A Two-Stage Distillation Framework for Mitigating Shortcut Learning via Decoupled Representation and Classifier Training

Alessandro Di Frenna
alessandro.difrenna@studenti.unipd.it

Margarita Shnaider
margarita.shnaider@studenti.unipd.it

Abstract

Knowledge distillation is a common technique for model compression, but student models can inherit shortcuts from their teachers. While methods like Contrastive Representation Distillation (CRD) align feature spaces, the learning process often remains entangled with the final classification objective. This paper explores a two-stage distillation strategy that decouples representation learning from classifier training to analyze its effects on generalization. In the first phase, we use CRD to align hierarchical feature representations between a teacher and student model without training the student's classifier. In the second phase, the feature extractor is frozen, and the classifier is trained independently using Kullback-Leibler (KL) divergence on the teacher's soft logits.

We evaluate this modular approach on the CIFAR-100 dataset using various ResNet architectures. Our experiments show that this decoupled strategy is effective, yielding improved generalization and stable training. A statistical analysis of the distilled model's predictions reveals distinct error patterns compared to the teacher, suggesting that the student learns different inductive biases. These findings indicate that separating feature and classifier learning is a viable and robust framework for knowledge distillation, offering a structured alternative to end-to-end training.

1. Introduction

Deep learning models, given their high parametric capacity, are particularly susceptible to shortcut learning [2]. This phenomenon occurs when a model exploits spurious correlations within the training dataset to achieve a minimal training error, yet fails to capture the underlying causal relationships necessary for accurate inference on unseen data. This inevitably leads to a high test error. While standard regularization techniques are effective, more targeted strategies can be employed within the context of knowledge distillation.

When considering knowledge distillation, it is essential to begin with its core motivation. During the training of deep learning models, the primary focus is typically on maximizing accuracy. However, in deployment scenarios efficiency becomes equally critical. This discrepancy between training and deployment objectives motivated the teacher-student paradigm introduced by Hinton et al. in 2015 [3], where a smaller student network is trained to mimic a larger teacher, aiming to retain accuracy while reducing computational cost.

A major breakthrough in recent years has come from contrastive learning, a self-supervised approach that enables models to learn powerful representations by distinguishing between similar and dissimilar pairs. Frameworks such as SimCLR [1] have demonstrated that, given a sufficiently large architecture and dataset, contrastive methods can achieve performance on par with supervised learning.

The success of contrastive learning naturally raises the question of how such high-capacity models can be made more efficient for deployment. This is the motivation behind Contrastive Representation Distillation, which uses contrastive learning to align the representations of teacher and student for individual samples, effectively transferring the semantic structure learned by a high-capacity teacher to a lightweight student network [5].

Our proposal builds on the hypothesis that separating representation learning from classification leads to more robust and generalizable models. To this end, we adopt a two-phase training strategy. In the first phase, we focus exclusively on learning semantically rich and transferable representations by aligning the student's intermediate features with those of a high-performing teacher using CRD. The classification objective is entirely removed at this stage to prevent the model from relying on spurious shortcuts. In the second phase, we freeze or conservatively fine-tune the feature extractor and train the final classifier using KL divergence and Cross Entropy between the soft predictions of teacher and student. This decomposition enables stable optimization, mitigates shortcut learning, and results in a more interpretable and effective training process.

We evaluate our two-phase training strategy on CIFAR-100 using multiple teacher-student pairs. The results indicate that two-phase training shows good results in both representation alignment and classification performance, where second-phase fine-tuning

yields additional gains, enhancing model accuracy. Statistical tests confirm that these improvements are robust and meaningful. The following sections provide detailed results and in-depth analysis.

The rest of this report is organised as follows. Firstly, in Section 2 we describe the theoretical foundations of the approach we explore. Descriptions of our methodology, dataset, and experimental setup are provided in Sections 3 and 4, respectively. The experimental results, including model performance comparisons are discussed in Sections 5 and 6. Finally, we conclude the report with remarks, future directions and key findings in Section 7.

2. Related Work

Knowledge Distillation. It is a widely used technique for model compression, which involves training a smaller student model to imitate the behaviour of a larger teaching model [3]. The core idea is to use the teacher's soft predictions to guide the student. Unlike hard labels, softened outputs provide richer supervision with inter-class similarities and decision boundaries. However, some traditional distillation methods focus on matching output distributions and does not explicitly guide the internal representations learned by the student.

