diverse set of incremental settings by varying the number of tasks, reducing the amount of training data available per class and introducing imbalance between classes; these experiments further demonstrate the practical robustness of our method. We further ablate design choices and patch aggregation strategies, and evaluate different parameter settings and backbone encoders. Overall, the results demonstrate that exploiting class-conditional patch structure yields a strong, robust, and scalable framework for class-incremental learning.

Our main contributions are:

- We identify and empirically study Gaussian structure in class-conditional patch representations of visual foundation models. We then propose a probabilistic model based on Gaussian likelihoods and cross-class entropy to quantify patch-level class uncertainty.
- Building on this model, we present PATCHBELL, a lightweight CIL framework that incorporates new classes without replay or parameter updates, scales to models of any size, and remains agnostic to language supervision and incremental task splits.
- PATCHBELL naturally obtains intrinsic spatial explainability directly from the same patch-level statistics used for classification, without requiring a separate post-hoc attribution procedure.
- We evaluate PATCHBELL across standard CIL benchmarks and multiple vision-language and vision-only foundation models, outperforming leading baselines and demonstrating strong, robust performance with reduced incremental update cost.

2 RELATED WORK

Visual foundation models have demonstrated that rich visual representations can be learned from large-scale pretraining and transferred effectively across a wide range of downstream tasks (Radford et al., 2021; Zhai et al., 2023; Assran et al., 2023; Siméoni et al., 2025). Theoretical and empirical work has suggested that foundation-model representations exhibit Gaussian structure in high-dimensional spaces (Balestriero & LeCun, 2025; Betser et al., 2026). This probabilistic view has enabled likelihood-based methods for downstream applications in both image and video (Betser et al., 2025; Ben Hayun et al., 2026) representation spaces. However, existing work has primarily considered global representations and has not explored this structure for class-conditional modeling. We extend this to class-conditional distributions of patch-level features, enabling finer-grained probabilistic modeling while retaining the simplicity and scalability of likelihood-based approaches.

Class-incremental learning. Classical approaches can be broadly categorized into replay-based methods, which store or synthesize samples from previous tasks (Rebuffi et al., 2017; Chaudhry et al., 2019; Buzzega et al., 2020); regularization-based methods that constrain parameter updates to mitigate forgetting (Kirkpatrick et al., 2017; Zenke et al., 2017); and parameter-isolation approaches that allocate task-specific model components (Rusu et al., 2016; Mallya & Lazebnik, 2018; Serra et al., 2018). While these methods have made significant progress, they generally rely on updating model parameters throughout incremental sessions, introducing additional computational cost and the risk of catastrophic forgetting.

Frozen-representation incremental learning. A complementary line of work keeps the feature extractor fixed and performs incremental learning directly in the learned representation space (Petit et al., 2023a;b; Wang et al., 2023). These approaches commonly represent classes through prototypes or distributional statistics; notably, FeCAM (Goswami et al., 2023) incorporates class-specific covariance information through Mahalanobis-based classification. In contrast, PATCHBELL models class-conditional distributions at the patch level and uses probabilistic likelihoods together with cross-class entropy to obtain image-level predictions with spatial explanations.

Foundation-model incremental learning. More recently, class-incremental learning has been studied with large pretrained foundation models, particularly vision-language models such as CLIP (Radford et al., 2021). Existing approaches typically rely on incremental parameter optimization, updating prompts, auxiliary modules, or selected parts of the pretrained model (Zheng et al., 2023; Huang et al., 2024; Liu et al., 2025; Huang et al., 2025; Wang et al., 2025; Zhou et al., 2025a; Zhao et al., 2026). Despite their success, these methods remain centered around per task optimization. In this work, we view incremental learning as a statistical modeling problem, where classes are represented directly in the representation space and new classes can be incorporated without modifying existing class models.

3 PATCH-LEVEL PROBABILISTIC MODELING

We first review the statistical tools used throughout our formulation, and then introduce our class-conditional patch-level model.

3.1 PRELIMINARIES

Gaussian modeling and whitening. Let $X \in \mathbb{R}^{d\times M}$ denote a collection of d-dimensional representations modeled as samples from some Gaussian, $x \sim \mathcal{N}(\mu, \Sigma)$. A PCA-based whitening transform W can be applied to center and rescale the representations so that the transformed data have zero mean and identity covariance; details of the construction of W are provided in Appendix A.2. The resulting whitened representations are

$$Y = W(X - \mu), \qquad Y \sim \mathcal{N}(0, I). \tag{1}$$

Dimensionality reduction. When working in a subspace of the full representation space, the whitening transform can be restricted to the top k principal directions. We denote the corresponding transform by $W^{(k)}$ and the resulting representation by $Y^{(k)} \in \mathbb{R}^{k\times M}$.

Likelihood estimation. For a standard multivariate Gaussian, as obtained after whitening Equation (1), the log-likelihood of a k-dimensional representation $y^{(k)}$ is

$$\ell(x) = -\frac{1}{2}\left\|y^{(k)}\right\|_2^2 - \frac{k}{2}\log(2\pi). \tag{2}$$

3.2 CLASS-CONDITIONAL PATCH MODELING

From patch distributions to classification. Let E be a frozen visual encoder, and let $x_p(I) \in \mathbb{R}^d$ denote the representation of patch p from its final encoder layer. For each class $c \in \mathcal{C}$, we collect the patch representations of its training images into a class-specific population X_c. As shown in Table 13 at the appendix, these populations exhibit approximately Gaussian structure.

Using this structure for patch-level classification introduces two distinct challenges: likelihoods from independently normalized class distributions can scale differently across classes, and patch-level representations contain both discriminative object regions and generic content shared across classes. We address these challenges through likelihood correction and cross-class uncertainty, respectively.

Comparable class-conditional likelihoods. For each class, we fit a Gaussian model to its patch population and whiten it in the top k principal directions according to Equation (1). However, each class-specific whitening transform induces a different volume change, making likelihoods evaluated under different class models not directly comparable. We therefore correct Equation (2) with the corresponding log-determinant Jacobian term:

$$\ell_{p,c} = -\frac{1}{2}\left\|W_c^{(k)}(x_p - \mu_c)\right\|_2^2 - \frac{k}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{k}\log\lambda_{c,j}, \tag{3}$$

where $\lambda_{c,j}$ are the retained PCA eigenvalues. This correction makes the class-conditional likelihoods directly comparable across classes. Further details are provided in Appendix A.3.

Cross-class patch uncertainty. Patch-level representations introduce a different challenge: many image patches correspond to background or generic visual content rather than class-discriminative regions. A discriminative patch tends to strongly favor a small number of classes, whereas a generic patch produces similar likelihoods across many classes. We quantify this behavior through the normalized entropy of the class-conditional likelihood distribution. Before computing the entropy, the likelihood scores are normalized as described in Appendix A.4:

$$q_p(c) = \frac{\exp(T_e\widetilde{\ell}_{p,c})}{\sum_{j\in\mathcal{C}}\exp(T_e\widetilde{\ell}_{p,j})}, \qquad H_p = -\frac{1}{\log|\mathcal{C}|}\sum_{c\in\mathcal{C}} q_p(c)\log q_p(c), \qquad w_p = 1 - H_p. \tag{4}$$

Thus, w_p assigns greater confidence to patches whose likelihoods provide discriminative class evidence and suppresses patches that are ambiguous across classes.

Together, likelihood correction and cross-class uncertainty provide comparable and discriminative patch-level evidence across classes. In the following section, we use this formulation to construct PATCHBELL for class-incremental learning.

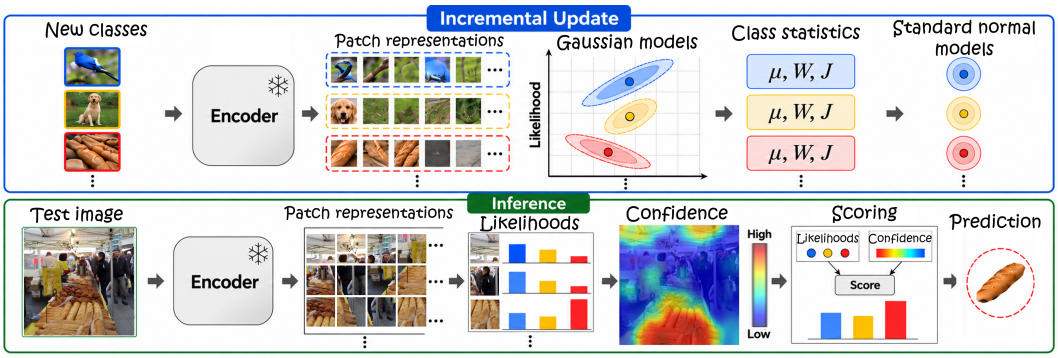


Figure 2: **Overview of PATCHBELL. Top:** new classes are independently modeled from their patch representations using class-specific Gaussian statistics, whitening, and Jacobian correction. **Bottom:** test patches are evaluated across class models, and entropy-based confidence weights guide image-level classification while providing spatial explanations.

4 PATCHBELL

Problem setting. We consider a visual system that continuously encounters new classes over time. In realistic settings, newly introduced classes may differ substantially in both the amount and diversity of available training data: some classes may be observed only a few times, while others may be represented by many heterogeneous examples. To reliably incorporate new knowledge under these conditions, a method should address four key challenges:

(i) incorporating new classes without revisiting previous data or repeatedly optimizing parameters;
(ii) learning reliable class representations from limited data;
(iii) remaining robust to differences in class statistics and data availability;
(iv) identifying the image regions that provide the discriminative evidence for each prediction.

PATCHBELL addresses these requirements using the class-conditional patch statistics introduced in Section 3. The method consists of two modes: (i) *Incremental update* - updating the set of class models when new classes are observed; (ii) *Inference* - evaluating a test image against all currently available class models. An overview of PATCHBELL is provided in Figure 2; full implementation details are in Appendix A.

