On Explaining Knowledge Distillation: Measuring and Visualising the Knowledge Transfer Process

Gereziher Adhane
Universitat Oberta de Catalunya
gradhane@uoc.edu

Mohammad Mahdi Dehshibi
Universidad Carlos III de Madrid
mohammad.dehshibi@yahoo.com

Dennis Vetter
Goethe University Frankfurt
vetter@em.uni-frankfurt.de

David Masip
Universitat Oberta de Catalunya
dmasipr@uoc.edu

Gemma Roig
Goethe University Frankfurt
roig@cs.uni-frankfurt.de

Abstract

Knowledge distillation (KD) remains challenging due to the opaque nature of the knowledge transfer process from a Teacher to a Student, making it difficult to address certain issues related to KD. To address this, we proposed UniCAM, a novel gradient-based visual explanation method, which effectively interprets the knowledge learned during KD. Our experimental results demonstrate that with the guidance of the Teacher's knowledge, the Student model becomes more efficient, learning more relevant features while discarding those that are not relevant. We refer to the features learned with the Teacher's guidance as distilled features and the features irrelevant to the task and ignored by the Student as residual features. Distilled features focus on key aspects of the input, such as textures and parts of objects. In contrast, residual features demonstrate more diffused attention, often targeting irrelevant areas, including the backgrounds of the target objects. In addition, we proposed two novel metrics: the feature similarity score (FSS) and the relevance score (RS), which quantify the relevance of the distilled knowledge. Experiments on the CIFAR10, ASIRRA, and Plant Disease datasets demonstrate that UniCAM and the two metrics offer valuable insights to explain the KD process.

1. Introduction

Knowledge Distillation (KD) has emerged as a crucial technique in deep learning, especially in computer vision. It aims to develop efficient models without compromising performance [4, 5, 19, 31, 44]. By transferring knowledge from typically a complex Teacher model to a simpler Student model, KD can potentially address the increasing demand for deploying robust yet lightweight models in practical scenarios [43]. Nevertheless, despite its widespread adoption, the underlying mechanisms of KD remain somewhat opaque, thus impeding its broader application and theoretical comprehension.

The current research in knowledge distillation (KD) is confronted with four main challenges, including (1) understanding the specific knowledge that is transferred from Teacher to Student [3]; (2) evaluating whether KD improves the Student's focus on task-relevant features compared to independent training [5, 40]; (3) measuring the importance of features adopted or ignored by the Student for the target task; (4) addressing and resolving KD failures, mainly when there are significant architectural differences between Teacher and Student models [29, 37, 38].

Existing visual explainability methods for Convolutional Neural Networks (CNNs), like Grad-CAM [35], are not equipped to tackle these KD-specific challenges. While effective for single-model predictions, these methods cannot capture the nuance of knowledge transfer between models or quantify the relevance of distilled knowledge. Specifically, Grad-CAM focuses on the importance of class-specific features within a single model. However, it does not distinguish between knowledge inherited from the Teacher and knowledge independently learned by the Student.

To address these issues, we introduce a new framework for improving the explainability of KD. We will first define the key terms used in this paper to ensure clarity. **Distilled features** are unique to the Student and are acquired through KD-based training, which the Student considers relevant to the task. **Residual features** are present in the Teacher (Base model) but are not adopted by the Student, as the Student finds them irrelevant to the task during KD-based training. We used **Unique features** to collectively refer to the distilled and residual features, as they are unique to the Student and the Base model, respectively. Throughout this text, we will use the term **Base model** to refer to the model which

has the same architecture as the Student but is trained using only raw data. When the Teacher and Student have similar architecture, then the Teacher will act as a suitable Base model.

Our framework consists of two main components:

- Visual explainability tool – We introduce UniCAM (Unique Class Activation Mapping), a gradient-based visual explanation method designed explicitly for KD scenarios. UniCAM distinguishes between distilled and residual features using partial distance correlation to isolate unique features learned by each model. (detailed methodology in Section 3.1).
- Knowledge transfer metrics – We introduce Feature Similarity Score (FSS) and Relevance Score (RS). The former, FSS, quantifies the alignment of attention patterns between Student and Teacher (Base model) using distance correlation. The latter, RS, evaluates the task relevance of distilled and residual features using distance correlation between the extracted features and BERT embeddings of the ground truth labels [12].

FSS and RS work in tandem to provide a comprehensive analysis of KD's effectiveness. While FSS measures the similarity of learned representations between the Base model and the Student, RS assesses their relevance to the target task. This dual approach is crucial because high similarity (FSS) does not always guarantee optimal task-specific learning (RS). For example, a Student might closely imitate a Teacher's attention patterns (high FSS) but struggle to differentiate between important and irrelevant features for the specific task (potentially low RS).

Our key contributions are fourfold:

- Introducing **Uni**que **C**lass **A**ctivation **M**aps (*UniCAM*), a novel method for visualising the knowledge transfer process in KD. This approach offers insights into how Students acquire relevant features and discard less important ones under the Teacher's guidance, addressing challenges 1 and 2.
- Proposing the complementary FSS and RS metrics, which enable quantitative analysis of KD effectiveness. This dual-metric approach captures both the similarity of learned representations (FSS) and their task-specific relevance (RS), addressing challenges 2 and 3.
- Conducting extensive experiments across various KD techniques and model architectures, demonstrating the broad applicability of our approach. Our evaluation encompasses standard image classification datasets (CIFAR-10 [28], Microsoft PetImages [15]) and a more challenging fine-grained plant disease classification task [24]. This comprehensive assessment validates the effectiveness of our methods across diverse scenarios.
- Providing a detailed examination of KD failure cases, particularly those arising from significant capacity gaps between Teacher and Student models. Through this analysis, we demonstrate how our method can guide the selection of appropriate Teacher models or intermediate architectures (Teacher assistants) to improve KD outcomes, addressing challenge 4.

