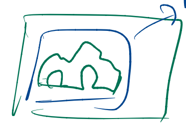
TAKE 5: INTERPRETABLE IMAGE CLASSIFICATION WITH A HANDFUL OF FEATURES

Anonymous authors
Paper under double-blind review

Fine-grain Image Cls



Birds, toyota

ABSTRACT

Deep Neural Networks use thousands of mostly incomprehensible features to identify a single class, a decision no human can follow. We propose an interpretable sparse and low dimensional final decision layer in a deep neural network with measurable aspects of interpretability and demonstrate it on fine-grained image classification. We argue that a human can only understand the decision of a machine learning model, if the input features are interpretable and only very few of them are used for a single decision. For that matter, the final layer has to be sparse and - to make interpreting the features feasible - low dimensional. We call a model with a Sparse Low-Dimensional Decision "*SLDD-Model*". We show that a *SLDD-Model* is easier to interpret locally and globally than a dense high-dimensional decision layer while being able to maintain competitive accuracy. Additionally, we propose a loss function that improves a model's feature diversity and accuracy. Our interpretable *SLDD-Model* only uses 5 out of just 50 features per class, while maintaining 97 % to 100 % of the accuracy on four common benchmark datasets compared to the baseline model with 2048 features.

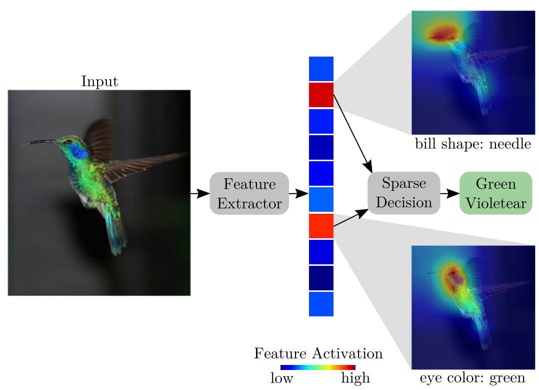


Figure 1: Local explanation by our *SLDD-Model*: The two features used for the predicted class, emerged without additional supervision, are aligned with human interpretable attributes and localized (described in App. D) adequately.

1 INTRODUCTION

Understanding the decision of a deep learning model is becoming more and more important. Especially for safety-critical applications such as the medical domain or autonomous driving, it is often either legally (Bibal et al., 2021) or by the practitioners required to be able to trust the decision and evaluate its reasoning (Molnar, 2020). Due to the high dimensionality of images, most previous work on interpretable models for computer vision combines the deep features computed by deep neural network with a method that is considered interpretable, such as a prototype based decision tree (Nauta et al., 2021). While approaches for measuring the interpretability without humans exist for conventional machine learning algorithms (Islam et al., 2020), they are missing for methods including deep neural networks. In this work, we propose a novel sparse and low-dimensional *SLDD-Model*

which offers measurable aspects of interpretability. The key aspect is a heavily reduced number of features, out of which only very few are considered per class. Humans can only consider 7 ± 2 aspects at once (Miller, 1956) and could therefore follow a decision that uses that many features. To be intelligible for all humans, we aim for an average of 5 features per class.
Having a reduced number of features makes it feasible to investigate every single feature and understand its meaning: We are able to align several of the learned features with human concepts post-hoc. The combination of reduced features and sparsity therefore increases both global *How does the model behave?* and local interpretability *Why did the model make this decision?*, demonstrated in Figure 1. Our proposed method generates the *SLDD-Model* by utilizing *glm-saga* (*Wong et al., 2021*) to compute a sparse linear classifier for precomputed, selected features. We apply feature selection instead of a transformation to reduce the computational load and preserve the original semantics of the features, which can improve interpretability (Tao et al., 2015), especially if a more interpretable model like *B-cos Networks* (Böhle et al., 2022) is used. Additionally, we propose a novel loss function for more diverse features, which is especially relevant when one class depends on very few features, since using more redundant features limits the total information available for the decision.

Our main **contributions** are as follows:

- We present a pipeline that ensures a globally and locally interpretable model that identifies a single class with just few, *e.g.* 5, features of its low-dimensional representation. We call the resulting model *SLDD-Model*.
- Our novel feature diversity loss ensures diverse features. This increases the accuracy for the extremely sparse case.
- We demonstrate the competitive performance of our proposed method on four common benchmark datasets in the domain of fine-grained image classification as well as ImageNet-1K (Russakovsky et al., 2015), and show that several learned features for algorithmic decision-making can be directly connected to attributes humans use.
- Code will be published upon acceptance.

2 Related Work

Fine-Grained Image Classification Fine-grained image classification describes the problem of differentiating similar classes from one another. It is more challenging compared to conventional image recognition tasks (Lin et al., 2015) since the differences between classes are much smaller. To tackle this difficulty, several adaptions to the common image classification approach have been applied. They usually involve learning more discriminative features by adding a term to the loss function (Chang et al., 2020; Liang et al., 2020; Zheng et al., 2020), introducing hierarchy to the architecture (Chou et al., 2022) or using expensive expert knowledge (Chen et al., 2018; Chang et al., 2021). Chang et al. (2020) divide the features into groups, s. t. every group is assigned to exactly one class. While training, an additional loss increases the activations of features for samples of their assigned class and reduces the overlap of feature maps in each group. Liang et al. (2020) tried to create class-specific filters by inducing sparsity in the features. Both (Chang et al., 2020) and (Liang et al., 2020) optimize for class-specific filters, which are neither suitable for the low-dimensional case when the number of classes exceeds the number of features nor interpretable, since it is unclear if the feature is already detecting the class rather than a lower level feature. The *Feature Redundancy Loss* (Zheng et al., 2020) (FRL) enforces the K most used features to be localized differently by reducing the normalized inner product between their feature maps. This adds a hyperparameter and does not optimize all features at once.

Interpretable Machine Learning Interpretable machine learning is a broad term and can refer to both models that are interpretable by design, and post-hoc methods that try to understand what the model has learned. Furthermore, interpretability can be classified as the interpretability of a single instance (local) or the entire model (global) (Molnar, 2020).
In this work, we present methods making models more interpretable by design but also utilize post-hoc methods to offer local and global interpretability. Common local post-hoc methods are saliency maps like Grad-CAM (Selvaraju et al., 2017) that aim to show what part of the input image is relevant for the prediction. While they can be helpful, they have to be cautiously interpreted, as they do not show many desired properties one would expect from an explanation like shift invariance (Kindermans et al.,

2019) or only producing reasonable explanations, when the model is working as intended (Adebayo et al., 2018). Another way of obtaining saliency maps is based on masking the input image and measuring the impact on the output (Zeiler & Fergus, 2014; Fong & Vedaldi, 2017; Jain et al., 2022). As a global post-hoc method, *Elude* (Ramaswamy et al., 2022) generates an explanation for a model by mimicking its behavior with a sparse model. This model uses additional attributes and main directions of the remaining feature space of the model as input. Instead of explaining a model, we directly train the interpretable model in this work. Another line of research tries to align learned representations with human understandable concepts from an additional labeled dataset (Kim et al., 2018; Bau et al., 2017; McGrath et al., 2021), increasing the global interpretability of the model. To our best knowledge, measuring the interpretability of a deep neural network is an open task, as previous work focuses on measuring the quality of explanations of black boxes (Rokade & Alluri, 2021) or on conventional machine learning algorithms (Islam et al., 2020), where increased interpretability is measured when model complexity is reduced, *e.g.* via the number of operations (Yang et al., 2017; Friedler et al., 2019; Rüping et al., 2006) or number of features (Rüping et al., 2006). The sparsity and low-dimensionality of our proposed *SLDD-Model* is motivated by these findings. Due to the limitations of post-hoc methods in explaining a deep neural network, models that are more interpretable by design are becoming more relevant. Nauta et al. (2021) and Yang et al. (2022) used a deep feature extractor in combination with a prototype based decision tree(s) to achieve state-of-the-art performance on fine-grained image classification while increasing interpretability. However, decision trees struggle to model linear dependencies, and it is hard to globally interpret an ensemble of deep decision trees. Kim et al. (2021) and Hoffmann et al. (2021) additionally indicate a gap between the perceived similarities of humans and prototype based models. *Concept bottleneck models* (*CBM*) (Koh et al., 2020) first predict concepts annotated in the dataset and then use a simple model to predict the target class from the concepts. *CBM-AUC* (Sawada & Nakamura, 2022) extended *CBM* by allowing unsupervised concepts to influence the decision. *PCBM* (Yuksekgonul et al., 2022) created a post-hoc *CBM* using *TCAV* (Kim et al., 2018) to compress high-dimensional learned features into a concept bottleneck. Margeloiu et al. (2021) and *Elude* (Ramaswamy et al., 2022) both suggest that training the *CBM* end-to-end leads to the encoding of additional information next to the concepts, which reduces the interpretability. In contrast to *CBM*, our proposed method does not require additional labels for training and leads to a very sparse decision process. While their features are generally more aligned with the given concepts, they also need to be analyzed thoroughly. Wong et al. (2021) developed *glm-saga*, a method to efficiently fit a heavily regularized sparse layer to the computed features of a backbone feature extractor, and showed that human understanding is more aligned with the decision process of the sparse model. Additionally, *glm-saga* reaches levels of sparsity that network wide sparsity methods do not obtain in the final layer with competitive accuracy (Gale et al., 2019). Since their aim is understanding the neural network, they fit the sparse layer to fixed features and do not finetune the features to the sparse layer, which requires different optimization strategies than dense networks (Tessera et al., 2021). In this work, we utilize *glm-saga* to create an interpretable model with competitive accuracy.