Incremental update. When a new class is introduced, PATCHBELL performs three steps:

- Extract patch representations from the new class images using the frozen encoder;
- Compute class statistics $(\mu_c, W_c^{(k)}, \log J_c)$ required for likelihood estimation (Equation (3));
- Append the resulting class statistics to the existing set of class models.

Because each image contributes many spatial patches, even a relatively small number of images provides a large population of feature samples for estimating the class distribution. Updating a class requires only a single forward pass over its images, with no parameter optimization, no revisiting of previous-class images, and no modification of previously stored class models. Since classes are modeled independently, the resulting models are also unaffected by how classes are grouped or partitioned across incremental tasks. The complete update procedure is given in Algorithm 1.

Inference. Given a test image, PATCHBELL performs four steps:

- Extract patch representations using the frozen encoder;
- Compute the class-conditional likelihoods ($\{\ell_{p,c}\}$) of each patch using Equation (3);
- Compute entropy H_p, quantifying the uncertainty of each patch across classes, and define its confidence (w_p) using Equation (4);

- Aggregate the confidence-weighted patch likelihoods into a class score:

$$S_c(I) = \sum_{p=1}^{P} w_p \, \ell_{p,c}, \qquad \hat{c} = \arg\max_{c \in \mathcal{C}} S_c(I). \tag{5}$$

Patches that strongly favor a particular class have low entropy and therefore contribute more to the final decision, while ambiguous patches are downweighted. Because these confidence weights retain the spatial organization of the patches, they also reveal which image regions provide the most discriminative evidence for the prediction. Examples of the resulting explanation maps are shown in Figure 6. The complete inference procedure is given in Algorithm 2.

5 EVALUATIONS

We evaluate PATCHBELL from several complementary perspectives. We first compare its class-incremental learning performance against existing methods on standard benchmarks. We then evaluate the method across a diverse set of incremental settings, including task partitioning, data availability, and class imbalance. Finally we study sensitivity through controlled ablations and examine the quality of the intrinsic spatial explanations produced by the patch-level formulation.

5.1 SETUP

Benchmarks and incremental protocol. We evaluate on ImageNet-derived class-incremental learning benchmarks, including ImageNet-R (Hendrycks et al., 2021), ImageNet100, a 100-class subset of ImageNet-1K, and ImageNet-1K (Deng et al., 2009). We follow the common class-incremental learning protocol, where the label space is divided into a sequence of disjoint *tasks*, each containing an equal subset of classes. At each task, only examples from the newly introduced classes are available, while evaluation is performed over all classes observed so far without access to task identity. Unless stated otherwise, we consider the non-replay setting.

Implementation details. We evaluate PATCHBELL using both vision-language and vision-only foundation encoders. Our main experiments use DINOv3 ViT-L/16 (Siméoni et al., 2025). The visual encoder remains frozen, and training images are processed in a single pass to estimate the class-wise patch statistics. We use a default whitening dimension of $k = 500$ in all main experiments, unless stated otherwise. Standard model-specific details, including patch count, embedding dimensions, and input pre-processing, follow the original backbone configurations and are summarized in Table 2 in the Appendix.

Baselines. We compare PATCHBELL against a broad set of recent CIL methods spanning data replay, feature replay, and non-replay settings. For RAPF (Huang et al., 2024), MG-CLIP (Huang et al., 2025), and RSIAT (Zhao et al., 2026), we use the official implementations and reproduce the results using the matched incremental-task configurations. For CLAP (Jha et al., 2024), we report the published results where available. For the remaining methods, we report published results under the corresponding benchmark protocols. Further implementation details for the reproduced baselines are provided in Appendix B.2.

Metrics. We report *Avg*, the average classification accuracy across incremental tasks, and *Last*, the accuracy after the final task. We regard *Last* as the primary metric, since it measures the final recognition performance over the complete set of classes encountered throughout the incremental process. *Avg* provides a complementary summary of performance across intermediate stages.

5.2 MAIN CIL EVALUATION

Table 1 presents the main CIL comparison, including replay and non-replay methods, as well as frozen-encoder baselines. In this experiment we follow the standard 10-task setting for all datasets, with 10 classes introduced per task for ImageNet100, 20 classes per task for ImageNet-R, and 100 classes per task for ImageNet-1K. For each dataset, the full training set of each newly introduced class is used, while evaluation is performed on the corresponding validation set. Across all three benchmarks, PATCHBELL achieves excellent performance in both *Avg* and *Last*, outperforming all compared replay, non-replay, and frozen-encoder baselines on ImageNet-R and ImageNet100. On

Table 1: **Class-incremental learning comparison.** Conducted on ImageNet-R, ImageNet100, and ImageNet-1K. *Avg* and *Last* denote the average accuracy across incremental tasks and the last accuracy, respectively. Best result per column is in **bold**, second best is underlined.

Method	Replay	ImageNet-R		ImageNet100		ImageNet-1K	
		Avg.	Last	Avg.	Last	Avg.	Last
PROOF (Zhou et al., 2025b)	Data replay	82.83	77.05	84.71	72.48	76.23	65.26
CLAP (Jha et al., 2024)		85.77	79.98	87.76	79.16	81.02	72.77
SLCA (Zhang et al., 2023)	Feature replay	75.92	70.37	78.63	59.92	79.10	68.27
RAPF (Huang et al., 2024)		86.07	80.27	88.18	79.98	81.57	72.35
L2P++ (Wang et al., 2022c)	Non-replay	81.67	75.98	80.51	67.22	79.30	69.60
DualPrompt (Wang et al., 2022b)		82.01	75.77	80.65	67.38	79.39	69.79
CODA (Smith et al., 2023)		78.00	67.52	64.13	34.76	76.99	66.96
Continual-CLIP (Thengane et al., 2022)		79.12	72.00	84.98	75.40	72.96	64.44
Aper-Adapter (Zhou et al., 2025a)		78.65	71.35	85.84	76.40	76.60	68.74
MOE4CL (Yu et al., 2024)		85.28	80.77	86.39	76.66	81.29	72.73
CLAP* (Jha et al., 2024)		81.22	75.80	81.07	72.00	75.85	67.36
MagMax (Marczak et al., 2024)		87.13	80.85	86.33	75.92	80.74	71.31
MG-CLIP (Huang et al., 2025)		87.82	82.98	87.41	78.10	81.87	73.78
RSIAT (Zhao et al., 2026)		87.55	82.98	91.05	83.60	**87.48**	**81.79**
Mahalanobis	Frozen Encoders	3.02	1.62	88.38	80.34	80.98	74.36
Text	Baselines	87.18	82.70	87.68	79.54	75.03	71.00
Patch Majority	Non-replay	75.19	70.33	77.24	67.84	76.18	74.41
PATCHBELL	Non-replay	**91.25**	**89.05**	**92.25**	**87.28**	85.13	81.62

ImageNet-1K, PATCHBELL performs comparably to RSIAT, the most recent competing method, while requiring substantially lower incremental update cost and exhibiting much lower sensitivity to task decomposition; see Figure 3A and Table 14.

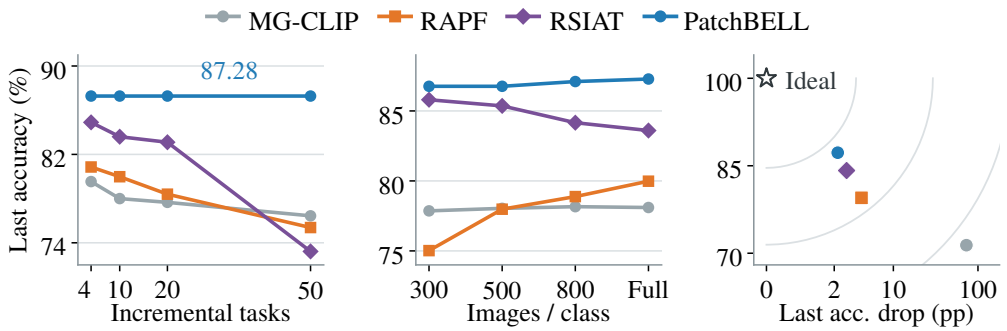
5.3 SENSITIVITY TO THE INCREMENTAL SETTING

Beyond the standard benchmark setting, we evaluate how PATCHBELL and other leading baselines behave under different incremental-learning conditions. We vary the task partition, the amount of training data available per class, and add class imbalance to test robustness beyond the default protocol. Figure 3 summarizes these complementary stress tests. The complete numerical results are reported in Tables 3, 4, 5, and 6 in the Appendix.

Number of incremental tasks. To test sensitivity to task partitioning, we keep the same 100 classes fixed while varying the number of incremental tasks, with fewer classes introduced per task as the partition becomes finer. This allows us to evaluate how performance changes as the incremental sequence becomes increasingly fine-grained. As shown in Figure 3A, PATCHBELL is agnostic to task partitions. In contrast, the other baselines, particularly RSIAT, degrade substantially as the number of tasks increases and fewer classes are introduced per task, indicating greater sensitivity to task granularity.

Training samples per class. We also examine the effect of training-set size on the performance of PATCHBELL and the competing methods. For each class, we vary the number of available training images from a limited-data regime of 300 samples per class up to the full training set. Figure 3B shows that most methods remain relatively stable as the amount of class-specific training data is reduced, with only modest degradation at smaller data sizes. RAPF is notably more sensitive, whereas PATCHBELL maintains strong performance across the evaluated data regimes.

Controlled class imbalance. We further evaluate robustness to class imbalance in a controlled setting. Out of 10 classes per task, one is restricted to only 100 training images, while the remaining classes use their full training sets. We use the same reduced classes and training samples for all methods, allowing a direct comparison of their sensitivity to underrepresented classes. Because only one of the ten classes in each task is underrepresented, overall accuracy can mask substantial degradation on the affected classes. We therefore report both overall *Last* accuracy and the *Last* accuracy drop on the underrepresented classes. As shown in Figure 3C, PATCHBELL achieves the highest overall accuracy while exhibiting the smallest degradation on the underrepresented classes, placing it closest to the ideal operating point of high accuracy and zero imbalance-induced drop.



