Can we induce desirable properties into models via Cross-Architecture Knowledge Distillation

Muhammad Huraira Anwer
Department of Computer Science, LUMS
25100314@lums.edu.pk

Adeen Ali Khan
Department of Computer Science, LUMS
25100154@lums.edu.pk

Abstract

This project explores knowledge distillation between heterogeneous architectures, focusing on transferring knowledge from attention-based models, such as Vision Transformers (ViT), to convolutional models, such as VGG or ResNets, and vice-versa. While traditional knowledge distillation techniques have predominantly been applied within homogeneous architectures, the challenges and potential insights of heterogeneous distillation remain underexplored. Using different distillation approaches,this work aims to address three key objectives. First, it investigates whether architectural biases, such as shape, texture, and locality biases are effectively distilled from teacher model, or do they exist as they normally would in the student model. Finally, the project evaluates if the permutation invariance property of teacher (transformer) is distilled into the student or not. Expected outcomes include partial transfer of biases, varying levels of attention alignment depending on the distillation method, and enhanced robustness to noisy or corrupted data for students distilled using CRD. This research aims to provide deeper insights into the interplay between architectural differences and distillation techniques, offering guidance for designing more robust and efficient student models across diverse architectures.

1 Introduction

Knowledge distillation has emerged as a powerful technique for transferring knowledge from a large, complex teacher model to a smaller, more efficient student model, enabling practical deployment of deep learning models in resource-constrained environments. However, most existing approaches assume homogeneity between teacher and student architectures, limiting their applicability to scenarios where architectural diversity is essential. This project addresses the challenge of knowledge distillation across heterogeneous architectures, specifically from attention-based models like Vision Transformers (ViT) to convolutional models like Resnet, and from ResNet back to ViT. This problem is particularly significant given the growing interest in combining the complementary strengths of attention and convolutional paradigms to achieve more robust and versatile models.

Despite the success of homogeneous distillation methods, the interplay between fundamentally different inductive biases, such as the global attention mechanisms in ViT and the localized feature extraction of convolutional networks, remains underexplored. Previous studies have shown that attention-based models exhibit a stronger shape bias and global reasoning capabilities, while convolutional networks tend to focus on texture information and local features. This discrepancy raises important questions about the extent to which these biases can be distilled and the effectiveness of existing distillation methods in bridging this gap. Furthermore, the robustness of distilled models to input noise and corruptions is an ongoing challenge, particularly in safety-critical applications.

This project aims to investigate these challenges through three complementary objectives. First, it will explore whether architectural biases, such as shape, texture, and locality biases can be effectively transferred across heterogeneous architectures. Finally, it will evaluate the robustness of the student

model (when distilling CNN from ViT) under adversarial conditions and corruptions, to see if it is permutation invariant or not, providing insights into the generalization capabilities of distilled models.

To achieve these objectives, the project will employ various knowledge distillation techniques, including Logit Matching [Hinton, 2015] and Contrastive Representation Distillation (CRD) [Tian et al., 2019] for both CNN-to-ViT and ViT-to-CNN. Moreover, due to architectural differences, we use 4 separate methods (2 for each hetergeoneous model pair distillation). For CNN-to-Vit, we employ Cumulative Spatial Knowledge Distillation (CSKD) [Zhao et al., 2023] and Visual-Linguistic Feature Knowledge distillation (VLFKD) [Zheng et al., 2023]. For Vit-to-CNN, we utilize 2 other methods namely Self Supervised Knowledge Distillation (SSKD) [Xu et al., 2020] and Cross Architecture Projection Distillation (CAPD) [Liu et al., 2022].

These methods have been chosen for their complementary strengths in aligning output distributions, intermediate feature representations, and learned representations in the latent space, respectively. Anticipated outcomes include improved understanding of the transferability of biases, deeper insights into attention alignment, and enhanced robustness for student models. By addressing these research questions, this work seeks to advance the state of knowledge distillation, offering a framework for effective transfer learning across heterogeneous architectures with potential applications in real-world scenarios requiring lightweight and robust models.

2 Related Work

Knowledge distillation (KD) has emerged as a powerful technique for transferring knowledge from a larger, high-capacity teacher model to a smaller, more efficient student model. [Hinton, 2015] first formalized KD, demonstrating its efficacy in compressing neural networks by aligning the logits of the teacher and student. This foundational work spurred numerous advancements in KD methods, including FitNets [Romero et al., 2014], which introduced intermediate layer supervision through hints, and Contrastive Representation Distillation (CRD) [Tian et al., 2019], which leverages contrastive learning to enhance the representation quality of student models. These approaches have laid the groundwork for applying KD across diverse architectures, including convolutional and attention-based models.

Vision Transformers (ViTs) [Dosovitskiy et al., 2020] have revolutionized computer vision by leveraging self-attention mechanisms [Vaswani et al., 2023] to capture global dependencies in images, showing state-of-the-art performance on tasks traditionally dominated by convolutional neural networks (CNNs) [O'Shea and Nash, 2015]. Subsequent research highlighted the unique inductive biases of ViTs, such as global attention and shape bias, contrasting with the local receptive fields and texture bias inherent to CNNs. While ViTs excel in robustness and generalization, their computational cost limits their practicality in many applications, motivating efforts to distill their strengths into more efficient architectures like CNNs.

