Discriminative and Consistent Representation Distillation

Nikolaos Giakoumoglou[1], **Tania Stathaki**[1]

[1]Imperial College London

Knowledge Distillation (KD) transfers knowledge from a large teacher model to a smaller student model. While contrastive objectives have proven effective for learning structured representations in self-supervised settings, their use in distillation is hindered by two practical shortcomings: the reliance on large external memory banks for negative sampling, and the class-collision problem, in which semantically similar instances are pushed apart due to uniform negative treatment. To address these limitations, we propose **D**iscriminative and **C**onsistent **D**istillation (DCD), a framework that combines contrastive instance discrimination with a consistency regularization term based on KL divergence over pairwise similarity distributions. This dual objective ensures the student captures fine-grained instance-level features while preserving the teacher's relational structure. Our method further introduces an efficient in-batch sampling strategy that removes the need for external memory banks, and learnable temperature and bias parameters that adapt during training to control the sharpness and scale of the distillation signal. The resulting method matches the training speed of standard KD while introducing only 66K additional parameters, compared to 12.8M for prior contrastive approaches. Through extensive experiments on CIFAR-100, ImageNet, and MS-COCO across 13 teacher-student pairs, we show that our approach achieves state-of-the-art performance in classification, object detection, and cross-dataset transfer, while substantially reducing memory consumption and training time compared to existing distillation methods.

Date: July 16, 2024
Correspondence: Nikolaos Giakoumoglou <nikos@imperial.ac.uk>
Code: https://github.com/giakoumoglou/rrd

IMPERIAL

1 Introduction

Knowledge Distillation (KD) enables the transfer of knowledge from large, high-capacity *teacher* models to compact *student* models [1]. As state-of-the-art vision models for image classification [2; 3], object detection [4; 5], and semantic segmentation [6; 7] continue to grow in size and computational cost [8; 9], efficient model compression has become a practical necessity [10; 11]. The foundational works of Buciluǎ et al. [10] and Hinton et al. [1] formulated distillation as minimizing the KL divergence between teacher and student output distributions. While this is natural when the output is a categorical distribution over classes, it does not capture the richer internal knowledge encoded in intermediate representations, including visual semantics and inter-class relations.

Representational knowledge is inherently *structured*: feature dimensions exhibit non-trivial correlations and higher-order dependencies that logit

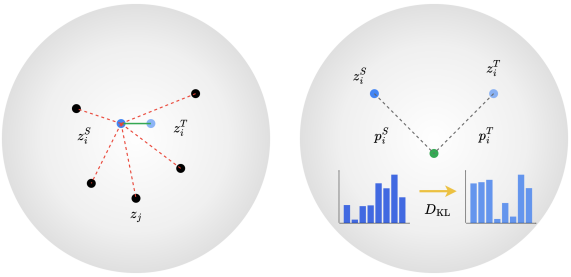


Figure 1 Overview of DCD. (left) Discriminative learning through contrastive distillation encourages student features (solid blue) to differentiate between instances by pulling them closer to their corresponding teacher features (transparent blue) while pushing away from other instances as negative samples (black dots). (right) Structural consistency through consistency regularization preserves the distributional relationship patterns captured by the teacher model by aligning the student and teacher feature similarities (represented by dotted lines) through KL divergence minimization.

matching alone cannot preserve. To address this, feature-based methods [12; 13; 14; 15] extend distillation to intermediate layers. However, Tian et al. [16] demonstrated that such approaches still fail to capture the full structural knowledge in the teacher's representations. Building on this observation, contrastive objectives [17; 18; 19] have been adapted for distilling structured knowledge between teacher and student networks [16; 20]. By treating each instance as its own class and learning to discriminate between them, contrastive distillation can transfer fine-grained representational information that goes beyond what logit-based or feature-matching methods capture.

However, contrastive distillation in its current form faces notable **limitations**. First, instance-discrimination objectives inevitably introduce a *class-collision problem*, formally identified by Giakoumoglou and Stathaki [21]: because negative samples are drawn uniformly, semantically similar instances are pushed apart just as aggressively as dissimilar ones [22; 23; 24]. In the distillation context, this weakens the alignment between teacher and student relational structures, preventing the student from learning a coherent semantic geometry. Second, existing contrastive distillation methods such as CRD [16] require large external memory banks to store negative representations, introducing significant memory overhead and implementation complexity. Third, fixed temperature hyperparameters limit the adaptability of the contrastive objective across different training stages and teacher-student pairs [25].

To overcome these limitations, we propose **D**iscriminative and **C**onsistent **D**istillation (DCD), a method that combines contrastive instance discrimination with a consistency regularization term. The contrastive component aligns teacher and student representations at the instance level, while the consistency term, formulated as KL divergence over pairwise similarity distributions, ensures that the student preserves the teacher's global relational structure. This dual objective mitigates the class-collision problem by explicitly encouraging the student to maintain the relative similarity patterns observed by the teacher. Furthermore, our method replaces external memory banks with an efficient in-batch sampling strategy, using only the negative samples that naturally co-exist within each mini-batch, reducing the additional parameter count from 12.8M to just 66K while matching the training speed of standard KD at 8ms per batch. We also introduce learnable temperature and bias parameters that adapt during training, automatically adjusting the sharpness and scale of the distillation signal rather than relying on fixed hyperparameters.