(A) Number of incremental tasks (B) Training images per class (C) Controlled class imbalance

Figure 3: **Sensitivity to the incremental settings on ImageNet100.** We evaluate sensitivity to (A) the number of incremental tasks, (B) the number of training images per class, and (C) class imbalance. In (C), each method is characterized by its overall *Last* accuracy and the corresponding *Last*-accuracy drop on the underrepresented classes; the ideal operating point is the top-left corner. Complete numerical results are reported in Tables 3, 4, 5, and 6 in the Appendix.

MG-CLIP is particularly sensitive to the imbalance, while the remaining methods show smaller but still more pronounced degradation than PATCHBELL.

Taken together, these experiments highlight distinct weaknesses of the competing methods: RSIAT is sensitive to task granularity, RAPF to limited class data, and MG-CLIP to class imbalance. In contrast, PATCHBELL remains stable across all three settings, indicating substantially lower sensitivity to the structure and data conditions of the incremental stream.

5.4 Ablations and Additional Analysis

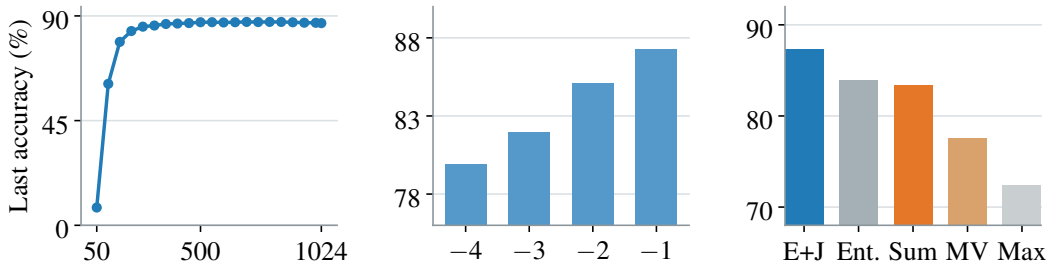
We further analyze PATCHBELL from multiple perspectives to better understand its individual components, design choices, and robustness. The main method ablations are summarized in Figure 4; complete numerical results and additional analyses are provided in the Appendix.

Whitening dimension. We vary the retained whitening dimension k from 50 up to the full DINOv3 ViT-L/16 representation dimension of 1024. A single value of k is used for all classes. As shown in Figure 4A, performance increases rapidly with the retained dimension and then largely saturates at higher dimensions. We therefore use $k = 500$ as the default setting, providing near-saturated performance while retaining a substantially more compact representation. Complete numerical results are reported in Table 8 in the Appendix.

Foundation-model backbone. We then evaluate PATCHBELL across foundation-model families and model scales, including CLIP (B/16 and L/14) (Radford et al., 2021), DINOv3 (B/16 and L/16) (Siméoni et al., 2025), SigLIP (Zhai et al., 2023), and I-JEPA (Assran et al., 2023). The results demonstrate that the same statistical formulation transfers across both vision-language and vision-only encoders and naturally benefits from stronger frozen representations. The complete backbone comparison is reported in Table 9 in the Appendix.

Encoder layer. We also estimate the class distributions using representations extracted from each of the last four encoder layers. As shown in Figure 4B, the final encoder layer achieves the best performance, while accuracy degrades gradually when moving to earlier layers. This indicates that later semantic representations are most effective for class-conditional patch modeling, although the method remains reasonably robust across depth. Complete numerical results are reported in Table 10 in the Appendix.

Jacobian correction. We further compare the proposed class-conditional likelihood model with and without the Jacobian correction in Equation (3). We also evaluate this correction under controlled class imbalance, where differences in class statistics and data availability make likelihood comparability particularly important. The results consistently support the correction, with its contribution becoming especially pronounced for underrepresented classes. The complete numerical comparison is reported in Table 11 in the Appendix.



(A) Whitening dimension (B) Encoder layer (C) Aggregation method

Figure 4: **Method ablations on ImageNet100.** We evaluate the effect of the retained whitening dimension (A), the encoder layer used for patch extraction (B), and the likelihood and patch-aggregation components of PATCHBELL. In (C), *Ent, E* denote entropy weighting, *J* the Jacobian correction, *Sum* an unweighted sum of patch likelihoods, *MV* majority voting over patch predictions, and *Max* the maximum patch likelihood. Complete numerical results are reported in Tables 8, 10, and 12, respectively, in the Appendix.

Patch aggregation. We compare the proposed entropy-weighted patch likelihood aggregation with alternative patch-level aggregation rules. As shown in Figure 4C, the complete formulation combining entropy weighting with the Jacobian correction provides the strongest behavior. Replacing the uncertainty-aware aggregation with simpler alternatives consistently degrades performance. These results show that PATCHBELL's gains rely on both comparable class-conditional likelihoods and uncertainty-aware patch weighting.

Additional analyses in Appendix B.5 show that whitening consistently moves the class-conditional patch distributions toward Gaussian behavior, both along individual representation coordinates and random one-dimensional projections. Table 14 further shows that PATCHBELL provides the fastest end-to-end incremental updates despite its larger method overhead. Additional frozen-encoder baselines show that the gains are not explained by the frozen backbone alone, but by the proposed class-conditional patch modeling and aggregation.

5.5 PREDICTION INTERPRETATION

A useful by-product of the patch-level formulation is an intrinsic spatial explanation of the prediction. We evaluate these explanations both with respect to PATCHBELL's own classification decisions and more generally as visual importance maps, comparing to baseline explanation methods. The resulting maps identify both sufficient and decision-critical regions, and retain this behavior when evaluated with external classifiers. The complete evaluation protocol and quantitative results, together with qualitative examples, are provided in Appendix C.

6 DISCUSSION AND CONCLUSIONS

We introduced PATCHBELL, a probabilistic framework for class-incremental learning that models class-conditional patch representations in frozen foundation models. Building on their Gaussian structure, PATCHBELL performs likelihood-based classification using a Jacobian correction for comparable class likelihoods and cross-class entropy for confidence-weighted patch aggregation. New classes are incorporated in a single pass, without replay or parameter optimization, while the same patch statistics provide intrinsic spatial explanations. Across ImageNet-R, ImageNet100, and ImageNet-1K, PATCHBELL achieves state-of-the-art performance while substantially reducing incremental update cost. It also remains robust across changes in task granularity, limited training data, and class imbalance, and is applicable to both vision-language and vision-only encoders regardless of backbone size. The current formulation assumes that the frozen representation space remains sufficiently informative for all newly encountered classes; extending the framework to settings with larger domain shifts or evolving representations is an important direction for future work. These results show that statistical modeling of frozen representations provides a simple, efficient, and scalable alternative to repeated parameter adaptation, while opening a broader direction for continual learning directly in pretrained representation spaces.

AI USE STATEMENT

Large language models were used for writing assistance, such as improving language clarity, correcting grammar, generating illustrations, and for code generation, limited to implementing specified functions or refactoring existing code according to explicit instructions. All experimental design choices, analyses, interpretations, and conclusions were made by the authors. The authors are fully responsible for all content, claims, and conclusions presented in this work.

REPRODUCIBILITY STATEMENT

We provide detailed algorithmic descriptions of PATCHBELL, together with the complete update and inference procedures, implementation details, hyperparameters, backbone configurations, and experimental protocols in the Appendix. These details are intended to enable faithful reproduction of the reported results. Upon acceptance, we will release the official implementation and the code required to reproduce the main experiments.

REFERENCES

Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4190–4197. Association for Computational Linguistics, 2020.

Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 15619–15629, 2023.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Omer Ben Hayun, Roy Betser, Meir Yossef Levi, Levi Kassel, and Guy Gilboa. Training-free detection of generated videos via spatial-temporal likelihoods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16299–16310, 2026.

Roy Betser, Meir Yossef Levi, and Guy Gilboa. Whitened CLIP as a likelihood surrogate of images and captions. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4069–4095. PMLR, 2025.

Roy Betser, Eyal Gofer, Meir Yossef Levi, and Guy Gilboa. Infonce induces gaussian distribution. In *The Fourteenth International Conference on Learning Representations*, 2026.

Walid Bousselham, Angie Boggust, Sofian Chaybouti, Hendrik Strobelt, and Hilde Kuehne. Legrad: An explainability method for vision transformers via feature formation sensitivity. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20336–20345, 2025.

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020.

Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, P Dokania, P Torr, and M Ranzato. Continual learning with tiny episodic memories. In *Workshop on Multi-Task and Lifelong Reinforcement Learning*, 2019.

Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 782–791, 2021.

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

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8340–8349, 2021.

Linlan Huang, Xusheng Cao, Haori Lu, and Xialei Liu. Class-incremental learning with clip: Adaptive representation adjustment and parameter fusion. In *European Conference on Computer Vision*, pp. 214–231. Springer, 2024.

Linlan Huang, Xusheng Cao, Haori Lu, Yifan Meng, Fei Yang, and Xialei Liu. Mind the gap: Preserving and compensating for the modality gap in clip-based continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3777–3786, 2025.

Saurav Jha, Dong Gong, and Lina Yao. CLAP4CLIP: Continual learning with probabilistic finetuning for vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 129146–129186, 2024.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Wenzhuo Liu, Fei Zhu, Longhui Wei, and Qi Tian. C-clip: Multimodal continual learning for vision-language model. In *The Thirteenth International Conference on Learning Representations*, 2025.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. MagMax: Leveraging model merging for seamless continual learning. In *European Conference on Computer Vision*, pp. 379–395. Springer, 2024.

Gregoire Petit, Adrian Popescu, Eden Belouadah, David Picard, and Bertrand Delezoide. Plastil: Plastic and stable exemplar-free class-incremental learning. In *Conference on lifelong learning agents*, pp. 399–414. PMLR, 2023a.

