This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation. Except for this watermark, it is identical to the accepted version; the final published version of the proceedings is available on IEEE Xplore.

DINO-QPM: Adapting Visual Foundation Models for Globally Interpretable Image Classification

Robert Zimmermann* Thomas Norrenbrock* Bodo Rosenhahn
Institute for Information Processing, L3S - Leibniz University Hannover
{zimmerro, norrenbr, rosenhahn}@tnt.uni-hannover.de

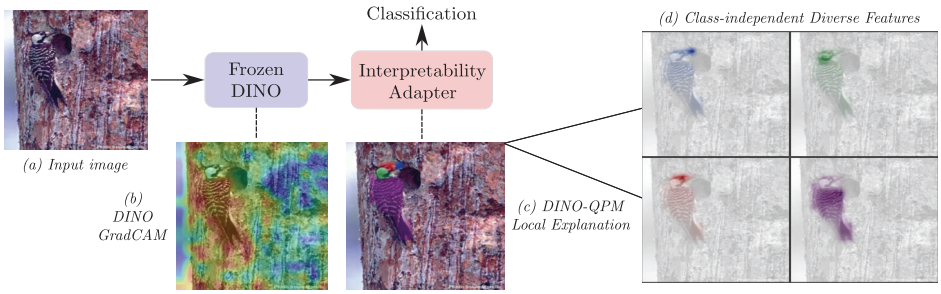


Figure 1. Overview of our proposed DINO-QPM. The pipeline processes the (a) input image using the frozen backbone to produce patch embeddings, which are transformed by the interpretability adapter to obtain a globally interpretable image classification. We compare the diffuse saliency map of (b) DINO GradCAM, extracted from a linear probed DINO model, with our (c) DINO-QPM local explanation. The local explanation can be further decomposed into its (d) class-independent diverse features. Compared to the baseline, we observe a drastic increase in localisation quality, showcasing how our interpretability adapter successfully isolates semantically meaningful features.

Abstract

Although visual foundation models like DINOv2 provide state-of-the-art performance as feature extractors, their complex, high-dimensional representations create substantial hurdles for interpretability. This work proposes DINO-QPM, which converts these powerful but entangled features into contrastive, class-independent representations that are interpretable by humans. DINO-QPM is a lightweight interpretability adapter that pursues globally interpretable image classification, adapting the Quadratic Programming Enhanced Model (QPM) to operate on strictly frozen DINO backbones. While classification with visual foundation models typically relies on the `CLS` token, we deliberately diverge from this standard. By leveraging average-pooling, we directly connect the patch embeddings to the model's features and therefore enable spatial localisation of DINO-QPM's globally interpretable features within the input space. Furthermore, we apply a sparsity loss to minimise spatial scatter and background noise, ensuring that explanations are grounded in relevant object parts. With DINO-QPM we make the level of interpretability of QPM available as an adapter while exceeding the accuracy of DINOv2 linear probe. Evaluated through an introduced Plausibility metric and other interpretability metrics, extensive experiments demonstrate that DINO-QPM is superior to other applicable methods for frozen visual foundation models in both classification accuracy and explanation quality.

1. Introduction

Visual foundation models such as DINOv2 [46] have shown great performance as powerful, general-purpose feature extractors across various image analysis benchmarks [15, 28, 31, 46, 68]. However, deploying these models in safety-critical domains requires a high degree of interpretability, which remains a significant challenge due to their complex, opaque architectures.

Thus interpretable-by-design approaches are getting more popular for such applications [53]. Inspired by human cognitive processes [51], one line of work computes the similarity to so-called prototypes to obtain an interpretable classification [10, 40, 54]. Turbé et al. [60], Zhu et al. [70], Ma et al. [37] and Turbé et al. [61] apply this idea to visual foundation models. However, prototypical model have a deceiving interpretability as their similarity is not restricted to

* Indicates equal contribution.

be similar to humans [4, 26, 33]. Therefore several other approaches utilise sparse, low-dimensional, and quantised decision layers to enforce compact class representations [41–44]. Unlike local interpretability methods that only explain individual predictions, these approaches aim for global interpretability, providing a holistic, transparent view of the model's entire decision-making process and how it defines classes across the dataset. This leads to diverse [3], contrastive [35], general and compact [50] feature representations ideally suited for generating human-interpretable explanations [39].

Many of the aforementioned methods are fully trained end-to-end and therefore require massive resources for training purposes. While recent works have begun exploring interpretability techniques like post-hoc concept mapping [45, 65, 66] on top of frozen backbones, these approaches struggle to reach a competitive level of accuracy. To address these limitations, our approach aligns with the objective of models such as QPM [44] and ChiQPM [43], aiming to represent classes through general, diverse and contrastive features. Translating the mathematically constrained compactness of sparse, quantised decision layers, like those used in QPM [44], to frozen visual foundation models for inherently interpretable image classification remains an open problem.

In this work, we address this problem by applying the Quadratic Programming Enhanced Model (QPM) [44] to the high-dimensional representations of DINOv2 and building a lightweight interpretability adapter on top of its frozen features, as suggested in Siméoni et al. [58]. The adapter transforms DINOv2's powerful, entangled representations using a sparse feature assignment into diverse, class-independent and contrastive features, yielding a globally interpretable solution for image classification.

After applying an MLP to the patch embeddings of the frozen backbone, we use average-pooling to obtain a feature vector, which is inherently connected to the problem-specific feature maps returned by the MLP. Although the standard choice in classification literature is to, at least partially, use the `CLS` token [12, 46], this direct connection to the feature maps enables high-fidelity spatial localisation of features in the image.

Inspired by the pointing game [67], we introduce a Plausibility metric, which measures the fraction of the cumulative feature map activation that falls within the object boundaries. We are able to quantify that DINO-QPM's interpretable features localise consistently on the relevant object parts, while the saliency map of a linear probe on the frozen features lacks in Plausibility, as visualised in Fig. 1 and quantitatively shown in Fig. 2. We further apply a sparsity loss to enhance spatial precision of our model. The sparsity loss effectively minimises spatial scatter and background noise, which ensures that model explanations are strictly grounded in relevant object parts.

Extensive validation across multiple datasets and backbones confirms that DINO-QPM outperforms state-of-the-art interpretable methods applied to visual foundation models in terms of both classification accuracy and explanation quality. To facilitate reproducibility and future research, the code is available at `https://github.com/RobertZimm/DINO-QPM`.

The main contributions of this work are:

- **Lightweight Interpretability Adapter for Frozen Backbones:** Our proposed lightweight interpretability adapter (DINO-QPM) is designed to function with frozen self-supervised backbones, such as DINOv2. This design facilitates inherently interpretable image classification without the need for full model fine-tuning or high computational overhead, delivering state-of-the-art interpretability on top of frozen visual foundation models while maintaining exceptional accuracy.
- **Spatial Localisation through Token Representations:** DINO-QPM leverages average-pooling across tokens to enable the spatial localisation of its features in the input space. This allows the generation of high-fidelity saliency maps, while beating the linear probe in accuracy and outperforming the dense average-pooled variant by more than 10% on CUB-2011 [64].
- **Enhanced Plausibility via Sparsity Loss:** In contrast to the initial feature maps from a DINOv2 linear probe, DINO-QPM has an exceptional localisation ability, which is further enhanced using a sparsity loss. We quantify this via our introduced Plausibility metric.

2. Related Work

Initial approaches to interpreting visual foundation models rely heavily on visualising attention maps and feature projections. For instance, Dosovitskiy et al. [18] and Caron et al. [8] visualise the last self-attention layer of the CLS token to understand feature extraction. Similarly, for DINOv2, Oquab et al. [46] visualise patch embeddings by filtering the foreground using the first PCA principal component, then transforming the embeddings via a second PCA to display their primary components as RGB channels. While these visualisations suggest that foundation models inherently learn well-localising features without explicit supervision [8], they serve primarily as qualitative observations rather than rigorous explanations.

The assumption that attention maps suffice for credible explanations is heavily criticised in NLP [30] and vision [7, 9]. Attention maps are inherently independent of downstream tasks, meaning they often neglect crucial information required for classification [7, 9]. Derived solely from query-key products, they ignore the highly influential value vectors and MLP blocks [9, 13].

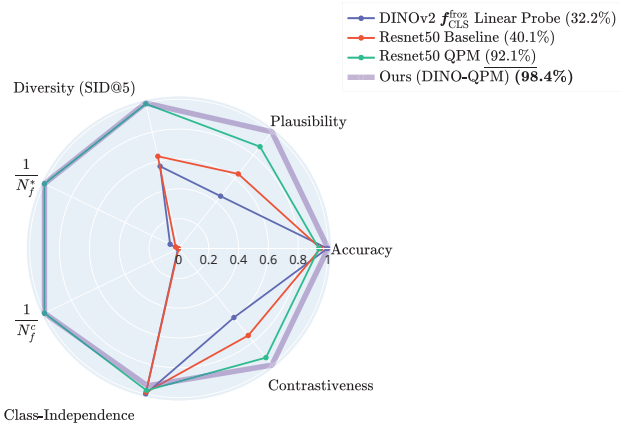


Figure 2. Radar Plot demonstrating the quality of DINO-QPM in a set of interpretability metrics and accuracy compared to non-frozen QPM and the corresponding baselines. DINO-QPM outperforms each of them reaching 98.4% of the maximal score calculated as the fraction of area of the heptagon reached by the respective model. We present rigorous insights in Sec. 5 including a more detailed presentation of the interpretability metrics.

While more advanced post-hoc methods have emerged to aggregate signals across the entire architecture—including Gradient Attention Rollout [2, 32], LRP [9], CDAM [7], and ViT-Shapley [14]—these techniques remain external approximations rather than inherent, faithful reflections of the model's decision-making process.

To overcome the limitations of post-hoc explanations, interpretable-by-design (ad-hoc) approaches integrate the explanation directly into the model's decision process. Current approaches utilise sparse [20, 21, 52], low-dimensional, and quantised decision layers to inherently increase interpretability [41–44]. Translating these inherent interpretability mechanisms to visual foundation models, however, presents unique challenges. Existing ad-hoc architectures for vision transformers include IA-RED^2 [47], the B-Cos alignment approach [5, 6], and prototype-based methods like ProtoViT [37]. While recent works like Zhu et al. [70] successfully achieve part-based interpretability, they depend on fine-tuning the backbone to enforce prototype clustering.

Post-hoc Concept Bottleneck Models (CBMs) maintain a completely frozen backbone, but they rely on textual concept supervision rather than providing direct spatial localisation [66]. While recent advancements in Post-hoc CBMs have automated concept selection via Large Language Models [45, 59, 69] or secondary segmentation networks [49], they still largely depend on external textual supervision or auxiliary trained modules.

To the best of our knowledge, DINO-QPM is the first approach to extract sparse, spatially localised, and class-independent part explanations directly from frozen DINOv2 features without requiring external concept banks, language models, or expensive backbone fine-tuning.

3. Fundamentals

3.1. QPM

Norrenbrock et al. [44] introduce the Quadratic Programming Enhanced Model (QPM) as a model which learns globally interpretable class representations. The QPM architecture is characterised by a decision layer that is both sparse (weight matrix $\boldsymbol{W}$ contains only a few non-zero entries) and low-dimensional.



Figure 3. Three-Stage Training Procedure for QPM

To transform a dense model (non-sparse weights, conventional neural network) into a QPM, the pipeline illustrated in Fig. 3 is employed. To reduce conceptual ambiguity between features, Norrenbrock et al. [41] introduced the Feature Diversity Loss, hereafter referred to as $\mathcal{L}_{\text{div}}$ and defined in Sec. 7. The objective of $\mathcal{L}_{\text{div}}$ is to encourage the representation of distinct, mutually independent concepts within the features, thereby enhancing the degree of model interpretability.

Initially, the dense model is trained using the $\mathcal{L}_{\text{div}}$ loss to ensure the emergence of diverse features. To enforce model sparsity and achieve the required low-dimensional feature space, the framework solves a quadratic program (QP). This QP is used to select N_f^* features from the N_f features of the dense model (where $N_f^* \ll N_f$). The selection is defined by a vector $\boldsymbol{s} \in \{0,1\}^{N_f}$, such that:

$$\sum_{d \in \mathcal{F}} s_d = N_f^* \tag{1}$$

where $d \in \mathcal{F} = \{1, \dots, N_f\}$ indexes the set of all features $\mathcal{F}$. Simultaneously each class $c \in \mathcal{C}$ is assigned a subset of these selected features $d \in \mathcal{F}^*$ consisting of N_f^c elements. This mapping is $\boldsymbol{W}^{\text{sparse}} \in \{0,1\}^{N_c \times N_f^*}$, where W_{cd}^{sparse} is 1 if class c is assigned feature d, and 0 otherwise. Consequently:

$$\sum_{d \in \mathcal{F}} W_{cd}^{\text{sparse}} = N_f^c \qquad \forall\, c \in \mathcal{C} \tag{2}$$

The assignment process relies on the optimisation of three components defined in [44]: Maximising the correlation between class c and its assigned feature activations, minimising similarity between selected features $\boldsymbol{s}$, and maximising the bias b_d to prioritise local features. For

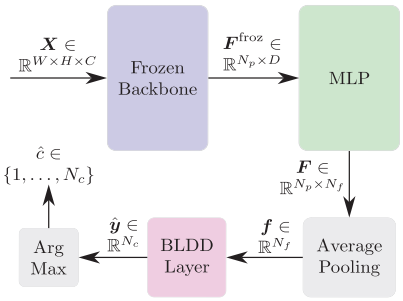


Figure 4. Architecture of our proposed DINO-QPM. Patch embeddings extracted using the frozen backbone are first projected via an MLP into the problem-specific feature space. Subsequently, our BLDD layer performs sparse feature assignment to yield a globally interpretable image classification.

a comprehensive derivation of the QP objective function, the reader is referred to the original work [44]. Finally the model is retrained with the constraint $\boldsymbol{W} = \boldsymbol{W}^{\text{sparse}}$, considering only the subset of selected features $\mathcal{F}^* \subset \mathcal{F}$ to arrive at the final QPM. When fine-tuning the features adapt to this forced assignment and finally become more interpretable (see Sec. 5.2).