Our main **contributions** are as follows: **(i)** We propose a unified distillation framework (Section 3) that combines contrastive instance discrimination (Section 3.2.1) with a KL divergence-based consistency regularization (Section 3.2.2), enabling the student to learn representations that are both discriminative and structurally aligned with the teacher. **(ii)** We introduce an efficient in-batch sampling strategy that eliminates the need for large external memory banks, reducing additional parameters by 195× compared to CRD [16] and matching the training throughput of simple logit-based methods (Section 3.3 and Table 5). **(iii)** We employ learnable temperature and bias parameters that dynamically adapt during training, providing flexible control over the distillation objective without manual tuning (Section 3.3 and Figure 4). **(iv)** We validate our method through extensive experiments on CIFAR-100 (Table 1), ImageNet (Table 2), and MS-COCO (Table 3) across 13 teacher-student pairs, demonstrating state-of-the-art results in classification, object detection, and cross-dataset transferability (Table 4).

2 Related Work

2.1 Knowledge Distillation

The original formulation of knowledge distillation [1] transfers knowledge through softened logit outputs using temperature scaling in the softmax function. Subsequent work has developed along two main branches: *logit-based* distillation, which operates on the output distributions of the network, and *feature-based* distillation, which leverages intermediate representations.

2.1.1 Logit-based Methods

Logit-based methods transfer knowledge by aligning the output distributions of the teacher and student networks. Several directions have been explored to improve upon the original KL divergence formulation. One line of work focuses on restructuring the distillation objective itself, including decoupling the KL

divergence into target-class and non-target-class components [26], reweighting soft labels from a bias-variance perspective [27], addressing the transfer gap through probability reweighting [28], and normalizing logits before applying softmax and KL divergence to improve gradient behavior [29]. Another direction explores adaptive temperature control, through fixed schedules with auxiliary networks [30] or relaxed distribution matching that accounts for teacher-student capacity differences [31]. Further efforts include test-time memory mechanisms that refine distillation at inference [32] and softmax regression-based representation learning [33]. Our method shares the use of temperature scaling with [1], but differs in that our temperature is a learnable parameter optimized jointly with the network, rather than a fixed or scheduled hyperparameter, closely following Radford et al. [25].

2.1.2 Feature-based Methods

Feature-based methods transfer knowledge from intermediate representations rather than output logits, allowing the student to learn richer structural information from the teacher's internal layers. Early work introduced intermediate feature hints [12] and spatial attention alignment [13] to guide student training. A substantial body of work has since focused on preserving structural relationships, including geometric relations between sample pairs such as distances and angles [34], correlation matrices across feature dimensions [15], pairwise similarity structures [35], and probabilistic embeddings [36]. Other approaches include factor transfer through paraphrasing [37], activation boundary transfer [38], flow of solution procedures [14], neuron selectivity transfer [39], variational information distillation [40], and comprehensive feature overhaul [41].

Recent methods introduced cross-stage review mechanisms [42], reuse of the teacher's classifier head [43], cross-layer semantic calibration [44], many-to-one representation matching [45], feature correlation-based distillation [46], and category-aligned attention transfer [47]. Of particular relevance to our work, CRD [16] adapted contrastive objectives to maximize mutual information between teacher and student representations, showing that feature-matching methods alone fail to capture the full structural knowledge in the teacher. However, CRD requires a large external memory bank (16k features per class) and relies on fixed hyperparameters. Extensions include Wasserstein-based contrastive distillation [48], complementary relational contrastive distillation [49], feature kernel distillation [50], and information-theoretic representation distillation [51].

Our method falls within the feature-based family but addresses the key limitations of existing contrastive approaches: we replace external memory banks with in-batch sampling, introduce learnable temperature and bias parameters, and augment instance-level contrastive alignment with a distributional consistency term that preserves the teacher's relational structure.

2.2 Instance Discrimination

Instance discrimination methods in self-supervised learning learn representations by treating each individual sample as its own class [52; 53], building on foundations in metric learning [54; 55] and noise contrastive estimation (NCE) [17]. These methods transform unsupervised learning into a classification problem, and theoretical analysis has shown that such objectives maximize a lower bound on mutual information between views [18; 22; 19]. Recent advances include momentum-based encoders [56], large-batch contrastive learning with stronger augmentations [57], and methods that eliminate explicit negative samples entirely through asymmetric architectures [58; 59; 60], redundancy reduction [61], or variance-covariance regularization [62]. Some approaches further explore invariance regularizers [63] or rethink the role of the momentum encoder itself [64]. Our method combines instance-level discrimination [52; 53] with a consistency regularization term that explicitly aligns pairwise similarity distributions between teacher and student, preserving relational structure while retaining the discriminative benefits of instance-level learning. Our approach avoids fixed negative samples or momentum encoders [64], using dynamic adaptation during training.

3 Methodology

Here, we introduce our objective which transfers knowledge from a pre-trained teacher network to a student network by combining instance-level contrastive alignment with distributional consistency regularization. Section 3.1 outlines the fundamental principles of knowledge distillation, Section 3.2 details the formulation of our dual objective, and Section 3.3 describes the implementation details.