3 Method

3.1 Problem Formulation

We apply the proposed *SLDD-Model* to the domain of fine-grained image classification. We consider the problem of classifying an image $\boldsymbol{I} \in \mathbb{R}^{3\times w\times h}$ of width w and height h into one class $c \in \{c_1, c_2, \dots, c_{n_c}\}$ using a trainable deep neural network Φ. This neural network extracts the feature maps $\boldsymbol{M} \in \mathbb{R}^{n_f \times w_M \times h_M}$ and aggregates them into the feature vector $\boldsymbol{f} \in \mathbb{R}^{n_f}$. Then it applies the trainable neural network C to obtain the final output $\boldsymbol{y} \in \mathbb{R}^{n_c}$ as $\boldsymbol{y} = C(\boldsymbol{f})$.

3.2 *SLDD-Model*

$f \in n_f \Rightarrow f \in n_f^* \Rightarrow y = wf + b$

We propose a flexible, generally applicable method for generating a locally and globally interpretable model with no need for additional labels and an adjustable tradeoff between interpretability and accuracy. We make the decision process more interpretable by only using n_f^* features with $n_f^* \ll n_f$ and using an interpretable classifier C. At the core of an interpretable classifier C lies a linear layer $\boldsymbol{y} = \boldsymbol{W}\boldsymbol{f} + \boldsymbol{b}$ with the weight matrix $\boldsymbol{W} \in \mathbb{R}^{n_c \times n_f^*}$ and bias $\boldsymbol{b} \in \mathbb{R}^{n_c}$. In order for it to be interpretable, $\boldsymbol{W}$ has to be very sparse, meaning the number of non-zero entries n_w has to be very

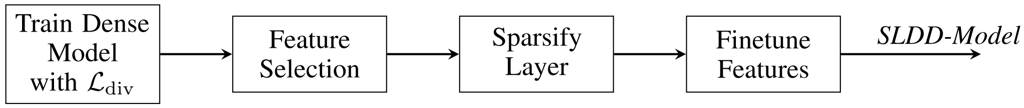


Figure 2: Overview of our proposed pipeline to construct a *SLDD-Model*

low. Miller (1956) showed that humans can handle 7 ± 2 cognitive aspects at once, which constitutes an appropriate upper bound on the average number of relevant features per class $n_{wc} = \frac{n_w}{n_c}$. In our work we focus on $n_{wc} \leq 5$.

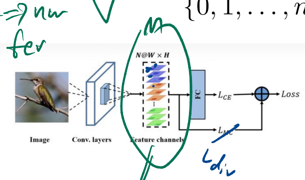
The pipeline of our approach is presented in Figure 2 and utilizes *glm-saga* for sparsification and feature selection. We first train a deep neural network with our proposed feature diversity loss $\mathcal{L}_{\text{div}}$ until convergence. Then the features $\boldsymbol{F}^{\text{train}} \in \mathbb{R}^{n_T \times n_f}$ for all n_T images in the training set are computed, which are the average pooled feature maps $\boldsymbol{M}$. Afterwards, the features are selected as described in Section 3.2.2 and *glm-saga* is used to calculate the regularization path. Finally, the solution with the desired sparsity is selected from the regularization path and the remaining layers get finetuned with the final layer set to the sparse model, s. t. the features adapt to it.

3.2.1 Feature Diversity Loss

The goal of the proposed feature diversity loss $\mathcal{L}_{\text{div}}$ is to enforce differently localized features in their feature maps, which requires a different computation of the features. The proposed loss is motivated by the *Mutual-Channel Loss* (MCL) (Chang et al., 2020) and the *Feature Redundancy loss* (FRL) (Zheng et al., 2020). In contrast to FRL, we use Cross-Channel-Max-Pooling (CCMP) (Goodfellow et al., 2013) over all weighted feature maps to optimize all features jointly and reduce the need for the hyperparameter K. MCL also uses CCMP but instead of grouping the channels into class-specific filters, we apply the diversity component to all feature maps $\boldsymbol{M}$, to aim for shared interpretable features. For notation, $w_{\hat{c}l}$ describes the entry in $\boldsymbol{W}$ that is assigned to the specific feature $l \in \{0, 1, \ldots, n_f - 1\}$ for the predicted class $\hat{c} = \arg\max(\boldsymbol{y})$ and $m_{ij}^l = \boldsymbol{M}_{l,i,j}$.

ij

$l \rightarrow$ nw fer



feature vector for $\mathbb{R}^n$ l

$$\hat{s}_{ij}^l = \frac{\exp(m_{ij}^l)}{\sum_{i'=1}^{h_M} \sum_{j'=1}^{w_M} \exp(m_{i'j'}^l)} \max f_l \frac{|w_{\hat{c}l}|}{\|\boldsymbol{w}_{\hat{c}}\|_2} \tag{1}$$

predict cls

$$\mathcal{L}_{\text{div}} = -\sum_{i=1}^{h_M} \sum_{j=1}^{w_M} \max(\hat{s}_{ij}^1, \hat{s}_{ij}^2, \ldots, \hat{s}_{ij}^{n_f}) \tag{2}$$

$W =$

L_{div}

Equation 1 uses the softmax to transform the feature maps $\boldsymbol{M}$ by normalizing their entries m_{ij}^l over the spatial dimensions and then scales the maps so that they focus on visible and important features by maintaining the relative mean of $\boldsymbol{M}$ while weighting them according to the predicted class. Equation 2 decreases the loss if the different weighted feature maps $\hat{S}^l$ attend to different locations. The final training loss is then $\mathcal{L}_{\text{final}} = \mathcal{L}_{CE} + \beta\mathcal{L}_{\text{div}}$ with the weighting factor $\beta \in \mathbb{R}_+$.

3.2.2 Feature Selection

For selecting the set of features N_{f^*} from the initial features N_f s. t. $|N_{f^*}| = n_f^*$, we run an adapted version of *glm-saga* until one solution $(\boldsymbol{W}_j^{\text{sparse}}, \boldsymbol{b}_j)$ of the regularization path uses a feature not already in N_{f^*}, which we then add to the set of selected features N_{f^*} and restart the adapted *glm-saga*. As adaptation, we extended the proximal operator of the group version of *glm-saga*, which operates on $\boldsymbol{w}_l = \boldsymbol{W}_{:,l}$, which are the entries in $\boldsymbol{W}$ that correspond to an entire feature l. Since $\|\boldsymbol{w}_l\|_2$ indicates the importance of l, we additionally only keep entries for features that have the maximum norm or are in N_{f^*}, s. t. exactly one feature is added per iteration. The resulting proximal operator is:

$$\text{Prox}_{\lambda_1,\lambda_2}(\boldsymbol{w}_i) = \begin{cases} \frac{\boldsymbol{w}_i(\|\boldsymbol{w}_i\|_2 - \lambda_1)}{(1+\lambda_2)\|\boldsymbol{w}_i\|_2} & \text{if } \|\boldsymbol{w}_i\|_2 > \lambda_1 \underline{\wedge \|\boldsymbol{w}_i\|_2 = \max_{j' \in N_f \setminus N_{f^*}} \|\boldsymbol{w}_{j'}\|_2 \vee i \in N_{f^*}} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{3}$$

feature only

feature

The extensions are underlined.

$w_i \left(\frac{1}{1+\lambda_2}\right)\left(1 - \frac{\lambda_1}{\|w_i\|_2}\right)$

Dataset	CUB-2011	Stanford Cars	FGVC-Aircraft	NABirds	ImageNet-1K
#Classes n_c	200	196	100	555	1 000
Training	5 994	8 144	6 667	23 929	1 281 167
Testing	5 774	8 041	3 333	24 633	50 000

Table 1: Overview of the number of classes, training and testing examples for the used datasets

3.3 *Glm-saga*

Glm-saga (*Wong et al.*, *2021*) is a solver for Elastic Net (Zou & Hastie, 2005) regularized models that combines the path algorithm of Friedman et al. (2010) with advancements in variance reduced gradient methods by Gazagnadou et al. (2019) to allow for GPU-accelerated fitting at *ImageNet*-scale. For precomputed and normalized features, *glm-saga* computes a series of n sparse linear classifiers

$$P = [(\boldsymbol{W}_1^{\text{sparse}}, \boldsymbol{b}_1), (\boldsymbol{W}_2^{\text{sparse}}, \boldsymbol{b}_2), \dots, (\boldsymbol{W}_n^{\text{sparse}}, \boldsymbol{b}_n)], \tag{4}$$

where the sparsity of $\boldsymbol{W}_i^{\text{sparse}}$ is decreasing with i. This series is called *regularization path*. Each of the models minimizes the elastic net loss

ℓ_2 ℓ_1

$$\mathcal{L} = \mathcal{L}_{\text{target}} + \lambda R(\boldsymbol{W}) \qquad\qquad R(\boldsymbol{W}) = (1-\alpha)\frac{1}{2}\|\boldsymbol{W}\|_F + \alpha\|\boldsymbol{W}\|_{1,1} \tag{5}$$

with the initial optimization goal $\mathcal{L}_{\text{target}}$, in our case the cross-entropy loss, and regularization strength λ, which decreases along the path. The regularization function R($\boldsymbol{W}$) with weighting factor $\alpha \in [0, 1]$ is known as Elastic Net (Zou & Hastie, 2005), a mix between ℓ_1 and ℓ_2 regularization, which leads to better results than pure ℓ_1 regularization ($\alpha = 1$), while causing a comparable sparsity. *Glm-saga* optimizes the problem iteratively, clipping entries in $\boldsymbol{W}$ with an absolute value below a threshold after each step, to ensure real sparsity. For reference, the pseudocode for *glm-saga* is included in Appendix C.