3.2. DINO

DINO (Self-**DI**stillation with **NO** labels), introduced by Caron et al. [8], is a framework for self-supervised representation learning [22, 24, 25] of visual embeddings. Unlike traditional contrastive learning approaches [11, 24, 56, 62], DINO does not rely on negative samples to distinguish the input from other instances and yet achieves remarkable results in various image analysis benchmarks [15, 28, 46, 68]. Beyond architectural changes, its successor DINOv2 utilises a novel pipeline for large-scale training data curation [46]. Darcet et al. [16] investigate a specific challenge exacerbated by the transition to DINOv2, though present in other ViT architectures: outlier tokens (or artifacts) in attention maps characterised by unusually high norms. Darcet et al. [16] conclude that the ViT utilises these tokens to store global context in areas of low information density. To mitigate this, the authors propose adding "register tokens", non-spatial tokens similar to the CLS token, to act as a storage for global information. These registers are discarded during downstream analysis tasks [16].

In our experiments (Sec. 5), we compare various sizes of DINOv2, both with and without register tokens, demonstrating their significant benefit for DINO-QPM. Further, evaluation is conducted on the original DINO [8].

4. Method

Our contribution is DINO-QPM, a model that achieves globally interpretable image classification by enforcing a structured decision process. Given an input image $\boldsymbol{X}$ from the input space $\mathcal{V} = \mathbb{R}^{W\times H\times C}$ where H and W are height and width of the image respectively and C is the number of image channels, a visual foundation model Φ, such as DINOv2, produces a global image representation $\boldsymbol{f}^{\text{froz}}_{\text{CLS}} \in \mathbb{R}^D$ and local image representations $\boldsymbol{F}^{\text{froz}} \in \mathbb{R}^{N_p\times D}$. $\boldsymbol{F}^{\text{froz}}$ can be interpreted either as a collection of D spatial feature maps of size N_p or as N_p individual patch representations embedded in $\mathbb{R}^D$. The objective is to construct a classifier $g : \mathcal{V} \to \mathcal{C}$ based on $\boldsymbol{F}^{\text{froz}}$ and $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$, which maps each input data point $\boldsymbol{X} \in \mathcal{V}$ to a corresponding class $c \in \mathcal{C}$. We introduce the superscript "froz" to indicate the frozen nature of the backbone's output.

To achieve this, our approach builds upon the QPM framework proposed by Norrenbrock et al. [44] to identify a subset of selected features $\mathcal{F}^* \subset \mathcal{F} = \{1,\dots,N_f\}$ and assign N^c_f features from this subset to each class. The exact procedure is shown in Fig. 4. Interestingly, it is sufficient to consider exclusively the frozen local feature maps $\boldsymbol{F}^{\text{froz}} \in \mathbb{R}^{N_p\times D}$ while discarding the global feature vector $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$, a choice we justify empirically in Sec. 5.3. This architectural choice is driven by the hypothesis that a global representation transparently built from local evidence is more inherently interpretable than the complex, pre-aggregated representation of the CLS token. By intentionally discarding $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$, we prevent its internal, opaque aggregation process from introducing features that cannot be inherently localised.

The purpose of the MLP is to facilitate a task-specific transformation MLP : $\mathbb{R}^D \to \mathbb{R}^{N_f}$ for the patch representations $\boldsymbol{F} \in \mathbb{R}^{N_p\times N_f}$. This transformation maps the initial D-dimensional space to N_f features, which is essential because a sparse Binary Low-Dimensional Decision (BLDD) layer alone provides no capacity for such a transformation.

The final feature vector $\boldsymbol{f} = \text{AvgPool}\big(\boldsymbol{F}\big) \in \mathbb{R}^{N_f}$ is then constructed as a direct average across the spatial dimensions. Consequently, $\boldsymbol{F}$ contains a saliency map of N_p elements for each feature $d \in \mathcal{F}$. These saliency maps can be upsampled to the original input image resolution for visualisation purposes (see Fig. 5) and directly highlight where evidence for each feature is found.

Subsequently, the BLDD layer BLDD : $\mathbb{R}^{N_f} \to \mathbb{R}^{N_c}$ is applied to the feature vector $\boldsymbol{f}$:

$$\text{BLDD}(\boldsymbol{f}) = \begin{cases} \boldsymbol{W}^{\text{sparse}}\boldsymbol{f}_{(\boldsymbol{s})}, & \text{if fine-tuning} \\ \boldsymbol{W}\boldsymbol{f}, & \text{otherwise} \end{cases}$$

During the dense training phase, the BLDD layer maps the feature vector $\boldsymbol{f}$ to the classification vector $\hat{\boldsymbol{y}}$ using a dense projection matrix $\boldsymbol{W} \in \mathbb{R}^{N_c\times N_f}$.

In the fine-tuning stage, the selection vector $\boldsymbol{s} \in \{0,1\}^{N_f}$ is used to extract N^*_f features from the N_f elements of the feature vector, yielding $\boldsymbol{f}_{(\boldsymbol{s})} \in \mathbb{R}^{N^*_f}$. This subset $\boldsymbol{f}_{(\boldsymbol{s})}$ is then mapped to the classification vector $\hat{\boldsymbol{y}}$ via the

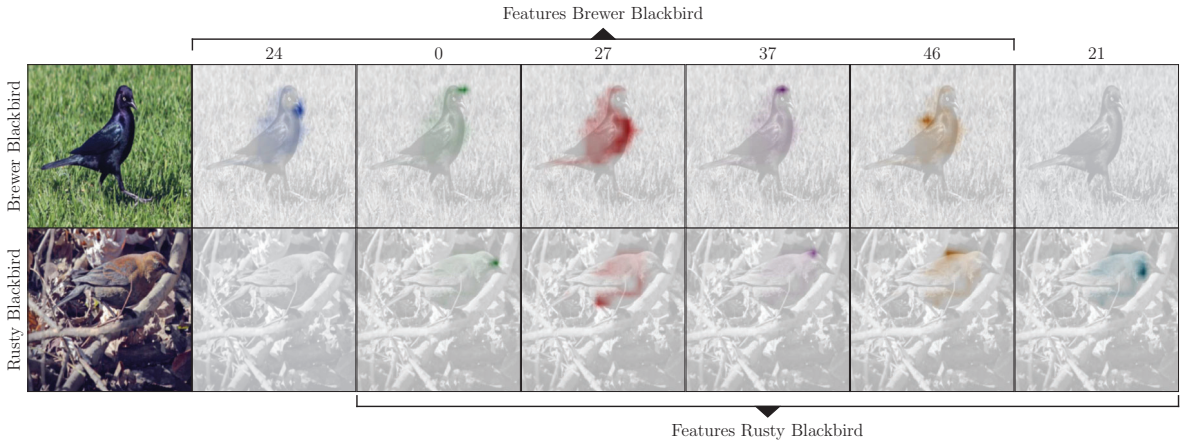


Figure 5. Comparison of a Brewer's Blackbird image with a Rusty Blackbird image. From the selected features $\mathcal{F}^*$, $N_f^{\hat{c}} = 5$ utilised features were selected for both classes using the QP; the corresponding feature maps from $\boldsymbol{F}$ are visualised as saliency maps. Both classes share 4 out of the 5 features and can thus be distinguished by the non-shared features. Notably, the model differentiates the Brewer's Blackbird using feature 24, which localises the beak. This aligns perfectly with established ornithological expertise, where beak morphology is considered a primary diagnostic trait [1, 55].

sparse projection matrix $\boldsymbol{W}^{\text{sparse}} \in \{0,1\}^{N_c \times N_f^*}$; notably, both $\boldsymbol{s}$ and $\boldsymbol{W}^{\text{sparse}}$ are derived using the QPM (Sec. 3.1). Specifically, each class $c \in \mathcal{C} = \{1, \ldots, N_c\}$ is assigned exactly N_f^c features. In both cases, the predicted class $\hat{c}$ is determined as the index of the maximum value in $\hat{\boldsymbol{y}}$.

For training purposes we use the exact pipeline described in Sec. 3.1, also to determine $\boldsymbol{s}$ and $\boldsymbol{W}^{\text{sparse}}$. Besides the Cross-Entropy loss $\mathcal{L}_{\text{CE}}$ and the already introduced $\mathcal{L}_{\text{div}}$, our loss function consists of two L1 sparsity losses, one for the feature vector $\mathcal{L}_{\text{L1-FV}} = \text{Mean}(\text{Abs}(\boldsymbol{f}))$ and $\mathcal{L}_{\text{L1-FM}} = \text{Mean}(\text{Abs}(\boldsymbol{F}))$ for the feature maps which are used to reduce spatial clutter and background noise in the feature maps to significantly increase accuracy and Plausibility. We explicitly show these benefits in Sec. 5.3 and Sec. 10 respectively.

5. Experiments

We evaluate our proposed DINO-QPM on the problem of fine-grained image classification, in line with previous work on inherently interpretable models [41–44]. Stanford Cars [34] and CUB-2011 [64] are used as datasets, as they are the default for this problem. For CUB-2011 we do not use cropping to the object of interest to demonstrate how DINO-QPM exploits the strong general features of its backbone. Additionally, CUB-2011 offers human annotated masks of the region of interests which enables a quantification of a models Plausibility. DINOv2 ViT-B/14 [46] with register tokens [16] serves as the primary backbone for the various experiments while we also test out different sizes with and without registers as well as DINOv1 [8]. Following the QPM [44], $N_f^* = 50$ features are selected from the initial

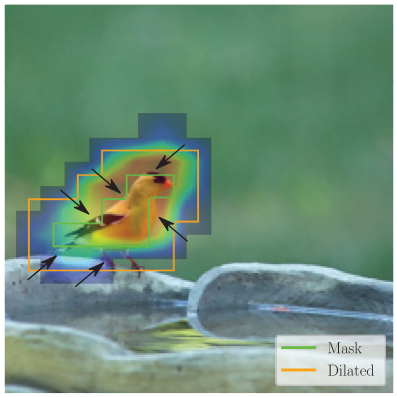


Figure 6. Visualisation of the Plausibility metric on a sample from the CUB2011 dataset (American Goldfinch). The metric quantifies the fraction of the cumulative weighted feature map activation $\widetilde{\boldsymbol{F}}$ that falls within the dilated object segmentation mask $\boldsymbol{M}^{\text{dil}}$. The arrows highlight the relevant features at the objects' edge that are missed by the non-dilated mask.

N_f features, with $N_f^c = 5$ features assigned to each class, unless otherwise noted. Further details on our implementation can be found in Sec. 9. For all experiments, we train on five different randomly chosen seeds and calculate statistics such as the mean and standard deviation (see Sec. 13).

5.1. Metrics

We evaluate our model following QPM [44] on their proposed metrics, using the default metrics for compact interpretable models, as well as introduce two metrics, *Plausibility* and *Patch Contextualisation*.

We desire our representations to be diverse [3], contrastive [35], general and compact [50] as such representations are ideally suited for generating human-interpretable explanations [39]. In order to quantify this, we utilise interpretability measures already introduced in Norrenbrock et al. [44]. We employ SID@5, which considers the spatial distinctiveness of feature maps to quantify diversity; meaning that high diversity implies feature maps activate diversely, at different spatial positions. Contrastiveness evaluates the overlap [27] between two components of a Gaussian Mixture Model (GMM) fit to the feature distribution; a feature is considered maximally contrastive if represented by two entirely non-overlapping distributions. Lastly, we evaluate Class-Independence, which measures the proportion of zero-based feature activations across the dataset that is not focused on the most relevant class. We use this as a proxy for feature generality, where high class-independence indicates that features capture broad, general concepts rather than being highly specific to a single class. The concrete definitions of the metrics are given in Sec. 8.