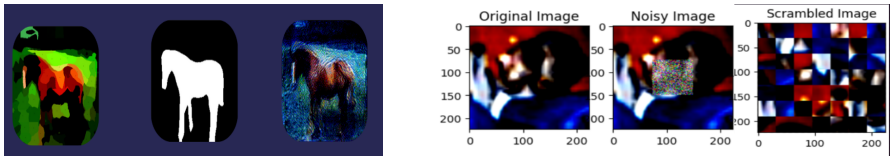
Despite significant progress in KD, existing studies primarily focus on homogenous architectures, such as transferring knowledge between two CNNs or two Transformers. Heterogeneous KD, particularly from ViTs to CNNs, remains vastly underexplored. Prior attempts in this area have often been limited by simplistic methods or inadequate evaluation criteria. Some works exist, but they are either restricted to specific usecases, like LLMs [Ralambomihanta et al., 2024] and Monocular Depth Estimation [Zheng et al., 2024] not image classification or they are limited to only logit matching distillation approach [Ahmadabadi et al., 2023]. In contrast, [Liu et al., 2022] propose a novel cross-architecture knowledge distillation method between Transformer and CNN models for image classification which, while effective, does not evaluate or test the properties and biases whose distillation we aim to explore in our work.

Moreover, recent studies have begun investigating architectural biases and invariances. [Geirhos et al., 2018] revealed the texture bias of CNNs compared to the shape bias of human vision, while [Naseer et al., 2021]. demonstrated the shape bias inherent in ViTs. However, it is unclear whether such biases can be effectively transferred across architectures through KD. Additionally, visualization techniques, such as Grad-CAM [Selvaraju et al., 2019], have been widely used to interpret CNNs (and the model's focus on local areas) but are less frequently applied in the context of ViTs.

This project aims to address these gaps by exploring the potential of heterogeneous KD to transfer not only predictive accuracy but also architectural biases and robustness characteristics from ViTs to CNNs. By leveraging 6 different KD methods, this work seeks to advance the understanding of cross-architecture KD while providing insights into shape and texture bias transfer, attention alignment, and robustness under noise and corruptions. In doing so, it positions itself as a significant step toward unifying the strengths of attention-based and convolution-based models in real-world applications.

3 Methodology

3.1 Datasets and Architectures



(a) SuperPixelated, Segmented, Stylized Images (b) Noisy and Scrambled Images

Figure 1: Example images of how the datasets were prepared. a) shows superpixelated, segmented , style transferred versions of the same image. b) shows how we are adding localised noise to the image, and how we are randomly rearranging patches of the image to create new images.

The CIFAR-10 test-set is manipulated using different techniques to create 5 different testsets(of 10,000 images each). The motivation to create all these datasets is to evaluate the trained students and independent student under the different distillation mechanisms, to get their accuracies in these extreme conditions, which will then be used to compute a ratio. The ratio would be a bias value for shape and texture and super-pixelated datasets i.e. shape bias = accuracy in shapes testset / accuracy in original testset. For the noisy and scrambled datasets, this ratio will be representing a measure of permutation invariance of the respective models, which will be used for further evaluation, and deriving conclusions.

A Fully Convolutional Network (FCN) pretrained for semantic segmentation i.e. ResNet-101 is used to isolate object shapes. Moreover, a pre-trained VGG-19 model is employed to synthesize textures, which are then overlaid onto the segmented object regions, ensuring that local features dominate the resulting images. These datasets provide a controlled setting to evaluate the biases inherent to each model (specifically, texture and shape biases) and the effectiveness of KD in transferring these biases.

Moreover, we created another dataset using the SLIC algorithm [Achanta et al., 2012] for efficient superpixel segmentation. It basically generates superpixels by clustering pixels based on their color similarity and proximity in the image plane, hence focusing on the local regions of the image. It kind of smoothens the image, and blurs out various global details, hence would serve as a benchmark dataset to evaluate if locality bias (which is very strong in CNNs, is it transferred to the ViT or not).

To evaluate the robustness of the teacher and student models under challenging conditions, two additional datasets were created: a noisy dataset and a permuted dataset, each designed to test specific aspects of model generalization. For the noisy dataset, Gaussian noise was added to a small, fixed region (a 32×32 pixel patch) of each image in the CIFAR-10 dataset. The same noise pattern was consistently applied across different images at the same location, ensuring a controlled testing environment. This setup simulates real-world scenarios where localized regions of an image might be corrupted or obscured, such as in low-quality or damaged photographs. By isolating the effects of noise to a specific region, we aim to assess how the inductive biases of Vision Transformers and Convolutional Networks influence their ability to handle such perturbations.

In the permuted dataset, images were divided into smaller patches, and their arrangement was randomly scrambled to investigate the impact of global structural distortions. This process disrupts the spatial coherence of the original image, requiring the models to rely on global reasoning to classify the images accurately. The task tests the models' capacity to capture the overall structure of an image, rather than relying solely on local cues. Both datasets provide a unique perspective on how

well the teacher and student architectures can handle distortions in spatial and structural information, highlighting the differences in their underlying inductive biases and generalization capabilities.