3.1 Preliminaries on Knowledge Distillation

Knowledge distillation involves transferring knowledge from a high-capacity teacher neural network f_θ^T to a more compact student neural network f_θ^S [1; 10]. Consider $\mathbf{x}_i$ as the input to these networks, typically an image. We represent the outputs at the penultimate layer (just before the final classification layer) as $\mathbf{z}_i^T = f_\theta^T(\mathbf{x}_i)$ and $\mathbf{z}_i^S = f_\theta^S(\mathbf{x}_i)$ for the teacher and student models, respectively. The primary objective of knowledge distillation is to enable the student model to approximate the performance of the teacher model. The overall distillation process can be mathematically expressed as:

$$\hat{\theta}_S = \arg\min_{\theta_S} \sum_{i=1}^{N} \left(\mathcal{L}_{\text{sup}}(\mathbf{x}_i, \theta_S, y_i) + \lambda \cdot \mathcal{L}_{\text{distill}}(\mathbf{x}_i, \theta_S, \theta_T) \right), \tag{1}$$

where y_i represents the true label for the input $\mathbf{x}_i$, θ_S and θ_T are the parameter sets for the student and teacher networks, and λ is a hyperparameter that balances the supervised loss and the distillation loss. The supervised loss $\mathcal{L}_{\text{sup}}$ is the task-specific alignment error between the network prediction and annotation. For image classification [65; 66; 11; 67], this is typically cross-entropy loss, while for object detection [68; 69], it includes bounding box regression. The distillation loss $\mathcal{L}_{\text{distill}}$ is the mimic error of the student network towards the teacher network, typically implemented as KL divergence between student and teacher outputs [1].

3.2 Discriminative and Consistent Distillation

Our method extends Equation (1) with an additional feature-based distillation term $\mathcal{L}_{\text{kd}}$:

$$\hat{\theta}_S = \arg\min_{\theta_S} \sum_{i=1}^{N} \left(\mathcal{L}_{\text{sup}}(\mathbf{x}_i, \theta_S, y_i) + \lambda \cdot \mathcal{L}_{\text{distill}}(\mathbf{x}_i, \theta_S, \theta_T) + \beta \cdot \mathcal{L}_{\text{kd}}(\mathbf{x}_i, \theta_S, \theta_T) \right), \tag{2}$$

where β weights our proposed loss $\mathcal{L}_{\text{kd}}$, which consists of two complementary terms:

$$\mathcal{L}_{\text{kd}} = \mathcal{L}_{\text{ctr}} + \alpha \cdot \mathcal{L}_{\text{cst}}, \tag{3}$$

where $\mathcal{L}_{\text{ctr}}$ is a contrastive objective (Section 3.2.1) for instance-level discrimination, $\mathcal{L}_{\text{cst}}$ is a consistency regularization term (Section 3.2.2) for preserving relational structure, and α balances the two components (*cf.* Section 4.4). The combination provides complementary supervision: KL divergence offers direct class-level guidance through logit-space alignment, while our proposed loss ensures feature-space consistency.

3.2.1 Contrastive Objective

We employ contrastive learning to align teacher and student representations at the instance level. For a mini-batch of N samples, each student representation $\mathbf{z}_i^S$ must identify its corresponding teacher representation $\mathbf{z}_i^T$ among all N teacher representations. This is formulated as an N-way classification problem using noise contrastive estimation [17]:

$$\mathcal{L}_{\text{ctr}} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{\exp(\ell_{ii})}{\sum_{j=1}^{N} \exp(\ell_{ij})}, \tag{4}$$

where $\ell_{ij} = \mathbf{z}_i^S \cdot \mathbf{z}_j^T \cdot \exp(\tau) + b$ is the scaled similarity between the i-th student feature and the j-th teacher feature, and τ and b are learnable temperature and bias parameters described in Section 3.3.2. The negative samples are all other teacher representations within the same mini-batch, eliminating the need for external memory banks. This formulation effectively transforms into a cross-entropy loss, where each student representation $\mathbf{z}_i^S$ must identify its corresponding teacher representation $\mathbf{z}_i^T$ among all other teacher representations in the batch. The objective essentially becomes a classification task: student embeddings must "classify" their matching teacher embeddings correctly, with the normalized similarities acting as logits and

positive pair indices as class labels. The parameters τ and b provide fine-grained control over this classification process, determining its sharpness and scale respectively.

3.2.2 Consistency Objective

The contrastive objective aligns individual instances but does not explicitly preserve the relational structure of the teacher's representation space. To address this, we introduce a consistency regularization term that enforces agreement between two complementary perspectives of the cross-model similarity. Given the similarity matrix $\ell_{ij} = \mathbf{z}_i^S \cdot \mathbf{z}_j^T \cdot \exp(\tau) + b$ from the contrastive objective, we define two distributions by normalizing along different axes. The student perspective $\mathbf{p}_i^S = [p_i^S(1), \ldots, p_i^S(N)]$ captures, for each student representation i, its relative affinity to all teacher representations, while the teacher perspective $\mathbf{p}_j^T = [p_j^T(1), \ldots, p_j^T(N)]$ captures, for each teacher representation j, the relative affinity of all student representations towards it:

$$p_i^S(j) = \frac{\exp(\ell_{ij})}{\sum_{k=1}^N \exp(\ell_{ik})}, \qquad p_j^T(i) = \frac{\exp(\ell_{ij})}{\sum_{k=1}^N \exp(\ell_{kj})}. \tag{5}$$

Intuitively, $\mathbf{p}_i^S$ answers "which teacher representation best matches student i?", while $\mathbf{p}_j^T$ answers "which student representation best matches teacher j?". For a well-aligned student, these two perspectives should agree: if student i strongly selects teacher j, then teacher j should also strongly select student i. The consistency objective enforces this by minimizing the KL divergence between the two views:

$$\mathcal{L}_{\text{cst}} = \frac{1}{N} \sum_{i=1}^N D_{\text{KL}}(\mathbf{p}_i^S \| \mathbf{p}_i^T) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^N p_i^S(j) \log \frac{p_i^S(j)}{p_j^T(i)}. \tag{6}$$

This term encourages the similarity matrix to be structurally consistent across both perspectives, promoting a coherent bidirectional matching between student and teacher representations. Unlike relational methods such as RKD [34] that preserve geometric relations between sample pairs, or CC [15] that maintains correlation matrices, our approach aligns complementary views of the same cross-model similarity structure, providing a softer and more informative supervisory signal. While RRD [21] also aligns similarity distributions via cross-entropy, it relies on an external memory bank with dual temperatures (τ_t, τ_s) and uses only the distributional objective.