4 Experiments

This section contains our experimental results. We validate our method using Resnet50 (He et al., 2016), DenseNet121 (Huang et al., 2017) and Inception-v3 (Szegedy et al., 2016) on four common benchmark datasets in the domain of fine-grained image classification. Additionally, we show the applicability of a *SLDD-Model* for large scale datasets like ImageNet-1K (Russakovsky et al., 2015). An overview of CUB-2011 (Wah et al., 2011), Stanford Cars (Krause et al., 2013), FGVC-Aircraft (Maji et al., 2013), NABirds (Van Horn et al., 2015) and ImageNet-1K is given in Table 1. Additionally, CUB-2011 contains labels for the images such as attributes (*e.g.* "red wing") as well as for the classes, which makes it easier to measure the alignment with understandable concepts. After the competitive accuracy and the impact of $\mathcal{L}_{\text{div}}$ is shown, the interpretability of the *SLDD-Model* is discussed and these attributes are used to show the alignment of the learned features. The implementation details can be found in Appendix B.1. Finally, the tradeoff between interpretability and accuracy is visualized.

4.1 Diversity Metric

To assess the impact of $\mathcal{L}_{\text{div}}$, we developed a measurement for the local diversity of the feature maps $\boldsymbol{M}$ that led to the decision, inspired by the diversity component of MCL (Chang et al., 2020), which entails a different way of computing the features. For that, we consider the k feature maps $\boldsymbol{M}_k$ that are weighted the highest for the predicted class $\hat{c}$ in $\boldsymbol{W}$. To only compare the localization, softmax is applied to the $\boldsymbol{M}_k$, yielding $\boldsymbol{S}_k$. With these distributions $\boldsymbol{S}_k$, we compute the diversity as

$$\text{diversity@k} = \frac{\sum_{i=1}^{h_M}\sum_{j=1}^{w_M} \max(s_{ij}^1, s_{ij}^2, \dots, s_{ij}^k)}{k} \tag{6}$$

with diversity@k $\in [\frac{1}{k}, 1]$ to measure how different and pronounced the $\boldsymbol{M}_k$ are localized. Since we focus on $n_{wc} \leq 5$, we set $k = 5$. We report the mean diversity@5 for all classes that use at least five features. Note that the proposed $\mathcal{L}_{\text{div}}$ is a weighted version of diversity@n_f.

Method	Additional Supervision required	Features trained end-to-end	n_f^* ↓	n_{wc} ↓	ImageNet-1K ↑	CUB-2011 ↑
CBM (Koh et al., 2020) *- joint*	✓	✓	112	112	-	80.1 (76.8*)
CBM (Koh et al., 2020) *- independent*	✓	✗	112	112	-	76.0
CBM-AUC (Sawada & Nakamura, 2022)	✓	✓	256	256	-	**82.3**
glm-saga (Wong et al., 2021)	✗	✓	n_f	≤ 5	58.0*	76.1 (78.0*)
SLDD-Model (**Ours**) with $n_f^* = n_f$	✗	✓	n_f	**5**	**76.7***	80.3 (**86.5***)
SLDD-Model (**Ours**)	✗	✓	**50**	**5**	**72.8***	78.3 (**84.0***)

Table 2: Comparison with competitors on accuracy in percent. For ease of comparison, we evaluated CUB-2011 on Inception-v3 ($n_f = 1024$) and ImageNet-1K with Resnet50 ($n_f = 2048$) (*denotes Resnet50 accuracy). The dense Resnet50 achieves $80.9\,\%$ on ImageNet-1K. For *glm-saga*, we selected the solution with maximum $n_{wc} \leq 5$. Arrows indicate generally preferable directions. For *CBM - joint*, Resnet50 results were created as described in Appendix B.1.1

23,-- --

div

	CUB-2011					FGVC-Aircraft					NABirds					Stanford Cars				
	$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$	
$\mathcal{L}_{\text{div}}$	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
✗	86.6	81.8	85.3	79.5	83.4	90.0	88.4	89.4	87.3	88.1	84.2	79.5	83.3	77.3	80.7	93.2	90.9	92.6	89.3	91.1
✓	86.6	**84.0**	**86.5**	**81.7**	**84.0**	**91.4**	**90.7**	**91.1**	**89.8**	**90.1**	**84.4**	**81.0**	**84.0**	**79.8**	**81.7**	**93.6**	**92.1**	**93.3**	**91.1**	**92.0**
MCL Chang et al. (2020)	86.1	81.9	85.1	79.4	82.8	90.1	88.4	89.0	87.2	88.1	-	-	-	-	-	93.1	91.0	92.5	89.0	90.7
FRL Zheng et al. (2020)	86.4	81.5	85.3	78.9	82.6	90.0	88.5	89.4	87.5	88.2	-	-	-	-	-	93.3	90.8	92.6	89.4	90.9

Table 3: Impact of the loss function on accuracy in percent for Resnet50. Best results per column are in bold.

4.2 Results

We report the accuracy on the test set for the dense model after training the pretrained model on the training data with $n_{wc} = n_f$, for the sparse model with $n_{wc} \leq 5$, and for the result of our whole pipeline, the model with finetuned features, obtained by training the sparse model on the training data, and still $n_{wc} \leq 5$. Every shown metric is the average over five (four for ImageNet-1K) randomly seeded runs. The standard deviations are included in the appendix.
Table 3 shows the competitive performance of our *SLDD-Model* to the dense Resnet50. It is evident that an extreme sparsity of $s = \frac{5}{2048}$ can be obtained in the final layer with just 0.1 to 0.4 percent points less accuracy. Additionally decreasing the number of features by $97.6\,\%$, resulting in just 50 instead of the previous 2048 features, only reduces the accuracy compared to the dense model by 1.3 to 2.7 percent points. Finally, our proposed $\mathcal{L}_{\text{div}}$ improves the accuracy for all sparse models. Table 4 shows the general applicability of our method with different backbones. However, we observed some instability and no increased accuracy when finetuning the DenseNet121 with $n_f^* = 2048$, showing a positive effect of sharing features. Table 2 compares our approach to competitors: Without requiring additional supervision, we achieve a competitive performance compared to *CBM* (Koh et al., 2020)-based methods, while achieving a lower dimensionality and higher sparsity. Additionally, we improve the accuracy of *glm-saga* with heavily reduced n_f^*. For ImageNet-1K, we skipped the dense training and directly used the pretrained model. The good scalability of our proposed method to this large dataset with a higher number of classes is displayed in Table 5. Table 6 shows the diversity@5: With $\mathcal{L}_{\text{div}}$ it is very close to the maximum value of $100\,\%$ in the dense case and still heavily increased in the sparse cases. This showcases that $\mathcal{L}_{\text{div}}$ is suitable to ensure a diverse localization of the used feature maps.

4.2.1 Comparison with Other Loss Functions

We compare our diversity loss $\mathcal{L}_{\text{div}}$ with the MCL (Chang et al., 2020) and the FRL (Zheng et al., 2020). The used hyperparameters for the loss functions are reported in Appendix B.1.1 and we focussed on three datasets to save computational resources. Table 3 shows that our $\mathcal{L}_{\text{div}}$ reaches

	CUB-2011					FGVC-Aircraft					NABirds					Stanford Cars				
	$n_f^* = n_f$			$n_f^* = 50$		$n_f^* = n_f$			$n_f^* = 50$		$n_f^* = n_f$			$n_f^* = 50$		$n_f^* = n_f$			$n_f^* = 50$	
Backbone	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
DenseNet121	86.3	76.2	82.9	75.7	83.1	**91.5**	88.2	89.8	88.1	90.0	84.1	72.8	64.6	71.0	80.5	93.3	87.3	91.7	85.8	91.4
Inception-v3	82.3	78.0	80.3	74.0	78.3	88.9	87.5	88.1	85.9	87.4	79.0	75.8	77.3	73.1	76.5	91.5	88.9	90.3	86.3	89.4
Resnet50	**86.6**	**84.0**	**86.5**	**81.7**	**84.0**	91.4	**90.7**	**91.1**	**89.8**	**90.1**	**84.4**	**81.0**	**84.0**	**79.8**	**81.7**	**93.6**	**92.1**	**93.3**	**91.1**	**92.0**

Table 4: Accuracy in percent dependent on backbone. Best results per column are in bold.

Dense ($n_f^* = 2048$)	Sparse ($n_f^* = 2048$)	Finet. ($n_f^* = 2048$)	Sparse ($n_f^* = 50$)	Finet. ($n_f^* = 50$)
80.9	62.2	76.7	44.8	72.8

Table 5: Accuracy in percent for Resnet50 on ImageNet-1K using the pretrained dense model.

	CUB-2011					FGVC-Aircraft					Stanford Cars				
	$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$	
$\mathcal{L}_{\text{div}}$	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
✗	50.2	46.0	43.4	48.0	46.5	46.5	43.8	40.9	45.9	44.0	45.0	41.7	39.1	43.6	43.7
✓	**98.9**	**69.9**	**71.9**	**65.2**	**72.6**	**98.8**	**85.7**	**86.6**	**69.3**	**73.9**	**99.2**	**72.4**	**74.6**	**63.7**	**74.8**
MCL Chang et al. (2020)	52.5	51.4	48.9	56.7	52.3	50.1	50.6	48.3	51.7	50.1	49.0	49.0	46.2	51.8	49.5
FRL Zheng et al. (2020)	51.1	47.1	44.1	49.0	46.3	48.2	44.6	41.2	44.9	43.1	46.2	43.0	40.0	43.0	41.7

Table 6: Impact of the loss function on diversity@5 in percent for Resnet50. Best results per column are in bold.

