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.

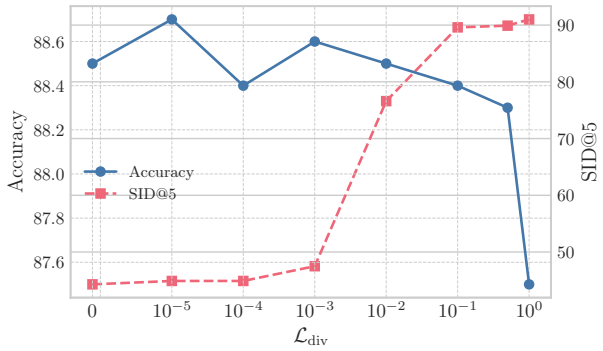

Figure 11. Accuracy and Feature Diversity (SID@5) on CUB-2011 across variations of the $\mathcal{L}_{\text{div}}$ weight during dense and finetuning training.

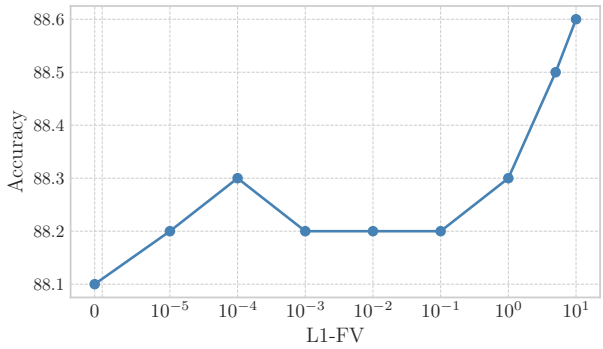

Figure 12. Impact of the $\mathcal{L}_{\text{L1-FV}}$ on CUB-2011 accuracy during finetuning.

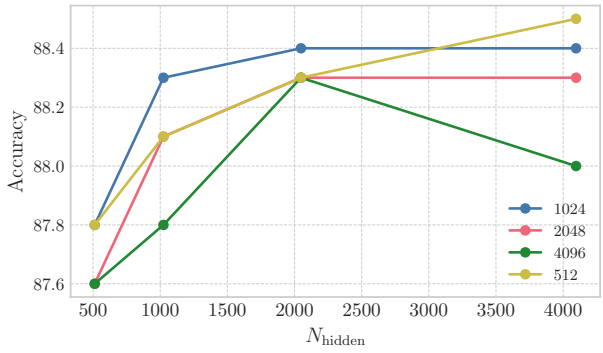

Figure 13. Mean finetuning accuracy on CUB-2011 for various numbers of features N_f across a range of hidden layer neurons N_{hidden} in the MLP. We observe small accuracy gains up until $N_{\text{hidden}} = 2048$ regardless of the number of features N_f.

12. Visualisations

12.1. Class Comparisons

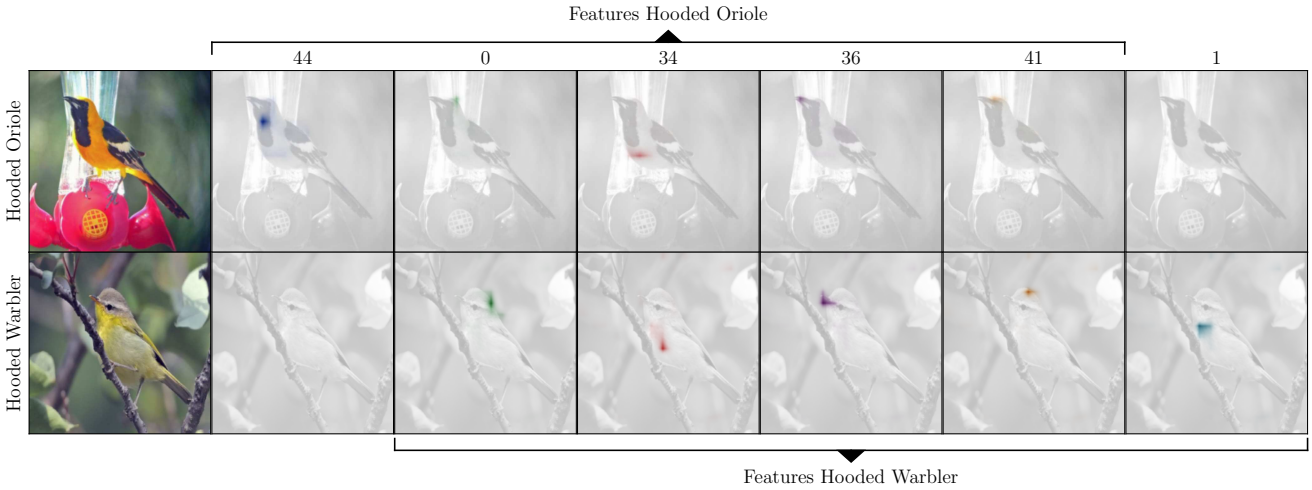

Figure 14. Faithful global interpretability on CUB-2011: DINO-QPM autonomously discovers the 5 diverse, generalisable features for each class used to represent the Hooded Oriole and Hooded Warbler, completely without external supervision. The probed QPM distinguishes them using their evidently different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