3.2.3 Relation to InfoNCE

Our contrastive objective in Equation (4) is closely related to the InfoNCE loss [18]. InfoNCE implements instance discrimination through N-way classification, separating different instances while enforcing similar representations for matching pairs. Our formulation extends this by sharing the learnable temperature and bias parameters with the consistency objective (Equation (6)), coupling the two terms through a common similarity scale. When $\alpha = 0$, our method reduces to a contrastive distillation objective with learnable temperature (*cf.* Section 4.4 for the $\alpha = 0$ case).

3.2.4 Relation to KL Divergence Distillation

The standard KL divergence distillation loss [1] aligns softmax distributions over C class logits, while our consistency objective (Equation (6)) aligns softmax distributions over N pairwise similarity scores. Both use KL divergence to transfer distributional knowledge, but operate in complementary spaces: the former captures class-level predictions, while the latter captures relational structure in the feature space.

3.3 Implementation Details

We implement the objective using mini-batch stochastic gradient descent. The representations $\mathbf{z}_i^T = f_\theta^T(\mathbf{x}_i)$ and $\mathbf{z}_i^S = f_\theta^S(\mathbf{x}_i)$ are obtained from the last layer of the teacher and student models, respectively. We further encode $\mathbf{z}_i^T$ and $\mathbf{z}_i^S$ using a projection head to match the dimensions. The projection head is trained using stochastic gradient descent as well, ensuring that the representations from both models are compatible for

comparison and alignment. Additionally, we ℓ_2-normalize the outputs $\mathbf{z}_i^T$ and $\mathbf{z}_i^S$ before computing the loss, so that the representations lie on a unit hypersphere.

3.3.1 Memory-efficient In-batch Sampling

Instead of maintaining a large memory buffer for negative sampling as in CRD [16] or RRD [21], we use the negative samples that naturally co-exist within the mini-batch. This approach significantly reduces memory requirements: while CRD's memory bank on ImageNet requires approximately 8MB per class (16k features $\times$ 128 dimensions $\times$ 4 bytes), our in-batch sampling requires only 0.13MB total ($256 \times 128 \times 4$ bytes for a batch size of 256). This also eliminates the complexity of memory bank management, including challenges related to feature staleness and queue maintenance [56], and ensures that all negative representations are up-to-date within the current training iteration. This efficiency extends to training time, *i.e.*, on a 4-GPU machine, our method completes ImageNet training in approximately 72 hours compared to CRD's 88 hours, representing an 18% reduction in training time.

3.3.2 Learnable Temperature and Bias

Unlike contrastive learning objectives that use a fixed temperature parameter, we parameterize the temperature using $\exp(\tau)$ where τ is a learnable parameter, along with a learnable bias b, closely following Radford et al. [25]. For a batch of normalized embeddings $\mathbf{z}_i^S$ and $\mathbf{z}_i^T$, the similarity matrix is computed as $\ell_{ij} = \mathbf{z}_i^S \cdot \mathbf{z}_j^T \cdot \exp(\tau) + b$, implemented efficiently as a scaled matrix multiplication. The exponential parameterization ensures the effective temperature remains positive while allowing unconstrained optimization of τ, which is clamped to $[0, \tau_{\max}]$ for numerical stability. The learnable bias b provides an additive degree of freedom that adjusts the logit scale. Both τ and b are shared across the contrastive and consistency objectives (Equations (4) and (6)), coupling the two terms through a common similarity parameterization. This adaptive approach allows the model to automatically tune the contrast level and logit scaling during training, leading to more robust knowledge transfer compared to fixed hyperparameter approaches. We ablate the learnable τ and b in Section 4.4.

4 Experiments

We evaluate our method[1] on both image classification and object detection tasks across multiple benchmarks and teacher-student configurations. Section 4.1 outlines the experimental setup, Section 4.2 presents quantitative results across benchmarks, and Section 4.3 analyzes the learned representations and structural properties. Ablations are discussed in Section 4.4.

4.1 Experimental Setup

We evaluate the proposed framework on both image classification and object detection tasks using five standard benchmarks: CIFAR-100 [70], ImageNet ILSVRC-2012 [71], STL-10 [72], Tiny ImageNet [71], and MS-COCO [73]. Following prior work [16], we experiment with 13 teacher-student combinations of varying capacity using ResNet [74], Wide ResNet (WRN) [75], VGG [76], MobileNet (MN) [77], and ShuffleNet (SN) [78] architectures. We follow the implementation protocol of [16] for image classification and [26; 42] for object detection. Both student and teacher outputs are projected to a 128-dimensional space using a projection head consisting of a single linear layer, followed by ℓ_2 normalization. We set $\alpha = 0.5$, $\beta = 1$, and $\tau_{\max} = 10.0$. The hyperparameter λ is set to 1.0 for the KL divergence loss to maintain consistency with [16; 44; 43], while we provide ablations in Section 4.4. We compare against a wide range of distillation methods spanning logit-based, feature-based, and contrastive approaches, including KD [1], FitNet [12], AT [13], SP [35], CC [15], VID [40], RKD [34], PKT [36], AB [38], FT [37], FSP [14], CRD [16], OFD [41], WSLD [27], IPWD [28], and CTKD [30], with more methods in Section B.1. Full implementation details are provided in Section B.