2. Related Work

Recent works on KD have focused on improving the performance of the Student [25], adapting the distillation process to specific tasks [13], or developing alternative methods for knowledge transfer [22]. In contrast to the performance-oriented works, some studies have explored KD explainability using various techniques. For instance, Cheng et al. [3] used information theory and mutual information to visualise and measure knowledge during KD. However, this method requires human annotations of the background and foreground objects, which limits its applicability and scalability. Moreover, using entropy to quantify randomness might be unreliable in scenarios with highly correlated data or multiple modes. Similarly, Wang et al. [42] used the KD and generative models to diagnose and interpret image classifiers, but this approach does not account for the knowledge acquired by the Student.

Existing visual explainability techniques offer valuable insights into how CNNs make decisions (e.g., [2, 7]). Applying these methods to KD could reveal if the Student focuses on the same input areas as a Base model and learns similar or superior features. For example, DeepVID [40] visually interprets and diagnoses image classifiers through KD. Haselhoff et al. [20] proposed a probability density encoder and a Gaussian discriminant decoder to describe how explainers deviate from concepts' training data in KD. However, existing visual explainability techniques cannot visualise the saliency maps of distilled and residual features. Similarity metrics offer a promising strategy to measure and identify features unique to one model [45], which could be useful to effectively quantify and explain the distilled knowledge.

Similarity measures have been widely employed across various disciplines, including machine learning [1,8,10,11], information theory [6, 9, 16], and computational neuroscience [27]. These measures offer valuable insights into how information is processed and encoded in different contexts. In deep learning, similarity metrics have been useful to (1) quantify how DNNs replicate the brain's encoding process [41], (2) compare vision transformers and convnets [26, 33], (3) gain insights into transfer learning [14, 30], and (4) explain the mechanisms behind deep model training [18]. In this study, we propose a novel gradient-based visual explainability technique and quantitative metrics that leverage similarity measures to isolate

unique features of each model and quantify their relevance, enhancing the transparency of the KD process.

3. Methodology

Given a Student model trained using KD and a Base model trained solely on data, our objective is to explain and quantify the amount of knowledge distilled during KD. Our approach leverages gradient-based explainability techniques to compute gradients with respect to input features, along with distance correlation (dCor) [39] and partial distance correlation (pdCor) [39, 45]. dCor measures the dependence between two random vectors that capture their multidimensional associations. Similarly, pdCor extends dCor to measure the association between two random vectors after adjusting for the influence of a third vector. It is computed by projecting the distance matrices onto a Hilbert space and taking the inner product between the U-centered matrices. Zhen et al. [45] used pdCor to condition multiple models and identify their unique features[1], which means removing the common features and assessing the remaining ones. This enables us to introduce a novel visual explanation and metrics to assess the knowledge the student learned (distilled features) and that it may have overlooked (residual features).

3.1. UniCAM: Unique Class Activation Mapping

Our goal is to generate saliency maps that highlight the *distilled* and *residual* features of the Student model, emphasising their importance and revealing their attention patterns to enable a deeper understanding of KD. Existing gradient-based visual explanations [2, 7, 35] generates saliency maps based on the gradients of the target class, effectively revealing the relevance of features for the target prediction. However, these techniques are not suited for KD, as they do not identify the distilled or residual features compared to the Base model. To overcome this limitation, we introduce *UniCAM*, a novel gradient-based explainability technique tailored for KD. *UniCAM* leverages pdCor to adjust feature representations and remove the shared features between the Student and the Base model. This process isolates the *distilled* and *residual* features, which correspond to the knowledge the Student has acquired or overlooked, providing insights into the relevance of these features for the target task.

Let x_s and x_b represent the features extracted from a specific convolutional layer of the Student and the Base model, respectively. The *UniCAM* method follows the following key steps: (1) First, we compute the pairwise distance matrices for both x_s and x_b, which capture the relationships between different feature vectors within the Student and Base models. (2) Next, we normalise these distance matrices to create adjusted distance matrices, denoted as P^s and P^b, which ensure the distance information is centred and standardised. (3) We then calculate the mutual influence between the Student and Base models' features and remove the shared components, effectively isolating the unique features that each model has learned. (4) Finally, we generate heatmaps of these unique features, which localise the importance and relevance of the distilled features compared to the Base model.

Following the approach explained in [39], we first compute the pairwise distance matrix $D^{(s)} = (D^{(s)}_{i,j})$ for the Student's feature set. The pairwise distance matrix captures the relationship between every pair of feature vectors within the Student model:

$$D^{(s)}_{i,j} = \sqrt{(x_i - x_j)^2 + \epsilon}, \tag{1}$$

where ϵ is a small positive number added for numerical stability. This matrix helps us quantify how closely related the feature vectors are, which forms the basis for identifying unique and shared features.