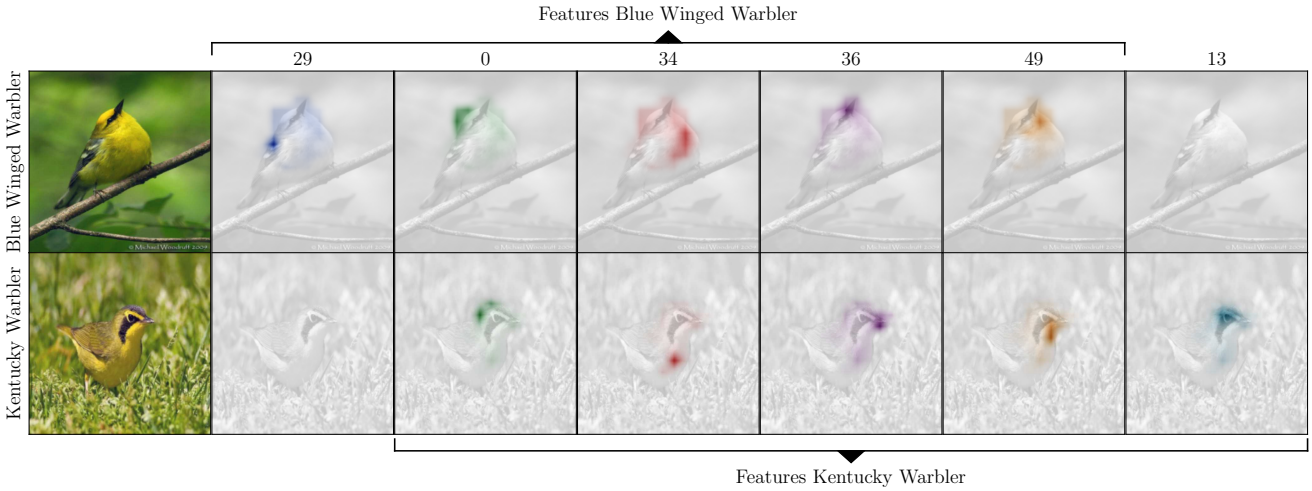

Figure 15. Faithful global interpretability on CUB-2011: DINO-QPM autonomously discovers the 5 diverse, generalisable features for each class used to represent the Blue Winged Warbler and Kentucky Warbler, completely without external supervision. The probed QPM distinguishes them using their evidently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

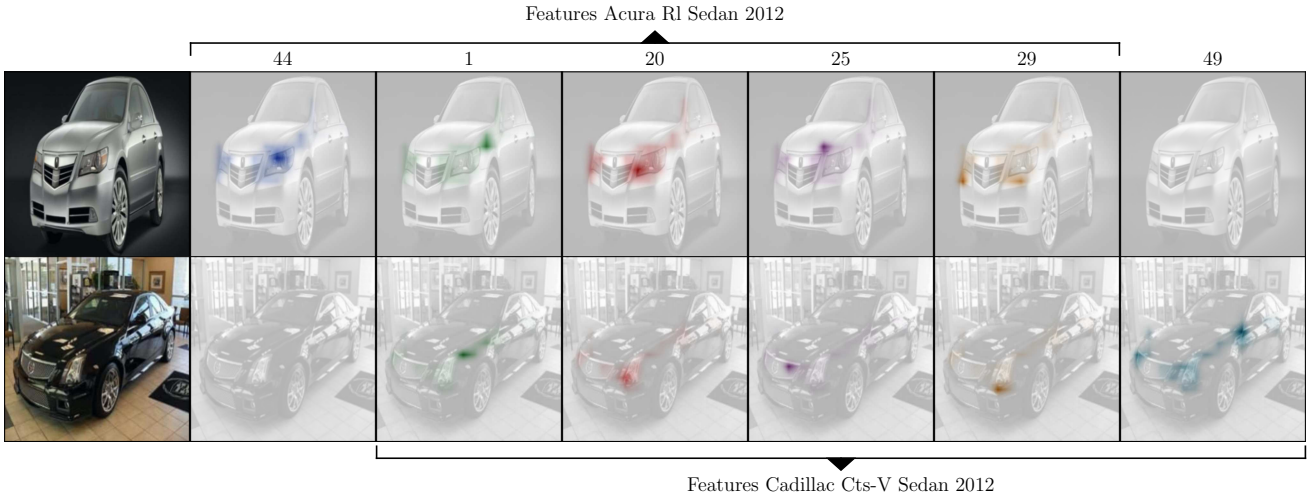

Figure 16. Faithful global interpretability on Stanford Cars: DINO-QPM autonomously discovers the 5 diverse, generalisable features for each class used to represent the Acura Rl Sedan 2012 and Cadillac Cts-V Sedan 2012, completely without external supervision. The probed QPM distinguishes them using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