[1] Following standard practice, we denote our method as "*DCD*" when using only supervised and proposed losses ($\lambda = 0$), and "*DCD+KD*" when incorporating all objectives ($\lambda \neq 0$).

Table 1 Main results on CIFAR-100. Test top-1 accuracy (%) for various teacher-student architecture combinations. Results for our method and those adapted from [16] are averaged over five independent runs. **Architecture abbreviations**: W: WideResNet, RN: ResNet, rn: resnet, MN: MobileNet, SN: ShuffleNet. See Section B.3 for details.

	Same architecture							Different architecture					
Teacher	WRN-40-2	WRN-40-2	rn-56	rn-110	rn-110	rn-32x4	VGG-13	VGG-13	RN-50	RN-50	RN-32x4	RN-32x4	WRN-40-2
Student	WRN-16-2	WRN-40-1	rn-20	rn-20	rn-32	rn-8x4	VGG-8	MN-v2	MN-v2	VGG-8	SN-v1	SN-v2	SN-v1
Teacher	75.61	75.61	72.34	74.31	74.31	79.42	74.64	74.64	79.34	79.34	79.42	79.42	75.61
Student	73.26	71.98	69.06	69.06	71.14	72.50	70.36	64.60	64.60	70.36	70.50	71.82	70.50
KD [1]	74.92	73.54	70.66	70.67	73.08	73.33	72.98	67.37	67.35	73.81	74.07	74.45	74.83
FitNet [12]	73.58	72.24	69.21	68.99	71.06	73.50	71.02	64.14	63.16	70.69	73.59	73.54	73.73
AT [13]	74.08	72.77	70.55	70.22	72.31	73.44	71.43	59.40	58.58	71.84	71.73	72.73	73.32
SP [35]	73.83	72.43	69.67	70.04	72.69	72.94	72.68	66.30	68.08	73.34	73.48	74.56	74.52
CC [15]	73.56	72.21	69.63	69.48	71.48	72.97	70.81	64.86	65.43	70.25	71.14	71.29	71.38
VID [40]	74.11	73.30	70.38	70.16	72.61	73.09	71.23	65.56	67.57	70.30	73.38	73.40	73.61
RKD [34]	73.35	72.22	69.61	69.25	71.82	71.90	71.48	64.52	64.43	71.50	72.28	73.21	72.21
PKT [36]	74.54	73.45	70.34	70.25	72.61	73.64	72.88	67.13	66.52	73.01	74.10	74.69	73.89
AB [38]	72.50	72.38	69.47	69.53	70.98	73.17	70.94	66.06	67.20	70.65	73.55	74.31	73.34
FT [37]	73.25	71.59	69.84	70.22	72.37	72.86	70.58	61.78	60.99	70.29	71.75	72.50	72.03
FSP [14]	72.91	n/a	69.95	70.11	71.89	72.62	70.33	58.16	64.96	71.28	74.12	74.68	76.09
CRD [16]	75.48	74.14	71.16	71.46	73.48	75.51	73.94	69.73	69.11	74.30	75.11	75.65	76.05
CRD+KD [16]	75.64	74.38	71.63	71.56	73.75	75.46	**74.29**	**69.94**	69.54	74.58	75.12	76.05	76.27
OFD [41]	75.24	74.33	70.38	n/a	73.23	74.95	73.95	69.48	69.04	n/a	75.98	76.82	75.8
WSLD [27]	n/a	73.74	71.53	n/a	73.36	74.79	n/a	n/a	68.79	73.80	75.09	n/a	75.23
IPWD [28]	n/a	74.64	71.32	n/a	**73.91**	**76.03**	n/a	n/a	**70.25**	**74.97**	**76.03**	n/a	76.44
CTKD [30]	75.45	73.93	71.19	70.99	73.52	n/a	73.52	68.46	68.47	n/a	74.78	75.31	75.78
DCD (ours)	74.99	73.69	71.18	71.00	73.12	74.23	73.22	68.35	67.39	73.85	74.26	75.26	74.98
DCD+KD (ours)	**76.06**	**74.76**	**71.81**	**72.03**	73.62	75.09	73.95	69.77	70.03	74.08	76.01	**76.95**	**76.51**

Table 2 Main results on ImageNet. Test top-1 accuracy (%) on the ILSVRC-2012 validation set across diverse distillation methods. We report results from a single run across representative teacher-student pairs.

	Teacher	*Student*	KD [1]	AT [13]	SP [35]	CC [15]	RKD [34]	CRD [16]	DCD	DCD+KD
RN-34→RN-18	73.31	69.75	70.67	71.03	70.62	69.96	70.40	71.17	71.10	**71.71**
RN-50→RN-18	76.16	69.75	71.29	71.18	71.08	n/a	n/a	71.25	71.38	**71.65**
RN-50→MN-v2	76.16	69.63	70.49	70.18	n/a	n/a	68.50	69.07	70.51	**71.55**

4.2 Main Results

We benchmark our method on image classification and object detection tasks. Comprehensive comparisons with additional methods are provided in Section C.