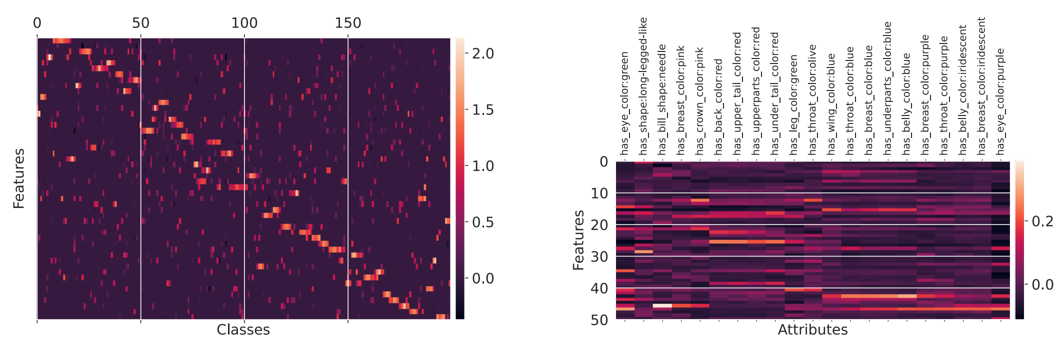
the highest accuracy across all datasets. Notably, the accuracy reported in (Chang et al., 2020) for the MC-Loss is achieved by a two-layer MLP plus additional techniques, whereas we only use one layer to ensure linearly separable representations for our *SLDD-Model*. Although it is expected that applying $\mathcal{L}_{\text{div}}$ has a positive effect on diversity@5 due to the similar formulation, we could observe a remarkable uplift in diversity@5 (Table 6) compared to MCL and FRL, which also optimize for differently localized features.

4.3 Interpretability

In this section, we discuss the interpretability of the proposed *SLDD-Model* using example models. The interpretability of the proposed *SLDD-Model* is based upon using very few (n_{wc}) features from a small pool of n_f^* to make a decision. A low n_f^* allows the analysis of the remaining features to try to align them with a human understandable concept, which is discussed in Section 4.3.1. Since the sparse linear layer is easily interpretable, the complete model with sufficiently well understood features is both **locally** and **globally** interpretable.

For **global interpretability**, the final layer of the *SLDD-Model* can be fully visualized and analyzed. Figure 3a shows $\boldsymbol{W}^{\text{sparse}}$. This allows the practitioner to verify the global behavior of the model. For example, the attribute aligned with "has-bill-shape:needle" in the presented model in Section 4.3.1 has a non-zero weight for all four classes that have the attribute in more than 30 % of examples. The visualization of the classes positively related to a specific attribute helps to trust the model. Additionally, $\boldsymbol{W}^{\text{sparse}}$ allows for further feature understanding, since it is possible to analyze the similarities between classes that share a feature. If the feature is aligned well, this leads to knowledge discovery.

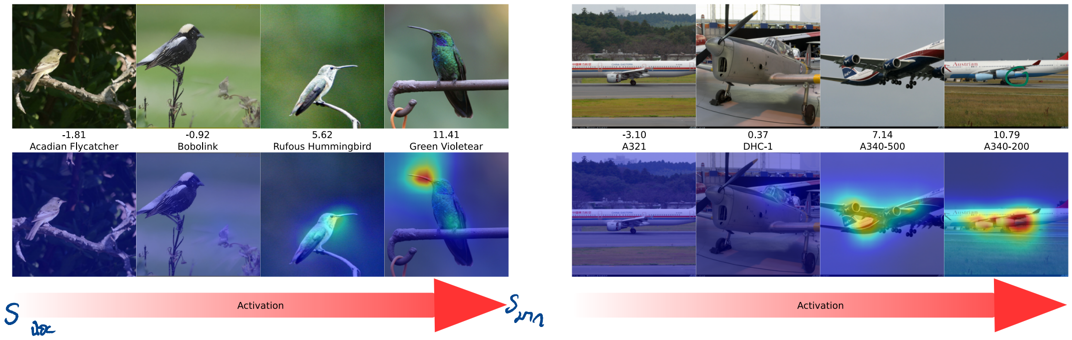
The **local interpretability** describes the explanation of a single decision made by the model. De-



(a) Exemplary $\boldsymbol{W}^{\text{sparse}}$. The alignment of the features with attributes in CUB-2011 is displayed in Figure 3b.

(b) Relationship between chosen features and attributes ($C > 20\%$) of CUB-2011 for the exemplary model. Higher values indicate that the feature describes the attribute.

Figure 3: Visualization of the sparse matrix and the feature alignment.



(a) Feature 45 with $C = 0.39$ for the attribute "has-bill-shape:needle": Higher activations are localized around the bill, and a needle-like bill is visible.

(b) Manually aligned feature of a model trained on FGVC-Aircraft: The feature is aligned with four engine aircraft.

Figure 4: Example images and localization $\boldsymbol{L}$, scaled to indicate feature activation, in ascending order for two models. The text between the rows describes the activation value for the image, which drops below 0 due to the normalization of *glm-saga*, and the class name.

cisions with sparsely connected features are inherently locally interpretable, if the features can be interpreted and localized, as shown in Figure 1. The practitioner can understand where and what was found in the image, and due to the full global interpretability also understand the behavior around the current example.

4.3.1 Feature Alignment

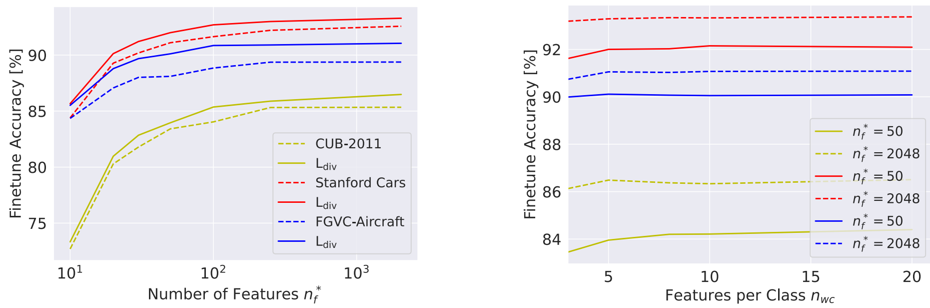
In this section, we demonstrate how the features of the proposed *SLDD-Model* can be interpreted.We describe how one can use additional labels or expert knowledge to interpret the features and demonstrate that several learned sparse features are directly connected to attributes relevant to humans. Thus, our model learns such abstract concepts directly from the data. Overall, due to the very limited number of used features n_f^*, the features can be thoroughly analyzed and interpreted. For feature localization, we follow a masking approach similar to Fong & Vedaldi (2017), which is described in Appendix D.

Alignment with Additional Data We use the attributes A contained in CUB-2011 to align the learned features with these labels after the finetuning. For each attribute $a \in A$ and feature j we compute a score C_{aj} that corresponds to a relative increase of the feature when the attribute is present:

$$\delta_{aj} = \frac{1}{|\rho_{a+}|}\sum_{i\in\rho_{a+}} \boldsymbol{F}^{\text{train}}_{i,j} - \frac{1}{|\rho_{a-}|}\sum_{i\in\rho_{a-}} \boldsymbol{F}^{\text{train}}_{i,j} \qquad C_{aj} = \frac{\delta_{aj}}{\max(\boldsymbol{F}^{\text{train}}_{:,j}) - \min(\boldsymbol{F}^{\text{train}}_{:,j})} \tag{7}$$

The set of indices whose images contain the attribute is denoted by ρ_{a+}. We considered an attribute to be present if the human annotated it with "probably" or "definitely". Annotations with "guessing" were neither included in the positive (ρ_{a+}) nor negative (ρ_{a-}) examples. For one exemplary model a part of the matrix of C values is displayed in Figure 3b. It is clear, that some features correspond to colors, some to specific shapes like "bill-shape:needle" and other features do not correlate with specific attributes. Figure 4a visually validates the connection that was implied in Figure 3b.

Alignment without Additional Data Without additional data one has to analyze the feature activations manually or with expert knowledge. Some useful aspects for understanding a feature are the localization, extreme examples, feature visualization (Olah et al., 2017) or which classes use that feature. One such alignment for a model trained on FGVC-Aircraft is displayed in Figure 4b. Aligning learned features with human understandable concepts is still challenging as a single feature can refer to multiple aspects and human understandable concepts do not need to be axis aligned (Szegedy et al., 2013). However, the low dimensionality of the remaining features allows for a sophisticated analysis in practice.



(a) Impact of changing n_f^* with $n_{wc} = 5$

(b) Impact of changing n_{wc} with $n_f^* = 50$ or 2048

Figure 5: Relationship between Finetune Accuracy and aspects related to interpretability for Resnet50.