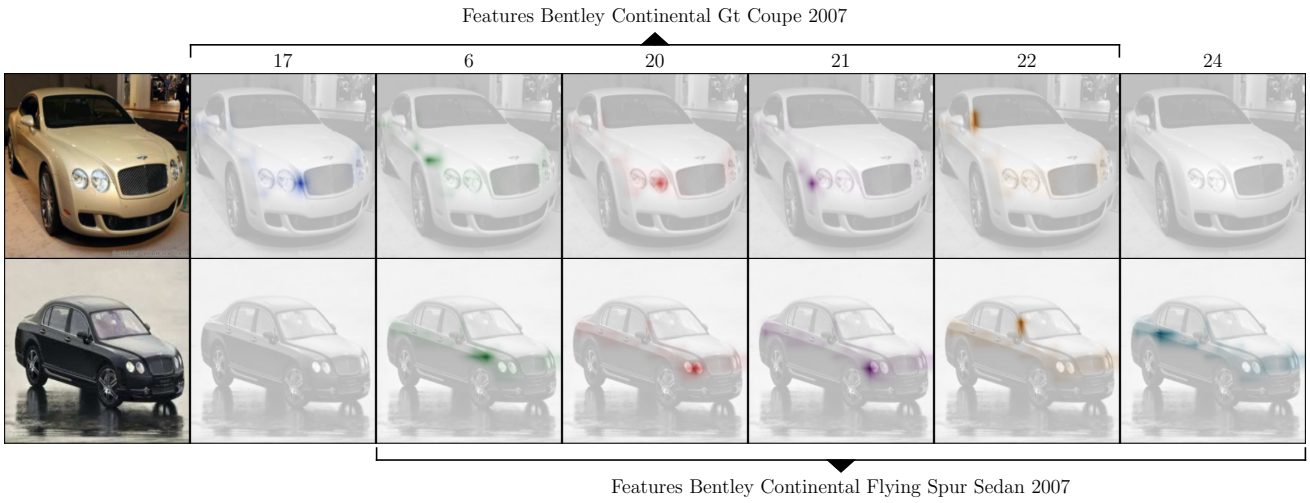

Figure 17. Faithful global interpretability on Stanford Cars: DINO-QPM autonomously discovers the 5 diverse, generalisable features for each class used to represent the Bentley Continental Gt Coupe 2007 and Bentley Continental Flying Spur Sedan 2007, completely without external supervision. The probed 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