4.2.1 Results on CIFAR-100

We evaluate top-1 classification accuracy on CIFAR-100 across 13 teacher-student pairs covering both same-architecture and cross-architecture configurations. Table 1 compares our method against existing distillation approaches. DCD combined with KD achieves superior performance, surpassing the teacher network by +0.45% in the same-architecture setting (WRN-40-2 to WRN-16-2) and by +0.90% in the cross-architecture setting (WRN-40-2 to ShuffleNet-v1). Over baseline students, the improvements reach +2.82% for same-architecture and +5.25% for cross-architecture pairs, outperforming CRD. DCD alone performs slightly below CRD, which relies on a large 16k-feature memory bank. However, combining DCD with KD yields stronger results, as the two objectives are complementary: KL divergence provides direct class-level supervision through logit-space alignment, while our proposed loss ensures feature-space consistency.

4.2.2 Results on ImageNet

We evaluate top-1 classification accuracy on ImageNet ILSVRC-2012 across multiple teacher-student configurations. Table 2 presents the results. Our method consistently surpasses baselines [1; 13; 35; 15; 34; 16]

Table 3 Object detection performance on MS-COCO. Evaluation using Faster R-CNN with an FPN backbone on the `val2017` set. We report mean Average Precision (AP) and AP at IoU thresholds of 0.5 and 0.75 for single-run experiments.

Method	RN-101 → RN-18			RN-101 → RN-50			RN-50 → MN-v2		
	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}	AP	AP_{50}	AP_{75}
Teacher	42.04	62.48	45.88	42.04	62.48	45.88	40.22	61.02	43.81
Student	33.26	53.61	35.26	37.93	58.84	41.05	29.47	48.87	30.90
KD [1]	33.97	54.66	36.62	38.35	59.41	41.71	30.13	50.28	31.35
FitNet [12]	34.13	54.16	36.71	38.76	59.62	41.80	30.20	49.80	31.69
ReviewKD [42]	36.75	56.72	39.00	40.36	60.97	44.08	33.71	53.15	36.13
DKD [26]	35.05	56.60	37.54	39.25	60.90	42.73	32.34	53.77	34.01
DCD (ours)	**37.12**	**57.58**	**39.93**	**40.48**	**61.14**	**44.21**	**33.89**	**53.84**	**36.28**

Table 4 Cross-dataset generalization performance. Test top-1 accuracy (%) of a WRN-16-2 student distilled from a WRN-40-2 teacher. Representations learned on CIFAR-100 are transferred and evaluated on the STL-10 and Tiny ImageNet datasets. **Abbreviations**: C: CIFAR, TIN: Tiny ImageNet.

	Teacher	*Student*	KD [1]	AT [13]	FitNet [12]	CRD [16]	CRD+KD [16]	DCD	DCD+KD
C-100→STL-10	68.6	69.7	70.9	70.7	70.3	71.6	72.2	71.2	**72.5**
C-100→TIN	31.5	33.7	33.9	34.2	33.5	35.6	35.5	35.0	**36.2**

Table 5 Efficiency analysis on CIFAR-100. Comparison of training latency (ms per batch) and parameter counts (in millions, M) for different distillation frameworks. We use a ResNet-32x4 teacher and ResNet-8x4 student within the `MDistiller` framework [26] on an NVIDIA RTX 6000 GPU.

	KD [1]	FitNet [12]	AT [13]	RKD [34]	CRD [16]	OFD [41]	RewiewKD [42]	DKD [26]	DCD
Time (ms)	8	8	10	17	19	20	14	9	8
Params (M)	0	0.017	0	0	12.866	0.087	1.809	0	0.066

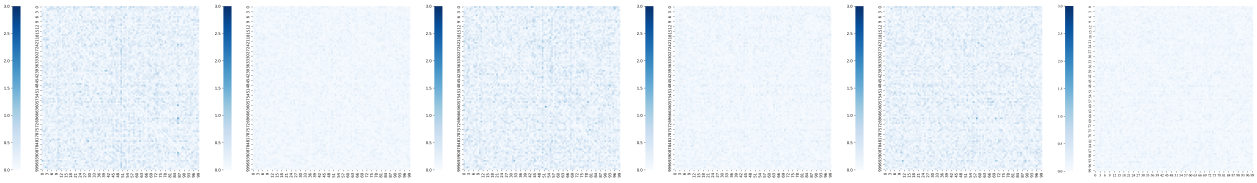
and achieves competitive performance across different architectures, including challenging cross-architecture transfer scenarios. These results confirm that the gains observed on CIFAR-100 extend to large-scale settings.

4.2.3 Results on COCO

We evaluate object detection performance on MS-COCO using Faster R-CNN [4] with Feature Pyramid Network (FPN) [5] as the detection framework, following [26]. We report AP, AP_{50}, and AP_{75} across three teacher-student scenarios: ResNet-101 to ResNet-18, ResNet-101 to ResNet-50, and ResNet-50 to MobileNet-V2. Table 3 presents the results. Our method consistently surpasses baselines [1; 26; 12; 42] across all three settings. These results demonstrate that the gains of our approach extend beyond classification to the more challenging object detection task, including cross-architecture transfer scenarios.