We are using pretrained *vit_base_patch16_224* and *resnet50* models to start with, which are then further trained depending on the method. The models are implemented using PyTorch. All models are trained on CIFAR-10 train-set, with the teacher finetuned for **1** epoch only, while the student is trained for **3** epochs in all cases. The teacher is trained for less epochs as starts to overfit and get **100%** accuracies by 2-3 epochs These settings are to ensure fair comparison while having a limited compute.

3.2 CNN to ViT Knowledge Distillation

The distillation of knowledge from a CNN teacher to a ViT student is not an easy task considering not only the architectural differences between the two but also the ways in which they process data, the number of features, and the features and representations that these models learn differ a lot.The following are some of the distillation methods that we employed to distill knowledge despite these various challenges:

3.2.1 Visual-Linguistic Feature Knowledge distillation (VLFKD)

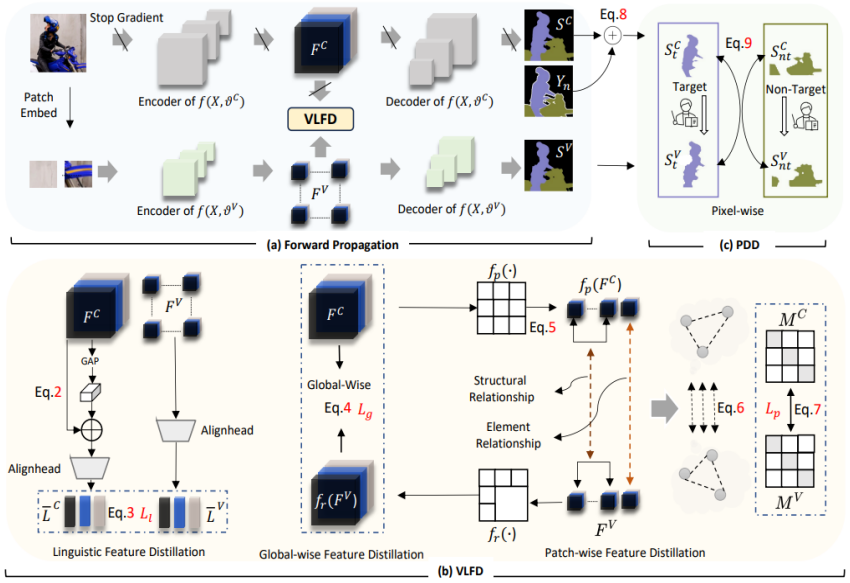


Figure 2: VLFKD block diagram

VLFKD [Zheng et al., 2023] addresses the challenges of distilling knowledge from convolutional models to vision transformers by employing several feature alignment methods, namely global, patch-level, and linguistic feature alignment with pixel-wise distillation.

To adapt the model for distillation, the classification head is replaced with a new fully connected layer matching the number of classes in CIFAR-10. Furthermore, the ViT is used in two forms: a full model for predictions and a truncated feature model for patch-level feature extraction.

Loss Functions

The overall loss function for VLFKD combines several components to align both global and patch-level features between the teacher and student models. The cumulative loss is defined as:

$$L_{\text{total}} = \lambda_g L_{\text{global}} + \lambda_p L_{\text{patch}} + \lambda_l L_{\text{linguistic}} + \lambda_d L_{\text{pixel-decoupled}} \tag{1}$$

where $\lambda_g, \lambda_p, \lambda_l, \lambda_d$: Weighting coefficients for each loss term.

Global Feature Alignment Loss (L_{global})

As CNN features summarize the entire image, while ViTs split images into patches, Aligning these global features is crucial to ensure that the student understands the overall context of the image. This is done by combining all patch-level features from the ViT student to create a global representation and then using KL-Divergence to minimize the difference between the teacher's global features and the rebuilt student features.

This loss ensures the student's global feature representation aligns with the teacher's global feature representation:

$$L_{\text{global}} = \text{KL}\left(\text{Softmax}(\text{Projteacher}(\text{GAP}(F\text{teacher}))) \parallel \text{Softmax}(\text{GAP}(F_{\text{student}}))\right) \tag{2}$$

Here, GAP is Global Average Pooling, and $\text{Proj}_{\text{teacher}}$ projects teacher features to match the student's feature dimensionality.

Patch-Level Feature Alignment Loss (L_{patch})

Since ViTs process images as patches, it's critical to ensure that features for individual patches and relationships between patches are correctly aligned. This is done by dividing the teacher's global features into smaller pieces that correspond to the student's patches and aligning them.

This loss aligns individual patches between the teacher and student models using Mean Squared Error (MSE):

$$L_{\text{patch}} = \text{MSE}\left(\text{Projteacher}(F\text{teacher}), \text{Resized}(F_{\text{student}})\right) \tag{3}$$

Here, F_{teacher} and F_{student} are teacher and student patch features, and resizing ensures spatial dimensions match.