4.4 Interpretability Tradeoff

In this section, we analyze the impact of changing n_f^* and n_{wc} on the finetuning accuracy of the model trained with $\mathcal{L}_{\mathrm{div}}$, shown in Figure 5. Figure 5a visualizes the impact of n_f^*: With decreasing n_f^* the accuracy drops slowly until a dataset-specific threshold is reached, at which a steep decline starts. Additionally, the proposed $\mathcal{L}_{\mathrm{div}}$ works regardless of n_f^*. Figure 5b shows the finetuning accuracy in relation to n_{wc}: The accuracy is rather insensitive to n_{wc}, only decreasing when $n_{wc} < 5$, which is the case for both dimensions and showcases that five features suffice for a competitive model even if the features are shared among classes. Figure 5 demonstrates the tradeoff that our *SLDD-Model* offers: Both n_{wc} and n_f^* can be drastically reduced with either a negligible or small impact on accuracy to adapt to the amount of interpretability needed.

5 Limitations and Future Work

As shown in Figure 5, the *SLDD-Model* cannot get arbitrarily low-dimensional or sparse with competitive accuracy via our proposed method. The optimal sparsity and dimensionality for a given problem are hard to predict and might require some experiments to determine the minimum values for competitive accuracy. Aligning all used features with human concepts is still difficult, albeit more feasible than without a *SLDD-Model*. Future work could use a more interpretable feature extractor like *B-cos Networks* (Böhle et al., 2022) to alleviate that problem. It seems promising to apply a *SLDD-Model* to other safety-critical domains, such as medical, where an expert can be utilized to align the features and follow the decision, as it can help bring the required interpretability and trustworthiness to the domain. Embodied autonomous agents can also benefit from it, as the entire decision process can be thoroughly analyzed. While more interpretable models could be used to more deliberately bring harm, they can disclose existing problems with machine learning models and open up the opportunity to build fair and trustworthy models. Finally, sparsity and dimensionality could be part of metrics used to quantify the trustworthiness of a model.

6 Conclusion

In this work, we proposed the interpretable sparse low-dimensional decision model (*SLDD-Model*) to allow a human to follow and understand the decision of a Deep Neural Network for image classification. Our proposed pipeline constructs a *SLDD-Model* with drastically increased global and local interpretability while still showing competitive accuracy. As demonstrated, a practitioner can manually configure the pipeline to set the tradeoff between accuracy and interpretability. Our novel loss increases the feature diversity and we showed that identifying a class with varied features can improve the accuracy. Finally, our *SLDD-Model* offers measurable aspects of interpretability, which allows future work to not just compare itself on accuracy but also on interpretability.

7 REPRODUCIBILITY

For reproducibility, we uploaded the code for both the feature selection and the feature diversity loss $\mathcal{L}_{\text{div}}$ as supplementary material. Additionally, we clearly reported the used hyperparameters and data augmentation in the implementation details, Appendix B.1.

REFERENCES

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *Advances in neural information processing systems*, 31, 2018.

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6541–6549, 2017.

Adrien Bibal, Michael Lognoul, Alexandre Streel, and Benoît Frénay. Legal requirements on explainability in machine learning. *Artificial Intelligence and Law*, 29, 06 2021. doi: 10.1007/s10506-020-09270-4.

Moritz Böhle, Mario Fritz, and Bernt Schiele. B-cos networks: Alignment is all we need for interpretability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10329–10338, 2022.

Dongliang Chang, Yifeng Ding, Jiyang Xie, Ayan Kumar Bhunia, Xiaoxu Li, Zhanyu Ma, Ming Wu, Jun Guo, and Yi-Zhe Song. The Devil is in the Channels: Mutual-Channel Loss for Fine-Grained Image Classification. *IEEE Transactions on Image Processing*, 29:4683–4695, 2020. ISSN 1057-7149. doi: 10.1109/tip.2020.2973812.

Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your "Flamingo" is My "Bird": Fine-Grained, or Not. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 00:11471–11480, 2021. doi: 10.1109/cvpr46437.2021.01131.

Tianshui Chen, Wenxi Wu, Yuefang Gao, Le Dong, Xiaonan Luo, and Liang Lin. Fine-grained representation learning and recognition by exploiting hierarchical semantic embedding. In *Proceedings of the 26th ACM international conference on Multimedia*, pp. 2023–2031, 2018.

Po-Yung Chou, Cheng-Hung Lin, and Wen-Chung Kao. A Novel Plug-in Module for Fine-Grained Visual Classification. *arXiv*, 2022.

Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

Ruth C Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE international conference on computer vision*, pp. 3429–3437, 2017.

Sorelle A Friedler, Chitradeep Dutta Roy, Carlos Scheidegger, and Dylan Slack. Assessing the local interpretability of machine learning models. 2019.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

Nidham Gazagnadou, Robert Gower, and Joseph Salmon. Optimal mini-batch and step sizes for saga. In *International conference on machine learning*, pp. 2142–2150. PMLR, 2019.

Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International conference on machine learning*, pp. 1319–1327. PMLR, 2013.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Adrian Hoffmann, Claudio Fanconi, Rahul Rade, and Jonas Kohler. This looks like that... does it? shortcomings of latent space prototype interpretability in deep networks. *arXiv preprint arXiv:2105.02968*, 2021.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Sheikh Rabiul Islam, William Eberle, and Sheikh K Ghafoor. Towards quantification of explainability in explainable artificial intelligence methods. In *The thirty-third international flairs conference*, 2020.

Saachi Jain, Hadi Salman, Eric Wong, Pengchuan Zhang, Vibhav Vineet, Sai Vemprala, and Aleksander Madry. Missingness bias in model debugging. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=Te5ytkqsnl.

Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, pp. 2668–2677. PMLR, 2018.

Sunnie SY Kim, Nicole Meister, Vikram V Ramaswamy, Ruth Fong, and Olga Russakovsky. Hive: evaluating the human interpretability of visual explanations. *arXiv preprint arXiv:2112.03184*, 2021.

Pieter-Jan Kindermans, Sara Hooker, Julius Adebayo, Maximilian Alber, Kristof T Schütt, Sven Dähne, Dumitru Erhan, and Been Kim. The (un) reliability of saliency methods. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, pp. 267–280. Springer, 2019.

Pang Wei Koh, Thao Nguyen, Yew Siang Tang, Stephen Mussmann, Emma Pierson, Been Kim, and Percy Liang. Concept bottleneck models. In *International Conference on Machine Learning*, pp. 5338–5348. PMLR, 2020.

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

Haoyu Liang, Zhihao Ouyang, Yuyuan Zeng, Hang Su, Zihao He, Shu-Tao Xia, Jun Zhu, and Bo Zhang. Training interpretable convolutional neural networks by differentiating class-specific filters. In *European Conference on Computer Vision*, pp. 622–638. Springer, 2020.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear cnn models for fine-grained visual recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 1449–1457, 2015.

S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

Andrei Margeloiu, Matthew Ashman, Umang Bhatt, Yanzhi Chen, Mateja Jamnik, and Adrian Weller. Do concept bottleneck models learn as intended? *arXiv preprint arXiv:2105.04289*, 2021.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of Chess Knowledge in AlphaZero. *arXiv*, 2021.

George A Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological review*, 63(2):81, 1956.

Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020.

Meike Nauta, Ron van Bree, and Christin Seifert. Neural prototype trees for interpretable fine-grained image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14933–14943, 2021.

Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Vikram V Ramaswamy, Sunnie S Y Kim, Nicole Meister, Ruth Fong, and Olga Russakovsky. ELUDE: Generating interpretable explanations via a decomposition into labelled and unlabelled features. *arXiv*, 2022. Decomposition in unlabelled / labelled features. Sparse explanation, not prediction as it uses ground truth attribute labels for explanation. Does not have to be faithful with actually done prediction.

Pratyush Rokade and BKSP Kumar Raju Alluri. Towards quantification of explainability algorithms. In *2021 The 5th International Conference on Advances in Artificial Intelligence (ICAAI)*, pp. 31–37, 2021.

Stefan Rüping et al. Learning interpretable models. 2006.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Yoshihide Sawada and Keigo Nakamura. Concept Bottleneck Model With Additional Unsupervised Concepts. *IEEE Access*, 10:41758–41765, 2022. ISSN 2169-3536. doi: 10.1109/access.2022.3167702.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Hong Tao, Chenping Hou, Feiping Nie, Yuanyuan Jiao, and Dongyun Yi. Effective discriminative feature selection with nontrivial solution. *IEEE transactions on neural networks and learning systems*, 27(4):796–808, 2015.

Kale-ab Tessera, Sara Hooker, and Benjamin Rosman. Keep the gradients flowing: Using gradient flow to study sparse network optimization. *arXiv preprint arXiv:2102.01670*, 2021.

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)*, pp. 595–604, 2015. doi: 10.1109/CVPR.2015.7298658.

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

Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In *International Conference on Machine Learning*, pp. 11205–11216. PMLR, 2021.

Hongyu Yang, Cynthia Rudin, and Margo Seltzer. Scalable bayesian rule lists. In *International conference on machine learning*, pp. 3921–3930. PMLR, 2017.

Peiyu Yang, Zeyi Wen, and Ajmal Mian. Multi-grained interpretable network for image recognition. In *26TH International Conference on Pattern Recognition*, 2022.

Mert Yuksekgonul, Maggie Wang, and James Zou. Post-hoc concept bottleneck models. In *ICLR 2022 Workshop on PAIR^2Struct: Privacy, Accountability, Interpretability, Robustness, Reasoning on Structured Data*, 2022. URL https://openreview.net/forum?id=HAMeOIRD_g9.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

Runkai Zheng, Zhijia Yu, Yinqi Zhang, Chris Ding, Hei Victor Cheng, and Li Liu. Learning Class Unique Features in Fine-Grained Visual Classification. *arXiv*, 2020.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2005.00503.x.