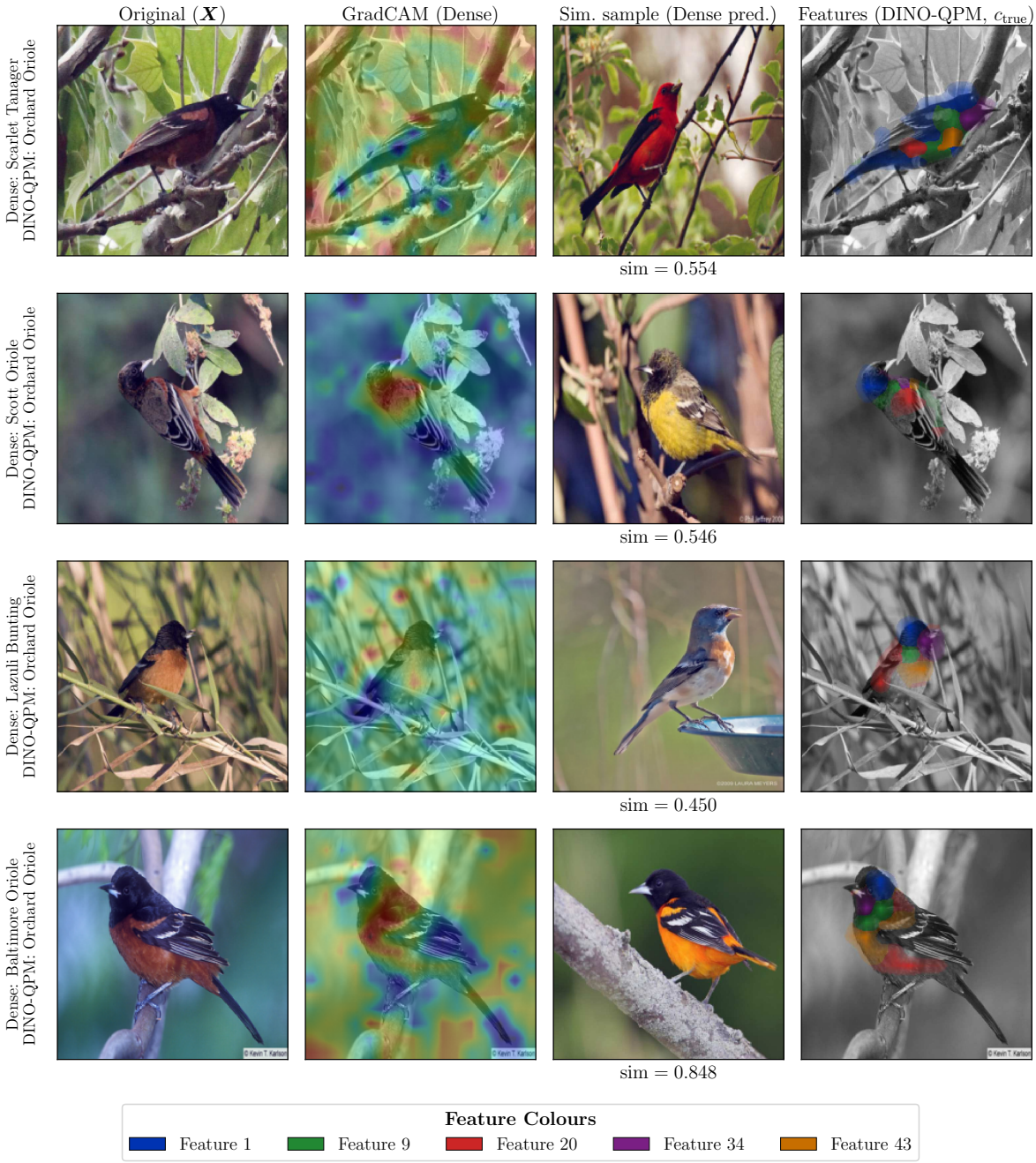

Figure 18. Comparison on the *Orchard Oriole* (CUB-2011). We show test samples where our DINO-QPM correctly classifies the image while the dense baseline fails. Columns from left to right: original image ($\boldsymbol{X}$), GradCAM activation map of the dense model, the most similar training sample from 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.