Grégoire Petit, Adrian Popescu, Hugo Schindler, David Picard, and Bertrand Delezoide. FeTrIL: Feature translation for exemplar-free class-incremental learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 3911–3920, 2023b.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Samyak Rawlekar, Amitabh Swain, Yujun Cai, Yiwei Wang, Ming-Hsuan Yang, and Narendra Ahuja. Finding distributed object-centric properties in self-supervised transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31326–31335, 2026.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

David Schinagl, Christian Fruhwirth-Reisinger, Alexander Prutsch, Samuel Schulter, and Horst Possegger. Learn to rank: Visual attribution by learning importance ranking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pp. 618–626, 2017.

Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International conference on machine learning*, pp. 4548–4557. PMLR, 2018.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11909–11919, 2023.

Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. CLIP model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022.

Qi-Wei Wang, Da-Wei Zhou, Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Few-shot class-incremental learning via training-free prototype calibration. *Advances in neural information processing systems*, 36: 15060–15076, 2023.

Yan Wang, Da-Wei Zhou, and Han-Jia Ye. Integrating task-specific and universal adapters for pre-trained model-based class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 806–816, 2025.

Yangtao Wang, Xi Shen, Shell Xu Hu, Yuan Yuan, James L. Crowley, and Dominique Vaufreydaz. Self-supervised transformers for unsupervised object discovery using normalized cut. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14543–14553, 2022a.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pp. 631–648. Springer, 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 139–149, 2022c.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23219–23230, 2024.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International conference on machine learning*, pp. 3987–3995. Pmlr, 2017.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Gengwei Zhang, Liyuan Wang, Guoliang Kang, Ling Chen, and Yunchao Wei. SLCA: Slow learner with classifier alignment for continual learning on a pre-trained model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19148–19158, 2023.

Jiarui Zhao, Libo Huang, Xiangqi Li, Zhulin An, Chuanguang Yang, Yu Wang, Boyu Diao, and Yongjun Xu. Representation-steered incremental adapter-tuning for class-incremental learning with pre-trained models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18010–18020, 2026.

Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 19125–19136, 2023.

Da-Wei Zhou, Zi-Wen Cai, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Revisiting class-incremental learning with pre-trained models: Generalizability and adaptivity are all you need. *International Journal of Computer Vision*, 133(3):1012–1032, 2025a.

Da-Wei Zhou, Yuanhan Zhang, Yan Wang, Jingyi Ning, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Learning without forgetting for vision-language models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 47(6):4489–4504, 2025b. doi: 10.1109/TPAMI.2025.3540889.

OVERVIEW

The Appendix is organized into three main parts. Appendix A provides additional implementation details for PATCHBELL, including patch extraction, class-specific whitening, likelihood correction, image-level normalization, and the complete update and inference algorithms. Appendix B provides additional experimental details and results, including backbone configurations, baseline implementations, sensitivity analyses, method ablations, Gaussianity analysis, and computational efficiency. Appendix C provides the complete explainability evaluation.

For convenient access to the main quantitative results, the sensitivity experiments are reported in Tables 3–7, the method ablations in Tables 8–12, computational efficiency in Table 14, Gaussianity in Table 13, and explainability in Tables 15 and 16.

A METHOD DETAILS

A.1 PATCH EXTRACTION

Given an input image I, we extract spatial patch representations from the frozen visual encoder E. In all main experiments, we use the spatial patch tokens from the final encoder output. The global class token and other non-spatial tokens are discarded.

We denote the resulting patch representations by

$$E(I) = \{x_p(I) \in \mathbb{R}^d\}_{p=1}^{P}, \tag{6}$$

where d is the representation dimension and P is the number of spatial patches.

For a class c with N_c training images, all patch representations are collected as columns in

$$X_c = \begin{bmatrix} x_1^{(c)} & \cdots & x_{M_c}^{(c)} \end{bmatrix} \in \mathbb{R}^{d \times M_c}, \qquad M_c = N_c P. \tag{7}$$

No parameters of the visual encoder are updated during this process.

A.2 PCA WHITENING AND DIMENSIONALITY REDUCTION

For each class c, we estimate the empirical mean of its patch population,

$$\mu_c = \frac{1}{M_c} \sum_{i=1}^{M_c} x_i^{(c)}, \tag{8}$$

and the empirical covariance

$$\Sigma_c = \frac{1}{M_c} \sum_{i=1}^{M_c} (x_i^{(c)} - \mu_c)(x_i^{(c)} - \mu_c)^\top. \tag{9}$$

Equivalently, defining the centered class matrix

$$\widehat{X}_c = X_c - \mu_c \mathbf{1}^\top, \tag{10}$$

we have

$$\Sigma_c = \frac{1}{M_c} \widehat{X}_c \widehat{X}_c^\top. \tag{11}$$

We compute the eigendecomposition

$$\Sigma_c = V_c \Lambda_c V_c^\top, \tag{12}$$

where the eigenvalues are ordered from largest to smallest.

To obtain a compact class representation, we retain the leading k principal directions. Let $V_{c,k} \in \mathbb{R}^{d \times k}$ contain the corresponding eigenvectors and let $\Lambda_{c,k} \in \mathbb{R}^{k \times k}$ contain their associated eigenvalues.

The class-specific PCA-whitening transform is defined as

$$W_c^{(k)} = \Lambda_{c,k}^{-1/2} V_{c,k}^\top \in \mathbb{R}^{k \times d}. \tag{13}$$

A patch x is transformed under class c according to

$$y_c^{(k)} = W_c^{(k)}(x - \mu_c) \in \mathbb{R}^k. \tag{14}$$

The retained dimension k is shared across classes. Unless stated otherwise, we use $k = 500$ in all main experiments. Its influence on classification performance is evaluated in Section 5.4.

A.3 Likelihood and Jacobian Correction

For a test patch x_p, its representation under the statistical model of class c is

$$y_{p,c}^{(k)} = W_c^{(k)}(x_p - \mu_c). \tag{15}$$

Following the Gaussian approximation in the whitened representation space, the retained PCA coordinates are modeled as approximately standard normal. The corresponding log-likelihood in the retained subspace is

$$\ell_{\text{white}}(x_p \mid c) = -\frac{1}{2} \left\| y_{p,c}^{(k)} \right\|_2^2 - \frac{k}{2} \log(2\pi). \tag{16}$$

Because each class is modeled using independently estimated statistics, likelihoods evaluated in different class-specific coordinate systems must also account for differences in covariance volume.

For the retained PCA coordinates, the corresponding volume correction is

$$\log J_c = -\frac{1}{2} \sum_{j=1}^{k} \log \lambda_{c,j}. \tag{17}$$

Consequently, the corrected class-conditional patch log-likelihood is

$$\ell_{p,c} = -\frac{1}{2} \left\| W_c^{(k)}(x_p - \mu_c) \right\|_2^2 - \frac{k}{2} \log(2\pi) - \frac{1}{2} \sum_{j=1}^{k} \log \lambda_{c,j}. \tag{18}$$

Equivalently,

$$\ell_{p,c} = -\frac{1}{2} \left\| y_{p,c}^{(k)} \right\|_2^2 - \frac{k}{2} \log(2\pi) + \log J_c. \tag{19}$$

Since k may be smaller than the original representation dimension d, the likelihood above is defined over the retained PCA subspace rather than through a full-dimensional invertible change of variables. The Jacobian term therefore corresponds to the covariance-volume correction within this retained subspace.

Implementation convention. The implementation computes the PCA transform using the standard sample covariance normalization $1/(M_c - 1)$ and estimates a scalar residual variance in the whitened coordinates. This is algebraically equivalent to the $1/M_c$ covariance convention used above. Specifically, if $\Sigma_c^{\text{sample}} = \frac{M_c}{M_c - 1} \Sigma_c$, then the corresponding whitened-coordinate variance is $\sigma_c^2 = (M_c - 1)/M_c$, such that

$$\sigma_c^2 \Sigma_c^{\text{sample}} = \Sigma_c. \tag{20}$$

Thus, the implemented likelihood is equivalent to the formulation above while using the conventional sample-covariance estimator internally.

A.4 IMAGE-LEVEL LIKELIHOOD NORMALIZATION

Before computing patch entropy, we normalize the likelihood values independently for each candidate class over the patches of the test image.

For image I and class c, let

$$\mu_c^{(I)} = \frac{1}{P} \sum_{p=1}^{P} \ell_{p,c}, \tag{21}$$

and

$$\sigma_c^{(I)} = \sqrt{\frac{1}{P-1} \sum_{p=1}^{P} \left(\ell_{p,c} - \mu_c^{(I)} \right)^2}. \tag{22}$$

We compute

$$\widetilde{\ell}_{p,c} = \frac{\ell_{p,c} - \mu_c^{(I)}}{\sigma_c^{(I)} + \epsilon}. \tag{23}$$

For numerical stability, if $\sigma_c^{(I)} = 0$, it is replaced by 1 before normalization.

This normalization is used only for computing the cross-class entropy and the resulting patch-confidence weights. The final image-level class score is computed using the original, unnormalized likelihood values $\ell_{p,c}$.

The normalized likelihoods are converted into a class distribution for each patch,

$$q_p(c) = \frac{\exp(T_e \widetilde{\ell}_{p,c})}{\sum_{j \in \mathcal{C}} \exp(T_e \widetilde{\ell}_{p,j})}, \tag{24}$$

where T_e controls the sharpness of the distribution.

In all experiments, we use $T_e = 5$ and $\epsilon = 10^{-6}$.

For numerical stability, the posterior probabilities are clipped from below by ϵ before evaluating the logarithm. The normalized entropy and confidence weight are then

$$H_p = -\frac{1}{\log |\mathcal{C}| + \epsilon} \sum_{c \in \mathcal{C}} q_p(c) \log q_p(c), \qquad w_p = 1 - H_p. \tag{25}$$

Thus, patches whose normalized likelihood distributions are concentrated on a small number of classes receive larger weights, while ambiguous patches are suppressed.