To evaluate the model's ability to localise relevant features within the object of interest, we introduce the Plausibility metric, following the terminology of Jacovi and Goldberg [29]. This metric quantifies the fraction of the cumulative feature map activation, weighted by feature relevance, that falls within the ground-truth object boundaries. For the CUB-2011 dataset, we utilise the provided segmentation masks [19] $\boldsymbol{M} \in \{0,1\}^{N_p}$ as a tokenised form to define these regions. Inspired by the poiting game [67] and following Grad-CAM by Selvaraju et al. [57], the feature maps $\boldsymbol{F}$ are weighted by their decisional relevance. First, the aggregate weighted feature map $\widetilde{\boldsymbol{F}}$ is calculated as:

$$\widetilde{\boldsymbol{F}} = \sum_{d \in \mathcal{F}} W_{\hat{c}d}\, \boldsymbol{F}_d \tag{3}$$

where $\hat{c}$ is the predicted class, and each feature map $\boldsymbol{F}_d$ is scaled by its relevance weight $W_{\hat{c}d}$. We define Plausibility as the proportion of the Grad-CAM map $\widetilde{\boldsymbol{F}}$ activating within the region of interest:

$$\text{Plausibility} = \frac{\sum_{p \in \mathcal{P}} \widetilde{F}_p \cdot M_p^{\text{dil}}}{\sum_{p \in \mathcal{P}} \widetilde{F}_p} \tag{4}$$

$$\boldsymbol{M}^{\text{dil}} = \boldsymbol{M} \oplus \mathbb{1}_{3,3} \tag{5}$$

where $p \in \mathcal{P} = \{1, \ldots, N_p\}$ indexes the set of all image patches. $\boldsymbol{M}^{\text{dil}}$ is generated by dilating the original mask with a 3×3 identity structuring element $\mathbb{1}_{3,3}$. This dilation provides a spatial margin for features that might otherwise overlap with the background due to the patch structure, ensuring that activations accurately capturing the object's silhouette are not unfairly penalised by the metric as visualised in Fig. 6. In order to obtain a dataset-wide representative value, the mean Plausibility is computed across all images $\boldsymbol{X} \in \mathcal{X}_{\text{train}}$. A higher Plausibility indicates the Grad-CAM visualisation matches human expectations by concentrating relevant features within the region of interest.

To quantify the degree to which the collective patch representations align with the global semantic information, we introduce the *Patch Contextualisation* metric. We first define the mean patch embedding $\overline{\boldsymbol{F}^{\text{froz}}}$ and then calculate its cosine similarity with the global class token $\boldsymbol{f}_{\text{CLS}}^{\text{froz}}$:

$$\overline{\boldsymbol{F}^{\text{froz}}} = \frac{1}{N_p} \sum_{p \in \mathcal{P}} \boldsymbol{F}_p^{\text{froz}} \tag{6}$$

$$\text{Patch Contextualisation} = \cos\left(\overline{\boldsymbol{F}^{\text{froz}}},\, \boldsymbol{f}_{\text{CLS}}^{\text{froz}}\right) \tag{7}$$

A higher score indicates that the average spatial representation is strongly aligned with $\boldsymbol{f}_{\text{CLS}}^{\text{froz}}$, suggesting that the patch embeddings generally share the same semantic direction as the global context. Conversely, a lower score implies a divergence, indicating that the patch representations contain less global information about the image. We utilise this metric to understand differences between different backbones and bring insights into how various DINO models differ.

5.2. Main Results

In this section, we present our main experimental results. In Tab. 1, we evaluate the interpretability metrics introduced in Sec. 5.1 alongside the accuracy of each approach, while Tab. 2 outlines compactness and training time. Across both datasets, DINO-QPM achieves the highest overall accuracy and outperforms the baselines and other approaches in plausibility, demonstrating a substantial increase in both metrics compared to dense representations. Furthermore, DINO-QPM exhibits strong interpretability, reaching high feature diversity (SID@5) and Class-Independence while securing the highest Contrastiveness. Crucially, it accomplishes this while extracting features that are localised by design and maintaining representational compactness, aligning with other interpretability-enhancing approaches (or even surpassing them, as seen with DINO-QPM Compact), but contrasting sharply with the dense baselines. From a computational perspective, relying on a frozen DINOv2 backbone drastically reduces the required training resources, particularly when compared to the ResNet-50 QPM, as it enables pre-computation of patch and `CLS` token embeddings. Notably, our linear probe reaches lower accuracy than reported in the original paper [46]. This discrepancy arises because they use a different evaluation scheme, which includes output of up to four layers of the ViT while allowing the concatenation of average-pooled patch embeddings to the `CLS` tokens. The supplementary material provides full results with standard deviations (Sec. 13) and further visualisations (Sec. 12) contrasting DINO-QPM's faithful explanations against the baseline's ungrounded attributions across successes, baseline failures, and interpretable failure cases.

Method	Local. Features	Accuracy ↑		Plausibil. ↑	SID@5 ↑		Class-Indep. ↑		Contrast. ↑	
		CUB	CARS		CUB	CARS	CUB	CARS	CUB	CARS
DINOv2 $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$ Linear Probe	✗	87.9	91.7	42.6	50.9	51.5	**99.2**	**99.1**	59.2	60.9
Dense $\boldsymbol{F}^{\text{froz}}$	✓	78.1	92.9	32.7	**91.8**	**93.1**	98.8	98.7	84.5	82.8
Resnet50 Baseline [44]	✓	83.9	92.5	60.7	57.1	51.5	98.0	97.9	74.6	75.1
Resnet50 QPM [44]	✓	82.9	92.1	82.9	89.6	88.2	96.8	97.8	93.6	97.1
DINO-SLDD	✓	84.6	92.9	78.0	88.7	90.9	94.4	93.9	93.0	94.9
DINO-QSENN	✓	85.4	93.3	86.0	91.5	92.6	93.6	94.0	94.4	94.9
DINO-QPM (Ours)	✓	**88.3**	**94.0**	**95.0**	90.1	91.7	93.7	93.7	**100.0**	**100.0**
DINO-QPM Compact (Ours)	✓	**88.3**	**94.0**	94.4	–	–	93.8	93.6	**100.0**	**100.0**

Table 1. Comparison with state-of-the-art interpretable models. We report Accuracy, Plausibility, SID@5, Class-Independence, and Contrastiveness (all metrics in %). Features of a model are localised if they have a direct connection to the feature vector used for classification. The Plausibility metric is evaluated only on CUB-2011 due to the availability of segmentation masks. Dense $\boldsymbol{F}^{\text{froz}}$ is the dense model of DINO-QPM and DINOv2 $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$ Linear Probe is a linear probe [11] trained on top of the frozen `CLS` representation. For DINO-SLDD and DINO-QSENN, we employ a pipeline closely resembling the one described in Sec. 4, with the exception of the feature selection mechanisms, which follow Norrenbrock et al. [41] and Norrenbrock et al. [42], respectively.

Method	# Features (N_f)	# Features per class (N^c_f)	Avg. Training Time per Epoch [s]
DINOv2 $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$ Linear Probe	768	768	3
DINOv2 $\boldsymbol{F}^{\text{froz}}$ Baseline	512	512	6
Resnet50 Baseline	2048	2048	40
Resnet50 QPM	50	5	40
DINO-SLDD	50	5	6
DINO-QSENN	50	5	6
DINO-QPM (Ours)	50	5	6
DINO-QPM Compact (Ours)	40	4	6

Table 2. Comparison of model complexity and training efficiency. We report the total number of features (N_f), features per class (N^c_f), and the average training time per epoch in seconds on the referenced hardware (Sec. 9).

Input Source	Loc. Feat.	Registers	Acc. ↑	Plausibil. ↑
$\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$	✗	✗	87.3	96.9
		✓	87.6	**99.2**
$\boldsymbol{F}^{\text{froz}}$	✓	✗	83.3	73.5
		✓	**88.3**	95.0

Table 3. Ablation study showing the impact of input source and register tokens on CUB-2011. The local features column indicates whether the input representation preserves spatial information.

5.3. Ablation Studies

In this section, we conduct a series of ablation studies to systematically evaluate the architectural and hyperparameter choices for DINO-QPM which influence predictive performance and model interpretability.

Backbone Architecture	Acc. ↑	Patch Context. ↑
DINO ViT-B/16	37.1	8.9
DINOv2 ViT-S/14 Reg.	83.4	42.9
DINOv2 ViT-B/14 Reg.	88.3	43.9
DINOv2 ViT-L/14 Reg.	86.5	2.2

Table 4. Comparison of different backbone architectures on CUB-2011. We compare DINO with DINOv2 and different ViT sizes and report accuracy alongside Patch Contextualisation.

To compare DINO-QPM with a variant utilising the CLS token, we define its predictive pipeline as follows: $\hat{c} = \arg\max\left\{\text{BLDD}(\text{MLP}(\boldsymbol{f}^{\text{froz}}_{\text{CLS}}))\right\}$. To maintain a feature map equivalent, we apply the same MLP to the frozen patch embeddings and utilise the output as our feature maps.

In Table 3, we show that when using registers, achieving high accuracy only requires patch embeddings ($\boldsymbol{F}^{\text{froz}}$), allowing us to discard the CLS representation ($\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$). Despite $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$ scoring high on Plausibility, it lacks a direct connection between the features and their maps (i.e., no localised features). Therefore, high Plausibility is less meaningful than for DINO-QPM ($\boldsymbol{F}^{\text{froz}}$), which possesses this property.

Comparing different backbone sizes for DINOv2 (Tab. 4), we observe our proposed approach can be applied to all of them. However, the accuracy declines with larger and smaller backbones. For ViT-L, this might be due to low Patch Contextualisation. The model has insufficient global context in its patch embeddings, which hinders higher accuracy. For ViT-S we tested DINOv2 $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$ Linear Probe which reaches similar accuracy. Therefore, we conclude that the disproportionate (compared to [46]) decline is not

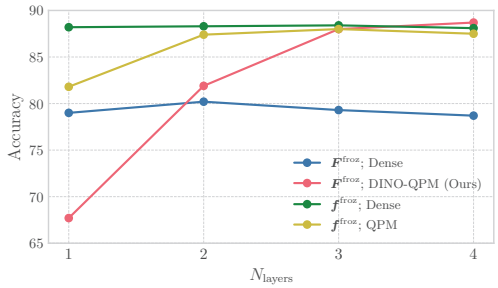
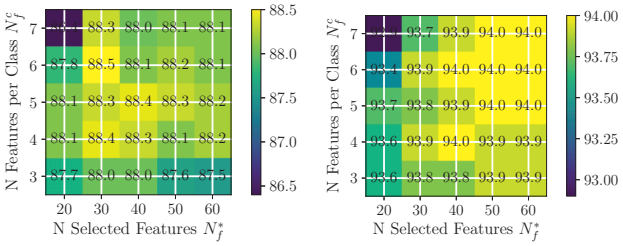


Figure 7. Impact of the number of MLP layers (N_{layers}) on classification accuracy. The plot compares the accurcy on CUB-2011 of using frozen patch-level feature maps ($\boldsymbol{F}^{\text{froz}}$) versus the global feature vector ($\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$) for both Dense and QPM



(a) CUB2011 (b) Stanford Cars

Figure 8. Impact of sparsity constraints on classification accuracy. We vary the total number of selected features N_f^* and the number of features assigned per class N_f^c. Compared to the QPM [44] we observe relatively low impact on accuracy.

attributable to DINO-QPM but their use of a different evaluation scheme (see Sec. 5.2). Hence, we demonstrate a previously unexplored difference between DINOv2 backbone sizes that affects interpretable downstream performance and may guide further research. Furthermore, DINO-QPM exhibits poor accuracy on DINO ViT-B/16 combined with low Patch Contextualisation. Frozen DINO in general does not perform well on CUB-2011 when evaluating a linear probe on $\boldsymbol{f}^{\text{froz}}_{\text{CLS}}$ [36, 38] and the low amount of global image information in the patch embeddings exacerbates this.

While the number of layers in the MLP has no measurable impact on the dense model, when introducing sparsity, especially using the patch embeddings, we observe a huge increase in model accuracy (Fig. 7). Notably, comparing dense and QPM accuracy for higher N_{layers}, the QPM is able to outperform its dense model by almost 10% using $\boldsymbol{F}^{\text{froz}}$.

We evaluate the compactness of DINO-QPM across Stanford Cars and CUB-2011 w.r.t the total number of selected features N_f^* and the features per class N_f^c (Fig. 8). We observe that the overall variation in accuracy across both datasets is remarkably low ($< 1\%$). This contrasts with the higher sensitivity reported in Norrenbrock et al. [44]. Hence, Dino-QPM enables higher accuracy at a higher compactness, e.g. using our Compact Model in Tab. 1.

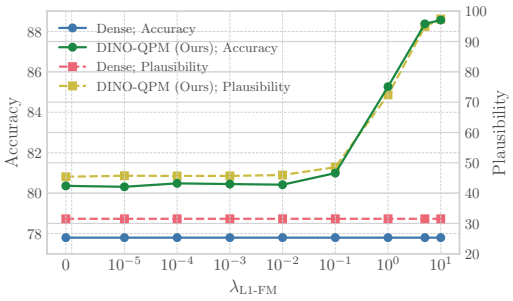


Figure 9. Effect of varying the $\mathcal{L}_{\text{L1-FM}}$ weight during finetuning compared to the dense model. Higher penalty weights yield substantial improvements in both accuracy and plausibility metrics.

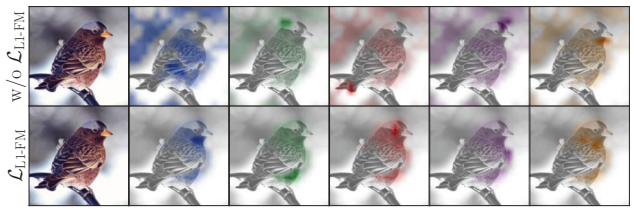


Figure 10. Qualitative ablation of $\mathcal{L}_{\text{L1-FM}}$ on the Gray-crowned Rosy Finch. Without $\mathcal{L}_{\text{L1-FM}}$ (top row), feature activations exhibit background noise and spatial scatter. Adding $\mathcal{L}_{\text{L1-FM}}$ (bottom row) suppresses this noise, resulting in distinct activations semantically localised to specific object parts.

A significant increase in model accuracy which highly correlates with Plausibility is observed when increasing $\lambda_{\text{L1-FM}}$ (Fig. 9), the weight of the L1 sparsity loss on the feature maps. A potential explanation is that the regularisation forces the model to concentrate its activation mass on object regions relevant for classification which then has a tremendously positive effect on accuracy (Fig. 10).