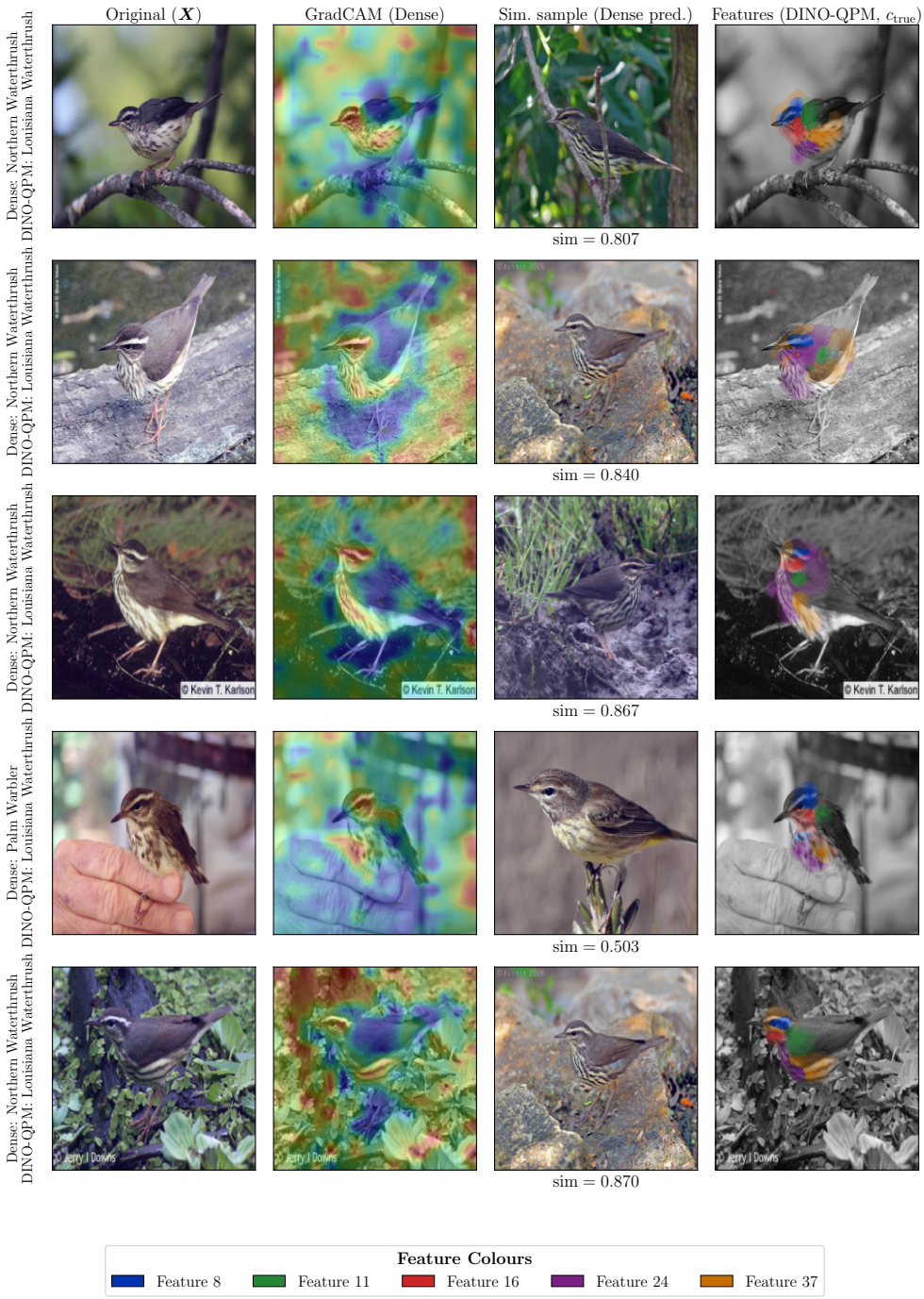

Figure 19. Comparison on the *Louisiana Waterthrush* (CUB-2011). We show test samples where our DINO-QPM correctly classifies the image while the dense baseline fails. Columns from left to right: original image ($\boldsymbol{X}$), GradCAM activation map of the dense model, the most similar training sample 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