A.5 ALGORITHM DETAILS

At incremental session s, let $\Delta\mathcal{C}_s$ denote the set of newly introduced classes and let

$$\mathcal{C}_s = \bigcup_{t=1}^{s} \Delta\mathcal{C}_t \tag{26}$$

denote the complete set of classes observed up to session s.

Algorithm 1 PATCHBELL incremental update

Require: Frozen visual encoder E, retained dimension k, new class set $\Delta\mathcal{C}_s$, current class statistics $\mathcal{M}$

for each new class $c \in \Delta\mathcal{C}_s$ **do**
    Extract the class patch population X_c from its training images
    Estimate the empirical mean μ_c and covariance Σ_c
    Compute the eigendecomposition of Σ_c
    Retain the leading k principal directions
    Construct the PCA-whitening transform $W_c^{(k)}$
    Compute the retained-subspace correction $\log J_c$
    Append $(\mu_c, W_c^{(k)}, \log J_c)$ to $\mathcal{M}$
end for
return updated class statistics $\mathcal{M}$

Algorithm 2 PATCHBELL inference

Require: Frozen visual encoder E, test image I, observed class set $\mathcal{C}_s$, class statistics $\mathcal{M}$

Extract patch representations $\{x_p\}_{p=1}^{P} \leftarrow E(I)$
for each patch p and class $c \in \mathcal{C}_s$ **do**
    Compute $\ell_{p,c}$ using Eq. (3)
end for
Normalize patch likelihoods for entropy computation (Appendix A.4)
for each patch p **do**
    Compute entropy H_p using Eq. (4)
    Set confidence weight $w_p \leftarrow 1 - H_p$
end for
for each class $c \in \mathcal{C}_s$ **do**
    Compute $S_c(I) \leftarrow \sum_{p=1}^{P} w_p \, \ell_{p,c}$
end for
$\widehat{c} \leftarrow \arg\max_{c \in \mathcal{C}_s} S_c(I)$
return prediction $\widehat{c}$ and patch-confidence map $\{w_p\}_{p=1}^{P}$

B EVALUATION DETAILS AND ADDITIONAL RESULTS

B.1 BACKBONE CONFIGURATIONS

We evaluate PATCHBELL using both vision-language and vision-only foundation encoders. For all backbones, the visual encoder remains frozen throughout incremental learning, and no backbone parameters are updated when new classes are introduced. We extract spatial patch representations from the final encoder layer and discard non-spatial tokens. Unless stated otherwise, PATCHBELL uses a retained whitening dimension of $k = 500$.

Table 2 summarizes the backbone configurations used in our experiments. Here, d denotes the dimensionality of each patch representation before whitening and dimensionality reduction, and P denotes the number of spatial patch tokens extracted from each image.

Table 2: Backbone configurations used in our experiments. d denotes the patch-representation dimension before whitening, and P denotes the number of spatial patch tokens for the default input resolution.

Backbone	**Input size**	**Patch size**	P	d
CLIP ViT-B/16	224×224	16×16	196	768
CLIP ViT-L/14	224×224	14×14	256	1024
DINOv3 ViT-B/16	224×224	16×16	196	768
DINOv3 ViT-L/16	224×224	16×16	196	1024
SigLIP ViT-B/16	224×224	16×16	196	768
I-JEPA ViT-H/14	224×224	14×14	256	1280

CLIP. For CLIP (Radford et al., 2021), we use the official pretrained ViT-B/16 and ViT-L/14 models together with their corresponding model-specific preprocessing. Spatial patch representations are extracted from the output of the final visual-transformer layer after the final layer normalization (ln_post). The CLS token is discarded, and only the spatial patch tokens are retained. For an input resolution of 224×224, ViT-B/16 produces a 14×14 patch grid, corresponding to 196 spatial patch representations, while ViT-L/14 produces a 16×16 grid, corresponding to 256 spatial patch representations.

DINOv3. For DINOv3 (Siméoni et al., 2025), our main experiments use the ViT-L/16 model pretrained on LVD-1689M, and we additionally evaluate the ViT-B/16 variant. Images are resized to 224×224 using bicubic interpolation, center-cropped, converted to tensors, and normalized using the standard ImageNet mean and standard deviation. With a patch size of 16×16, both models produce a 14×14 spatial grid, corresponding to 196 patch representations per image. DINOv3 additionally produces one CLS token and four register tokens; these non-spatial tokens are discarded, and only the 196 spatial patch tokens are used for class-conditional modeling.

SigLIP. For SigLIP (Zhai et al., 2023), we use the ViT-B/16 vision encoder at an input resolution of 224×224. The model uses 16×16 patches, resulting in a 14×14 grid of 196 spatial patch representations, each with dimension $d = 768$. Input preprocessing follows the configuration associated with the pretrained SigLIP checkpoint, and only spatial vision-transformer representations are retained for class modeling.

I-JEPA. For I-JEPA (Assran et al., 2023), we use the ViT-H/14 encoder pretrained at an input resolution of 224×224. The model uses 14×14 patches, producing a 16×16 grid of 256 spatial representations, each with dimension $d = 1280$. We use the frozen encoder representations from the final layer for class-conditional patch modeling.

After extraction, patch representations from all backbones are processed using the same class-conditional PCA whitening, dimensionality reduction, and likelihood estimation procedure described in Appendix A.2. Thus, no backbone-specific classifier or adaptation mechanism is introduced, and differences between backbone experiments arise only from the frozen representation model itself.

B.2 Baseline Implementation Details

We compare against CLAP (Jha et al., 2024), RAPF (Huang et al., 2024), MG-CLIP (Huang et al., 2025), and RSIAT (Zhao et al., 2026). RAPF, MG-CLIP, and RSIAT are rerun using their official implementations. For CLAP (Jha et al., 2024), we report the published results for ImageNet-R and ImageNet100, and our reproduction for ImageNet-1K. Across all matched experiments, we use the same dataset splits, RAPF class order, sampled training subsets, and incremental-task configurations for all methods, while preserving each method's original training procedure.

In the standard setting, ImageNet100 is evaluated using 10 tasks of 10 classes, ImageNet-R using 10 tasks of 20 classes, and ImageNet-1K using 10 tasks of 100 classes. At each incremental task, evaluation is performed jointly over all classes observed up to that point without providing task identity at inference.

CLAP. CLAP (Jha et al., 2024) is a CLIP-based continual-learning method that uses probabilistic adaptation together with replay. We evaluate both the replay-based CLAP variant and CLAP*, its replay-free variant, under the same incremental protocol used for PatchBELL.

RAPF. RAPF (Huang et al., 2024) performs class-incremental learning with a pretrained CLIP model through adaptive representation adjustment and parameter fusion. RAPF uses feature replay and therefore differs from PatchBELL in its access to information retained from previous incremental tasks.

MG-CLIP. MG-CLIP (Huang et al., 2025) is a non-replay CLIP-based continual-learning approach that explicitly models and compensates for the modality gap during incremental adaptation.

RSIAT. RSIAT (Zhao et al., 2026) performs representation-steered incremental adapter tuning on top of a pretrained model. For ImageNet100, RSIAT is trained for 30 epochs in all evaluated settings, including the task-sensitivity, training-size, and controlled-imbalance experiments.

For the controlled data-availability experiments, including the few-shot and class-imbalance settings, the sampled training images are fixed before running the methods. The same per-class training subsets are provided to all evaluated methods, while the evaluation data remain unchanged. Reported Avg and Last accuracies are computed using the same incremental evaluation protocol described in Section 5.1.

B.3 Sensitivity Results

We evaluate the robustness of the incremental-learning methods to several changes in the ImageNet100 protocol. Unless stated otherwise, these experiments use the standard 10-task ImageNet100 setting, and PatchBELL uses DINOv3 ViT-L/16 with $k = 500$. For each comparison, only the experimental factor under study is changed, while the remaining evaluation protocol is kept fixed. The same task partitions and sampled training subsets are used across the compared methods whenever applicable.

Number of incremental tasks. To isolate sensitivity to task granularity, we keep the same 100 ImageNet100 classes and vary only their partition into incremental tasks. We consider 4, 10, 20, and 50 tasks, corresponding to 25, 10, 5, and 2 new classes per task, respectively. The underlying training and evaluation data are unchanged; only the number of incremental updates and the number of classes introduced at each update vary. Each baseline follows its original optimization procedure under the corresponding task partition, whereas PatchBELL independently estimates the class statistics of the newly introduced classes.

Table 3: Sensitivity to the number of incremental tasks on ImageNet100. Each setting contains the same 100 classes and differs only in how they are partitioned across tasks. We report average incremental accuracy (Avg) and last accuracy (Last).

		MG-CLIP		RAPF		RSIAT		PATCHBELL	
# Tasks	Classes/task	Avg	Last	Avg	Last	Avg	Last	Avg	Last
4	25	87.04	79.54	87.44	80.86	91.16	84.90	**91.51**	**87.28**
10	10	87.41	78.10	88.18	79.98	91.05	83.60	**92.25**	**87.28**
20	5	87.69	77.66	88.44	78.40	90.98	83.10	**92.65**	**87.28**
50	2	86.32	76.44	87.50	75.38	89.06	73.22	**92.95**	**87.28**

Table 4: Effect of the number of training images per class on ImageNet100. We report the average incremental accuracy (Avg) and the last accuracy after all classes have been introduced (Last).