6. Conclusion

In this work, we introduced DINO-QPM as a compactness-based interpretability adapter applied on top of frozen backbones like DINOv2 [46] to achieve globally interpretable image classification. By establishing a direct connection between the feature vector and its corresponding maps via a non-standard average-pooling approach, DINO-QPM achieves exceptional spatial localisation, substantiating its highly faithful decision process. Paired with inherently contrastive, diverse, and general features, our approach outperforms the uninterpretable DINOv2 linear probe, as well as other baselines and applicable approaches, on fine-grained image classification, delivering state-of-the-art interpretability on top of frozen visual foundation models while maintaining exceptional accuracy.

Acknowledgements

Financial support for this research was provided by the MWK of Lower Saxony through the Hybrint (VWZN4219) and LCIS (VWZN4704) projects. Furthermore, funding was granted by the Deutsche Forschungsgemeinschaft (DFG) as part of Germany's Excellence Strategy for the PhoenixD (EXC2122) and Quantum Frontiers (EXC2123) Clusters of Excellence, and by the European Union under grant agreement no. 101136006 – XTREME.

References

[1] Rusty Blackbird Identification, All About Birds, Cornell Lab of Ornithology. https://www.allaboutbirds.org/guide/Rusty_Blackbird/id. 5

[2] Samira Abnar and Willem Zuidema. Quantifying Attention Flow in Transformers, 2020. 3

[3] David Alvarez-Melis and Tommi S. Jaakkola. Towards robust interpretability with self-explaining neural networks. *CoRR*, abs/1806.07538, 2018. 2, 6

[4] Hubert Baniecki and Przemyslaw Biecek. Birds look like cars: adversarial analysis of intrinsically interpretable deep learning. *Machine Learning*, 114(12), 2025. 2

[5] Moritz Bohle, Mario Fritz, and Bernt Schiele. B-cos Networks: Alignment is All We Need for Interpretability. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10319–10328, New Orleans, LA, USA, 2022. IEEE. 3

[6] Moritz Böhle, Navdeeppal Singh, Mario Fritz, and Bernt Schiele. B-cos Alignment for Inherently Interpretable CNNs and Vision Transformers, 2024. 3

[7] Lennart Brocki, Jakub Binda, and Neo Christopher Chung. Class-Discriminative Attention Maps for Vision Transformers, 2024. 2, 3

[8] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. DinoV1: Emerging Properties in Self-Supervised Vision Transformers, 2021. 2, 4, 5

[9] Hila Chefer, Shir Gur, and Lior Wolf. Transformer Interpretability Beyond Attention Visualization, 2021. 2, 3

[10] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This Looks Like That: Deep Learning for Interpretable Image Recognition, 2019. 1

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations, 2020. 4, 7, 12

[12] Xiaokang Chen, Mingyu Ding, Xiaodi Wang, Ying Xin, Shentong Mo, Yunhao Wang, Shumin Han, Ping Luo, Gang Zeng, and Jingdong Wang. Context Autoencoder for Self-Supervised Representation Learning, 2023. 2

[13] Minjae Chung, Jong Bum Won, Ganghyun Kim, Yujin Kim, and Utku Ozbulak. Evaluating Visual Explanations of Attention Maps for Transformer-based Medical Imaging. pages 110–120. 2025. 2

[14] Ian Covert, Chanwoo Kim, and Su-In Lee. Learning to Estimate Shapley Values with Vision Transformers, 2023. 3

[15] Beilei Cui, Mobarakol Islam, Long Bai, and Hongliang Ren. Surgical-dino: Adapter learning of foundation models for depth estimation in endoscopic surgery, 2024. 1, 4

[16] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision Transformers Need Registers, 2024. 4, 5

[17] Aaron Defazio, Xingyu Alice Yang, Harsh Mehta, Konstantin Mishchenko, Ahmed Khaled, and Ashok Cutkosky. The Road Less Scheduled, 2024. 2

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2021. 2

[19] Ryan Farrell. CUB-200-2011 Segmentations, 2022. Segmentation masks for the CUB-200-2011 dataset. 6

[20] Patrick Glandorf and Bodo Rosenhahn. Pruning by block benefit: Exploring the properties of vision transformer blocks during domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3006–3016, 2025. 3

[21] Patrick Glandorf, Timo Kaiser, and Bodo Rosenhahn. Hypersparse neural networks: Shifting exploration to exploitation through adaptive regularization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1234–1243, 2023. 3

[22] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised Learning, 2020. 4

[23] Gurobi Optimization, LLC. Gurobi optimizer reference manual, 2024. 2

[24] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality Reduction by Learning an Invariant Mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, pages 1735–1742, 2006. 4

[25] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum Contrast for Unsupervised Visual Representation Learning, 2020. 4

[26] Adrian Hoffmann, Claudio Fanconi, Rahul Rade, and Jonas Kohler. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. *CoRR*, abs/2105.02968, 2021. 2

[27] Henry F. Inman and Edwin L. Bradley Jr. The overlapping coefficient as a measure of agreement between probability distributions and point estimation of the overlap of two normal densities. *Communications in Statistics - Theory and Methods*, 18(10):3851–3874, 1989. 6, 2

[28] Sergio Izquierdo and Javier Civera. Optimal transport aggregation for visual place recognition, 2023. 1, 4

[29] Alon Jacovi and Yoav Goldberg. Towards Faithfully Interpretable NLP Systems: How should we define and evaluate faithfulness?, 2020. 6

[30] Sarthak Jain and Byron C. Wallace. Attention is not Explanation, 2019. 2

[31] Timo Kaiser, Thomas Norrenbrock, and Bodo Rosenhahn. Uncertainsam: Fast and efficient uncertainty quantification of the segment anything model. In *Forty-second International Conference on Machine Learning*. 1

[32] Rojina Kashefi, Leili Barekatain, Mohammad Sabokrou, and Fatemeh Aghaeipoor. Explainability of Vision Transformers: A Comprehensive Review and New Perspectives, 2023. 3

[33] Sunnie S. Y. Kim, Nicole Meister, Vikram V. Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: Evaluating the human interpretability of visual explanations, 2022. 2

[34] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3D object representations for fine-grained categorization. In *4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13)*, Sydney, Australia, 2013. 5

[35] Peter Lipton. Contrastive explanation. *Royal Institute of Philosophy Supplement*, 27:247–266, 1990. 2, 6

[36] Yiming Liu, Yuhui Zhang, Dhruba Ghosh, Ludwig Schmidt, and Serena Yeung-Levy. Data or Language Supervision: What Makes CLIP Better than DINO?, 2025. 8

[37] Chiyu Ma, Jon Donnelly, Wenjun Liu, Soroush Vosoughi, Cynthia Rudin, and Chaofan Chen. Interpretable Image Classification with Adaptive Prototype-based Vision Transformers. 2024. 1, 3

[38] Markus Marks, Manuel Knott, Neehar Kondapaneni, Elijah Cole, Thijs Defraeye, Fernando Perez-Cruz, and Pietro Perona. A Closer Look at Benchmarking Self-Supervised Pretraining with Image Classification, 2024. 8

[39] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019. 2, 6

[40] Meike Nauta, Ron van Bree, and Christin Seifert. Neural Prototype Trees for Interpretable Fine-grained Image Recognition, 2021. 1

[41] Thomas Norrenbrock, Marco Rudolph, and Bodo Rosenhahn. Take 5: Interpretable Image Classification with a Handful of Features, 2023. 2, 3, 5, 7, 1, 12

[42] Thomas Norrenbrock, Marco Rudolph, and Bodo Rosenhahn. Q-SENN: Quantized Self-Explaining Neural Networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21482–21491, 2024. 7, 1, 12

[43] Thomas Norrenbrock, Timo Kaiser, Sovan Biswas, Neslihan Kose, Ramesh Manuvinakurike, and Bodo Rosenhahn. CHiQPM: Calibrated Hierarchical Interpretable Image Classification, 2025. 2

[44] Thomas Norrenbrock, Timo Kaiser, Sovan Biswas, Ramesh Manuvinakurike, and Bodo Rosenhahn. QPM: Discrete Optimization for Globally Interpretable Image Classification, 2025. 2, 3, 4, 5, 6, 7, 8, 1, 12

[45] Tuomas Oikarinen, Subhro Das, Lam M. Nguyen, and Tsui-Wei Weng. Label-Free Concept Bottleneck Models, 2023. 2, 3

[46] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision, 2024. 1, 2, 4, 5, 6, 7, 8

[47] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. IA-REDˆ2: Interpretability-Aware Redundancy Reduction for Vision Transformers. In *Advances in Neural Information Processing Systems*, pages 24898–24911. Curran Associates, Inc., 2021. 3

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc., 2019. 2

[49] Katharina Prasse, Patrick Knab, Sascha Marton, Christian Bartelt, and Margret Keuper. DCBM: Data-Efficient Visual Concept Bottleneck Models, 2025. 3

[50] Stephen J. Read and Amy Marcus-Newhall. Explanatory coherence in social explanations: A parallel distributed processing account. *Journal of Personality and Social Psychology*, 65(3):429–447, 1993. 2, 6

[51] Eleanor Rosch. Principles of categorization. In *Cognition and categorization*, pages 27–48. Lawrence Erlbaum Associates, Hillsdale, NJ, 1978. 1

[52] Bodo Rosenhahn. Optimization of sparsity-constrained neural networks as a mixed integer linear program. *Journal of Optimization Theory and Applications*, 199(3):931–954, 2023. (open access). 3

[53] Cynthia Rudin. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215, 2019. 1

[54] Dawid Rymarczyk, Łukasz Struski, Jacek Tabor, and Bartosz Zieliński. ProtoPShare: Prototype Sharing for Interpretable Image Classification and Similarity Discovery. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021. 1

[55] Carl Savignac. *COSEWIC Assessment and Status Report on the Rusty Blackbird, Euphagus Carolinus, in Canada*. Committee on the Status of Endangered Wildlife in Canada, Ottawa, 2006. 5

[56] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015. 4

[57] 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. *International Journal of Computer Vision*, 128(2):336–359, 2020. 6

[58] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov,

Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3, 2025. 2

[59] Chung-En Sun, Tuomas Oikarinen, Berk Ustun, and Tsui-Wei Weng. Concept Bottleneck Large Language Models, 2025. 3

[60] Hugues Turbé, Mina Bjelogrlic, Gianmarco Mengaldo, and Christian Lovis. ProtoS-ViT: Visual foundation models for sparse self-explainable classifications, 2024. 1

[61] Hugues Turbé, Mina Bjelogrlic, Gianmarco Mengaldo, and Christian Lovis. Tell me why: Visual foundation models as self-explainable classifiers, 2025. 1

[62] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation Learning with Contrastive Predictive Coding, 2019. 4

[63] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 595–604, 2015. 10

[64] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 2, 5

[65] Yue Yang, Artemis Panagopoulou, Shenghao Zhou, Daniel Jin, Chris Callison-Burch, and Mark Yatskar. Language in a Bottle: Language Model Guided Concept Bottlenecks for Interpretable Image Classification, 2023. 2

[66] Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc Concept Bottleneck Models, 2023. 2, 3

[67] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pages 543–559. Springer, 2016. 2, 6

[68] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence, 2023. 1, 4

[69] Delong Zhao, Qiang Huang, Di Yan, Yiqun Sun, and Jun Yu. Partially Shared Concept Bottleneck Models, 2025. 3

[70] Zhijie Zhu, Lei Fan, Maurice Pagnucco, and Yang Song. Interpretable Image Classification via Non-parametric Part Prototype Learning, 2025. 1, 3

DINO-QPM: Adapting Visual Foundation Models for Globally Interpretable Image Classification

Supplementary Material

7. Feature Diversity Loss

To reduce conceptual ambiguity between features, Norrenbrock et al. [41] introduced the Feature Diversity Loss, hereafter referred to as $\mathcal{L}_{\text{div}}$. The objective of $\mathcal{L}_{\text{div}}$ is to encourage the representation of distinct, mutually independent concepts within the features, thereby enhancing the degree of model interpretability. Let $i \in \mathcal{I} = \{1, \dots, W_f\}$ and $j \in \mathcal{J} = \{1, \dots, H_f\}$ denote the spatial dimensions of the feature map $\boldsymbol{F}^d$ associated with feature $d \in \mathcal{F} = \{1, \dots, N_f\}$. Furthermore, let $\boldsymbol{W}_{(\hat{c})}$ represent the row of the weight matrix $\boldsymbol{W}$ corresponding to the predicted class $\hat{c}$, and $W_{\hat{c}d}$ represent the specific entry for class $\hat{c}$ and feature d. The diversity loss $\mathcal{L}_{\text{div}}$ is defined by the following equations:

$$\mathcal{L}_{\text{div}} = -\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \max_{d \in \mathcal{F}} \hat{S}_{ij}^d \tag{8}$$

where the weighted diversity maps $\hat{S}_{ij}^d$ are computed for all $i \in \mathcal{I}, j \in \mathcal{J}$, and $d \in \mathcal{F}$ according to:

$$\hat{S}_{ij}^d = \frac{\exp(F_{ij}^d)}{\sum_{i' \in \mathcal{I}} \sum_{j' \in \mathcal{J}} \exp(F_{i'j'}^d)} \frac{f_d}{\max_{d' \in \mathcal{F}} f_{d'}} \frac{|W_{\hat{c}d}|}{\|\boldsymbol{W}_{(\hat{c})}\|_2} \tag{9}$$

Eq. (9) employs the softmax function to normalize $\boldsymbol{F}^d$ across its spatial dimensions i and j. Simultaneously, the feature map is weighted in the feature dimension d: first, by the value of feature d relative to the maximum of the feature vector, and second, by scaling $W_{\hat{c}d}$ relative to the L_2-norm of the weights for all features associated with the predicted class. These components serve to highlight decision-relevant features. Eq. (8) then ensures that the normalized feature maps $\hat{\boldsymbol{S}}^d$ focus on distinct spatial regions. Overall, $\mathcal{L}_{\text{div}}$ acts as a regularizer to the standard cross-entropy loss, resulting in a total objective function:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \beta \mathcal{L}_{\text{div}} \tag{10}$$

where $\beta \in \mathbb{R}_+$ is a weighting hyperparameter.