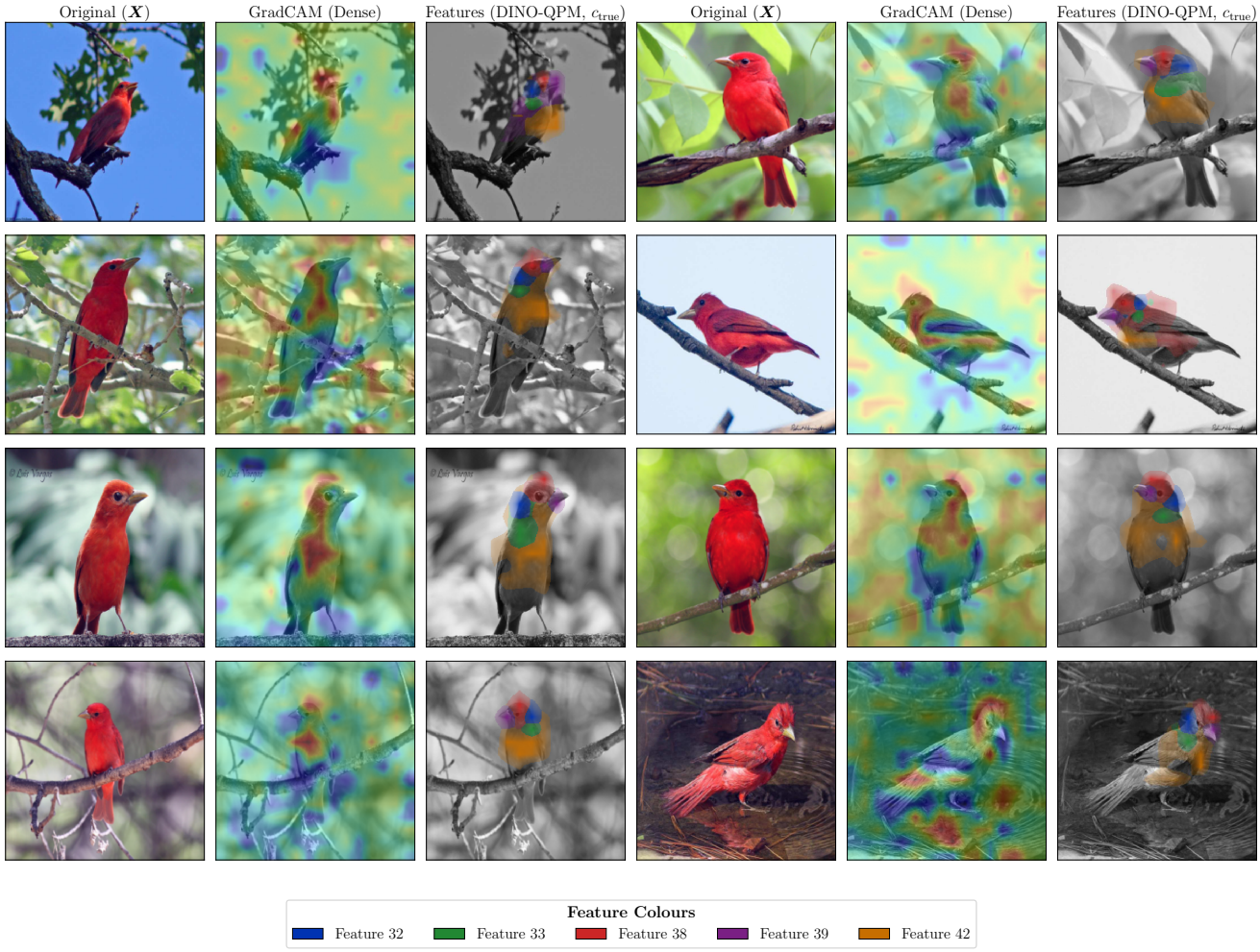

Figure 20. Comparison on the *Summer Tanager* (CUB-2011). We compare the dense baseline and DINO-QPM on eight test images, correctly classified by both models. Each sample is shown as a triplet: the original image ($\boldsymbol{X}$), the GradCAM attribution of the dense model, and the local explanation 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.