	CUB-2011					FGVC-Aircraft					NABirds					Stanford Cars				
		$n_f^* = 2048$		$n_f^* = 50$			$n_f^* = 2048$		$n_f^* = 50$			$n_f^* = 2048$		$n_f^* = 50$			$n_f^* = 2048$		$n_f^* = 50$	
$\mathcal{L}_{\text{div}}$	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
✗	**86.6** ±0.4	81.8 ±0.3	85.3 ±0.2	79.5 ±0.3	83.4 ±0.2	90.0 ±0.3	88.4 ±0.3	89.4 ±0.2	87.3 ±0.4	88.1 ±0.3	84.2 ±0.1	79.5 ±0.3	83.3 ±0.1	77.3 ±0.3	80.7 ±0.2	93.2 ±0.1	90.9 ±0.2	92.6 ±0.1	89.3 ±0.3	91.1 ±0.1
✓	**86.6** ±0.2	**84.0** ±0.2	**86.5** ±0.1	**81.7** ±0.2	**84.0** ±0.3	**91.4** ±0.2	**90.7** ±0.3	**91.1** ±0.2	**89.8** ±0.4	**90.1** ±0.1	**84.4** ±0.2	**81.0** ±0.2	**84.0** ±0.0	**79.8** ±0.2	**81.7** ±0.1	**93.6** ±0.2	**92.1** ±0.3	**93.3** ±0.1	**91.1** ±0.1	**92.0** ±0.2

Table 7: Accuracy in percent dependent on $\mathcal{L}_{\text{div}}$ for Resnet50. Best results per column are in bold and $\pm$ indicates the standard deviation across five runs.

	CUB-2011					FGVC-Aircraft					NABirds					Stanford Cars				
		$n_f^* = n_f$		$n_f^* = 50$			$n_f^* = n_f$		$n_f^* = 50$			$n_f^* = n_f$		$n_f^* = 50$			$n_f^* = n_f$		$n_f^* = 50$	
Backbone	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
DenseNet121	86.3 ±0.0	76.2 ±0.5	82.9 ±0.5	75.7 ±0.7	83.1 ±0.1	**91.5** ±0.1	88.2 ±0.5	89.8 ±0.4	88.1 ±0.2	90.0 ±0.4	84.1 ±0.1	72.8 ±0.4	64.6 ±22.8	71.0 ±0.5	80.5 ±0.3	93.3 ±0.1	87.3 ±0.4	91.7 ±0.1	85.8 ±0.3	91.4 ±0.2
Inception-v3	82.3 ±0.1	78.0 ±0.3	80.3 ±0.4	74.0 ±0.7	78.3 ±0.4	88.9 ±0.1	87.5 ±0.2	88.1 ±0.2	85.9 ±0.3	87.4 ±0.2	79.0 ±0.1	75.8 ±0.2	77.3 ±0.3	73.1 ±0.2	76.5 ±0.1	91.5 ±0.1	88.9 ±0.2	90.3 ±0.2	86.3 ±0.2	89.4 ±0.2
Resnet50	**86.6** ±0.2	**84.0** ±0.2	**86.5** ±0.1	**81.7** ±0.2	**84.0** ±0.3	91.4 ±0.2	**90.7** ±0.3	**91.1** ±0.2	**89.8** ±0.4	**90.1** ±0.1	**84.4** ±0.2	**81.0** ±0.2	**84.0** ±0.0	**79.8** ±0.2	**81.7** ±0.1	**93.6** ±0.2	**92.1** ±0.3	**93.3** ±0.1	**91.1** ±0.1	**92.0** ±0.2

Table 8: Accuracy in percent dependent on backbone. Best results per column are in bold and $\pm$ indicates the standard deviation across five runs.

A Appendix

In this appendix, we provide implementation details and standard deviations for the experiments. Additionally, the pseudocode for *glm-saga* (*Wong et al., 2021*) is shown. Finally, we present the feature visualization technique and more ablations on $\mathcal{L}_{\text{div}}$.

B Detailed Results

The full results of Section 4.2 with the standard deviations are presented in Tables 7 to 13. The reported standard deviations are, except for DenseNet121 on NABirds, as mentioned in Section 4.2, generally rather small compared to the differences in means, which supports our conclusions.

B.1 Implementation Details

We use Pytorch (Paszke et al., 2019) to implement our methods and on ImageNet pretrained models as backbone feature extractor. We utilized *glm-saga* and *robustness* (Engstrom et al., 2019). The images are resized to 448×448 (299×299 for Inception-v3, 224×224 for ImageNet-1K), normalized, randomly horizontally flipped and jitter is applied. The model is finetuned using stochastic gradient descent on the specific dataset for 150 (100 for NABirds) epochs with a batch size of 16 (64 for ImageNet-1K), starting with $5 \cdot 10^{-3}$ as learning rate for the pretrained layer and 0.01 for the final linear layer. Both get multiplied by 0.4 every 30 epochs. Additionally, we used momentum of 0.9, ℓ_2-regularization of $5 \cdot 10^{-4}$ and apply a dropout rate of 0.2 on the features to reduce dependencies. β was set to 0.196 for Resnet50, 0.098 for DenseNet121 and 0.049 for Inception-v3. For the feature selection, we set $\alpha = 0.8$ and reduce the regularization strength λ by $90\,\%$ as we found it sped up the process without decreasing performance.
We use *glm-saga* to compute the regularization path with $\alpha = 0.99$ and all other parameters set to default with a lookbehind of $T = 5$. From this path, the solution with maximum $n_{wc} \leq 10$ is selected. Then the non-zero entries with the lowest absolute value get zeroed out until we are left

ImageNet-1K				
	$n_f^* = 2048$		$n_f^* = 50$	
Dense	Sparse	Finet.	Sparse	Finet.
80.9	62.2 ±0.0	76.7 ±0.0	44.8 ±0.1	72.8 ±0.1

Table 9: Accuracy in percent for Resnet50 on ImageNet-1Kusing the pretrained dense model.. Best results per column are in bold and $\pm$ indicates the standard deviation across four runs.

Algorithm 1 Pseudocode from *glm-saga* (Wong et al., 2021)

Initialize table of scalars $a'_i = 0$ for $i \in [n]$
Initialize average gradient of table $g_{avg} = 0$ and $g_{0avg} = 0$
for minibatch $B \subset [n]$ **do**
  for $i \in B$ **do**
    $a_i = x_i^T \beta + \beta_0 - y_i$
    $g_i = a_i \cdot x_i$ *// calculate new gradient information*
    $g'_i = a'_i \cdot x_i$ *// calculate stored gradient information*
  end for
  $g = \frac{1}{|B|} \sum_{i \in B} g_i$
  $g' = \frac{1}{|B|} \sum_{i \in B} g'_i$
  $g_0 = \frac{1}{|B|} \sum_{i \in B} a_i$
  $g'_0 = \frac{1}{|B|} \sum_{i \in B} a'_i$
  $\beta = \beta - \gamma(g - g' + g_{avg})$
  $\beta_0 = \beta_0 - \gamma(g_0 - g'_0 + g_{0avg})$
  $\beta = \text{Prox}_{\gamma\lambda\alpha, \gamma\lambda(1-\alpha)}(\beta)$
  for $i \in B$ **do**
    $a'_i = a_i$ *// update table*
    $g_{avg} = g_{avg} + \frac{|B|}{n}(g - g')$ *// update average*
    $g_{0avg} = g_{0avg} + \frac{|B|}{n}(g_0 - g'_0)$
  end for
end for

with $n_{wc} = 5$, as we empirically found that they do not improve test accuracy after finetuning. This selected solution replaces the final layer of our model. Then we train for 40 epochs, starting with the final learning rate of the initial training multiplied by 100 ($\frac{1}{100}$ of that for ImageNet-1K), and decrease it by $60\,\%$ every 10 epochs. Dropout on the features was set to 0.1 and momentum was increased to 0.95. Note that, while the increased momentum has been important for the stability of the final training, the hyperparameters were not thoroughly optimized for the sparse case.

B.1.1 Competitors

For creating the accuracy for Resnet50 and *CBM* (Koh et al., 2020) - *joint* in Table 2 we resized the images to 448×448 and used a batch size of 16. The remaining used hyperparameters were almost identical to the *CBM* experiments with Inception-v3, but we only trained for up to 400 epochs, as 650 led to decreased accuracy (-0.8 percent points). Additionally, the learning rate was not decayed, mirroring the published code. The reported accuracy stems from three runs with a standard deviation of 0.7.
For MCL, we used the reported hyperparameters of $\mu = 0.005$ and $\lambda = 10$. For finetuning, we assigned every feature to every class that was using it. We optimized the hyperparameters for FRL based on accuracy, leading to $K = 10$ and $\lambda = 0.01$.