Linguistic Alignment Loss ($L_{\text{linguistic}}$)

Vision Transformers (ViTs) treat image patches like words in language processing. Their features behave like linguistic tokens while CNNs don't process data this way, so we need to align CNN features with the ViT's linguistic-like features. We do this by extracting the final-layer features of the CNN teacher, which represent the whole image and then to summarize overall context we use a global average pooling (GAP) layer on the teacher's features followed by passing through self attention layer to make them linguistically compatible.

This loss ensures alignment in linguistic feature space by projecting both teacher and student patch embeddings and minimizing KL divergence:

$$L_{\text{linguistic}} = \text{KL}\left(\text{Softmax}(\text{Projteacher}(F\text{teacher-patch})) \parallel \text{Softmax}(\text{Projstudent}(F\text{student-patch}))\right) \tag{4}$$

Pixel-Wise Decoupled Distillation Loss ($L_{\text{pixel-decoupled}}$)

This focuses on correcting teacher errors and ensuring the student learns robustly, even if the teacher occasionally makes wrong predictions. Simply copying the teacher's wrong predictions would transfer these mistakes to the student, hurting its performance.

This loss separates target and non-target class predictions, aligning the teacher's and student's predictions for each:

$$L_{\text{pixel-decoupled}} = \text{MSE}(S_{\text{target}}, T_{\text{target}}) + \text{MSE}(S_{\text{non-target}}, T_{\text{non-target}}) \tag{5}$$

3.2.2 Cumulative spatial knowledge distillation (CSKD)

Unlike the previous method where a lot of feature alignment happens, CSKD [Zhao et al., 2023] aims to teach ViTs to understand spatial information (like local image regions) without requiring intermediate feature alignment between CNNs and ViTs. The student model (ViT), is extended to include dense outputs in addition to the standard global logits.

The architecture consists of two key components, namely Global Logits nad Dense Logits. Global logits are derived from the class token (CLS) and provide the high-level image classification prediction. Dense Logits are generated from the patch tokens of the ViT, corresponding to sub-regions of the

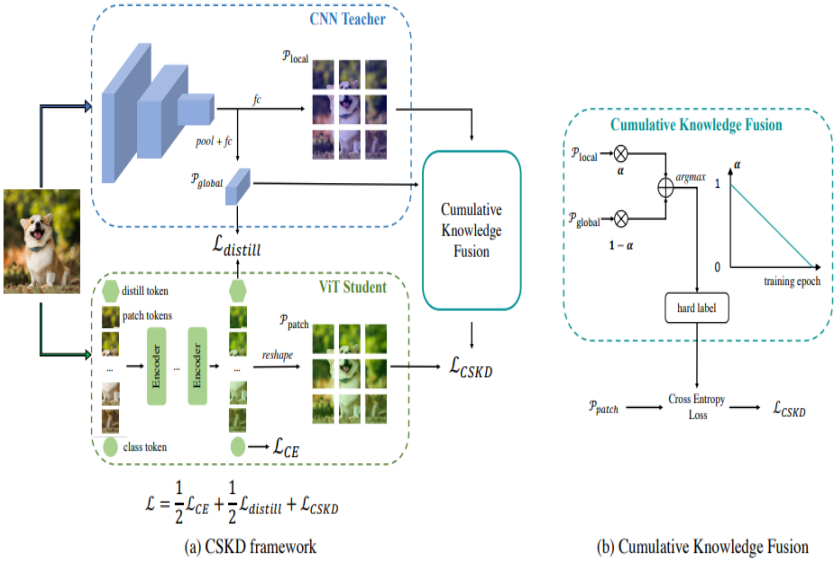


Figure 3: CSKD block diagram

input image, providing spatial predictions. The teacher model (Resnet) also outputs both dense logits (for each spatial position in the image) and global logits (for the entire image).

The student model (ViT) is trained to predict both global and dense logits. Initially, it focuses on local spatial knowledge from the teacher's dense logits, and later, it shifts its focus to the global logits for a more holistic image understanding. To control the shift from local to global learning, a decay ratio α is computed, which gradually decreases over the course of training:

$$\text{Decay Ratio} = 1 - \left(\frac{\text{epoch}}{\text{max epoch}}\right)$$

The decay ratio can be adjusted using different functions, such as linear decay, quadratic decay, or cosine decay. This ensures that during the later stages of training, the student model focuses more on the global context.

Loss Functions

The total loss for the CSKD framework is a weighted combination of the base classification loss, DeiT distillation loss, and CSKD loss:

$$L_{\text{total}} = (1-\alpha) \cdot L_{\text{base}} + \alpha \cdot L_{\text{deit}} + \gamma \cdot L_{\text{cskd}} \tag{6}$$

where α is a hyperparameter controlling the balance between the base classification loss and the DeiT distillation loss and γ is a hyperparameter controlling the weight of the CSKD loss in the final total loss.

Base Classification Loss (L_{base})

The base classification loss is computed using the standard cross-entropy loss:

$$L_{\text{base}} = -\sum_{i} \text{labels}_i \log(\text{outputs}_i) \tag{7}$$

where 'labels' are the true class labels and 'outputs' are the model's predictions.