Next, we normalise the distance matrix using Eq. 2 to obtain the adjusted distance matrix $P^{(s)}$. This normalisation is a U-centred projection, which centres the matrix around the mean and adjusts it for the overall distribution of distances.

$$P^{(s)}_{i,j} = \begin{cases} D^{(s)}_{i,j} - \dfrac{1}{n-2}\sum_{l=1}^{n} D^{(s)}_{i,l} - \dfrac{1}{n-2}\sum_{k=1}^{n} D^{(s)}_{k,j} \\ \qquad + \dfrac{1}{(n-1)(n-2)}\sum_{k=1}^{n}\sum_{l=1}^{n} D^{(s)}_{k,l}, & i \neq j; \\ 0, & i = j. \end{cases} \tag{2}$$

This step ensures that the distance information is properly centred and scaled, making it easier to compare features across models. To isolate the unique features learned by the Student model, we adjust for the mutual influence between the Student and Base models. This step subtracts the shared features between the Student and Base model:

$$x_{s|unique} = P^{(s)} - \frac{\langle P^{(s)}, P^{(b)} \rangle}{\langle P^{(b)}, P^{(b)} \rangle} \cdot P^{(b)}. \tag{3}$$

Here, we compute the inner product between the adjusted distance matrices of the Student and Base models, $\langle P^{(s)}, P^{(t)} \rangle$, which captures their shared information as:

$$\langle P^{(s)}, P^{(b)} \rangle = \frac{1}{n(n-3)} \sum_{i \neq j} \left(P^{(s)}_{i,j} \cdot P^{(b)}_{i,j} \right). \tag{4}$$

We subtract the common features to isolate the unique ones in each model (distilled and residual features) and reflect what it has learned beyond the Base model.

[1] Unique features are the features specific either to the Base model (residual features) or to the Student (distilled features).

Algorithm 1 UniCAM: Unique Class Activation Mapping

Require: x_s, x_b – Features from Student and Base models.
Ensure: *UniCAM* maps for distilled and residual features.
Compute pairwise distance matrix for x_s:
for $i, j = 1$ to n **do**
  $D_{i,j}^{(s)} = \sqrt{(x_{s_i} - x_{s_j})^2 + \epsilon}$
end for
Normalise the distance matrix to compute $P^{(s)}$ using Eq. 2
Extract the distilled features by adjusting for the mutual influence:
$x_{s|unique} = P^{(s)} - \frac{\langle P^{(s)}, P^{(b)} \rangle}{\langle P^{(b)}, P^{(b)} \rangle} \cdot P^{(b)}$
Compute the importance of the distilled features:
$\beta_k^{(x_{s|unique}, c)} = \frac{1}{N} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^{(x_{s|unique})}}$
Generate *UniCAM*:
$L_{UniCAM}^{(x_{s|unique}, c)} = \text{ReLU}\left(\sum_k \beta_k^{(x_{s|unique}, c)} A^{(x_{s|unique})}\right)$
return *UniCAM* maps for $x_{s|unique}$

Once the unique features are extracted, we compute their importance for the target task prediction by calculating the gradients of the prediction with respect to these features. The gradient-based importance of each unit k for class c is given by:

$$\beta_k^{(x_{s|unique}, c)} = \frac{1}{N} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^{(x_{s|unique})}}, \tag{5}$$

where A_{ij} represents the activation at position (i, j) in the feature map, y^c is the output score for class c, $\beta_k^{(x_{s|unique}, c)}$ is the weight of unit k for class c calculated from the unique features $x_{s|unique}$, and N is a normalisation factor.

Finally, we generate the *UniCAM* saliency map by combining these weights and applying a ReLU function to highlight the most important areas of the input image:

$$L_{UniCAM}^{(x_{s|unique}, c)} = \text{ReLU}\left(\sum_k \beta_k^{(x_{s|unique}, c)} A^{(x_{s|unique})}\right). \tag{6}$$

This process creates a heatmap that visualises the important features the Student learned from the Teacher, making the KD process more explainable. To identify the residual features, we follow the same procedure, but instead of the Student, we consider the Base model in the previous steps. These residual features represent areas where the Student, with the guidance of the Teacher's knowledge, has determined that certain features (such as background elements or irrelevant parts of the object) are not important for the target task. Analysing residual features is crucial to understanding whether the Student is effectively ignoring irrelevant features or potentially overfitting. The above procedure is summarised in the algorithm 1.

3.2. Quantitative analysis of KD features

While visualising the heatmaps using *UniCAM* provides insights to make the KD process transparent, it is equally important to quantify the relevance and significance of the distilled and residual features. To this end, we introduce two novel metrics: Feature Similarity Score (*FSS*) and Relevance Score (*RS*). These metrics allow us to evaluate both the overall features learned by the Student compared to the Base model, as well as the distilled and residual features.

To compute these metrics, we first extract the relevant features from the salient regions identified by *UniCAM*. Next, we apply a perturbation technique proposed by Rong et al. [34], which modifies image pixels based on their prediction relevance. This perturbation preserves the most important pixels and replaces the irrelevant ones with the weighted average of their neighbours. As a result, the perturbed images retain the most salient features while reducing noise and redundancy. Fig. 1 shows this process with examples of input images, *UniCAM* explanations, and perturbed images.

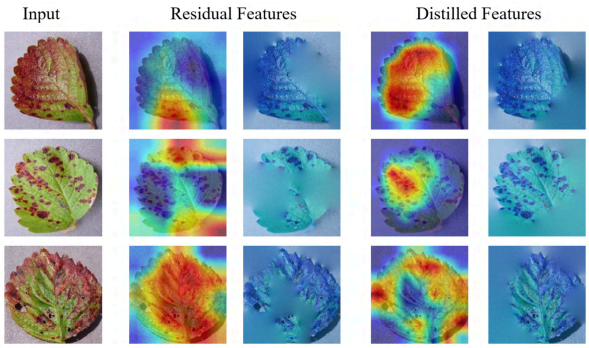


Figure 1. Residual and distilled features after perturbation.

The feature extraction function takes the perturbed images in Fig. 1 as input and extract the features (See Eq. 7) from the corresponding layer as:

$$\hat{x}_s = f_s(I \odot \mathcal{H}),\ \hat{x}_b = f_b(I \odot \mathcal{H}), \tag{7}$$

where f_s and f_b are the feature extractor functions for the Student and Base model, respectively, I is the input image, $\mathcal{H}$ is the heatmap generated by *UniCAM*, $\odot$ is the element-wise multiplication operator. These features are the numerical representations of the perturbed image that capture the essential information for prediction, and their dimension depends on the number of filters and the size of the activation maps in each layer. Hence, using these features, we quantify the similarity of the attention patterns and the relevance of distilled and residual features.