C *Glm-saga*

This section includes the Pseudocode for *glm-saga* (Wong et al., 2021) in algorithm 1. The proximal operator $\text{Prox}_{\lambda_1,\lambda_2}(\beta)$ is defined as:

$$\text{Prox}_{\lambda_1,\lambda_2}(\beta) = \begin{cases} \frac{\beta - \lambda_1}{1+\lambda_2} & \text{if } \beta > \lambda_1 \\ \frac{\beta + \lambda_1}{1+\lambda_2} & \text{if } \beta < \lambda_1 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

	CUB-2011					FGVC-Aircraft					NABirds					Stanford Cars				
	$n_f^*=2048$			$n_f^*=50$		$n_f^*=2048$			$n_f^*=50$		$n_f^*=2048$			$n_f^*=50$		$n_f^*=2048$			$n_f^*=50$	
$\mathcal{L}_{\text{div}}$	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
✗	50.2 ±0.2	46.0 ±0.2	43.4 ±0.3	48.0 ±0.5	46.5 ±0.3	46.5 ±0.5	43.8 ±0.7	40.9 ±0.4	45.9 ±0.6	44.0 ±0.6	41.5 ±0.2	38.4 ±0.1	34.9 ±0.1	40.8 ±0.6	35.1 ±0.7	45.0 ±0.2	41.7 ±0.3	39.1 ±0.1	43.6 ±0.6	43.7 ±0.4
✓	**98.9** ±0.1	**69.9** ±0.5	**71.9** ±0.4	**65.2** ±1.4	**72.6** ±0.3	**98.8** ±0.2	**85.7** ±1.5	**86.6** ±1.4	**69.3** ±0.8	**73.9** ±1.3	**98.7** ±0.1	**69.5** ±1.1	**81.0** ±1.2	**70.7** ±1.9	**85.3** ±1.0	**99.2** ±0.1	**72.4** ±2.0	**74.6** ±1.5	**63.7** ±1.3	**74.8** ±0.9

Table 10: diversity@5 in percent dependent on $\mathcal{L}_{\text{div}}$ for Resnet50. Best results per column are in bold and $\pm$ indicates the standard deviation across five runs.

	CUB-2011					FGVC-Aircraft					NABirds					Stanford Cars				
	$n_f^*=n_f$			$n_f^*=50$		$n_f^*=n_f$			$n_f^*=50$		$n_f^*=n_f$			$n_f^*=50$		$n_f^*=n_f$			$n_f^*=50$	
Backbone	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
DenseNet121	98.5 ±0.1	69.1 ±1.7	64.2 ±1.2	**76.2** ±1.0	71.2 ±0.8	**99.0** ±0.1	40.4 ±0.7	39.9 ±0.9	64.3 ±2.0	62.6 ±2.1	98.6 ±0.1	45.7 ±1.1	42.3 ±3.5	63.1 ±3.1	66.2 ±2.2	98.7 ±0.1	47.4 ±1.3	46.3 ±1.0	**71.8** ±1.5	68.0 ±1.2
Inception-v3	86.3 ±0.5	**74.9** ±0.9	65.1 ±0.6	53.2 ±0.8	52.7 ±0.4	95.1 ±0.2	**87.2** ±1.6	72.2 ±1.4	53.7 ±1.1	56.1 ±0.9	81.8 ±1.0	47.9 ±1.0	47.0 ±0.8	42.7 ±1.6	45.6 ±1.0	92.0 ±0.3	**76.3** ±0.6	63.7 ±0.6	52.0 ±1.2	50.7 ±0.8
Resnet50	**98.9** ±0.1	69.9 ±0.5	**71.9** ±0.4	65.2 ±1.4	**72.6** ±0.3	98.8 ±0.2	85.7 ±1.5	**86.6** ±1.4	**69.3** ±0.8	**73.9** ±1.3	**98.7** ±0.1	**69.5** ±1.1	**81.0** ±1.2	**70.7** ±1.9	**85.3** ±1.0	**99.2** ±0.1	72.4 ±2.0	**74.6** ±1.5	63.7 ±1.3	**74.8** ±0.9

Table 11: diversity@5 in percent dependent on backbone. Best results per column are in bold and $\pm$ indicates the standard deviation across five runs.

D Visualization of Features

For feature visualization, we follow a masking approach. We systematically blur, following (Fong & Vedaldi, 2017), one patch of size $p \times p$ of the image and measure the difference in feature activation between the augmented image and not augmented image. The actual localization map $\boldsymbol{L}_p \in \mathbb{R}^{n_f^* \times \frac{w}{p} \times \frac{h}{p}}$ for that square size is computed by

$$\boldsymbol{L}_{pxy} = \text{ReLU}(\boldsymbol{f}(I) - \boldsymbol{f}(I_{pxy})) \tag{9}$$

where I_{pxy} indicates the image where a p-sized patch starting at position $(x*p, y*p)$ is blurred and the ReLU suppresses parts that increased the feature activation, since blur should not be injecting a feature. The final localization map is the combination of different square sizes $p \in \{28, 56, 64, 112, 224\}$ to accommodate for differently sized features:

$$\boldsymbol{L} = \sum_p \frac{\boldsymbol{L}_p}{\max(\boldsymbol{L}_p)} \tag{10}$$

Notably, $\boldsymbol{L}_p$ has to be resized according to the smallest p and we only show $\boldsymbol{L}^i$ for one feature i.

E Ablations on Feature Diversity Loss

In this section, we present an additional analysis of the factors in $\mathcal{L}_{\text{div}}$ and the impact of β.

E.1 Factor Importance

We analyzed the impact of the two factors in Equation 1 with for accuracy optimized β, shown in Tables 14 and 15. The label *w/o Class-Specific* indicates not using the weights of the predicted class

	CUB-2011					FGVC-Aircraft					Stanford Cars				
	$n_f^*=2048$			$n_f^*=50$		$n_f^*=2048$			$n_f^*=50$		$n_f^*=2048$			$n_f^*=50$	
$\mathcal{L}_{\text{div}}$	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
✗	**86.6** ±0.4	81.8 ±0.3	85.3 ±0.2	79.5 ±0.3	83.4 ±0.2	90.0 ±0.3	88.4 ±0.3	89.4 ±0.2	87.3 ±0.4	88.1 ±0.3	93.2 ±0.1	90.9 ±0.2	92.6 ±0.1	89.3 ±0.3	91.1 ±0.1
✓	**86.6** ±0.2	**84.0** ±0.2	**86.5** ±0.1	**81.7** ±0.2	**84.0** ±0.3	**91.4** ±0.2	**90.7** ±0.3	**91.1** ±0.2	**89.8** ±0.4	**90.1** ±0.1	**93.6** ±0.2	**92.1** ±0.3	**93.3** ±0.1	**91.1** ±0.1	**92.0** ±0.2
MCL (Chang et al., 2020)	86.1 ±0.2	81.9 ±0.3	85.1 ±0.2	79.4 ±0.2	82.8 ±0.1	90.1 ±0.1	88.4 ±0.2	89.0 ±0.2	87.2 ±0.5	88.1 ±0.1	93.1 ±0.1	91.0 ±0.2	92.5 ±0.2	89.0 ±0.5	90.7 ±0.3
FRL (Zheng et al., 2020)	86.4 ±0.2	81.5 ±0.2	85.3 ±0.2	78.9 ±0.5	82.6 ±0.5	90.0 ±0.1	88.5 ±0.2	89.4 ±0.4	87.5 ±0.2	88.2 ±0.2	93.3 ±0.1	90.8 ±0.3	92.6 ±0.2	89.4 ±0.2	90.9 ±0.2

Table 12: Accuracy in percent for Resnet50 compared to other loss functions. Best results per column are in bold and $\pm$ indicates the standard deviation across five runs.

	CUB-2011					FGVC-Aircraft					Stanford Cars				
	$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$	
$\mathcal{L}_{\text{div}}$	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
✗	50.2 ±0.2	46.0 ±0.2	43.4 ±0.3	48.0 ±0.5	46.5 ±0.3	46.5 ±0.5	43.8 ±0.7	40.9 ±0.4	45.9 ±0.6	44.0 ±0.6	45.0 ±0.2	41.7 ±0.3	39.1 ±0.1	43.6 ±0.6	43.7 ±0.4
✓	**98.9** ±0.1	**69.9** ±0.5	**71.9** ±0.4	**65.2** ±1.4	**72.6** ±0.3	**98.8** ±0.2	**85.7** ±1.5	**86.6** ±1.4	**69.3** ±0.8	**73.9** ±1.3	**99.2** ±0.1	**72.4** ±2.0	**74.6** ±1.5	**63.7** ±1.3	**74.8** ±0.9
MCL (Chang et al., 2020)	52.5 ±0.3	51.4 ±0.7	48.9 ±0.8	56.7 ±1.0	52.3 ±1.0	50.1 ±0.7	50.6 ±0.8	48.3 ±1.2	51.7 ±1.8	50.1 ±1.9	49.0 ±0.6	49.0 ±0.9	46.2 ±0.8	51.8 ±1.2	49.5 ±1.3
FRL (Zheng et al., 2020)	51.1 ±0.3	47.1 ±0.5	44.1 ±0.4	49.0 ±0.7	46.3 ±0.6	48.2 ±0.3	44.6 ±0.3	41.2 ±0.4	44.9 ±0.8	43.1 ±0.5	46.2 ±0.1	43.0 ±0.4	40.0 ±0.4	43.0 ±1.1	41.7 ±1.1

Table 13: diversity@5 in percent for Resnet50 compared to other loss functions. Best results per column are in bold and ± indicates the standard deviation across five runs.