DeiT Distillation Loss (L_{deit})

The DeiT distillation loss ensures that the student learns from the teacher's logits. There are two options for this loss: soft targets using KL-Divergence or hard targets using cross-entropy.

using Soft targets (using KL-Divergence):

$$L_{\text{deit}} = \text{KL}(\text{softmax}(\text{student logits})||\text{softmax}(\text{teacher logits}))$$

using Hard targets (using cross-entropy):

$$L_{\text{deit}} = \text{CrossEntropy}(\text{student logits}, \text{argmax}(\text{teacher logits}))$$

CSKD Loss (L_{cskd})

The CSKD loss is computed by aligning the student's dense logits with the teacher's dense logits using a weighted combination of dense and global logits.

$$y^T = \text{argmax}_c \left(\alpha p^T_{\text{global}} + (1-\alpha) p^T_{\text{global}}\right) \tag{8}$$

where α is a coefficient that negatively correlates with the training duration (e.g., $\alpha = 1 - t/t_{\text{max}}$, where t denotes the current epoch and t_{max} denotes the number of total epochs).

Thus, the loss function for our CSKD can be written as:

$$L_{\text{CSKD}} = \text{CE}(p_{\text{S}}, y^T) \tag{9}$$

3.3 ViT to CNN Knowledge Distillation

Due to architectural differences, and since the previous methods (CNN-to-ViT) were heavily focused on aligning the features of the teacher-student in a certain manner, those methods weren't applicable for ViT-to-CNN. Hence, we explored two other methods that deal with this specific setting.

3.3.1 Cross Architecture Projection distillation (CAPD)

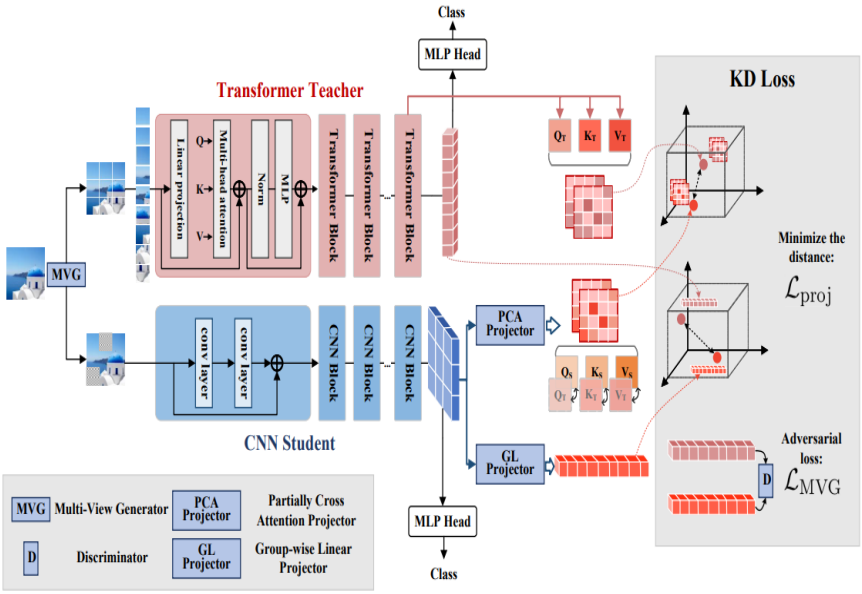


Figure 4: CAPD block diagram

CAPD [Liu et al., 2022] prevents the student from directly mimicking the output/intermediate features of the teacher, instead it uses 2 projectors. First one maps student features to attention space, and then

aligns that with teacher attention space. The other one maps student features into teacher's feature space and aligns them with teacher's features.

Moreover, to improve the robustness and stability of the student, a cross-view robust training scheme is proposed using adversarial learning. Multi-view samples are generated by a multi-view generator which randomly conducts some transformations and generates mask and noise adding to the inputs. Fed with the multi-view inputs, the student generates different features. A multi-view adversarial discriminator is constructed to distinguish the teacher features and the student features in the transformer feature space. Then the goal is to puzzle the discriminator, in doing so, the student's features get as close as possible to teacher's features.

Loss Functions

The overall loss function for the student in CAPD combines the projection losses as well as the multi-view generator (MVG) loss. The cumulative loss is defined as:

$$L_{\text{total}} = L_{\text{proj1}} + L_{\text{proj2}} + \lambda L_{\text{MVG}} \tag{10}$$

where λ is a regularizer.

Moreover, the loss for the multi-adversarial discriminator (MAD) is defined as follows:

$$\mathcal{L}_{MAD} = \frac{1}{m}\sum_{k=1}^{m}\left[-\log D(h_T^{(k)}) - \log\left(1 - D(h_S^{(k)})\right)\right]. \tag{11}$$

Projection Loss 1 (L_{proj1})

The Q,K,V matrices are computed for the student as well, in order to map it into the attention space. Then, loss of the Partially cross attention (PCA) projector is computed as the Mean Squared Error of aligned teacher-student attentions and teacher-student *value* (V) matrices:

$$\mathcal{L}_{proj1} = \|\text{Attn}_T - \text{PCAttn}_S\|_2^2 + \left\|\frac{V_T \cdot V_T}{\sqrt{d}} - \frac{V_S \cdot V_S}{\sqrt{d}}\right\|_2^2. \tag{12}$$

Projection Loss 2 (L_{proj2})