3.2.1 Feature similarity score (FSS)

FSS is designed to quantify the degree of alignment between the features learned between the Student and Base model at a specific layer. Since the Student is trained with the guidance of the Teacher's knowledge, *FSS* provides insight into how much the Student's focus has shifted or aligned with

the Base model features. A higher *FSS* value suggests that the Student and Base models are focusing on similar regions of the input, indicating that the Teacher's knowledge has not significantly altered the core feature focus of the Student or that the task is such that both models naturally converge on similar important features. Conversely, a lower *FSS* would suggest that the Student has diverged, potentially learning a more refined or generalised feature representation due to the knowledge from the Teacher. The *FSS* is computed as follows:

$$FSS = R^2(\hat{x}_s, \hat{x}_b) = \frac{1}{k}\sum_{i=1}^{k} \text{dCor}(\hat{x}_{s_i}, \hat{x}_{b_i}), \tag{8}$$

where k is the number of batches, $\hat{x}_{s_i}$ and $\hat{x}_{b_i}$ are the minibatch features of the Student and Base model. *FSS* ranges from 0 to 1, where 0 means no similarity and values close to 1 indicate higher attention pattern similarity.

3.2.2 Relevance score (RS)

While *FSS* measures the similarity between the features learned by the Student and Base model, it does not quantify how relevant these features are to the target task. To address this, we propose the Relevance Score (*RS*), which evaluates the relevance of the distilled and residual features for the target task.

To capture the semantic information of the target task more effectively, we use a pre-trained *BERT* embedding of the ground truth labels [12]. Unlike traditional one-hot encodings, which provide limited information, *BERT* embedding represents the labels in a high-dimensional space that captures richer semantic relationships between different labels. This allows us to compute meaningful correlations between the feature vectors encoded by the models and the ground truth, offering a more robust measure of relevance for the task. Hence, we compute the *RS* as follows:

$$RS = R^2(\hat{x}_s, gt) = \frac{1}{k}\sum_{i=1}^{k} \text{dCor}(\hat{x}_{s_i}, gt_i), \tag{9}$$

where $\hat{x}_{s_i}$ is the features extracted by the Student for each minibatch, and gt_i is the ground truth *BERT* embeddings for the corresponding targets in each batch. To compute the *RS* for the Base model, we replace $\hat{x}_{s_i}$ with $\hat{x}_{b_i}$.

Both *FSS* and *RS* provide a comprehensive quantitative technique to evaluate the similarity of the attention patterns and the relevance of the features learned during KD. This helps us understand whether the Student is acquiring features that are both similar and meaningful for the target task, offering deeper insights into explaining the KD process.

4. Experiments

4.1. Datasets and Implementation Details

We evaluate the proposed method on three public datasets for image classification: ASIRRA (Microsoft PetImages) [15], CIFAR10 [28] and Plant disease classification dataset [24]. ASIRRA contains 25,000 images of cats and dogs, while CIFAR10 contains 60,000 images of 10 classes. These datasets are widely used as benchmarks for image classification tasks and have different levels of complexity and diversity. Plant disease classification has a more challenging and realistic problem than fine-grained image classification, where the differences between classes are subtle and require more attention to detail. More results of plant disease classification are provided in the supplementary material **Sec. A**.

We performed various experiments to analyse and explain the KD process. First, we used ResNet-50 [21] as both the Student and Teacher models, effectively making the Teacher a Base model. This allows us to isolate the effects of KD without introducing the complexity bias that can arise when using a more powerful Teacher model. It ensures that any observed differences are due to the KD process itself rather than architectural disparities between the Teacher and Student. This experiment addresses key questions (1)-(3) by analysing the performance of the Student and the Base model, the similarity in attention patterns, and the relevance of the distilled and residual features. In the second experiment, we analysed different combinations of ResNet-18, ResNet-50, and ResNet-101 as Teacher and Student models to address the *fourth* question, which explores the impact of architecture differences on KD. We applied our approach to three state-of-the-art KD methods for classification: response-based KD [23], overhaul feature-based KD [22], and attention-based KD [31]. We implemented[2] the proposed method using PyTorch [32] and open source libraries from KD [36], pdCor [45] and Grad-CAM [17].

4.2. Results

We trained the models using 5-fold cross-validation. Details about each training setting are given in the supplementary material **Sec. D**. We assessed the performance and visual explanations of the Student and Base model trained with different KD. As shown in Fig. 2, the models trained with KD achieved higher accuracy compared to the equivalent Base model.

4.2.1 Comparison of Student and Base model attention patterns

We hypothesise that KD enhances the Student model's ability to learn more relevant features while discarding irrelevant ones. To test this hypothesis, we begin by using Grad-CAM, a widely adopted explainability technique, to provide an initial qualitative comparison of the feature localisation in both the Student and Base model across different layers.

[2]Code is available https://github.com/gadhane/UniCAM

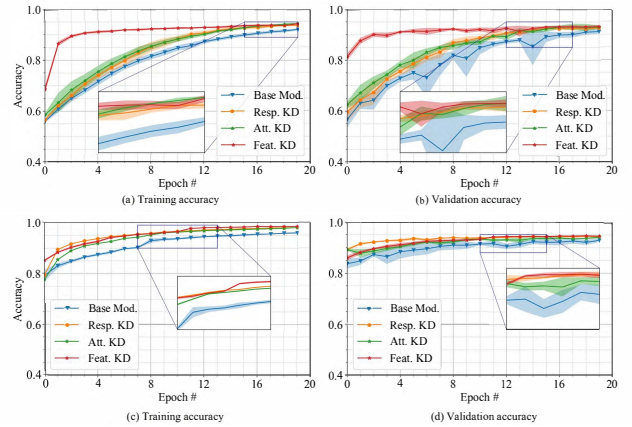


Figure 2. The training and validation accuracy((a) and (b) ASIRRA, (c) and (d) CIFAR10. The shaded region is the standard deviation.