	CUB-2011					FGVC-Aircraft					Stanford Cars				
	$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$	
Loss	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
$\mathcal{L}_{\text{div}}$	**86.6** ±0.2	**84.0** ±0.2	**86.5** ±0.1	**81.7** ±0.2	**84.0** ±0.3	**91.4** ±0.2	**90.7** ±0.3	**91.1** ±0.2	**89.8** ±0.4	**90.1** ±0.1	**93.6** ±0.2	**92.1** ±0.3	**93.3** ±0.1	**91.1** ±0.1	**92.0** ±0.2
w/o Rescaling	86.3 ±0.2	82.7 ±0.4	85.6 ±0.3	80.2 ±0.5	83.4 ±0.3	90.8 ±0.4	89.7 ±0.3	90.1 ±0.2	88.7 ±0.4	89.1 ±0.3	93.2 ±0.1	91.0 ±0.2	92.8 ±0.2	89.9 ±0.1	91.5 ±0.2
w/o Class-Specific	86.3 ±0.2	82.0 ±0.2	85.4 ±0.3	79.0 ±0.2	83.1 ±0.2	90.2 ±0.3	88.6 ±0.4	89.6 ±0.3	87.3 ±0.2	88.5 ±0.4	93.2 ±0.2	90.8 ±0.2	92.6 ±0.2	89.1 ±0.3	91.0 ±0.3

Table 14: Impact of factors in $\mathcal{L}_{\text{div}}$ on accuracy in percent for Resnet50. Best results per column are in bold and ± indicates the standard deviation across five runs.

	CUB-2011					FGVC-Aircraft					Stanford Cars				
	$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$	
Loss	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
$\mathcal{L}_{\text{div}}$	98.9 ±0.1	**69.9** ±0.5	**71.9** ±0.4	65.2 ±1.4	72.6 ±0.3	98.8 ±0.2	**85.7** ±1.5	**86.6** ±1.4	**69.3** ±0.8	**73.9** ±1.3	99.2 ±0.1	**72.4** ±2.0	**74.6** ±1.5	63.7 ±1.3	**74.8** ±0.9
w/o Rescaling	**99.3** ±0.0	67.6 ±0.9	66.9 ±0.5	**70.5** ±2.0	**73.3** ±1.8	**99.7** ±0.0	72.4 ±1.2	74.6 ±1.6	68.5 ±4.3	73.6 ±2.1	**99.7** ±0.0	64.9 ±0.8	65.3 ±0.6	**68.3** ±3.6	73.5 ±2.0
w/o Class-Specific	50.6 ±0.3	46.2 ±0.3	43.5 ±0.2	48.1 ±1.1	46.8 ±0.6	47.6 ±0.6	44.3 ±0.5	41.2 ±0.3	45.0 ±1.7	43.7 ±1.6	45.5 ±0.2	42.2 ±0.3	39.5 ±0.2	42.7 ±1.1	42.5 ±1.1

Table 15: Impact of factors in $\mathcal{L}_{\text{div}}$ on diversity@5 in percent for Resnet50. Best results per column are in bold and ± indicates the standard deviation across five runs.

	CUB-2011					FGVC-Aircraft					Stanford Cars				
	$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$	
β	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
0	**86.6** ±0.4	81.8 ±0.3	85.3 ±0.2	79.5 ±0.3	83.4 ±0.2	90.0 ±0.3	88.4 ±0.3	89.4 ±0.2	87.3 ±0.4	88.1 ±0.3	93.2 ±0.1	90.9 ±0.2	92.6 ±0.1	89.3 ±0.3	91.1 ±0.1
0.00196	86.4 ±0.2	82.0 ±0.3	85.5 ±0.3	79.3 ±0.3	83.3 ±0.2	90.2 ±0.2	88.7 ±0.3	89.5 ±0.3	87.6 ±0.2	88.5 ±0.2	93.1 ±0.1	90.9 ±0.3	92.6 ±0.1	89.0 ±0.2	90.8 ±0.2
0.0196	86.4 ±0.2	82.2 ±0.3	85.3 ±0.3	79.4 ±0.6	83.2 ±0.3	90.6 ±0.4	89.0 ±0.3	89.7 ±0.3	87.4 ±0.4	88.5 ±0.4	93.1 ±0.1	91.0 ±0.2	92.6 ±0.2	89.0 ±0.2	90.9 ±0.2
<u>0.196</u>	**86.6** ±0.2	**84.0** ±0.2	**86.5** ±0.1	**81.7** ±0.2	**84.0** ±0.3	**91.4** ±0.2	**90.7** ±0.3	**91.1** ±0.2	**89.8** ±0.4	**90.1** ±0.1	**93.6** ±0.2	**92.1** ±0.3	**93.3** ±0.1	**91.1** ±0.1	**92.0** ±0.2

Table 16: Accuracy in percent dependent on β for Resnet50. Best results per column are in bold and $\pm$ indicates the standard deviation across five runs. Our used β is underlined.

	CUB-2011					FGVC-Aircraft					Stanford Cars				
	$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$		$n_f^* = 2048$			$n_f^* = 50$	
β	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.	Dense	Sparse	Finet.	Sparse	Finet.
0	50.2 ±0.2	46.0 ±0.2	43.4 ±0.3	48.0 ±0.5	46.5 ±0.3	46.5 ±0.5	43.8 ±0.7	40.9 ±0.4	45.9 ±0.6	44.0 ±0.6	45.0 ±0.2	41.7 ±0.3	39.1 ±0.1	43.6 ±0.6	43.7 ±0.4
0.00196	51.0 ±0.1	46.1 ±0.2	43.7 ±0.2	49.2 ±0.6	47.6 ±0.4	48.1 ±0.4	44.6 ±0.4	41.5 ±0.4	45.6 ±0.5	43.8 ±0.3	46.3 ±0.0	42.6 ±0.3	39.7 ±0.2	43.0 ±0.7	43.3 ±0.5
0.0196	64.0 ±0.6	50.8 ±0.6	47.8 ±0.4	50.2 ±1.0	48.6 ±0.9	75.4 ±0.5	54.4 ±0.7	50.2 ±0.6	50.2 ±1.9	48.5 ±1.0	66.1 ±0.7	48.7 ±0.6	45.2 ±0.4	46.6 ±2.0	46.0 ±1.3
<u>0.196</u>	**98.9** ±0.1	**69.9** ±0.5	**71.9** ±0.4	**65.2** ±1.4	**72.6** ±0.3	**98.8** ±0.2	**85.7** ±1.5	**86.6** ±1.4	**69.3** ±0.8	**73.9** ±1.3	**99.2** ±0.1	**72.4** ±2.0	**74.6** ±1.5	**63.7** ±1.3	**74.8** ±0.9

Table 17: diversity@5 in percent dependent on β for Resnet50. Best results per column are in bold and $\pm$ indicates the standard deviation across five runs. Our used β is underlined.

and *w/o Rescaling* refers to not maintaining their relative mean. We used $\beta = 0.001 \cdot \frac{196}{2048} \approx 1\text{e}{-5}$ for *w/o Class-Specific*, since we use the size of the feature maps with $196 = w_M \cdot h_M$ and the number of features of the baseline model $n_f = 2048$ as scaling factors in order to be less dependent of model architecture and image size, and $\beta = 0.1$ for *w/o Rescaling*. Only the combination of both factors leads to an improved accuracy, validating our idea that it is important to only enforce diversity of features that are found in the input and used in conjunction.

E.2 Loss Weighting

This section is concerned with the impact of the weighting factor β for the feature diversity loss $\mathcal{L}_{\text{div}}$. For our proposed method, we use $\beta = 0.196$. Tables 16 and 17 show that $\mathcal{L}_{\text{div}}$ improves the diversity@5 and accuracy across all datasets in the sparse case with increasing β up to a maximum roughly around $\beta = 1$. Setting the value higher leads to $\mathcal{L}_{\text{div}}$ dominating the training. To ensure that the network is still mainly optimized for classification, we choose $\beta = 0.196$, even though in some cases we could still observe a slight gain in accuracy with a slight increase of β. The positive relation between diversity@5 and accuracy, visualized in Figure 6, supports our approach of enforcing varied features for the extremely sparse case.

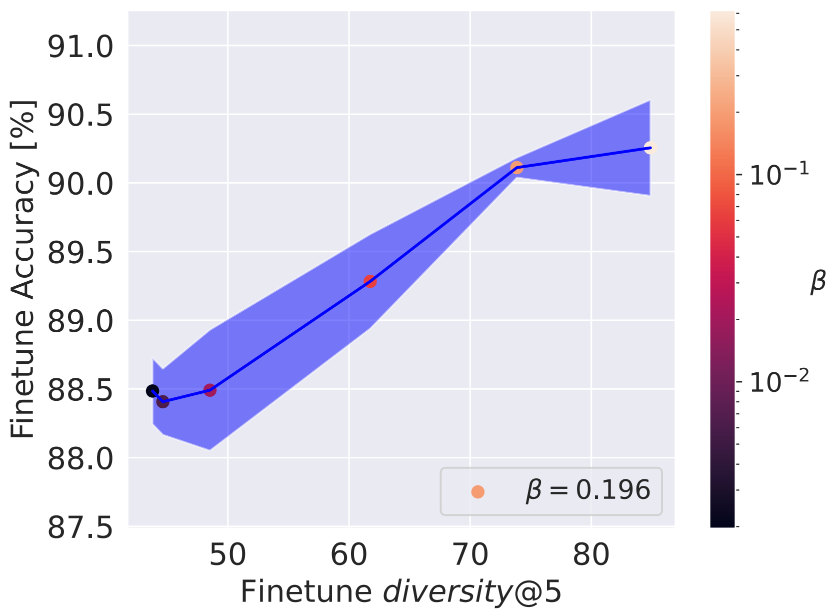


Figure 6: Relationship between finetuned diversity@5 and accuracy for varying β for Resnet50, portrayed via color, on FGVC-Aircraft. Each dot represents an increase by a factor of $\sqrt{10}$ and the standard deviation is indicated by the shaded area. $\beta = 1.96$ $\mathcal{L}_{\text{div}}$ is not shown, as it dominates the training.