8. Definition of Additional Interpretability Metrics

To assess model interpretability, we apply several metrics following Norrenbrock et al. [41, 42, 44]. Since interpretability is multifaceted, multiple metrics addressing distinct concepts are necessary.

Throughout this section, we utilise the following notation for index sets: $i \in \mathcal{I} = \{1, \dots, W_f\}$ and $j \in \mathcal{J} = \{1, \dots, H_f\}$ denote spatial dimensions, $d \in \mathcal{F} = \{1, \dots, N_f\}$ denotes the feature indices, $c \in \mathcal{C} = \{1, \dots, N_c\}$ denotes class indices, and $x \in \mathcal{X}_{\text{train}}$ represents samples from the training dataset.

8.1. SID@k

Similar to the $\mathcal{L}_{\text{div}}$ presented in Sec. 7, we utilise the Scale-Invariant Diversity (SID) from Norrenbrock et al. [44]. This metric measures the distinctiveness between the feature maps $\boldsymbol{F}^d$ of each feature d.

$$\begin{aligned} \widehat{F}_{ij}^d &= \frac{1}{F_{\text{avg}}^d} F_{ij}^d \\ \text{with} \quad F_{\text{avg}}^d &= \frac{1}{W_f H_f} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \left| F_{ij}^d \right| \end{aligned} \tag{11}$$

First, the feature maps $\boldsymbol{F}^d$ are normalized by their absolute mean F_{avg}^d for all $d \in \mathcal{F}$ (Eq. (11)).

$$\begin{aligned} \widehat{S}_{ij}^d = \frac{\exp\left(\widehat{F}_{ij}^d\right)}{\sum_{i' \in \mathcal{I}} \sum_{j' \in \mathcal{J}} \exp\left(\widehat{F}_{i'j'}^d\right)} \\ \forall\, i \in \mathcal{I}, j \in \mathcal{J}, d \in \mathcal{F} \end{aligned} \tag{12}$$

A softmax function is then applied to the normalized feature maps $\widehat{\boldsymbol{F}}^d$ to obtain $\widehat{\boldsymbol{S}}^d$.

$$\begin{aligned} \hat{S}_{ij}^{\max} = \max_{d \in \mathcal{F}_k} \hat{S}_{ij}^d \\ \forall\, i \in \mathcal{I}, j \in \mathcal{J} \end{aligned} \tag{13}$$

Subsequently, along the feature dimension, the maximum of the k highest-weighted, normalised feature maps $\widehat{\boldsymbol{S}}^d$ is computed for each spatial element. Here, $\mathcal{F}_k \subset \mathcal{F}$ denotes the subset of exactly those k features associated with the highest weights. The SID@k is defined as the sum over all $\hat{S}_{ij}^{\max}$, normalized by k.

$$\text{SID@k} = \frac{1}{k} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} \hat{S}_{ij}^{\max} \tag{14}$$

8.2. Class-Independence τ

Norrenbrock et al. [44] propose Class-Independence τ as a measure of whether features represent a general or a class-specific concept. For this purpose, the individual feature values f_x^d per data point and feature are first normalized over the entire training dataset such that their minimum is 0.

$$\begin{aligned} f_{x,\text{norm}}^d &= f_x^d - f_{\min}^d \\ \text{with} \quad f_{\min}^d &= \min_{x' \in \mathcal{X}_{\text{train}}} f_{x'}^d \end{aligned} \tag{15}$$

The resulting $f_{x,\text{norm}}^d$ values (for all $x \in \mathcal{X}_{\text{train}}, d \in \mathcal{F}$) are then used in conjunction with the label vector $\boldsymbol{l}^c$—where $l_x^c = 1$

if x belongs to class c, and 0 otherwise—to obtain φ^{cd}. This term indicates how strongly feature d focuses on class c.

$$\varphi^{cd} = \frac{\sum_{x \in \mathcal{X}_{\text{train}}} l_x^c \cdot f_{x,\text{norm}}^d}{\sum_{x \in \mathcal{X}_{\text{train}}} f_{x,\text{norm}}^d} \quad \forall\, c \in \mathcal{C}, d \in \mathcal{F} \tag{16}$$

By selecting the class c on which each feature d focuses most strongly and averaging these values, the Class-Dependence is obtained. The Class-Independence τ is then defined as the complement of the Class-Dependence relative to 1.

$$\tau = 1 - \frac{1}{N_f} \sum_{d \in \mathcal{F}} \max_{c \in \mathcal{C}} \varphi^{cd} \tag{17}$$

8.3. Contrastiveness

Let the empirical feature distribution $\hat{p}(f_x^d)$ be a normalized histogram over the vector $\boldsymbol{f}^d$, containing the feature values of a feature d for all training data. To measure contrastiveness, a Gaussian Mixture Model (GMM) with two components is constructed for each feature distribution $\hat{p}(f_x^d)$, yielding two normal distributions $\mathcal{N}_1^d$ and $\mathcal{N}_2^d$. The first component models the non-activation region, while the second approximates the activation region.

$$\text{Contrastiveness} = 1 - \frac{1}{N_f} \sum_{d \in \mathcal{F}} \text{Overlap}\left(\mathcal{N}_1^d, \mathcal{N}_2^d\right) \tag{18}$$

Contrastiveness results as the expected non-overlap [27] of the two distributions $\mathcal{N}_1^d$ and $\mathcal{N}_2^d$. Thus, a feature is considered (maximally) contrastive if and only if it can be represented by a bimodal distribution of two non-overlapping distribution functions.

9. Implementation Details

All input images are resized to 224×224 pixels and normalised according to the dataset mean values.

Unless otherwise specified, the Multi-Layer Perceptron (MLP) consists of four layers featuring ReLU activation and batch normalisation. The number of features is set to $N_f = 512$, and the number of neurons in the hidden layers is $N_{\text{hidden}} = 2048$. To manage the learning rate, a schedule-free approach following Defazio et al. [17] is employed in combination with Adam as our optimiser. In dense training we train for 40 epochs using a weight decay of $7 \cdot 10^{-4}$ with batch size 32 and a start learning rate of 10^{-3}. In our BLDD layer we use a dropout of 0.2. Besides $\mathcal{L}_{\text{CE}}$ we use $\mathcal{L}_{\text{div}}$ with $\lambda_{\text{div}} = 0.5$.
In fine-tuning training we train for 50 epochs with a weight decay of $8 \cdot 10^{-4}$ with batch size 32 and a start learning rate of $5 \cdot 10^{-3}$. We explicitly do not use dropout in our BLDD layer when fine-tuning. Besides $\mathcal{L}_{\text{CE}}$ we use $\mathcal{L}_{\text{div}}$ with $\lambda_{\text{div}} = 1$, $\mathcal{L}_{\text{L1-FM}}$ with $\lambda_{\text{L1-FM}} = 5$ and $\mathcal{L}_{\text{L1-FV}}$ with $\lambda_{\text{L1-FV}} = 1$.
The QP is solved using Gurobi [23], while the neural network architectures are implemented in PyTorch [48]. For measuring the training time in Tab. 2, we used an NVIDIA GeForce RTX 3090 GPU combined with an 11th Gen Intel(R) Core(TM) i9-11900K @ 3.50GHz CPU.

We report the mean and standard deviation across 5 random seeds for all models, with the exception of our DINO-QPM base configuration, which was evaluated over 15 seeds due to a configuration oversight. This larger sample size provides a more precise estimate of the mean without introducing any bias into the comparison.

10. Impact of Auxiliary Losses

The $\mathcal{L}_{\text{div}}$ loss, as proposed by Norrenbrock et al. [41] and introduced in detail in Sec. 7, is analysed here. Fig. 11 illustrates the influence of $\mathcal{L}_{\text{div}}$ on accuracy and SID@5. Notably, increasing the weight of this loss has a strong positive correlation with SID@5. Hence, the lightweight interpretability adapter can be steered similarly to the end-to-end trained models.

In the finetuning stage, besides the aforementioned $\mathcal{L}_{\text{L1-FM}}$ an additional L1 regularization loss $\mathcal{L}_{\text{L1-FV}}$ on the feature vector is introduced alongside $\mathcal{L}_{\text{div}}$. Looking at Fig. 12 we observe its positive impact on accuracy.

11. Impact of MLP Depth

Fig. 13 illustrates the accuracy plotted against the number of neurons in the MLP's hidden layers N_{hidden}. Small accuracy gains are observed up to $N_{\text{hidden}} = 2048$, regardless of the number of features N_f which is why we chose $N_{\text{hidden}} = 2048$ and $N_f = 512$, obtaining optimal accuracy while minimising compactness.

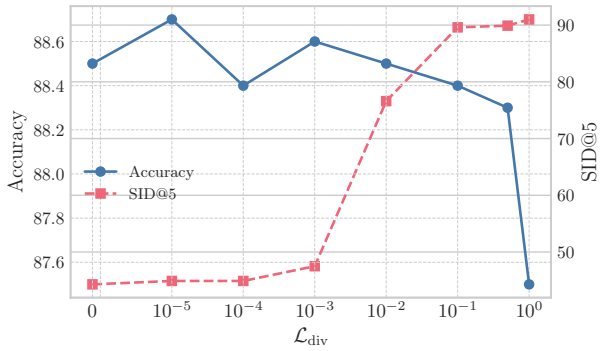


Figure 11. Accuracy and Feature Diversity (SID@5) on CUB-2011 across variations of the $\mathcal{L}_{\text{div}}$ weight during dense and finetuning training.

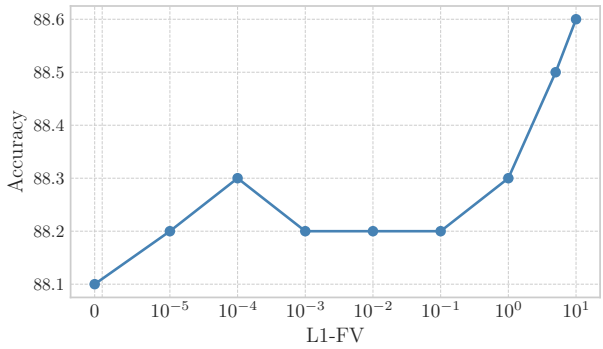


Figure 12. Impact of the $\mathcal{L}_{\text{L1-FV}}$ on CUB-2011 accuracy during finetuning.

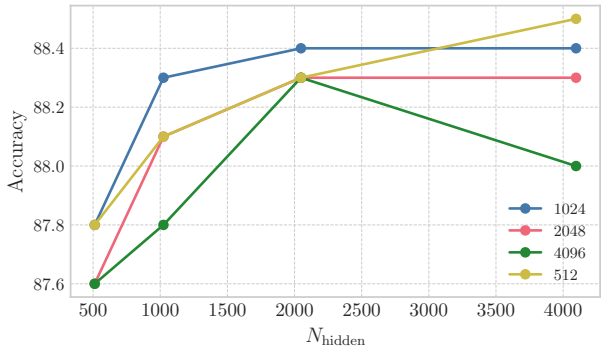


Figure 13. Mean finetuning accuracy on CUB-2011 for various numbers of features N_f across a range of hidden layer neurons N_{hidden} in the MLP. We observe small accuracy gains up until $N_{\text{hidden}} = 2048$ regardless of the number of features N_f.

12. Visualisations

12.1. Class Comparisons

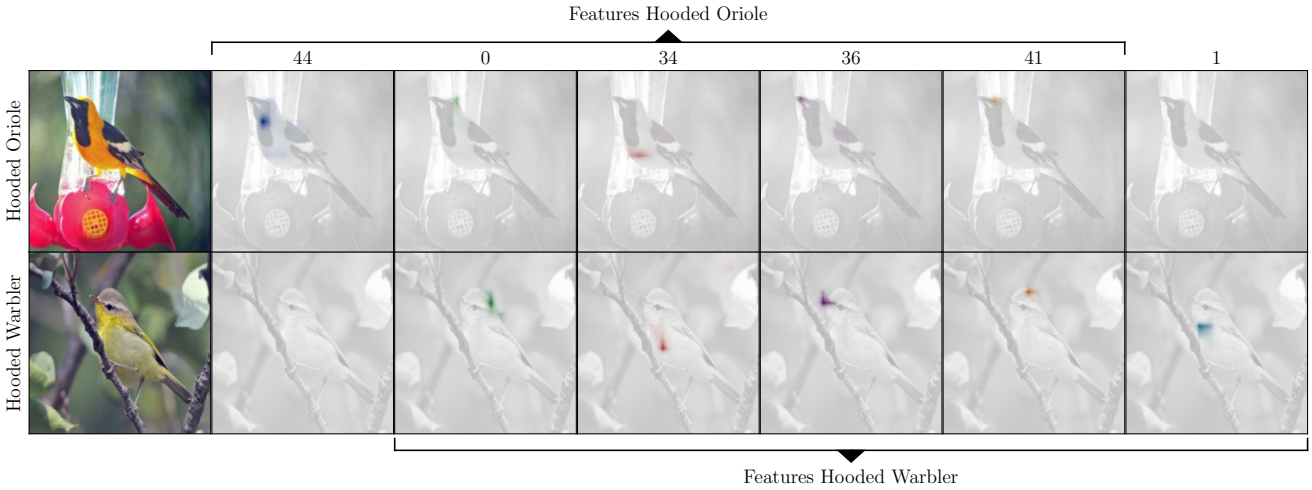


Figure 14. Faithful global interpretability on CUB-2011: DINO-QPM autonomously discovers the 5 diverse, generalisable features for each class used to represent the Hooded Oriole and Hooded Warbler, completely without external supervision. The probed QPM distinguishes them using their evidently different throat.