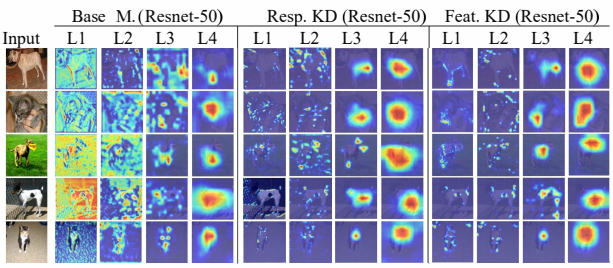


Figure 3. Grad-CAM visualisation of the Base model and Student trained with various KD techniques.

While Grad-CAM alone does not explain the KD process, it serves as a first step to visually compare which model encodes more relevant features based on attention maps. This initial visualisation offers insight into the general behaviour of the models, which is then complemented by our proposed quantitative analysis using *FSS* and *RS*. These metrics allow us to precisely measure the similarity and relevance of the learned features for the target task.

Fig. 3 shows the Grad-CAM visualisation at L1, L2, L3, and L4 of the last residual blocks in the four layers of the ResNet-50. The Base model relies on low-level features such as edges and spreads the attention over the entire image, including the background, in the first and intermediate layers. The saliency maps generated by Student models, however, localise more salient regions and focus on the object in all layers. This suggests that KD helps a model learn better features and improve its localisation ability by directing attention to more salient features earlier in the network.

We then use *FSS* and *RS* metrics to quantify the attention pattern similarity and relevance of the features between the Student and Base models. Fig. 4 (a) and (b) show the feature similarity of the attention patterns (*FSS*) and their relevance score (*RS*) between the Base model and Student across different layers, respectively. The *FSS* is higher for the deeper layers than for the input and intermediate layers, indicating that the Student models either learn more salient features in the input layers or fail to learn more irrelevant features. However, the Grad-CAMs in Fig. 3 show that the Student models have localised far better salient features than the Base model, especially in the input and intermediate layers. Therefore, the lower *FSS* at input and intermediate layers suggests that the Students have learned more relevant features that the Base model has not learned yet. Moreover, the Student models achieve higher *RS* than the Base model across all layers, implying that the models trained with KD have learned more relevant features with the guidance of the Teacher knowledge.

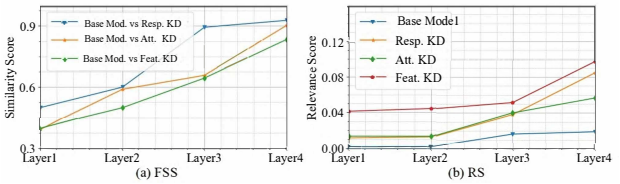


Figure 4. (a) FSS and (b) RS between Student (ResNet-50) and Base model (ResNet-50), localised by Grad-CAM.

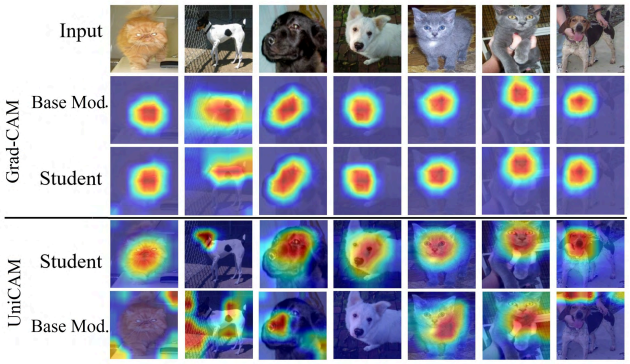
In general, Fig. 3 and Fig. 4 demonstrate that KD enables the Student models to encode more relevant features, which enhance the prediction accuracy and ability to generalise.

4.2.2 Visualising and quantifying distilled knowledge

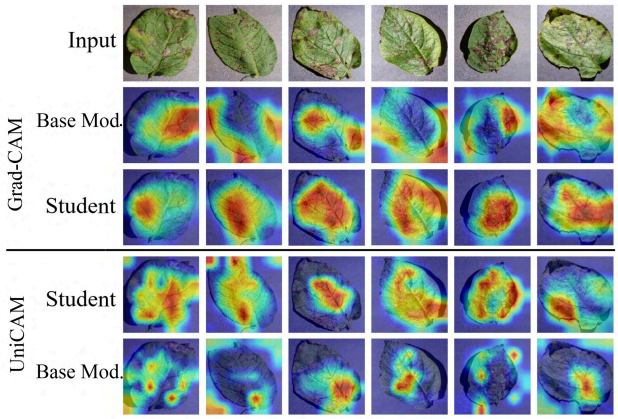
Here, we use our proposed method, *UniCAM*, to visualise the distilled and residual knowledge during KD. The saliency maps generated using *UniCAM* show that KD is not a simple feature copying process from the Teacher to the Student but a guided training process where the Teacher's knowledge assists the Student to learn existing or new features. This is illustrated in Fig. 5, where the distilled features mainly focus on the primary object, whereas the residual features localise regions in the background or seemingly less relevant parts of the object. In certain cases, *UniCAM* does not highlight any part of the object in the Base model. This occurs because, after the removal of common features, the remaining features from the Base model are less significant or lack relevance to the target task. In the plant disease classification[3], distilled features accurately identify segments of leaves essential for disease classification, demonstrating that KD helps models learn more relevant features.