A pixel-by-pixel mapping is done from student feature space to the transformer's feature space, using fully connected layers which are shared over a 4 X 4 neighbourhood. Then, loss of the Group-wise Linear (GWL) projector is computed as the Mean Squared Error of aligned teacher-student features:

$$\mathcal{L}_{proj2} = \|h_T - h_S\|_2^2. \tag{13}$$

Multi View Generator Loss (L_{MVG})

The MVG assigns with probability *p* (sampled uniformly) either the same image, or a transformed version of the image where the transformed version includes common transformations like color jitter, random crop, rotation, patch-wise mask. This encourages the student (generator) to confuse the discriminator from classifying it correctly as teacher's feature or student's feature, hence obtaining a robust student feature.

$$\mathcal{L}_{MVG} = \frac{1}{m}\sum_{k=1}^{m}\log\left(1 - D(h_S^{(k)})\right)]. \tag{14}$$

3.3.2 Self Supervised Knowledge distillation (SSKD)

What SSKD [Xu et al., 2020] does, is that, after finetuning the teacher to the new dataset, it wraps it with a pretext task head which is a 2-layer MLP, whose goal is to map the feature representations of teacher into a latent space, on which contrastive loss is computed (as self-supervised task). In the

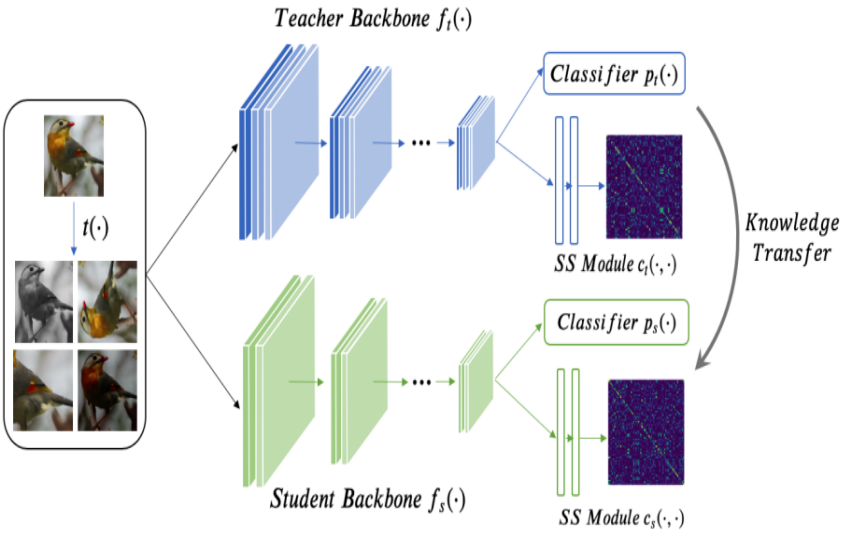


Figure 5: SSKD block diagram

contrastive loss, the positive pairs are an image x_i and its corresponding transformed image $\tilde{x_i}$, and negative pairs are image x_i and some other transformed image $\tilde{x_j}$ where j $\neq$ i. It is important to note, that the goal is not to increase training samples for teacher, because if done so, it may result in teacher losing the semantic representation of the original data, hence classification loss is not computed on the transformed data.

After training the teacher, we then distill this knowledge from the teacher to the student by encouraging it to mimick not only the teacher's classification outputs but also the teacher's self supervision output, in addition to the standard cross entropy loss. The self-supervised task helps the teacher learn a richer representation of the semantics of the data, resulting in effective transfer of this hidden information from the teacher to the student.

Loss Functions

The overall loss function for the student in SSKD combines the KL divergence losses of classification (as used in Logit Matching), KL divergence of self-supervised outputs, in addition to the cross entropy loss. The cumulative loss is defined as:

$$L = \lambda_1 \mathcal{L}_{ce} + \lambda_2 \mathcal{L}_{kd} + \lambda_3 \mathcal{L}_{ss} + \lambda_4 \mathcal{L}_T, \tag{15}$$

KL divergence of Classification Outputs ($L_{\mathbf{kd}} + L_{\mathbf{T}}$)

There are two terms, where the first one, L_{kd} represents logit matching loss function of logits of teacher, student on the normal data x_i, with temperature T. The second term L_{T} represents logit matching loss function of logits of teacher, student on the transformed data $\tilde{x_i}$, with temperature T. Even if the teacher is sometimes predicting wrongly for the transformed samples, it is still useful to distil this knowledge to the student. However, in order, to avoid huge amounts of incorrect knowledge transfer, only top k % of incorrect predictions are distilled, in addition to all correct predictions.

KL divergence of Self Supervised Outputs ($L_{\mathbf{ss}}$)

From the similarity matrices A^t and A^s, softmax is applied with temperature τ to obtain probability matrices B^t and B^s where $B_{i,j}$ represents the probability of $\tilde{x_i}$ and x_j being a positive pair. The following loss function is then applied on these matrices, that encourages the student's self-supervised task outputs to be as close as possible to the teacher.

$$\mathcal{L}_{ss} = -\tau^2 \sum_{i,j} B^t_{i,j} \log(B^s_{i,j}). \tag{16}$$

4 Experimental Design

The experiments are designed to assess the effectiveness of knowledge distillation in transferring properties like shape bias, locality bias, permutation invariance between heterogeneous architectures. They are driven by two primary research questions. First, to what extent can architectural biases, such as shape and texture bias, be effectively transferred between the models using different knowledge distillation techniques? And how robust are the teacher and student models to localized noise and spatial distortions introduced in the noisy and permuted datasets, respectively?