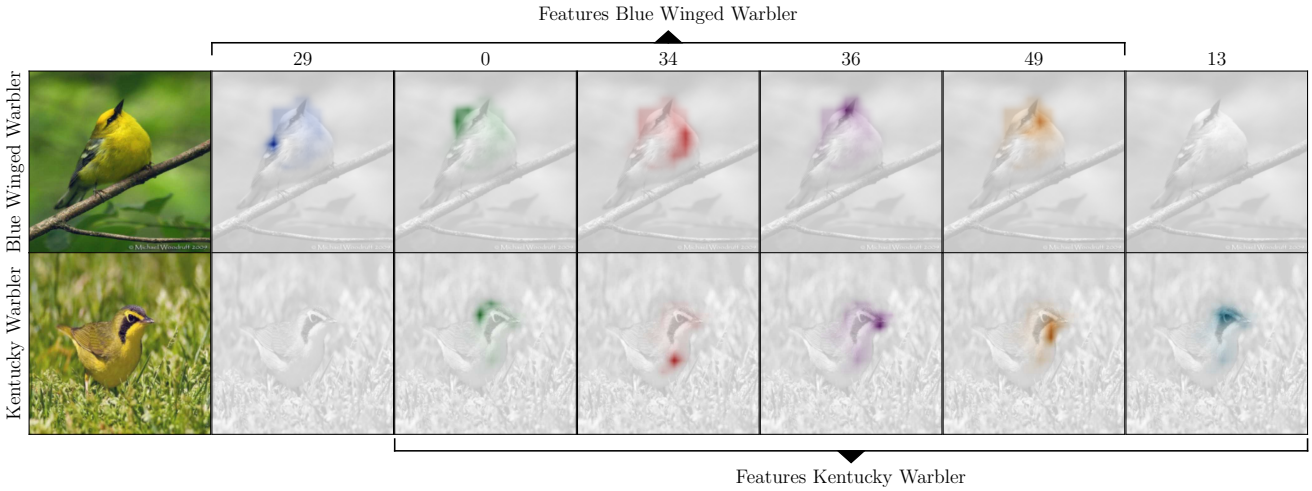


Figure 15. Faithful global interpretability on CUB-2011: DINO-QPM autonomously discovers the 5 diverse, generalisable features for each class used to represent the Blue Winged Warbler and Kentucky Warbler, completely without external supervision. The probed QPM distinguishes them using their evidently different eye area.

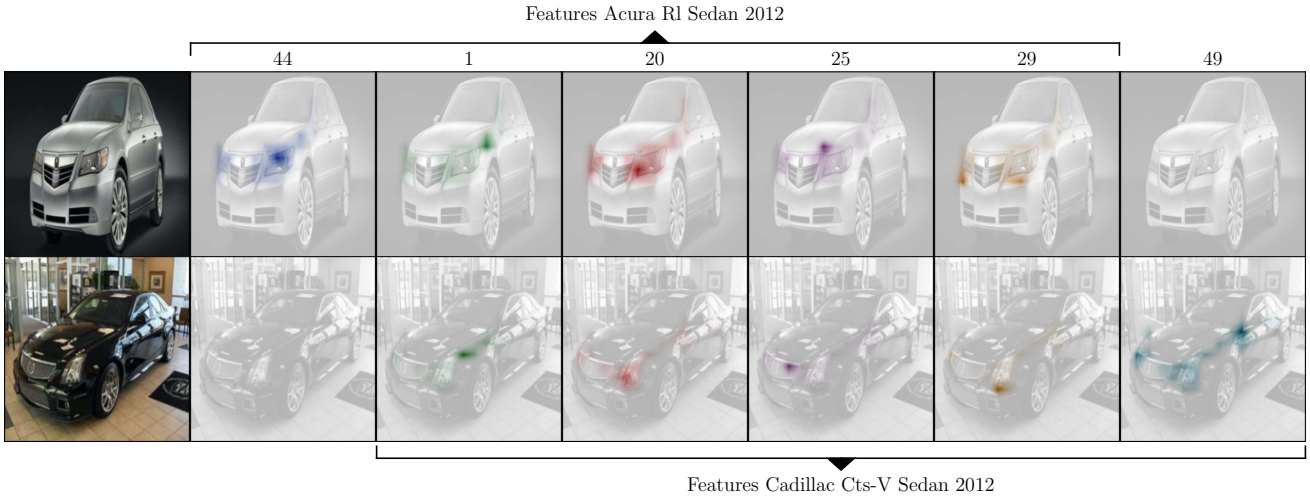


Figure 16. Faithful global interpretability on Stanford Cars: DINO-QPM autonomously discovers the 5 diverse, generalisable features for each class used to represent the Acura Rl Sedan 2012 and Cadillac Cts-V Sedan 2012, completely without external supervision. The probed QPM distinguishes them using their evidently different headlights.

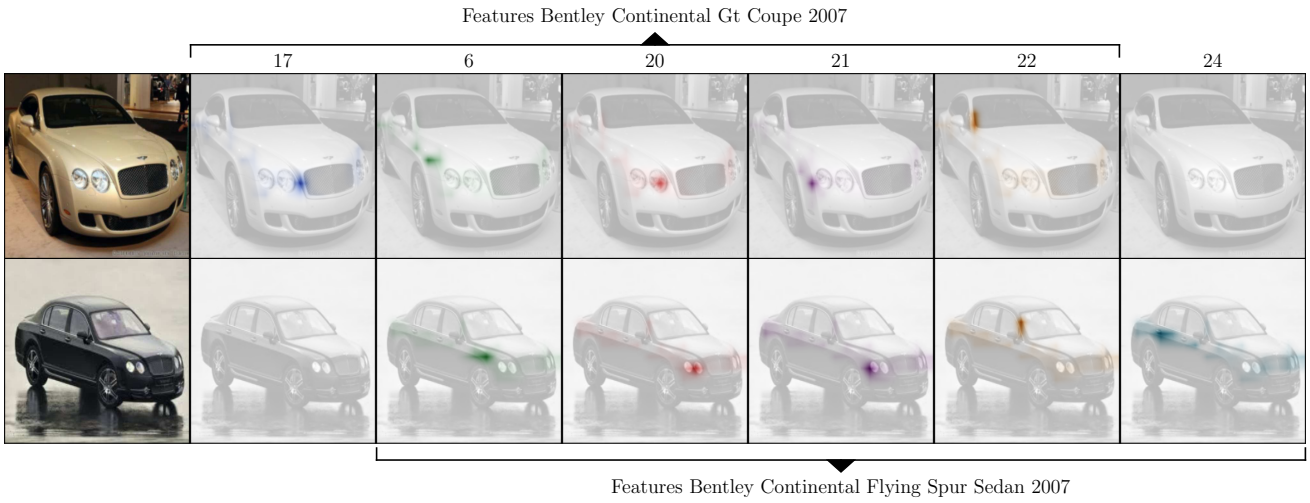


Figure 17. Faithful global interpretability on Stanford Cars: DINO-QPM autonomously discovers the 5 diverse, generalisable features for each class used to represent the Bentley Continental Gt Coupe 2007 and Bentley Continental Flying Spur Sedan 2007, completely without external supervision. The probed QPM distinguishes them using their evidently different door configurations. The probed QPM distinguishes them using their evidently different door configurations. As the most prominent distinguishing factor is the number of doors, the model's non-overlapping features for the Flying Spur (Sedan) specifically highlight the rear doors and rear door handles, which the GT (Coupe) lacks

12.2. Dense F^{froz} Failure vs. DINO-QPM Correct Classification

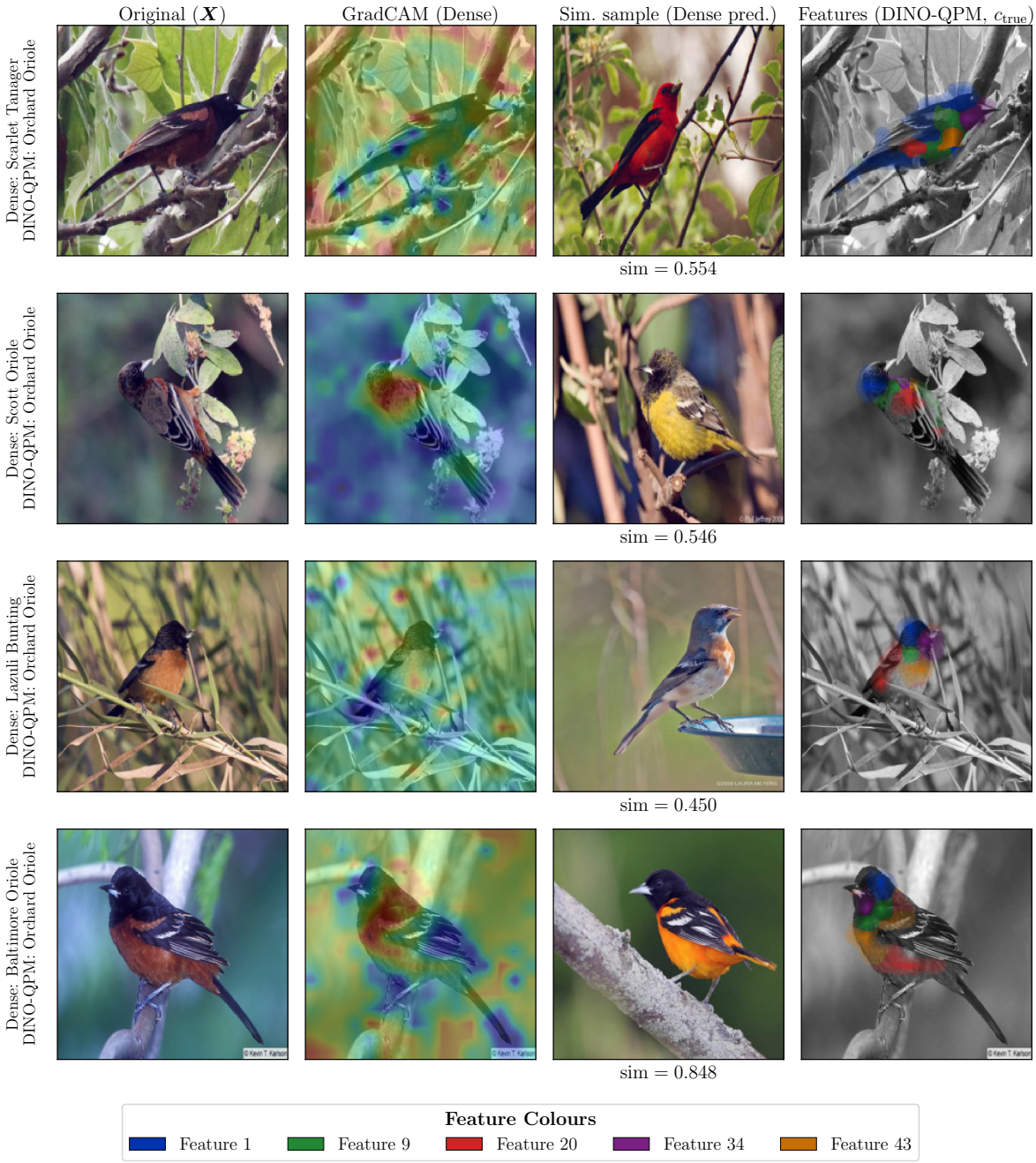


Figure 18. Comparison on the *Orchard Oriole* (CUB-2011). We show test samples where our DINO-QPM correctly classifies the image while the dense baseline fails. Columns from left to right: original image ($\boldsymbol{X}$), GradCAM activation map of the dense model, the most similar training sample from the dense-predicted class alongside its cosine similarity score (sim $= \max_{s \in \mathcal{S}_{c_{\text{pred}}}} \text{CosSim}(\boldsymbol{f}^{\text{froz}}_{\text{CLS}}(\boldsymbol{X}), \boldsymbol{f}^{\text{froz}}_{\text{CLS}}(s))$), and the colour-coded local explanation of DINO-QPM for the true class. Row labels indicate the dense prediction (top) and the DINO-QPM prediction (bottom). The dense model consistently confuses Orchard Orioles with visually similar species by attending to non-discriminative regions such as foliage and branches. In contrast, DINO-QPM correctly localises diverse features strictly on the bird's body, enabling accurate classification despite the visual similarity to other species.