	MG-CLIP		**RAPF**		**RSIAT**		**PATCHBELL**	
Images/class	**Avg**	**Last**	**Avg**	**Last**	**Avg**	**Last**	**Avg**	**Last**
300	86.78	77.86	85.85	75.02	92.01	85.80	**92.22**	**86.76**
500	86.90	78.04	86.80	77.98	91.72	85.36	**92.29**	**86.76**
800	87.25	78.16	87.73	78.88	91.38	84.16	**92.30**	**87.10**
Full	87.41	78.10	88.18	79.98	91.05	83.60	**92.25**	**87.28**

As shown in Table 3, PATCHBELL is largely insensitive to task granularity. In particular, its last accuracy remains unchanged as the same classes are divided into progressively more incremental updates, whereas the competing methods become more sensitive as the sequence becomes longer. This behavior is consistent with the class-independent update mechanism of PATCHBELL: introducing a class requires estimating only its own statistics and does not modify the representations or previously stored class models.

Number of training images per class. We next study sensitivity to the amount of data available when each class is introduced. We retain the standard 10-task ImageNet100 protocol and restrict the training set of every class to 300, 500, or 800 images, in addition to the full-data setting. For each data regime, the subset is sampled once and the same per-class images are supplied to all methods. The test set and incremental evaluation protocol remain unchanged, so the experiment isolates the effect of the number of examples available for learning each new class.

As shown in Table 4, PATCHBELL remains stable across the evaluated training-set sizes and retains strong performance even when substantially fewer examples are available for estimating each class distribution. This suggests that the class-conditional statistics can be estimated reliably without requiring the full ImageNet100 training set and that the method does not rely on repeated optimization over large per-class datasets.

Controlled class imbalance. To examine whether the method is sensitive to unequal amounts of training data across classes, we construct a controlled imbalance setting from the standard 10-task ImageNet100 protocol. In every task, one of the ten newly introduced classes is restricted to 100 training images, while the remaining classes retain their full training sets. The identities of the reduced classes and the sampled examples are fixed before training and are identical for all methods. Evaluation is performed on the original test set.

We report two complementary views of this experiment. First, we measure overall Avg and Last accuracy and compare them with the corresponding full-data runs. Second, we isolate the ten underrepresented classes and compute their per-class accuracy drop between the full and controlled settings. This separates changes in overall performance from the direct effect of imbalance on the classes whose training data were reduced.

Table 5: Overall performance under controlled class imbalance on ImageNet100 and in the corresponding full-data setting. Drop is computed as Full − Controlled; negative values indicate higher performance in the controlled setting. Lower drop is better.

	Controlled		Full		Drop	
Method	Avg ↑	Last ↑	Avg ↑	Last ↑	Avg ↓	Last ↓
MG-CLIP	82.19	71.38	87.41	78.10	5.22	6.72
RAPF	87.72	79.50	88.18	79.98	0.45	0.48
RSIAT	91.34	84.18	91.05	83.60	-0.29	-0.58
PATCHBELL	**92.16**	**87.24**	**92.25**	**87.28**	0.10	0.04

Table 6: Sensitivity of the underrepresented classes to controlled class imbalance. We report the average per-class accuracy drop, Full − Controlled, over the 10 classes restricted to 100 training images. Lower is better.

Method	Reduced-class Avg drop ↓	Reduced-class Last drop ↓
MG-CLIP	70.82	73.00
RAPF	7.92	4.20
RSIAT	3.48	2.80
PATCHBELL	**2.47**	**2.20**

As shown in Tables 5 and 6, PATCHBELL remains nearly unchanged overall and exhibits the smallest performance drop on the reduced classes. RSIAT slightly improves in overall accuracy, while remaining below PATCHBELL, but experiences a larger drop on the reduced classes. The other competing methods exhibit larger degradation both overall and on the reduced classes. These results demonstrate the robustness of PATCHBELL to both limited data and class imbalance.

B.3.1 SENSITIVITY TO CLASS ORDER

Finally, we test whether performance depends on the order in which classes are introduced. Starting from the standard 10-task ImageNet100 protocol, we generate five independently sampled random class orders. Each order contains the same 100 classes and preserves ten classes per task; only the assignment and order of classes across the incremental sequence are changed. The same five random orders are used for all compared methods, while all other experimental settings remain fixed. We report the original class order used in the main experiments together with the mean and standard deviation across the five random orders.

As shown in Table 7, unlike the competing methods, whose performance changes more noticeably across class orders, PATCHBELL remains essentially unchanged relative to the original ordering. In particular, its last accuracy is invariant across all evaluated random orders. Since the final set of independently estimated class models is the same regardless of the order in which those classes are introduced, this result is consistent with the order-independent nature of the proposed update mechanism.

Table 7: **Sensitivity to class order on ImageNet100.** Original denotes the standard class order used in the main experiments. Random reports mean ± standard deviation over five randomly sampled class orders. *Avg* and *Last* denote the average accuracy across incremental tasks and the final-task accuracy, respectively.

	Original		**Random (5)**	
Method	Avg.	Last	Avg.	Last
MG-CLIP (Huang et al., 2025)	87.41	78.10	86.55 ± 0.84	77.36 ± 0.66
RAPF (Huang et al., 2024)	88.18	79.98	87.69 ± 0.94	79.54 ± 0.76
RSIAT (Zhao et al., 2026)	91.05	83.60	89.73 ± 0.93	82.38 ± 2.45
PATCHBELL	**92.25**	**87.28**	$\mathbf{92.12 \pm 0.66}$	$\mathbf{87.28 \pm 0.00}$

B.4 Additional Ablation Results

We provide the complete numerical results corresponding to the method ablations summarized in Figure 4. Unless stated otherwise, all ablations use the standard 10-task ImageNet100 protocol with DINOv3 ViT-L/16, representations from the final encoder layer, a whitening dimension of $k = 500$, and entropy-weighted patch-likelihood aggregation with the Jacobian correction. In each experiment, we vary only the component under study while keeping the remaining pipeline unchanged. We additionally report the foundation-model backbone comparison and the controlled-imbalance Jacobian analysis, which are omitted from the main text for compactness.

Whitening dimension. We first study the effect of the retained PCA-whitening dimension k. Using DINOv3 ViT-L/16, we vary k from 50 to the full representation dimension of 1024 while keeping the feature extraction, class-conditional modeling, Jacobian correction, and entropy-weighted aggregation unchanged. A single value of k is used for all classes in each run.

Table 8: Whitening-dimension ablation on ImageNet100 using DINOv3 ViT-L/16. We report Avg and Last accuracy for entropy-weighted likelihood aggregation with the Jacobian correction.

Whitening dim.	**Avg**	**Last**
50	10.23	7.64
100	67.03	60.82
150	84.41	78.84
200	88.82	83.50
250	90.64	85.38
300	91.07	85.86
350	91.52	86.54
400	91.65	86.72
450	91.99	86.92
500	92.25	87.28
550	92.32	87.30
600	92.32	87.18
650	92.46	87.28
700	92.44	**87.42**
750	**92.50**	87.36
800	92.45	**87.42**
850	92.46	87.40
900	92.38	87.26
950	92.30	87.08
1000	92.27	87.04
1024	92.19	86.90

As shown in Table 8, performance increases rapidly as more dimensions are retained and then largely saturates at higher dimensionalities. This motivates the default choice $k = 500$, which lies in the

Table 9: Effect of the foundation-model backbone on ImageNet100. We report the average incremental accuracy (A_{avg}) and the last accuracy after all classes have been introduced (A_{last}).

Foundation model	**Backbone**	$\boldsymbol{A_{\text{avg}}} \uparrow$	$\boldsymbol{A_{\text{last}}} \uparrow$
CLIP	ViT-B/16	83.29	73.84
CLIP	ViT-L/14	89.15	82.36
DINOv3	ViT-B/16	90.02	83.42
DINOv3	ViT-L/16	**92.25**	**87.28**
SigLIP	ViT-B/16	84.73	75.54
I-JEPA	ViT-H/14	87.01	79.84

Table 10: Encoder-layer ablation on ImageNet100 using DINOv3 ViT-L/16 with $k = 500$, post-normalization features, and entropy-weighted likelihood aggregation with the Jacobian correction. We report Avg and Last accuracy when extracting patch representations from each of the last four encoder layers.

Encoder layer	**Avg**	**Last**
-4	87.22	79.88
-3	88.76	81.98
-2	90.74	85.12
-1	**92.25**	**87.28**

saturated regime while retaining a substantially more compact representation than the full DINOv3 feature dimension.

Foundation-model backbone. To test whether the proposed statistical formulation depends on a particular representation model, we repeat the standard ImageNet100 experiment using CLIP, DINOv3, SigLIP, and I-JEPA encoders of different scales. For each backbone, its visual encoder remains completely frozen and the corresponding spatial patch representations are passed through the same class-conditional modeling and image-level aggregation pipeline. No backbone-specific adaptation or learned classifier is introduced. The extraction details for each model are provided in Appendix B.1.

As shown in Table 9, the method transfers across all evaluated foundation-model families without changing the underlying classifier. Performance generally improves with stronger frozen representations, indicating that PATCHBELL can directly exploit advances in pretrained representation quality rather than depending on a particular backbone family.

Encoder layer. We next examine which representation depth is most suitable for the class-conditional model. Using DINOv3 ViT-L/16, we independently repeat the ImageNet100 experiment with patch representations extracted from each of the final four encoder layers. For all four runs, we use post-normalization features, retain $k = 500$ dimensions, and keep the likelihood computation and entropy-weighted aggregation unchanged.

As shown in Table 10, performance improves progressively toward the final encoder layer. This suggests that the later, more semantic patch representations are better suited to class-conditional statistical modeling, while the consistent performance across the final layers indicates that the method is not tied to a single intermediate representation.

Jacobian correction. We isolate the contribution of the change-of-variables correction by comparing the complete method with an otherwise identical variant that omits the Jacobian term. All other components, including the frozen DINOv3 ViT-L/16 features, class-specific whitening, $k = 500$, and entropy-weighted aggregation, are kept unchanged. We perform this comparison both under the standard ImageNet100 protocol and under the controlled-imbalance setting described above. For the latter, we additionally separate last accuracy on the classes restricted to 100 images from that of the full-data classes.