The evaluation of the models employs several metrics to provide a comprehensive analysis. Classification accuracy is measured on the all the datasets to assess the transfer of biases from the teacher to the student. Moreover, bias values are computed for each dataset, and are reported. Furthermore, runtimes of the models were also measured for analyzing and comparing model efficiencies. In all scenarios, the independent student model serves as a baseline comparison with all distilled models.

5 Results and Findings

5.1 CNN-to-ViT Results

Method	Texture Bias	Shape Bias	Locality Bias	Scramble Bias	Noise Bias
lm	0.3674	0.1759	**0.4124**	0.2519	0.6631
crd	0.4037	0.1585	0.3506	0.2155	0.6991
vlfkd	0.3493	**0.1806**	0.3583	**0.2770**	0.6400
cskd	**0.4241**	0.1779	0.3976	0.2279	**0.7340**
independent	0.4044	0.1757	0.3506	0.2301	0.7066

Table 1: Bias Values in ResNet (Teacher) to Vit (Student) Distillation

Table 1 presents the bias values for different knowledge distillation methods such as LM, CRD, VLFKD, CSKD, and Independent (control) across several bias types, including Texture Bias, Shape Bias, Locality Bias, Scramble Bias and Noise Bias. Scramble Bias and Noise Bias are not really bias values, rather they depict the permutation invariance property.

The results indicate that CSKD excels in Texture Bias **(0.4241)** and Noise Bias **(0.7340)**, suggesting that Cumulative Spatial Knowledge Distillation is particularly effective at transferring texture and noise-related information, which are essential for accurate predictions, especially in noisy environments.

In contrast, Shape Bias and Scramble Bias are less relevant for distillation, as these features are typically better learned directly by the model during training. The Shape Bias value of CSKD **(0.1779)** supports this, showing minimal focus on shape-based features. Similarly, Scramble Bias reflects that models do not rely on scrambled patterns for learning, as they are not useful for meaningful visual understanding.

From the results, we conclude that CSKD is superior for transferring relevant spatial and texture information, and that distillation should prioritize Texture Bias and Noise Bias while downplaying irrelevant biases like Shape Bias and Scramble Bias.

Method	Time per Epoch (seconds)
Logit Matching	815.8
CRD	1748
VLFKD	1604.2
CSKD	756

Table 2: Distillation Time Per Epoch for Different Methods

We also did some analysis for the computation time required for 1 epoch of these different distillation methods. The results are shown in 2. There are significant differences in the computation time required for CSKD versus CRD and VLFKD. CSKD requires the least amount of time while CRD and VLFKD take more than twice that time. This comparison further backs CSKD as not only a better knowledge distillation method but also an efficient one.

5.2 ViT-to-CNN Results

Method	Texture Bias	Shape Bias	Locality Bias	Scramble Bias	Noise Bias
lm	0.2796	0.1056	0.1010	0.1061	0.4055
crd	**0.717**	**0.679**	**0.665**	**0.586**	0.724
capd	0.99	0.99	0.997	0.997	0.99
sskd	0.596	0.589	0.583	0.574	**0.742**
independent	0.352	0.1217	0.125	0.15	0.48

Table 3: Bias Values in ViT (Teacher) to ResNet (Student) Distillation

Table 3 presents the bias values for different knowledge distillation methods such as LM, CRD, CAPD, SSKD, and Independent (control) across several bias types, including Texture Bias, Shape Bias, Locality Bias, Scramble Bias and Noise Bias. Scramble Bias and Noise Bias are not really bias values, rather they depict the permutation invariance property.

Firstly, to interpret, that 0.99 and close to 1 bias values in CAPD is not because of their extremely good accuracy, however , it depicts that it has more or less the same accuracy in all test sets which is **10%** which is equal to random guessing since the dataset has 10 classes. This is due to 1 epoch setting was not enough for it to learn anything meaningful at all, considering the adversarial setup and the feature alignment loss functions.

Now, the results indicate that CRD excels in distilling Shape Bias while SSKD is either close to or better than CRD in distilling Permutation Invariance (through Noise / Scrambled Bias), suggesting that both these methods are highly effective at transferring these properties from the ViT to the student network. Moreover, the comparison to the independent student's bias values highlight this difference even more, as it always almost doubles the bias value (as compared to the student), sometimes even more.

In contrast, Texture Bias and Locality Bias are less relevant for distillation, as these features are typically better learned directly by the model during training. The higher values of these than independent student are attributed to the more number of epochs given to training the distilled students than the independent student which was only finetuned for 1-2 epochs.

From the results, we conclude that CRD is superior for transferring relevant shape bias and permutation invariance information to the students as visible by the stark gap with baseline model comparison, while a close second is SSKD.