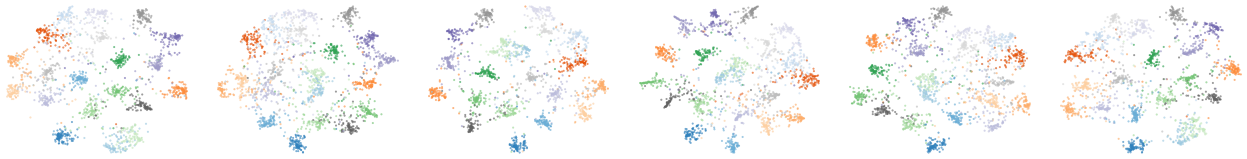
4.2.4 Transferability of Representations

We evaluate the cross-dataset transferability of distilled representations by training a WRN-16-2 student (distilled from WRN-40-2 teacher) on CIFAR-100, then using it as a frozen feature extractor with a linear classifier on STL-10 and Tiny ImageNet. Table 4 reports the top-1 test accuracy for each distillation method. Our method, both standalone and combined with KD, consistently improves transferability over baselines [1; 13; 12; 16], indicating that the learned representations capture generalizable features rather than overfitting to the training distribution.



(a) Vanilla Mean: 0.24 Max: 1.66 | **(b)** KD [1] Mean: 0.09 Max: 0.49 | **(c)** CRD [16] Mean: 0.23 Max: 1.56 | **(d)** CRD+KD [16] Mean: 0.10 Max: 0.57 | **(e)** DCD Mean: 0.22 Max: 1.59 | **(f)** DCD+KD Mean: 0.09 Max: 0.51

Figure 2 Logit correlation analysis. Matrix of the average logit difference between teacher and student outputs on CIFAR-100 (lower values indicate higher similarity). Results are based on a WRN-40-2 teacher and a WRN-40-1 student.



(a) Teacher | **(b)** Vanilla | **(c)** KD [1] | **(d)** CRD [16] | **(e)** DCD | **(f)** DCD+KD

Figure 3 t-SNE visualization of embedding spaces. Comparison of feature distributions from the teacher and student networks for the first 20 classes of CIFAR-100. We use a WRN-40-2 teacher and a WRN-40-1 student.

4.2.5 Efficiency

We compare the computational cost of our method against existing approaches in terms of training latency and parameter overhead. Table 5 presents the results using a ResNet-32x4 teacher and ResNet-8x4 student on CIFAR-100. Our method matches KD and FitNet as the fastest approach at 8ms per batch, while requiring only 65,794 additional parameters attributed solely to the projection heads. By contrast, CRD requires 12.87M parameters to maintain its external memory bank and runs at 19ms per batch, making our method over 2× faster with 195× fewer parameters. Compared to other feature-based methods such as ReviewKD [42] (1.81M parameters, 14ms) and OFD [41] (0.087M parameters, 20ms), our approach is both lighter and faster. Our method achieves the efficiency of simple logit-based methods like KD and DKD while retaining the accuracy gains of contrastive feature-based distillation.

4.3 Visualizations

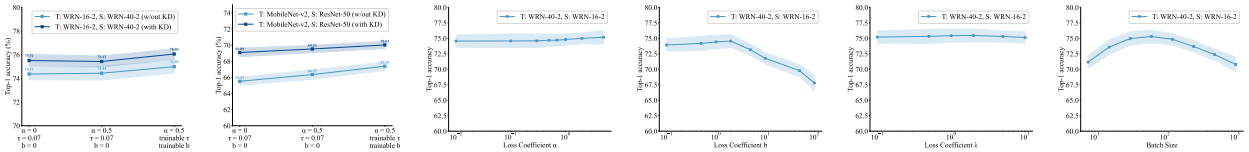
We analyze the learned representations through qualitative visualizations that illustrate the knowledge transfer patterns of different distillation approaches. Additional figures are provided in Section C.

4.3.1 Inter-class Correlations

We visualize the logit correlation matrices [16] on CIFAR-100 using WRN-40-2 (teacher) and WRN-40-1 (student). Figure 2 shows the difference between teacher and student correlation structures for various distillation methods. Compared to vanilla training and standard KD, our method substantially reduces the discrepancy between teacher and student correlations, demonstrating stronger and more consistent alignment in logit space.

4.3.2 t-SNE Visualization

We visualize the feature embeddings on CIFAR-100 using t-SNE [79] for the same teacher-student pair. Figure 3 shows that our method achieves closer alignment of student features with the teacher while preserving meaningful semantic class structure across clusters, outperforming standard training, KD, and several strong contrastive distillation baselines.



(a) WRN-16-2 to WRN-40-2 **(b)** MobileNet-v2 to ResNet-50 **(c)** Effect of varying α **(d)** Effect of varying β **(e)** Effect of varying λ **(f)** Effect of varying N

Figure 4 Ablation study on CIFAR-100. (a–b) Component analysis across architecture pairs. (c–f) Sensitivity to hyperparameters α, β, λ, and batch size. All results are averaged over five runs. Curves in (c–f) are smoothed with Savitzky–Golay filtering.

4.4 Ablation Study

We analyze our method through ablation experiments on CIFAR-100, presented in Figure 4. We first study the individual contributions of the consistency objective and learnable temperature (Figures 4a and 4b), then investigate the sensitivity to the loss coefficients α, β, λ in Equations (2) and (3) and to batch size (Figures 4c to 4f).