In addition, Table 1 quantifies the relevance of the distilled, residual, and layer-specific features. Distilled and

[3]More results for plant disease experiments are presented in **Sec. A** of the supplementary material



(a) Pet Images



(b) Potato Early Blight

Figure 5. Sample visualisation of Distilled and residual features on PetImages and Plant disease dataset.

residual features are extracted from the areas localised by *UniCAM*, while layer-specific features are extracted from regions localised using Grad-CAM. Distilled features indicate how relevant the knowledge learned from the Teacher is for the Student. In contrast, residual features represent features the Student deemed less useful for the task and thus ignored. Layer-specific features provide an overall view of features encoded at each layer, helping assess which model has better localisation and a higher *RS*, indicating more relevant features. Models trained with various KD techniques show higher *RS* than their equivalent Base models, with overhaul feature distillation achieving the best performance by transferring intermediate feature representations, allowing the Student to learn more fine-grained and diverse features.

4.2.3 Exploring the capacity gap impact

The Student's performance often declines when there is a large architecture (capacity) gap between the Teacher and the Student [29, 38]. The drop in the Student's performance may stem from either its own challenges in learning relevant features or the overwhelming knowledge of the Teacher. To investigate this issue, we employ two distillation strategies in our experiments using ResNet-101 as the Teacher and ResNet-18 as the Student, which have a significant capacity disparity. In the first approach, we conduct direct distillation from ResNet-101 to ResNet-18. The second approach introduces an intermediate "Teacher assistant" [37] to help bridge the capacity gap between ResNet-101 and ResNet-18. We use *UniCAM* and *RS* to analyse the KD process in these settings, with a focus on how well the smaller model manages to learn relevant features.

Using the proposed methods, we first examine the impact of a large capacity gap on the knowledge transfer between Teacher and Student. We use ResNet-101 as the Teacher and ResNet-18 as the Student and apply KD to train the Student model. Fig. 6 demonstrates that, in this setting, the Base Model captures more relevant features than the Student model. This suggests that a large capacity gap impedes knowledge transfer, as the Student model cannot effectively learn from the complex Teacher's knowledge.

To bridge the capacity gap, we use an intermediate Teacher assistant [29] to enable a more effective and focused knowledge transfer from ResNet-101 to ResNet-18 via ResNet-50. Figure 7 compares the saliency maps of the distilled features learned by two Students: ResNet-18 directly distilled from ResNet-101 (R18-R101) and ResNet-18 distilled from ResNet-101 through Teacher assistant ResNet-50 (R18-R50-R101). The saliency maps, visualised using *UniCAM*, reveal that the Teacher assistant helps learn more relevant features that highlight the object parts. In contrast, R18-R101 learns some irrelevant features and misses the salient features for the gt prediction. In fact,

Table 1. Relevance of features (RS) learned by Student (ResNet-50) and Base model (ResNet-50).

Dataset	KD-Technique	Layer#	Layer-Specific Features		Residual / Distilled Features	
			Base Model	Student	Base Model	Student
ASIRRA [15]	Response-based	L1	0.0092	**0.0189**	**0.0024**	0.0017
		L2	0.0054	**0.0130**	0.0001	**0.0040**
		L3	0.0100	**0.0365**	0.0007	**0.008**
		L4	0.0141	**0.0861**	0.0043	**0.006**
	Attention-based	L1	0.0092	**0.0107**	**0.0049**	0.0047
		L2	0.0054	**0.0189**	0.0022	**0.0035**
		L3	0.0100	**0.0431**	0.0045	**0.0100**
		L4	0.0141	**0.0583**	0.0082	**0.0102**
	Feature-based	L1	0.0092	**0.0465**	0.0063	**0.0101**
		L2	0.0054	**0.0453**	0.0027	**0.0048**
		L3	0.0100	**0.0570**	0.0036	**0.0196**
		L4	0.0141	**0.0953**	0.0012	**0.0258**
CIFAR10 [28]	Response-based	L1	0.0063	**0.0304**	0.0040	**0.0155**
		L2	0.0133	**0.0378**	0.0090	**0.0148**
		L3	0.0282	**0.0432**	0.0046	**0.0113**
		L4	0.0417	**0.0585**	0.0090	**0.0106**
	Attention-based	L1	0.0063	**0.0232**	0.0043	**0.0136**
		L2	0.0133	**0.0280**	0.0099	**0.0101**
		L3	**0.0282**	0.0256	**0.0087**	0.0063
		L4	0.0417	**0.0437**	0.0017	**0.0021**
	Feature-based	L1	0.0063	**0.0311**	0.0028	**0.0185**
		L2	0.0133	**0.0388**	0.0017	**0.0153**
		L3	0.0282	**0.0457**	0.0070	**0.0117**
		L4	0.0417	**0.0794**	0.0024	**0.0150**

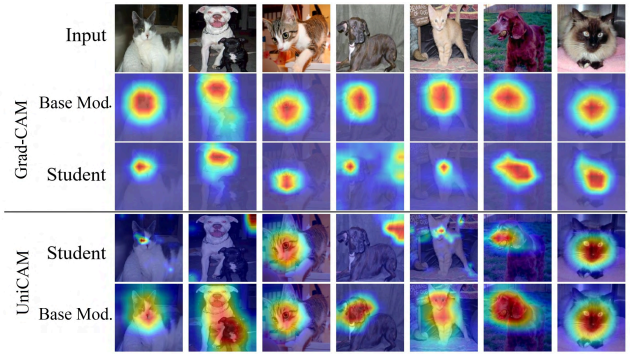


Figure 6. Relevant features learned by Student (ResNet-18) distilled from ResNet-101 compared to Base Model.

incorporating the Teacher assistant model facilitates the Student model in learning compatible knowledge from the complex Teacher and provides more appropriate supervision and feedback.