Contrastive Representation Distillation. This technique extends knowledge distillation into the representation space by using contrastive learning to align student and teacher features [5]. CRD trains the student to bring its feature vector close to the teacher's for the same input while contrasting it against negatives sampled from other instances. In this method the assumption is that the representations are hierarchical and the objective is to capture the correlations and to find the higher order dependencies for each one of the individual data points corresponding to a sample (e.g. to find inter-sample relations). Some works have shown that transferring mutual similarities rather than the actual representations yields better distillation [4].In CRD, the pairwise relationships among samples are first computed in the teacher's feature space by measuring their similarities. These are then converted into probability distributions, and the student is trained to match these distributions in its own feature space using losses such as L2.

Two-Stage and Modular Training. Our approach builds on the idea of decoupling representation learning from classification. This principle is reflected in transfer learning pipelines, where pretrained feature extractors are frozen and a lightweight classifier is trained on top [7]. Similar strategies are also adopted in self-supervised learning, such as SimCLR [1], where a linear classifier is trained after learning representations with a contrastive objective. Such decoupled strategies are shown to reduce overfitting and improve generalization.

Shortcut Learning. Deep neural networks often rely on spurious correlations in the data to minimize training loss, a phenomenon referred to as shortcut learning [2]. These shortcuts lead to poor generalization under distributional shift. Several recent works propose regularization techniques or architectural interventions to force models to develop more semantically meaningful features.

3. Methodology

The training strategy is articulated in two distinct, sequential phases to decouple representation learning from classification, thereby mitigating shortcut learning and improving generalization.

3.1. Phase 1: Hierarchical Representation Alignment via CRD

In the first phase, we aim to transfer the structural knowledge from a high-capacity *teacher* model to a more compact *student* model using Contrastive Representation Distillation. Unlike traditional distillation approaches that emphasize output logits, CRD focuses on aligning intermediate representations.

To encourage the student to mirror the relational structure of the teacher's feature space, we apply CRD loss at two hierarchical levels of the feature extractor. We extract features from the output of the penultimate and final residual blocks of the ResNet backbone, denoted as f_{pen} and f_{last}, respectively. The contrastive representation distillation loss, denoted as $\mathcal{L}_{\text{CRD}}$, is employed to align the sample-wise relationships in the feature space of the student with those of the teacher. As it was introduced in the work by Tian *et al.* [6].

Given a batch of anchor samples $\{x_i\}$, their positive counterparts $\{x_i^+\}$ from the teacher, and negative samples $\{x_j^-\}$, the loss is defined as:

$$\mathcal{L}_{\text{CRD}} = -\sum_{i=1}^{N} \log \frac{e^{\frac{\text{sim}(z_i, z_i^+)}{\tau}}}{e^{\frac{\text{sim}(z_i, z_i^+)}{\tau}} + \sum_{j \neq i} e^{\frac{\text{sim}(z_i, z_j)}{\tau}}}$$

where z_i and z_i^+ are the projected representations of the student and teacher, $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, τ is a temperature scaling parameter, and the denominator includes all negatives in the batch.

The contrastive representation distillation loss, denoted as $\mathcal{L}_{\text{CRD}}$, is employed to align the sample-wise relationships in the feature space of the student with those of the teacher [6]. The antepenultimate layer captures more abstract features, while the penultimate layer focuses on discriminative representations. The combined loss for this phase is:

$$\mathcal{L}_{\text{Stage1}} = \lambda_1 \mathcal{L}_{\text{CRD}}^{(L-2)} + \lambda_2 \mathcal{L}_{\text{CRD}}^{(L-1)}$$

Here, $\lambda_2 > \lambda_1$ emphasizes the role of discriminative features, while still leveraging abstract features. During this phase, the student's classifier is not optimized directly.

Table 1. CRD Hyperparameters

Parameter	Value
Feature dimension	256
Mode	exact
Number of negative samples	16 384
Temperature	0.1
Momentum (non-parametric updates)	0.5

For the optimization we use Stochastic Gradient Descent (SGD) with an initial learning rate of 0.03, exponential decay, and a momentum of 0.9.

The hyperparameters for our CRD framework are detailed in Table 1. Following prior work [5], we used a feature dimension of 256, 16 384 negative samples, a temperature of 0.1, a momentum of 0.5, and the 'exact' contrastive mode. These settings were empirically validated on our tasks.