Method	Time per Epoch (seconds)
LM	604.2
CRD	919.3
CAPD	2513.5
SSKD	3595.2

Table 4: Distillation Time Per Epoch for Different Methods

We also did some analysis for the computation time required for 1 epoch of these different distillation methods. The results are shown in 4. There are significant differences in the computation time required for CAPD and SSKD versus CRD and LM. The former 2 methods take almost 45 mins to an hour, on one epoch, while the latter 2 methods finish their epochs within 10-15 mins.

5.3 Insights and Implications

For the CNN-to-ViT distillations, it is evident that CSKD excels in distilling Texture Bias into the students. It distills Noise Bias however, which isnt a more desired property of the CNN architecture,

and it does not distill Locality Bias that well either (Logit Matching does it better). So even though, it is an efficient distillation method, it isn't an entirely desirable one.

Moreover, since the biases being distilled are also quite weak considering their difference from the baseline comparison (independent student), it isn't a quite significant increase in the bias, so the knowledge is being distilled at quite a low rate (in the given compute power of less than 5 epochs atleast).

On the other hand, for the ViT-to-CNN distillations, it is quite clear that CRD is both efficient and effective in distilling all the strong and required inductive biases (both shape bias and permutation invariance property) into the student, and it's distillation effect is quite significant, in terms of the difference with the baseline comparison.

6 Conclusion

To conclude, I would mention that even though CNN-to-ViT distillation is effective at some level while being efficient as well, in distilling the texture bias. However, it doesn't distil other desirable properties like locality bias, and it can be labelled as a 'poor' teacher in terms of distilling these properties as the comparison with independent model doesn't highlight a stark steering of the bias in one direction. More work can be done on this, by exploring this approach for way more epochs until convergence is observed, as well as experimenting with a broader range of datasets, architecture pairs, different distillation methods would result in a better conclusion.

However, the ViT-to-CNN is quite effective and efficient in distilling the desired properties of a transformer into a CNN. That being said, one thing to be taken into consideration is, if run for more epochs, what if it is at the same distilling the undesirable properties of a transformer (low texture bias and low locality bias) into a CNN, to work on strategies to prevent that for happening, since we want the student to keep its share of good biases, and learn some biases from the teacher as well, in order to become an all-round player. Again, to cement this conclusion, extensive experimentation needs to be done.

References

Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 34, pages 2274–2282, 2012. URL `http://www.kev-smith.com/papers/SLIC_Superpixels.pdf`.

Hamid Ahmadabadi, Omid Nejati Manzari, and Ahmad Ayatollahi. Distilling knowledge from cnn-transformer models for enhanced human action recognition. *arXiv preprint*, 2023. URL `https://arxiv.org/abs/2311.01283`. Subject: Computer Vision and Pattern Recognition (cs.CV).

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Matthias Dehghani, Georg Minderer, Sylvain Heigold, Jakob Uszkoreit, et al. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL `https://arxiv.org/abs/2010.11929`.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018. URL `https://arxiv.org/abs/1811.12231`.

Vinyals O. Dean J. Hinton, G. Distilling the knowledge in a neural network, 2015. URL `https://arxiv.org/abs/1503.02531`.

Yufan Liu, Jiajiong Cao, Bing Li, Weiming Hu, Jingting Ding, and Liang Li. Cross-architecture knowledge distillation. *arXiv preprint*, 2022. URL `https://arxiv.org/abs/2207.05273`. Accepted by ACCV 2022, Subject: Computer Vision and Pattern Recognition (cs.CV).

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*, 2021. URL `https://arxiv.org/abs/2105.10497`. NeurIPS'21 (Spotlight).

Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint*, 2015. URL https://arxiv.org/abs/1511.08458. 10 pages, 5 figures.

Tokiniaina Raharison Ralambomihanta, Shahrad Mohammadzadeh, Mohammad Sami Nur Islam, Wassim Jabbour, and Laurence Liang. Scavenging hyena: Distilling transformers into long convolution models. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2401.17574. 9 pages, 2 figures.

Adriana Romero, Nicolas Ballas, Samuel E. Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2014. URL https://arxiv.org/abs/1412.6550.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision (IJCV)*, 2019. URL https://arxiv.org/abs/1610.02391. A previous version was presented at ICCV'17.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation, 2019. URL https://arxiv.org/abs/1910.10699.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Guodong Xu, Ziwei Liu, Xiaoxiao Li, and Chen Change Loy. Knowledge distillation meets self-supervision. *arXiv preprint arXiv:2006.07114*, 2020. URL https://arxiv.org/abs/2006.07114. ECCV'20.

Borui Zhao, Renjie Song, and Jiajun Liang. Cumulative spatial knowledge distillation for vision transformers, 2023. URL https://arxiv.org/abs/2307.08500. Accepted by ICCV 2023.

Xu Zheng, Yunhao Luo, Pengyuan Zhou, and Lin Wang. Distilling efficient vision transformers from cnns for semantic segmentation, 2023. URL https://arxiv.org/abs/2310.07265.

Zhimeng Zheng, Tao Huang, Gongsheng Li, and Zuyi Wang. Promoting cnns with cross-architecture knowledge distillation for efficient monocular depth estimation. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2404.16386. Subject: Computer Vision and Pattern Recognition (cs.CV).