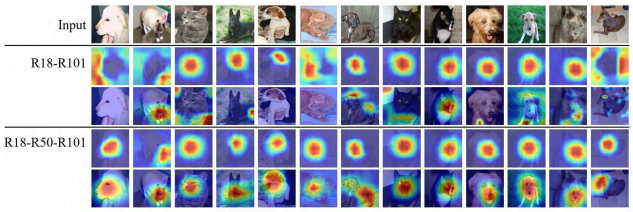


Figure 7. Grad-CAM (2^{nd} and 4^{th} rows) and *UniCAM* (3^{rd} and 5^{th} rows) visualisations.

We compare the relevance of features learned by the Student trained with the Teacher assistant and its equivalent Base model. We used *UniCAM* to generate the saliency maps of the distilled and residual features of each model. Fig. 8 shows that the saliency maps of the distilled features are more focused on the salient regions of the input images, while the residual features are more dispersed.

Table 2. RS of Students trained using Response-based KD with varying Teacher complexity level.

Layer#	Layer-Specific Features			Residual / Distilled Features		
	Base model	R18-R101	R18-R50-R101	Base model	R18-R101	R18-R50-R101
Layer 1	0.0037	0.0022	**0.0052**	0.0014	0.0007	**0.005**
Layer 2	0.0039	0.0035	**0.005**	0.0016	0.0014	**0.0031**
Layer 3	0.0057	0.0045	**0.0074**	0.0012	0.0008	**0.006**
Layer 4	0.0063	0.0052	**0.0082**	0.0018	0.0011	**0.0076**

Finally, Table 2 quantifies the relevance of the features learned by the Base model and equivalent Student model at different layers. The model trained with the Teacher assistant has learned more relevant features compared to the model directly distilled from ResNet-101 and the Base model.

The empirical findings presented above indicate that the capacity gap between the Teacher and Student models influences the quality and efficiency of KD. We demonstrate the benefit of our methods to explain the Student model's behaviour, both when it succeeds and fails to learn relevant knowledge from the Teacher. Therefore, our visual explanation and metrics can help to select the optimal Teacher-Student pairs for improved performance.

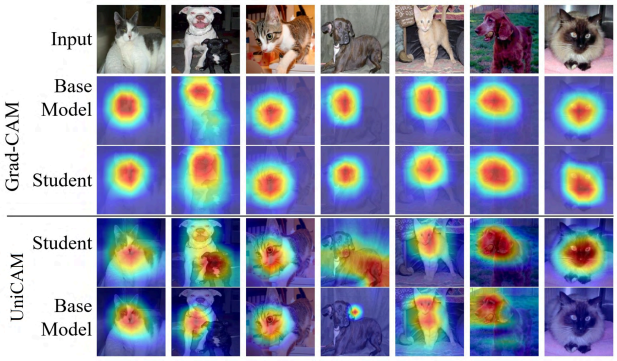


Figure 8. Comparison of distilled and residual features between Student (R18-R50-R101) and Base Model.

5. Discussion and Future Works

This paper presented novel techniques to explain and quantify the knowledge during KD. We proposed *UniCAM*, a gradient-based visual explanation method to explain the distilled knowledge and residual features during KD. Our experimental results show that *UniCAM* provides a clear and comprehensive visualisation of the features acquired or missed by the Student during KD. We also proposed two metrics: *FSS* and *RS* to quantify the similarity of the attention patterns and the relevance of the distilled knowledge and residual features. The proposed method has certain limitations. The experiments were exclusively conducted on classification tasks, which is one of the many potential applications of KD. In addition, we acknowledge the added computational cost introduced by the need to compute pairwise distances, gradients for feature localisation, and the proposed metrics. As part of future work, we aim to extend *UniCAM* to more complex datasets and explore its applicability to tasks beyond classification to enhance the robustness and versatility of the proposed method.

Acknowledgement

This research was supported by "PID2022-138721NB-I00" grant from the Spanish Ministry of Science, Research National Agency and FEDER (UE).

References

[1] Gereziher Adhane, Mohammad Mahdi Dehshibi, and David Masip. Incorporating reinforcement learning for quality-aware sample selection in deep architecture training. In *2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–5. IEEE, 2022. 2

[2] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58:82–115, 2020. 2, 3

[3] Xu Cheng, Zhefan Rao, Yilan Chen, and Quanshi Zhang. Explaining knowledge distillation by quantifying the knowledge. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12925–12935, 2020. 1, 2

[4] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4794–4802, 2019. 1

[5] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. In *International Conference on Machine Learning*, pages 2006–2015. PMLR, 2020. 1

[6] Mohammad Mahdi Dehshibi and Andrew Adamatzky. Electrical activity of fungi: Spikes detection and complexity analysis. *Biosystems*, 203:104373, 2021. 2

[7] Mohammad Mahdi Dehshibi, Mona Ashtari-Majlan, Gereziher Adhane, and David Masip. ADVISE: ADaptive feature relevance and VISual Explanations for convolutional neural networks. *The Visual Computer*, 40(8):5407–5419, 2024. 2, 3

[8] Mohammad Mahdi Dehshibi, Bita Baiani, Gerard Pons, and David Masip. A Deep Multimodal Learning Approach to Perceive Basic Needs of Humans From Instagram Profile. *IEEE Transactions on Affective Computing*, 14(2):944–956, 2023. 2