3.2. Phase 2: Classifier Optimization via KL Divergence and Cross Entropy

Students trained using KL divergence after CRD show improved generalization compared to end-to-end trained baselines. Additionally, fine-tuning the penultimate layer with a very low learning rate yields a modest yet consistent performance gain, without compromising previously learned features, preventing the phenomenon of "*Catastrophic Forgetting*". During this stage, we used a weighted combination of losses: more than half the weight loss to KL divergence compared to the cross-entropy one, allowing the student to balance learning from the teacher's soft targets with a grounding in the true labels.

In cases where the teacher acts as an excellent classifier, serving effectively as a perfect feature extractor, the Kullback-Leibler divergence alone may be not only necessary but also sufficient for effective student learning. This setup could help prevent overfitting behaviors that could be introduced by the Cross Entropy loss with one-hot labels.

Once the feature extractor has learned to generate aligned representations, we freeze its weights and shift focus to training the classifier. We minimize the Kullback-Leibler divergence between the softened probability distributions (using temperature scaling) of the teacher and the student:

$$\mathcal{L}_{\text{Stage2}} = D_{\text{KL}}\left(\sigma(\mathbf{z}_{\text{teacher}}/T) \parallel \sigma(\mathbf{z}_{\text{student}}/T)\right)$$

This approach forces the student to mimic not only the correct class but also the teacher's uncertainty over all classes, discouraging overconfident and brittle predictions [5].

To perform optimization during the second phase we use the Adam optimizer with a learning rate of 1×10^{-4} for the classifier. In a fine-tuning variant, we allow micro-adjustments in the penultimate feature layer using a smaller learning rate of 1×10^{-5}.

The temperature for softmax is set to $T = 4$.

4. Dataset and Experimental Setup

We evaluate our method using the CIFAR-100 dataset, which contains 100 classes of natural images sized 32×32. The dataset is split into 70% for training and 30% for testing.

To assess the generality of our approach, we experimented with a range of teacher-student configurations involving networks of varying depth and capacity. This includes large-capacity pairs such as ResNet-50 $\rightarrow$ ResNet-32, intermediate ones like ResNet-34 $\rightarrow$ ResNet-20, as well as more extreme compression settings exemplified by ResNet-18 $\rightarrow$ ResNet-8. These configurations were chosen to evaluate the effectiveness of our distillation strategy under different levels of representational complexity.

The teacher networks were fine-tuned to achieve strong accuracy on CIFAR-100 (e.g., 70.65% for ResNet-50), and all students were trained under identical setups described earlier.

5. Experiments

Our experiments are designed to evaluate both the representational quality of the student model and its final classification performance. The training follows the two-phase strategy outlined in the previous Section 3.

5.1. Evaluation

During Phase 1, we evaluate the alignment between teacher and student features using cosine similarity, which measures the cosine of the angle between two non-zero vectors in a multi-dimensional space, providing a normalized score of their orientation. For the cosine similarity, a value close to 1 indicates that the feature vectors point in nearly the same direction, signifying high representational similarity, while a value near 0 suggests orthogonality or dissimilarity.

In Phase 2, we assess the student's classification performance using simply Top-1 accuracy on the test set. Our evaluation protocol employs a fine-tuned feature extractor with a carefully configured training regime: the penultimate layer is optimized using a reduced learning rate (1×10^{-5}) during classifier training.

5.2. Teacher-Student Pair Comparison

We focus on several teacher-student comparisons, notably ResNet-50 $\rightarrow$ ResNet-32, ResNet-34 $\rightarrow$ ResNet-20, and ResNet-18 $\rightarrow$ ResNet-8. In addition to conventional evaluation metrics such as accuracy, we perform statistical analyses to assess the significance of observed performance differences and the dependencies in prediction errors between the teacher and student models. This evaluation helps to better understand how model capacity affects knowledge trans-

fer and to what extent student models can replicate teacher behavior under varying degrees of architectural compression.

We use statistical metrics to evaluate whether the accuracy difference is statistically significant, as this helps determine if a teacher model truly outperforms a student model beyond what might occur by random fluctuation. In particular, **McNemar's test**, which was designed for paired nominal data, making it ideal for comparing two classifiers on the same test set. It operates on a 2×2 contingency table of prediction outcomes (correct/incorrect). The test specifically evaluates the null hypothesis that the two models have the same error rate by focusing only on the discordant pairs, cases where one model is correct and the other is incorrect. A statistically significant result ($p < 0.05$) indicates that the observed difference in performance between the two models is not due to random chance.

We also use a **chi-square test of independence** to assess whether the prediction errors of the models are correlated. The analysis is conducted on 10 000 samples.