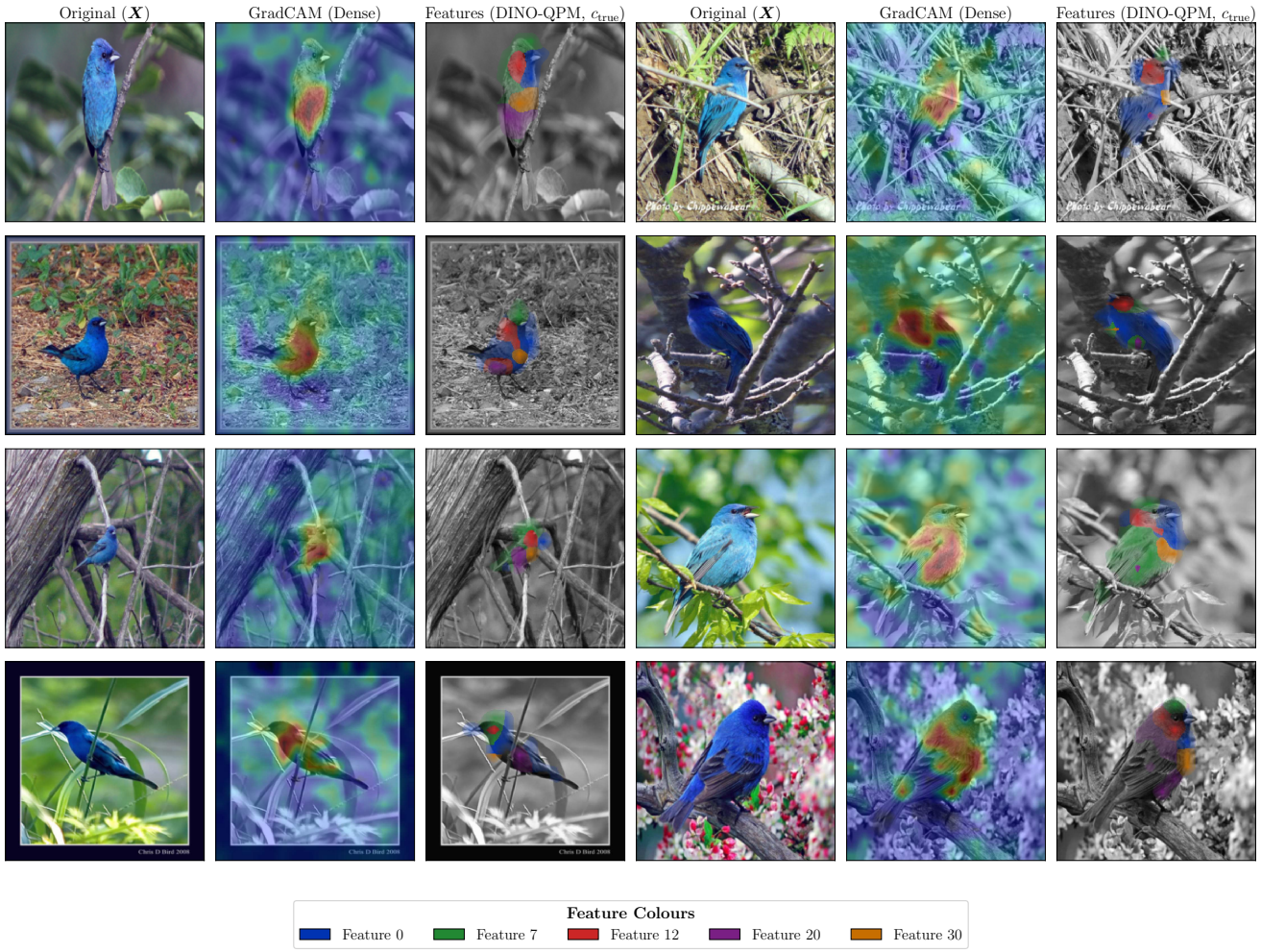

Figure 21. Indigo Bunting samples from the CUB-2011 test set. We compare the dense baseline and DINO-QPM on eight test images, correctly classified by both models. Each sample is shown as a triplet: the original image ($\boldsymbol{X}$), the GradCAM attribution of the dense model, and the local explanation 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

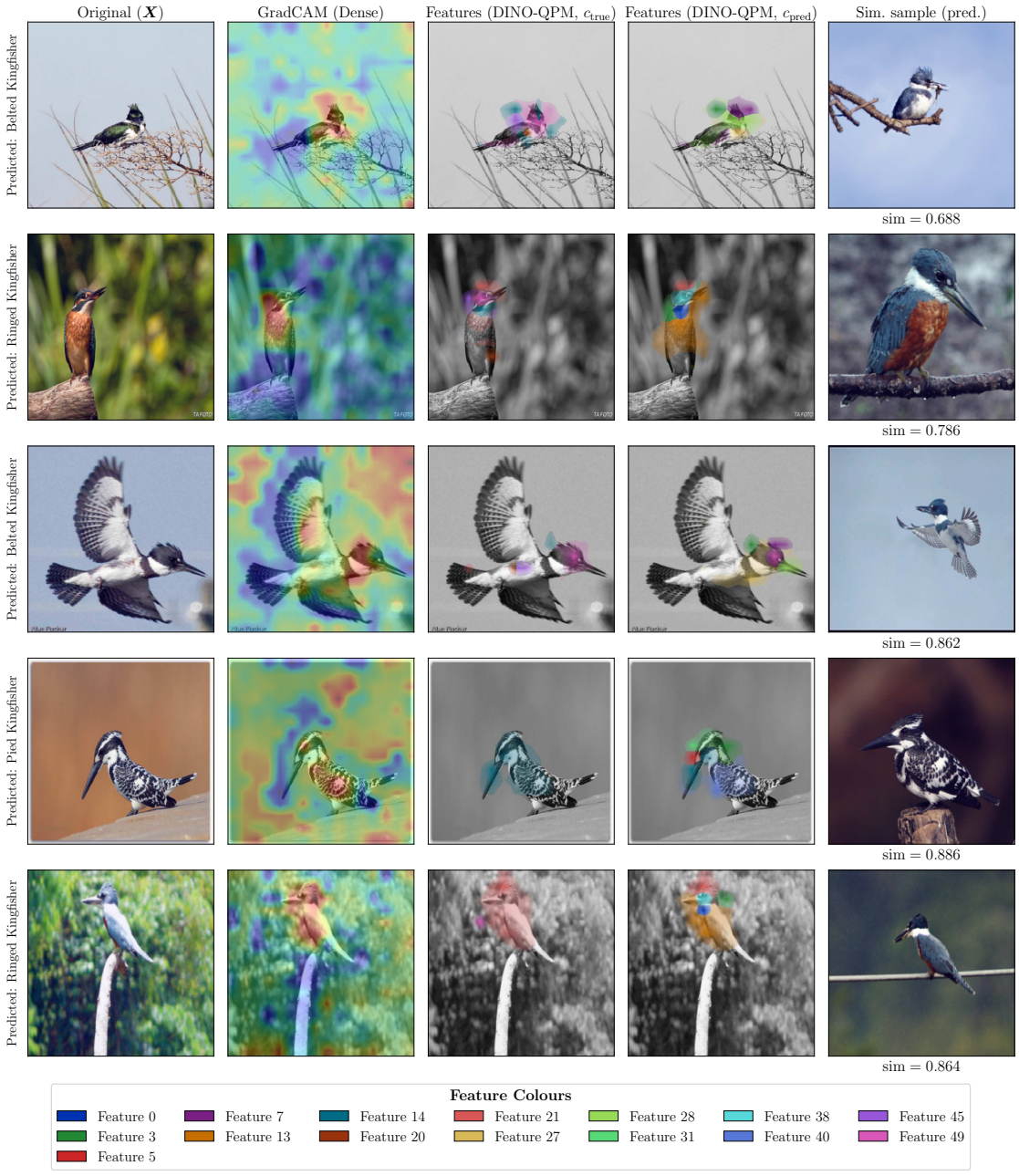

Figure 22. Failure analysis on the *Green Kingfisher* (CUB-2011). Comparison of test samples misclassified by both models. Columns (left to right): original image ($\boldsymbol{X}$); dense model GradCAM; DINO-QPM local explanations for both the true and predicted classes; and the nearest training sample 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.