[9] Mohammad Mahdi Dehshibi, Alessandro Chiolerio, Anna Nikolaidou, Richard Mayne, Antoni Gandia, Mona Ashtari-Majlan, and Andrew Adamatzky. Stimulating Fungi Pleurotus ostreatus with Hydrocortisone. *ACS Biomaterials Science & Engineering*, 7(8):3718–3726, 2021. 2

[10] Mohammad Mahdi Dehshibi and David Masip. BEE-NET: A deep neural network to identify in-the-wild Bodily Expression of Emotions, 2024. 2

[11] Mohammad Mahdi Dehshibi, Temitayo Olugbade, Fernando Diaz-de Maria, Nadia Bianchi-Berthouze, and Ana Tajadura-Jiménez. Pain Level and Pain-Related Behaviour Classification Using GRU-Based Sparsely-Connected RNNs. *IEEE Journal of Selected Topics in Signal Processing*, 17(3):677–688, 2023. 2

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. 2, 5

[13] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *Computer Vision – ECCV 2020*, pages 86–102. Springer International Publishing, 2020. 2

[14] Kshitij Dwivedi and Gemma Roig. Representation similarity analysis for efficient task taxonomy & transfer learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12387–12396, 2019. 2

[15] Jeremy Elson, John (JD) Douceur, Jon Howell, and Jared Saul. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., October 2007. 2, 5, 7

[16] Neda Gholami, Mohammad Mahdi Dehshibi, Andrew Adamatzky, Antonio Rueda-Toicen, Hector Zenil, Mahmood Fazlali, and David Masip. A Novel Method for Reconstructing CT Images in GATE/GEANT4 with Application in Medical Imaging: A Complexity Analysis Approach. *Journal of Information Processing*, 28:161–168, 2020. 2

[17] Jacob Gildenblat and contributors. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam, 2021. 5

[18] Akhilesh Gotmare, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. A closer look at deep learning heuristics: Learning rate restarts, warmup and distillation. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. 2

[19] Jianping Gou, Liyuan Sun, Baosheng Yu, Shaohua Wan, Weihua Ou, and Zhang Yi. Multilevel attention-based sample correlations for knowledge distillation. *IEEE Transactions on Industrial Informatics*, 19(5):7099–7109, 2023. 1

[20] Anselm Haselhoff, Jan Kronenberger, Fabian Kuppers, and Jonas Schneider. Towards Black-Box Explainability with Gaussian Discriminant Knowledge Distillation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 21–28. IEEE, 2021. 2

[21] 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*, pages 770–778, 2016. 5

[22] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *International Conference on Computer Vision (ICCV)*, 2019. 2, 5

[23] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. In *NIPS Deep Learning and epresentation Learning Workshop*, 2015. 5

[24] David P. Hughes and Marcel Salathé . An open access repository of images on plant health to enable the development

of mobile disease diagnostics through machine learning and crowdsourcing. *CoRR*, abs/1511.08060, 2015. 2, 5

[25] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174. Association for Computational Linguistics, 2020. 2

[26] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 2

[27] Nikolaus Kriegeskorte and Pamela K. Douglas. Cognitive computational neuroscience. *Nature Neuroscience*, 21(9):1148–1160, 2018. 2

[28] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada, 2009. 2, 5, 7

[29] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 5191–5198, 2020. 1, 7

[30] Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. What is being transferred in transfer learning? In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 512–523. Curran Associates, Inc., 2020. 2

[31] Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 35, pages 13657–13665, 2021. 1, 5

[32] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017. 5

[33] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 12116–12128. Curran Associates, Inc., 2021. 2

[34] Yao Rong, Tobias Leemann, Vadim Borisov, Gjergji Kasneci, and Enkelejda Kasneci. A consistent and efficient evaluation strategy for attribution methods. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18770–18795. PMLR, 17–23 Jul 2022. 4

[35] 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*, pages 618–626, 2017. 1, 3

[36] Het Shah, Avishree Khare, Neelay Shah, and Khizir Siddiqui. Kd-lib: A pytorch library for knowledge distillation, pruning and quantization, 2020. 5

[37] Wonchul Son, Jaemin Na, Junyong Choi, and Wonjun Hwang. Densely guided knowledge distillation using multiple teacher assistants. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9395–9404, October 2021. 1, 7

[38] Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. Does knowledge distillation really work? In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 6906–6919. Curran Associates, Inc., 2021. 1, 7

[39] Gábor J Székely and Maria L Rizzo. Partial distance correlation with methods for dissimilarities. *The Annals of Statistics*, 42(6):2382–2412, 2014. 3

[40] Junpeng Wang, Liang Gou, Wei Zhang, Hao Yang, and Han-Wei Shen. DeepVID: Deep Visual Interpretation and Diagnosis for Image Classifiers via Knowledge Distillation. *IEEE Transactions on Visualization and Computer Graphics*, 25(6):2168–2180, 2019. 1, 2

[41] Haiguang Wen, Junxing Shi, Yizhen Zhang, Kun-Han Lu, Jiayue Cao, and Zhongming Liu. Neural encoding and decoding with deep learning for dynamic natural vision. *Cerebral cortex*, 28(12):4136–4160, 2018. 2

[42] Mengqi Xue, Jie Song, Xinchao Wang, Ying Chen, Xingen Wang, and Mingli Song. Kdexplainer: A task-oriented attention model for explaining knowledge distillation. In *International Joint Conference on Artificial Intelligence*, 2021. 2

[43] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017. 1

[44] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018. 1

[45] Xingjian Zhen, Zihang Meng, Rudrasis Chakraborty, and Vikas Singh. On the versatile uses of partial distance correlation in deep learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 327–346. Springer, 2022. 2, 3, 5