6. Results

6.1. Representation Quality

Figure 1 shows both cosine similarity and Centered Kernel Alignment (CKA) metrics tracked during Phase I training. While cosine similarity captures more local alignment and converges rapidly, CKA, which evaluates global structural similarity between representation spaces, generally requires significantly more training epochs to reflect meaningful alignment [8]. Therefore, we chose to track cosine similarity throughout training, which confirms that CRD successfully aligns student representations with those of the teacher.

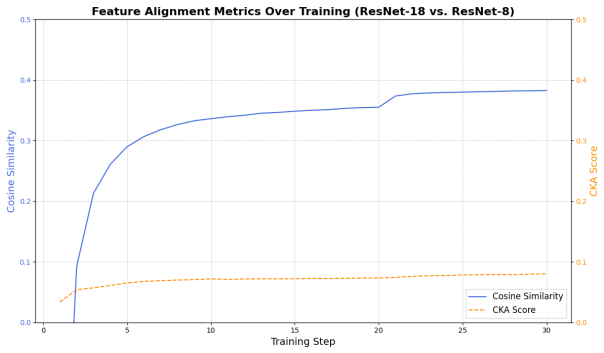

Figure 1. Cosine similarity and CKA during Phase I. Cosine similarity was used as the main monitoring metric due to faster convergence.

6.2. Classification Performance

Students trained during the Phase 2 after CRD shows good generalization. Additionally, fine-tuning the penultimate layer with a low learning rate yields a modest yet consistent performance gain, without compromising previously learned features (Figure 2).

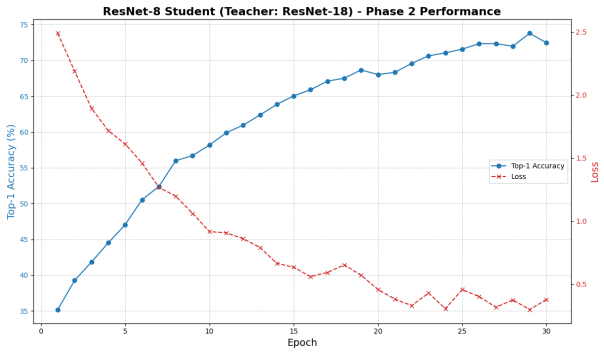

Figure 2. Top-1 accuracy and training loss for the ResNet-8 student model during Phase 2.

6.3. Statistical Analysis of Teacher-Student Pairs

Statistical Formulations:

Given the 2×2 contingency table:

	B Correct	**B Incorrect**
A Correct	a	b
A Incorrect	c	d

Table 2. Contingency table comparing prediction outcomes of two models.

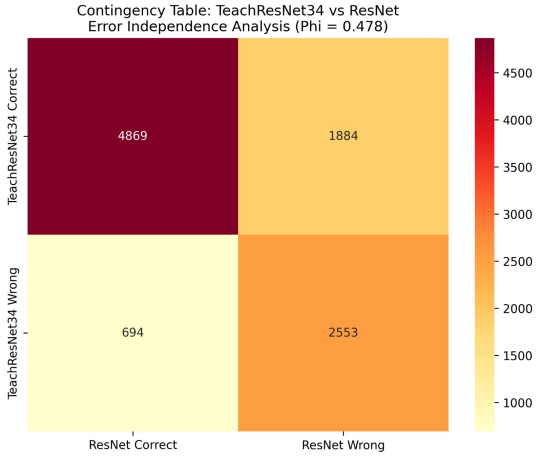

Figure 3. Contingency table used for McNemar's test to compare student (ResNet20) and teacher (ResNet34) predictions.

Chi-Square Test of Independence: The chi-square statistic is computed as:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{2} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

where O_{ij} is the observed frequency and E_{ij} is the expected frequency under the null hypothesis of independence, computed as:

$$E_{ij} = \frac{(\text{row total})_i \cdot (\text{column total})_j}{n}$$

Phi Coefficient: The phi coefficient (φ), which measures the strength of association between the two binary classifiers, is given by:

$$\varphi = \frac{ad - bc}{\sqrt{(a+b)(c+d)(a+c)(b+d)}}$$

This coefficient ranges from -1 (perfect negative association) to $+1$ (perfect positive association), with 0 indicating no association.

Error Separation Index. The Error Separation Index (ESI) measures the relative exclusivity of misclassifications between two models. It is defined as:

$$\text{ESI} = \frac{|b - c|}{b + c}$$

This metric ranges from 0 (errors are evenly shared between models) to 1 (errors are completely disjoint), offering insight into potential complementarity in ensemble settings.