Table 11: Ablation of the Jacobian correction on ImageNet100. The controlled imbalance setting restricts one class per task to 100 training images. Δ denotes the improvement obtained by adding the Jacobian correction.

Setting	Metric	w/o Jacobian	with Jacobian	Δ
Standard ImageNet100	Avg	90.21	**92.25**	+2.04
	Last	83.94	**87.28**	+3.34
Controlled imbalance	Avg	89.52	**92.16**	+2.64
	Last	82.88	**87.24**	+4.36
Reduced classes (100 images)	Last	73.80	**89.60**	+15.80
Full-data classes	Last	83.89	**86.98**	+3.09

As shown in Table 11, the Jacobian correction consistently improves performance and becomes particularly important when the amount of data differs across classes. The much stronger effect on the underrepresented classes supports its intended role: correcting for class-dependent transformations makes likelihoods obtained from different class-specific whitened spaces more comparable before they are combined for prediction.

Representation and aggregation rule. Finally, we examine whether the gains arise simply from using frozen representations or from the proposed patch-level evidence aggregation. We compare global CLS-based statistical classifiers with several patch-based alternatives. For the patch representation, we evaluate prototype voting, unweighted aggregation of patch log-likelihoods, majority voting over patch likelihood predictions, maximum patch likelihood, and entropy-weighted likelihood aggregation with and without the Jacobian correction. All variants use the same frozen representation backbone and the same ImageNet100 incremental protocol.

Table 12: Comparison of patch-level aggregation rules and global-representation baselines on ImageNet100. We report average incremental accuracy (A_{avg}) and final accuracy (A_{last}).

Representation	Classifier / aggregation	$A_{\mathbf{avg}} \uparrow$	$A_{\mathbf{last}} \uparrow$
CLS	Whitened Gaussian likelihood	91.49	85.60
CLS	Prototype Mahalanobis	88.38	80.34
Patches	Prototype + majority vote	77.24	67.84
Patches	Unweighted sum of log-likelihoods	89.49	83.40
Patches	Majority vote over likelihood predictions	84.75	77.60
Patches	Maximum patch likelihood	81.02	72.44
Patches	Entropy-weighted likelihood w/o Jacobian	90.21	83.94
Patches	Entropy-weighted likelihood + Jacobian	**92.25**	**87.28**

As shown in Table 12, using patch representations alone is not sufficient: simple voting or maximum-evidence rules lose much of the information contained in the patch population. Aggregating likelihood evidence across patches is substantially more effective, and explicitly emphasizing patches that are discriminative across the currently observed classes further improves the classifier. Together with the Jacobian correction, the entropy-weighted formulation provides the strongest patch-based configuration and motivates the complete PATCHBELL inference rule.

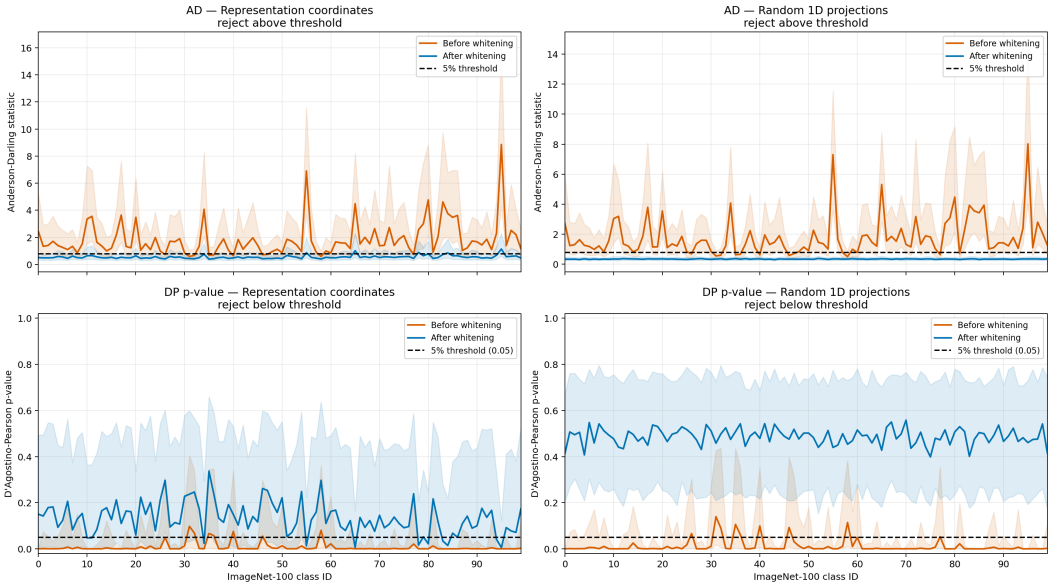


Figure 5: **Gaussianity analysis of class-conditional patch representations using DINOv3 ViT-L/16 on ImageNet100.** Anderson–Darling statistics (top) and D'Agostino–Pearson p-values (bottom) are evaluated before and after whitening, both along individual representation coordinates (left) and random one-dimensional projections (right). Lines denote the class-wise medians and shaded regions denote the 25th–75th percentiles. Dashed lines indicate the corresponding rejection thresholds.

B.5 Gaussianity Measurements

We provide additional analysis of the approximately Gaussian structure of the class-conditional patch representations used by PatchBELL.

For each class, we collect the patch population X_c and apply the PCA-based whitening procedure described in Appendix A.2. We assess the resulting distributions using two complementary normality tests: the Anderson–Darling (AD) statistic and the D'Agostino–Pearson (DP) test.

We perform each test from two complementary perspectives. First, we examine individual representation coordinates, directly testing the marginal distributions in the learned representation basis. Second, we evaluate randomly sampled one-dimensional projections, providing a view of the distributional structure that is not tied to a particular coordinate system. For every ImageNet100 class, the figure summarizes the distribution of test statistics across coordinates or projections before and after whitening.

As shown in Figure 5 and Table 13, whitening produces a consistent shift toward Gaussian behavior across classes. The Anderson–Darling statistics become substantially smaller after whitening, while the D'Agostino–Pearson p-values move away from the strong rejection regime. Importantly, the same qualitative behavior is observed both along the original representation coordinates and along random one-dimensional projections.

To provide a complementary quantitative summary, we define the *Gaussianity Score* as

$$\text{Gaussianity Score} = \frac{\%\text{AD pass} + \%\text{DP pass}}{2}. \tag{27}$$

At a significance level of 5%, samples drawn from an ideal Gaussian distribution are expected to pass each individual normality test approximately 95% of the time. Thus, a Gaussianity Score close to 95%, rather than 100%, represents the nominal Gaussian reference. The score is used only as a comparative summary of normality-test pass rates and should not be interpreted as the probability that a distribution is Gaussian.

Table 13: **Quantitative Gaussianity analysis on ImageNet100 using random one-dimensional projections.** We report the percentage of evaluated class-conditional projections that pass the Anderson–Darling (AD) and D'Agostino–Pearson (DP) normality tests at a 5% significance level, together with the percentage that pass both tests. The Gaussianity Score is the average of the AD and DP pass rates. Under an ideal Gaussian distribution, the nominal pass rate of each individual test is approximately 95%.

Backbone	State	AD pass	DP pass	Both pass	Gaussianity Score
DINOv3 ViT-L/16	Before	30.4%	26.0%	21.7%	28.2%
	After	**95.6%**	**93.9%**	**91.2%**	**94.7%**

Table 13 shows that whitening substantially increases the normality-test pass rates for DINOv3 ViT-L/16. The Gaussianity Score increases from 28.2% before whitening to 94.7% after whitening, nearly matching the nominal 95% pass rate expected under Gaussian data.

The random-projection analysis provides complementary evidence that the improvement is not restricted to the coordinate system in which whitening is performed, but remains visible when the representation space is examined along independently sampled directions. These results support the approximately Gaussian class-conditional model used by PATCHBELL and demonstrate that whitening substantially improves the Gaussian structure of the patch representations.

B.6 COMPUTATIONAL EFFICIENCY

We compare the computational requirements of PATCHBELL with RAPF (Huang et al., 2024), CLAP (Jha et al., 2024), MG-CLIP (Huang et al., 2025), and RSIAT (Zhao et al., 2026) under the ImageNet100 protocol with 10 incremental tasks of 10 classes each. Since the original methods do not report computational cost under a common FLOP convention, we analytically estimate all methods using a unified accounting protocol based on their corresponding architectures and training procedures.

We count one multiply-add as two FLOPs and include the dominant matrix operations of the visual backbone, attention and MLP layers, adapters, classification heads, replay mechanisms, and method-specific operations, including backward matrix multiplications for all trainable paths. Lower-order elementwise operations, evaluation metrics, diagnostics, visualization, data loading, memory traffic, and I/O are excluded. End-to-end update cost includes all backbone forward passes and method-specific operations required to incorporate new classes.

Table 14: **Computational efficiency on ImageNet100 (10 tasks $\times$ 10 classes).** All values are analytical FLOP estimates obtained under the same accounting convention. End-to-end update includes all computation required to incorporate the new classes.

Method	Training protocol	End-to-end update / task
RAPF	15 epochs, cached text features	7.273 PFLOPs
CLAP	5 epochs + 2 consolidation epochs	2.498 PFLOPs
MG-CLIP	2 epochs	3.780 PFLOPs
RSIAT	30 epochs + 10 CA epochs	40.311 PFLOPs
PATCHBELL	Closed-form, single pass	**1.634 PFLOPs**

Table 14 shows that PATCHBELL has the lowest end-to-end incremental-update cost among the evaluated methods. Despite using the substantially larger DINOv3 ViT-L/16 backbone, its closed-form, single-pass update requires less computation than the iterative optimization used by RAPF, CLAP, MG-CLIP, and RSIAT. Each new class is modeled once from its frozen representations, avoiding repeated backbone evaluation and backpropagation across multiple epochs.