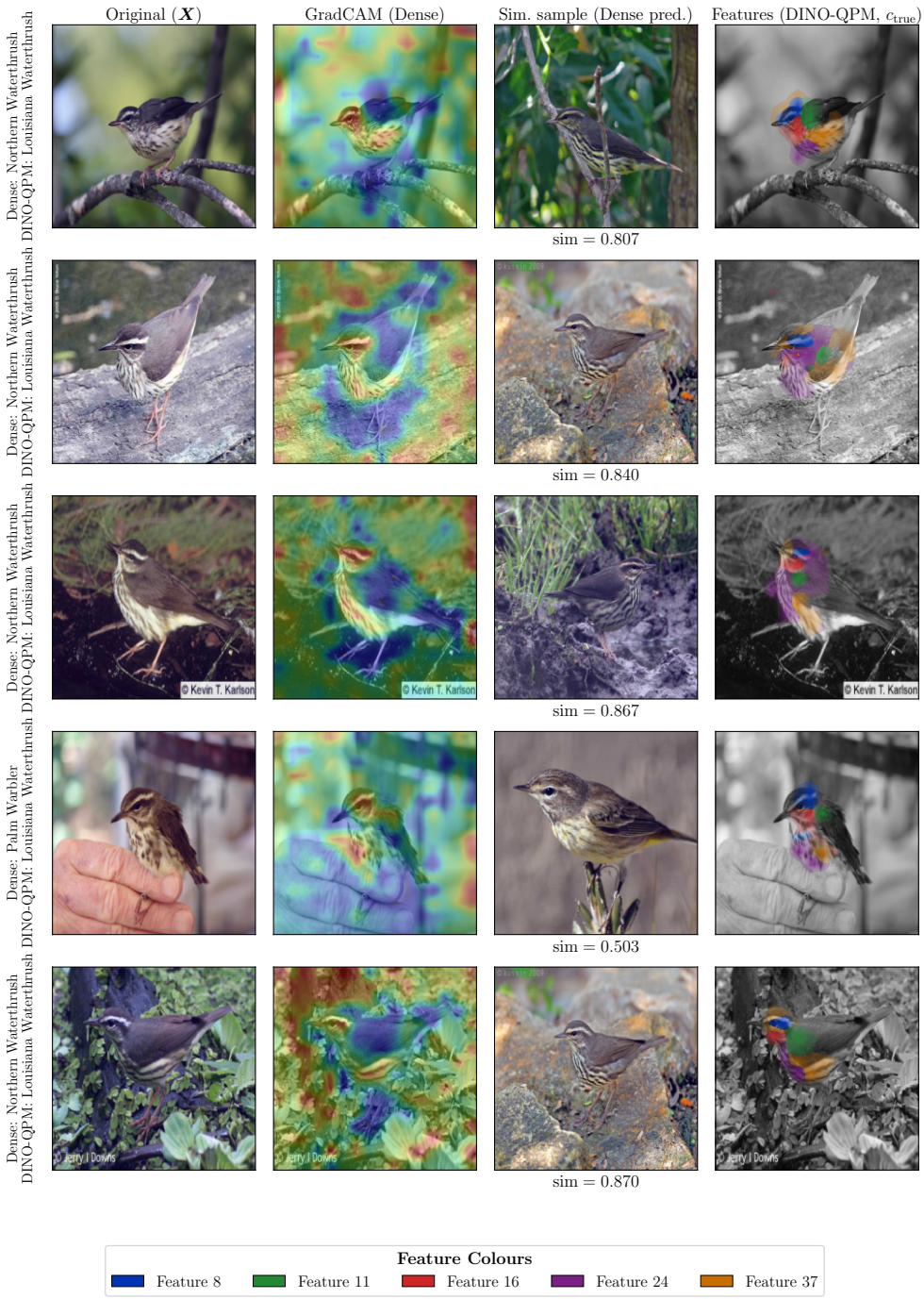


Figure 19. Comparison on the *Louisiana Waterthrush* (CUB-2011). We show test samples where our DINO-QPM correctly classifies the image while the dense baseline fails. Columns from left to right: original image ($\boldsymbol{X}$), GradCAM activation map of the dense model, the most similar training sample from the dense-predicted class alongside its cosine similarity score ($\text{sim} = \max_{s \in \mathcal{S}_{c_{\text{pred}}}} \text{CosSim}(\boldsymbol{f}^{\text{froz}}_{\text{CLS}}(\boldsymbol{X}),\ \boldsymbol{f}^{\text{froz}}_{\text{CLS}}(s))$), and the colour-coded local explanation of DINO-QPM for the true class. Row labels indicate the dense prediction (top) and the DINO-QPM prediction (bottom). The dense model consistently confuses the Louisiana Waterthrush with extremely similar species (e.g., Northern Waterthrush or Palm Warbler), often attending to less discriminative regions. In contrast, DINO-QPM correctly localises diverse features strictly on the bird's body, enabling accurate classification despite the extreme visual overlap between these species.

12.3. Dense F^{froz} vs. DINO-QPM Correct Classification

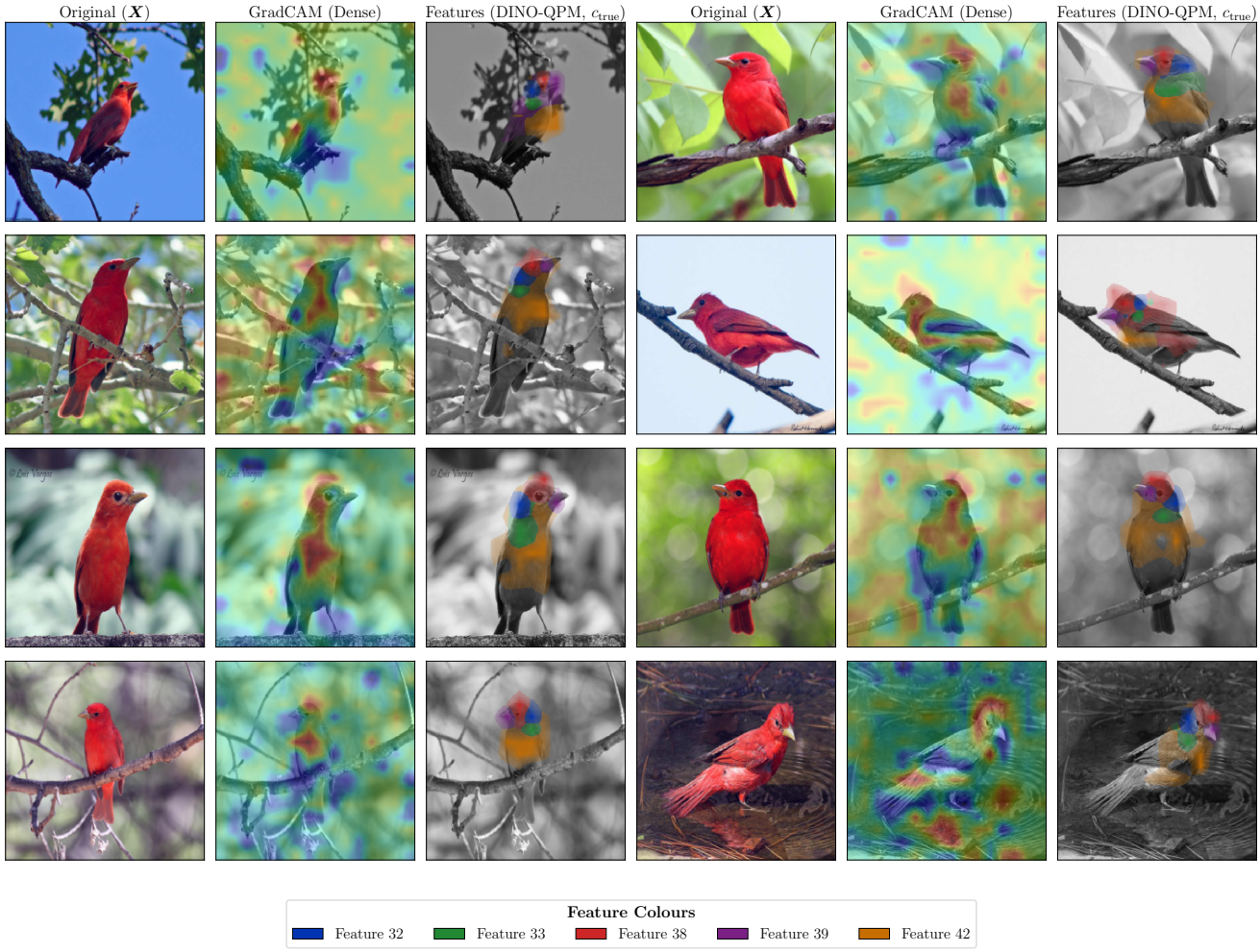


Figure 20. Comparison on the *Summer Tanager* (CUB-2011). We compare the dense baseline and DINO-QPM on eight test images, correctly classified by both models. Each sample is shown as a triplet: the original image ($\boldsymbol{X}$), the GradCAM attribution of the dense model, and the local explanation of DINO-QPM for the true class c_{true}. The GradCAM attributions of the dense model frequently spread across the background or miss the bird entirely (e.g., samples on the right), demonstrating inconsistent localisation despite correct predictions. In contrast, DINO-QPM's local explanation consistently focuses on the bird, decomposing it into interpretable parts such as the red body plumage (Feature 42), the upper head (Feature 32), and the eye region (Feature 38). This illustrates that DINO-QPM not only localises more reliably but also provides explicit part-level evidence for its decisions, whereas the dense model relies on diffuse, poorly grounded visual cues.

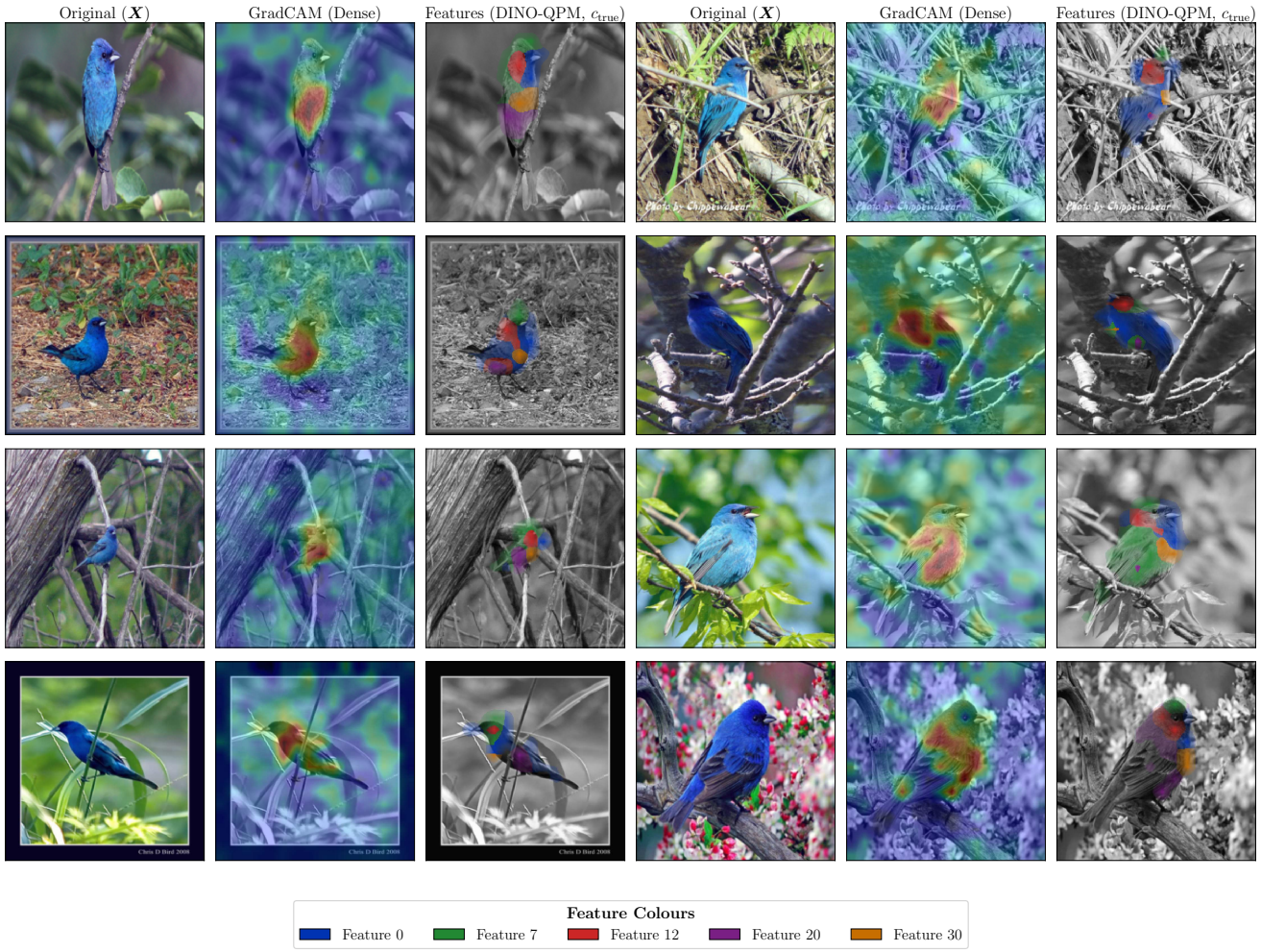


Figure 21. Indigo Bunting samples from the CUB-2011 test set. We compare the dense baseline and DINO-QPM on eight test images, correctly classified by both models. Each sample is shown as a triplet: the original image ($\boldsymbol{X}$), the GradCAM attribution of the dense model, and the local explanation of DINO-QPM for the true class c_{true}. While both models localise the bird reliably across varying poses and backgrounds, the key difference lies in *what* each model communicates. The dense model focuses on a single discriminative region, resulting in a non-diverse localisation. In contrast, DINO-QPM decomposes its focus into semantically distinct parts—e.g. the belly (Feature 20), the mantle and back (Feature 7), and the head region (Feature 12)—offering a richer, more diverse, part-level explanation of *why* the prediction is made.