Interpretation of metrics

We report the following results from the paired analysis on the classification task across three teacher-student pairs. Statistical analysis results are presented in Table 3.

All teacher-student pairs show statistically significant differences in performance based on the McNemar test ($p < 0.00001$). The teacher consistently outperforms the student, with the gap particularly wide in the ResNet18/ResNet8 pair.

In all three pairs, the chi-square test of independence is significant, indicating that the models' errors are not independent. Phi coefficients ranging from 0.344 to 0.515 suggest a strong positive correlation, meaning that both models tend to make errors on similar samples.

The error separation index across pairs hovers around 0.33–0.38, indicating a moderate overlap in error patterns. While this limits the potential gains from naive ensembling, it still suggests room for techniques that exploit error diversity.

In the case of ResNet18/ResNet8, the student is too shallow to effectively learn from the teacher, leading to a drastic performance drop (33.64% vs. 63.49%). This underlines the importance of sufficient student capacity for successful knowledge transfer. In contrast, deeper student models like ResNet32 exhibit better alignment and generalization, though still lag behind their more powerful teachers.

Overall, the results affirm the robustness of our distillation setup, particularly when the student network is sufficiently expressive to absorb the teacher's knowledge.

7. Conclusion

We introduced a two-phase training strategy for knowledge distillation that explicitly separates representation learning from classifier optimization. Our experiments on CIFAR-100 demonstrate that this decoupled approach successfully transfers knowledge, creating a compact student model that learns from a larger teacher. While the distilled student does not surpass the teacher's performance, our statistical analysis provides deeper insights. The strong positive correlation in prediction errors (with φ coefficients ranging from 0.344 to 0.515, increasing with capacity of the models) indicates that the student effectively mimics the teacher's decision-making process, especially on challenging samples. However, the analysis of discordant pairs (Figure 3) and the error separation index also confirm that the student develops unique error patterns, rather than being a simple replica. This highlights the method's ability to produce a model with a predictable, yet distinct, behavioral profile.

It is important to note that our experiments were constrained by computational costs, which limited the number of training epochs. The literature on contrastive learning suggests that optimal feature representations often require significantly longer training schedules [1]. Therefore, we hypothesize that the performance of both the teacher and the distilled student could be substantially improved with more extensive fine-tuning and a greater number of training epochs, allowing the models to converge more fully.

Furthermore, the implementation of this two-stage framework presented considerable practical challenges. The complexity of the codebase, coupled with limited computational and memory resources, demanded careful management. The process was also marked by a high sensitivity to hyperparameter selection, which required extensive and time-consuming experimentation to achieve stable and meaningful results.

Key Takeaways

- Decoupling feature learning from classification provides a stable and structured distillation framework.
- The student model is not a simple subset of the teacher; it develops unique error patterns, creating opportunities for analysis and ensembling.
- Fine-tuning the feature extractor with a low learning rate offers a consistent performance boost over a fully frozen approach.

This modular training strategy offers a promising direction for developing more efficient and interpretable deep learning models through distillation.

Table 3. Statistical metrics for paired model comparisons across different architectures.

Metric	**ResNet34/20**	**ResNet50/32**	**ResNet18/8**
Sample Size (n)	10,000	10,000	10,000
Teacher Accuracy	67.53%	70.65%	63.49%
Student Accuracy	55.63%	56.98%	33.64%
McNemar Statistic (χ^2)	548.38	780.41	2297.87
p-value (McNemar)	< 0.00001	< 0.00001	< 0.00001
Chi-square (Independence)	2283.92	2646.67	1183.88
p-value (Independence)	< 0.00001	< 0.00001	< 0.00001
Phi Coefficient (φ)	0.478	0.515	0.344
Error Separation Index	0.335	0.330	0.380

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 1597–1607. PMLR, July 2020.

[2] Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

[3] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *CoRR*, abs/1503.02531, 2015.

[4] Wonmin Park, Dongju Kim, Yang Lu, and Minsu Cho. Relational knowledge distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3962–3971, Long Beach, CA, USA, 2019.

[5] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020.

[6] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations (ICLR)*, 2020.

[7] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3320–3328, 2014.

[8] Zikai Zhou, Yunhang Shen, Shitong Shao, Linrui Gong, and Shaohui Lin. Rethinking centered kernel alignment in knowledge distillation, 2024.