Class-specific covariance estimation and whitening add computation, but are performed only once per new class, avoiding repeated optimization.

C PREDICTION INTERPRETATION

A useful by-product of the proposed patch-level formulation is a spatial confidence map obtained directly from the quantities used for classification. Importantly, PATCHBELL is not designed as a dedicated XAI method and does not optimize an explanation objective; the same cross-class entropy used for classification naturally assigns higher confidence to discriminative regions and suppresses ambiguous content.

We evaluate these maps from two complementary perspectives: transfer to an independent classifier and faithfulness to PATCHBELL's own decisions. In both settings, we compare against a set of explanation baselines, including Attention Rollout (Abnar & Zuidema, 2020), GradCAM (Selvaraju et al., 2017), Chefer (Chefer et al., 2021), TokenCut (Wang et al., 2022a), and the more recent LeGrad (Bousselham et al., 2025), Object-DINO (Rawlekar et al., 2026) and AHA (Schinagl et al., 2026).

C.1 EVALUATION PROTOCOL

We follow the image-space deletion and insertion protocol used by AHA (Schinagl et al., 2026). All methods are evaluated on the same fixed set of 5,000 ImageNet100 images in 224×224 RGB space. Patch-level maps are bilinearly resized to the input resolution before constructing a dense pixel ranking.

For each image, the target class is determined from the original image and held fixed throughout the perturbation trajectory. The target classifier is rerun after every perturbation and the probability assigned to this class is recorded.

We evaluate 21 perturbation levels,

$$0,\ 0.05,\ 0.10, \ldots, 1.0.$$

For deletion, highly ranked pixels are replaced first, such that lower deletion AUC is better. Insertion follows the official AHA convention by applying the same perturbation operator to the negated importance map, such that low-ranked pixels are added first; higher insertion AUC is better.

We use three reference images: black, ImageNet mean, and a Gaussian-blurred version of the input using a 37×37 kernel with standard deviation 37. For each reference, AUC is computed over the complete 21-point probability curve using trapezoidal integration. We report *Avg-3*, the mean AUC over the three references, together with

$$\text{Ins.} - \text{Del.} = \text{Insertion} - \text{Deletion}, \tag{28}$$

where larger values are better.

DINOv3-derived baseline maps are denoted by the suffix `_dino`; non-spatial tokens are excluded when constructing spatial maps.

C.2 TRANSFER TO AN INDEPENDENT CLASSIFIER

We first test whether regions identified by PATCHBELL remain informative for an independent classifier. The target model is the ViT-B/16 classifier used in the AHA evaluation, `vit_base_patch16_224.orig_in21k_ft_in1k`, with its original 1000-class ImageNet-1K head. Its top-1 prediction on the original image is fixed as the target throughout the perturbation sequence.

We compare against Raw Attention, Attention Rollout, GradCAM, LeGrad, an adapted Chefer-style gradient-weighted attention attribution, and the released AHA $T = 0$ explainer. AHA (Schinagl et al., 2026) is a learned attribution method that directly optimizes deletion and insertion rankings; the $T = 0$ variant produces the attribution map without additional test-time refinement.

We additionally evaluate maps derived from the frozen DINOv3 ViT-L/16 representation used by PATCHBELL. Attention Rollout_dino propagates attention across layers, TokenCut_dino (Wang et al., 2022a) uses a spectral partition of DINO features, and Object-DINO_dino (Rawlekar et al., 2026) derives an object-centric spectral ranking from distributed DINO representations.

Table 15: **Image-space perturbation with the independent ViT-B/16 classifier used by AHA.** Results are Avg-3 AUCs on the same 5,000 ImageNet100 images using the 21-step perturbation protocol. Lower deletion and higher insertion and Ins.-Del. are better. Best result in each column is in bold, second best is underlined.

Method	Deletion ↓	Insertion ↑	Ins.–Del. ↑
Random	0.3472	0.3464	-0.0008
Raw Attention	0.2514	0.5287	0.2773
Attention Rollout (Abnar & Zuidema, 2020)	0.1758	0.6268	0.4510
Attention Rollout_dino (Abnar & Zuidema, 2020)	0.3134	0.4416	0.1282
GradCAM (Selvaraju et al., 2017)	0.2882	0.5315	0.2433
Chefer-style (Chefer et al., 2021)	0.1546	0.6436	<u>0.4890</u>
TokenCut_dino (Wang et al., 2022a)	0.1752	<u>0.6596</u>	0.4844
LeGrad (Bousselham et al., 2025)	0.1747	0.5805	0.4059
Object-DINO_dino (Rawlekar et al., 2026)	0.3534	0.4007	0.0472
AHA $T = 0$ (Schinagl et al., 2026)	**0.1166**	**0.6877**	**0.5711**
PATCHBELL	<u>0.1545</u>	0.6409	0.4863

Table 16: **Image-space perturbation with PATCHBELL as the target classifier.** The same 5,000 images, references, and 21-step protocol as in Table 15 are used. Lower deletion and higher insertion and Ins.-Del. are better. Best result in each column is in bold, second best is underlined.

Method	Deletion ↓	Insertion ↑	Ins.–Del. ↑
Random	0.7215	0.7206	-0.0009
Raw Attention_dino	0.3606	**0.8788**	<u>0.5182</u>
Attention Rollout_dino (Abnar & Zuidema, 2020)	0.6465	0.7739	0.1274
GradCAM_dino (Selvaraju et al., 2017)	0.5159	0.7865	0.2705
TokenCut_dino (Wang et al., 2022a)	<u>0.3514</u>	0.8626	0.5112
LeGrad_dino (Bousselham et al., 2025)	0.4800	0.8069	0.3269
Object-DINO_dino (Rawlekar et al., 2026)	0.7128	0.7742	0.0614
PATCHBELL	**0.3388**	<u>0.8722</u>	**0.5334**

As shown in Table 15, PATCHBELL's intrinsic confidence map provides strong perturbation-based attribution, achieving the second-best deletion score and competitive insertion and Ins.–Del. performance across the evaluated methods. Notably, the map is derived from PATCHBELL's DINOv3 representations but evaluated with an independent supervised ViT-B/16 classifier, demonstrating that the identified visual evidence transfers across models. AHA achieves the strongest overall performance, while PATCHBELL remains highly competitive despite producing its map directly as a by-product of the classification formulation, with no dedicated attribution objective.

C.3 FAITHFULNESS TO THE PATCHBELL CLASSIFIER

We next evaluate the same spatial rankings with PATCHBELL itself as the target classifier. For each original image, the predicted PATCHBELL class is fixed as the perturbation target. For perturbation evaluation, we apply a softmax over the 100 class scores and track the probability assigned to this fixed target class.

The complete PATCHBELL inference procedure is rerun for every perturbed image, including feature extraction, class-conditional likelihood computation, normalization, entropy weighting, and final class scoring. Only the spatial ranking used to determine the perturbation order is fixed from the original image. The intrinsic PATCHBELL map is the original confidence $w_p = 1 - H_p$, computed exactly as in the standard inference procedure described in Appendix A.4. Target-specific gradient baselines are computed with respect to the fixed PATCHBELL target-class score.

As shown in Table 16, PATCHBELL's intrinsic confidence map achieves the best deletion and Ins.–Del. performance among all evaluated DINOv3-derived methods, while remaining within 0.7 points of the best insertion score. These results indicate that the same patch-level confidence used for classification also provides a strong ranking of visually important regions.

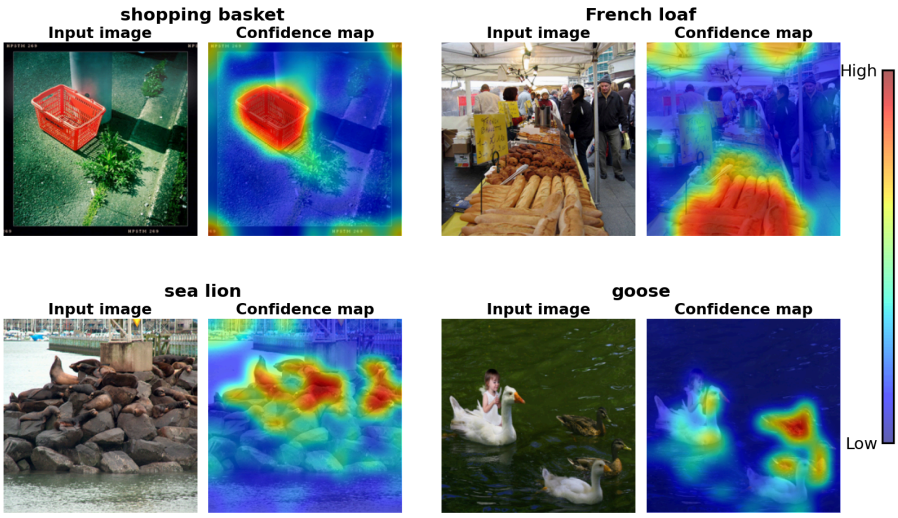


Figure 6: **Qualitative examples of intrinsic PATCHBELL confidence maps.** Each image is shown together with the spatial confidence $w_p = 1 - H_p$ used during classification. Warmer regions denote higher-confidence, more discriminative patches. No separate explanation model or post-hoc attribution procedure is used.

C.4 QUALITATIVE CONFIDENCE MAPS

Figure 6 shows representative ImageNet validation images together with the confidence maps produced by PATCHBELL. Across the examples, high-confidence regions largely coincide with visually distinctive foreground content, while generic background regions receive lower confidence. In the lower-right example of Figure 6, the image contains multiple objects belonging to more than one plausible class, and high confidence is assigned to all of them, showing that the map can simultaneously highlight multiple discriminative objects within the same scene.

Together, these results show that the confidence mechanism used for classification also provides informative spatial cues. We therefore view the resulting maps as an intrinsic interpretability signal of the proposed formulation rather than as a separate XAI method.