12.4. DINO-QPM Failure Analysis

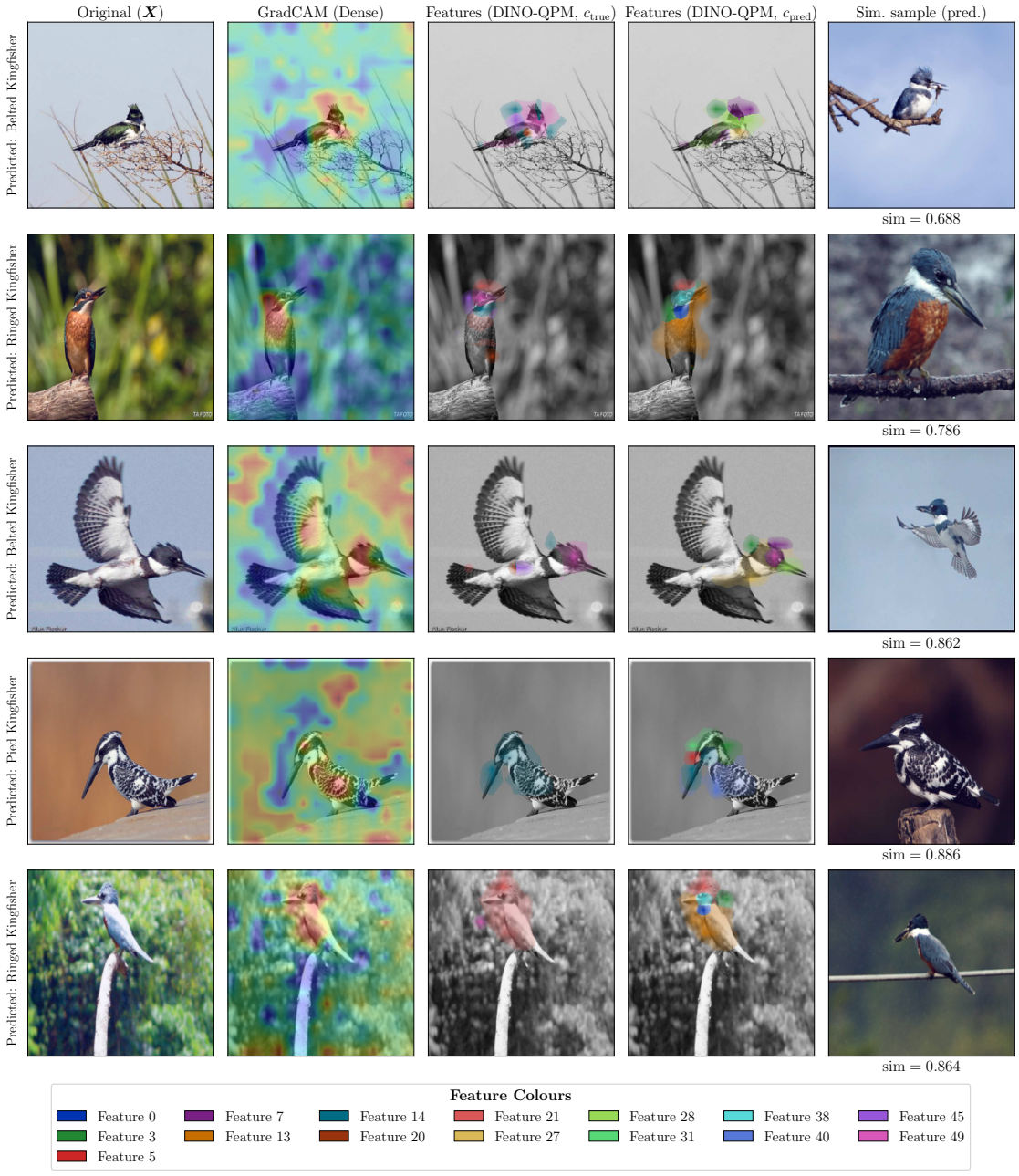


Figure 22. Failure analysis on the *Green Kingfisher* (CUB-2011). Comparison of test samples misclassified by both models. Columns (left to right): original image ($\boldsymbol{X}$); dense model GradCAM; DINO-QPM local explanations for both the true and predicted classes; and the nearest training sample from the predicted class with its cosine similarity ($\text{sim} = \max_{s \in \mathcal{S}_{c_{\text{pred}}}} \text{CosSim}(\boldsymbol{f}^{\text{froz}}_{\text{CLS}}(\boldsymbol{X}), \boldsymbol{f}^{\text{froz}}_{\text{CLS}}(s))$). Although both models struggle to distinguish these highly fine-grained kingfisher species, their failure modes differ significantly. The dense model provides no meaningful insight into its errors, whereas DINO-QPM transparently communicates the source of its confusion through faithful, part-level local explanations. Notably, some of these misclassifications might be due to incorrect ground-truth labels [63]. For example, the fourth sample appears to be incorrectly annotated, demonstrating how our concept-based explanations can assist in auditing dataset quality.

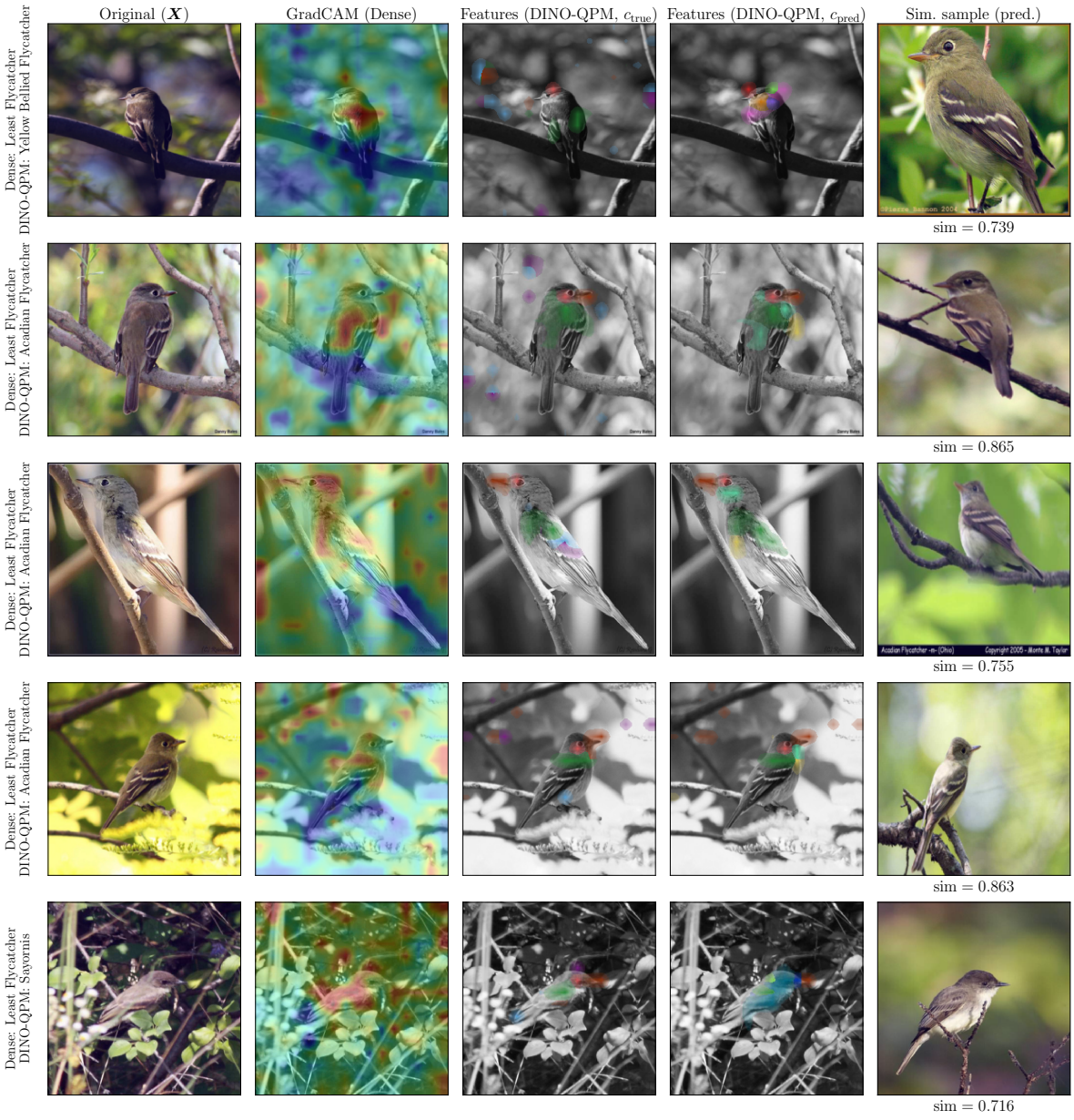


Figure 23. Failure analysis on the *Least Flycatcher* (CUB-2011). We show test samples where the dense baseline classifies correctly but DINO-QPM does not. Columns from left to right: original image ($\boldsymbol{X}$), GradCAM attribution of the dense model, DINO-QPM local explanations for the true and predicted classes, and the most similar training sample from the predicted class with its cosine similarity score ($\text{sim} = \max_{s \in \mathcal{S}_{c_{\text{pred}}}} \text{CosSim}(\boldsymbol{f}^{\text{froz}}_{\text{CLS}}(\boldsymbol{X}),\ \boldsymbol{f}^{\text{froz}}_{\text{CLS}}(s))$). While we previously demonstrated that DINO-QPM successfully overcomes poorly localised dense representations, this strict feature decomposition can occasionally induce errors. In this failure case, although the dense baseline correctly predicts the target class using ungrounded cues, DINO-QPM's refusal to exploit these uninterpretable shortcuts leads to confusion among visually similar flycatcher and *Sayornis* species.

Method	Local. Features	Accuracy ↑		Faithful. ↑	SID@5 ↑		Class-Indep. ↑		Contrast. ↑	
		CUB	CARS		CUB	CARS	CUB	CARS	CUB	CARS
DINOv2 f^{froz}_{CLS} Linear Probe	✗	<u>87.9</u> ± 0.1	91.7 ± 0.1	42.6 ± 0.2	50.9 ± 0.2	51.5 ± 0.1	**99.2** ± 0.0	**99.1** ± 0.0	59.2 ± 0.0	60.9 ± 0.0
Dense F^{froz}	✓	78.1 ± 0.3	92.9 ± 0.1	32.7 ± 0.2	**91.8** ± 0.7	**93.1** ± 0.1	<u>98.8</u> ± 0.0	<u>98.7</u> ± 0.0	84.5 ± 0.3	82.8 ± 0.1
Resnet50 Baseline [44]	✓	83.9 ± 0.4	92.5 ± 0.2	60.7 ± 0.2	57.1 ± 0.4	51.5 ± 0.2	98.0 ± 0.0	97.9 ± 0.0	74.6 ± 0.1	75.1 ± 0.1
Resnet50 QPM [44]	✓	82.9	92.1 ± 0.2	82.9	89.6	88.2 ± 0.5	96.8	97.8 ± 0.0	93.6	<u>97.1</u> ± 0.2
DINO-SLDD	✓	84.6 ± 0.4	92.9 ± 0.1	78.0 ± 0.9	88.7 ± 0.3	90.9 ± 0.8	94.4 ± 0.1	93.9 ± 0.2	93.0 ± 0.3	94.9 ± 0.5
DINO-QSENN	✓	85.4 ± 0.5	<u>93.3</u> ± 0.1	86.0 ± 0.4	<u>91.5</u> ± 0.5	<u>92.6</u> ± 0.4	93.6 ± 0.4	94.0 ± 0.1	<u>94.4</u> ± 0.3	94.9 ± 0.1
DINO-QPM (Ours)	✓	**88.3** ± 0.3	**94.0** ± 0.2	**95.0** ± 0.6	90.1 ± 0.0	91.7 ± 0.2	93.7 ± 0.1	93.7 ± 0.1	**100.0** ± 0.0	**100.0** ± 0.0
DINO-QPM Compact (Ours)	✓	**88.3** ± 0.3	**94.0** ± 0.1	<u>94.4</u> ± 0.6	–	–	93.8 ± 0.1	93.6 ± 0.1	**100.0** ± 0.0	**100.0** ± 0.0

Table 5. Comparison with state-of-the-art interpretable models. We report Accuracy, Faithfulness, SID@5, Class-Independence, and Contrastiveness (all metrics in %). Features of a model are localised if they have a direct connection to the feature vector used for classification. The Faithfulness metric is evaluated only on CUB-2011 due to the availability of segmentation masks. Dense F^{froz} is the dense model of DINO-QPM and DINOv2 f^{froz}_{CLS} Linear Probe is a linear probe [11] trained on top of the frozen `CLS` representation. For DINO-SLDD and DINO-QSENN, we employ a pipeline closely resembling the one described in Sec. 4, with the exception of the feature selection mechanisms, which follow Norrenbrock et al. [41] and Norrenbrock et al. [42], respectively. For Resnet50 QPM [44] on CUB-2011 we cannot provide standard deviation, as we use the original model provided by authors (https://github.com/ThomasNorr/Qpm).