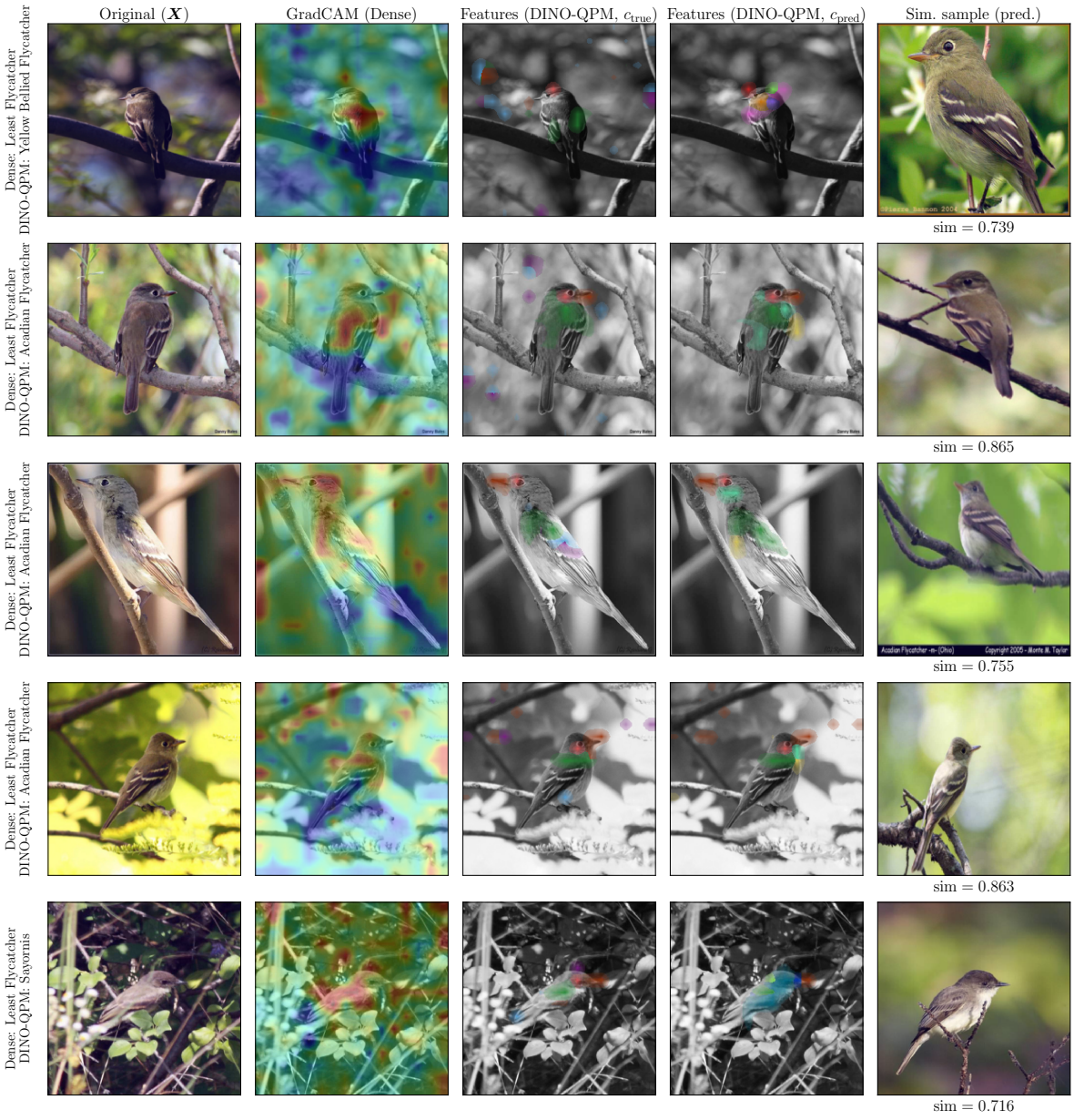

Figure 23. Failure analysis on the *Least Flycatcher* (CUB-2011). We show test samples where the dense baseline classifies correctly but DINO-QPM does not. Columns from left to right: original image ($\boldsymbol{X}$), GradCAM attribution of the dense model, DINO-QPM local explanations for the true and 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).