4.4.1 Objective Analysis

We evaluate the individual contributions of the consistency regularization $\mathcal{L}_{\text{cst}}$ and the learnable temperature τ across different network architectures (Figures 4a and 4b). Starting from a purely contrastive variant ($\alpha = 0$, fixed temperature), we incrementally add each component. Adding consistency regularization with fixed temperature improves performance across all tested architecture pairs: $+0.62\%$ without KD and $+0.55\%$ with KD for WRN-40-2 to WRN-16-2 transfers, and $+1.87\%$ without KD and $+0.94\%$ with KD for cross-architecture transfers from ResNet-50 to MobileNet-v2. Enabling the learnable temperature further improves results by up to $+1.69\%$ without KD and $+2.21\%$ with KD, confirming that both consistency regularization $\mathcal{L}_{\text{cst}}$ and the learnable temperature τ contribute to the overall performance.

4.4.2 Hyperparameter Sensitivity

We study the sensitivity to each loss coefficient in Equations (2) and (3) using WRN-40-2 to WRN-16-2 transfers. The coefficient α balances $\mathcal{L}_{\text{ctr}}$ and $\mathcal{L}_{\text{cst}}$ within the proposed objective (Equation (3)). Testing $\alpha \in \{0.01, 0.1, 0.3, 0.5, 0.7, 1, 2, 5\}$ with $\beta = 1$ and $\lambda = 0$, Figure 4c shows stable performance across the entire range with no significant differences between low and high values. This robustness stems from the learnable temperature, which automatically adjusts contrast levels regardless of the chosen α.

The coefficient β weights the proposed objective $\mathcal{L}_{\text{kd}}$ relative to the supervised loss in Equation (2). Varying β from 0.1 to 100 with $\alpha = 0.5$ and $\lambda = 0$, Figure 4d shows that extremely high values cause performance degradation as the proposed loss overwhelms the supervised term, while very low values slightly decrease performance. The optimal range lies between $\beta = 0.5$ and $\beta = 10$.

The coefficient λ weights the standard KL divergence distillation loss $\mathcal{L}_{\text{distill}}$ in Equation (2). Testing $\lambda \in \{0.1, 0.5, 1, 2, 5, 10\}$ with $\alpha = 0.5$ and $\beta = 1$, Figure 4e shows stable performance across all tested values, with the best results at $\lambda = 1.0$. Higher values ($\lambda = 50$, $\lambda = 100$) lead to training collapse, confirming that $\lambda = 1.0$ as used in prior work [16; 44; 43] is also optimal for our method.

Finally, Figure 4f shows the effect of batch size on performance. Since our method relies on in-batch negative sampling, the batch size directly determines the number of negatives available for contrastive learning. Optimal results occur at batch size 64, with degradation at smaller sizes (insufficient negatives) and larger sizes (diminishing returns from a diluted learning signal).

5 Conclusion & Discussion

We presented an objective that improves knowledge transfer by combining instance-level contrastive alignment with distributional consistency regularization. The contrastive objective provides fine-grained instance discrimination, while the consistency term preserves the teacher's relational geometry through KL divergence over pairwise similarity distributions. By incorporating learnable temperature and bias parameters, the distillation process adapts its sharpness and scale during training without manual tuning. Replacing external memory banks with in-batch negative sampling reduces the additional parameter count from 12.8M (CRD [16]) to just 66K, while matching the training speed of simple logit-based methods at 8ms per batch, making our approach the fastest among all feature-based distillation methods we evaluated. Experiments across various datasets validate that our method achieves state-of-the-art results in classification, object detection, and cross-dataset transfer, providing a practical and efficient alternative to existing distillation techniques.

Our method differs from prior work in several ways. Unlike CRD [16], which requires a large memory bank and fixed hyperparameters, our approach operates entirely within the mini-batch and uses learnable parameters for adaptive control. Compared to relational methods such as RKD [34] that preserve geometric relations between sample pairs, or CC [15] that maintains correlation matrices, our consistency objective aligns full similarity distributions, providing a softer supervisory signal. While RRD [21] also aligns similarity distributions, it relies on an external memory bank with dual temperatures and lacks an explicit contrastive term; our method replaces the memory bank with in-batch sampling, uses a single shared learnable temperature, and adds direct instance-level supervision through the contrastive objective. Attention-based methods like AT [13] and feature-matching approaches like FitNets [12] each target specific aspects of the teacher's knowledge, whereas our framework provides a unified objective that captures both discriminative and structural information. Methods such as ReviewKD [42] and SimKD [43] achieve strong results through architectural innovations (cross-stage review paths and classifier reuse), while our gains come from algorithmic refinements that require no architectural modifications and add minimal overhead.

Acknowledgements

We acknowledge the computational resources and support provided by the Imperial College Research Computing Service (http://doi.org/10.14469/hpc/2232), which enabled our experiments.

References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[2] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021.

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[5] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017.

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, 2017.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017.

[8] Priya Goyal, Dhruv Mahajan, Abhinav Gupta, and Ishan Misra. Scaling and benchmarking self-supervised visual representation learning, 2019.

[9] Simon Kornblith, Jonathon Shlens, and Quoc V. Le. Do better imagenet models transfer better?, 2019.

[10] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[11] Antonio Polino, Razvan Pascanu, and Dan Alistarh. Model compression via distillation and quantization. In *International Conference on Learning Representations*, 2018.

[12] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *Proceedings of the 4th International Conference on Learning Representations*, 2014.

[13] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *Proceedings of the 5th International Conference on Learning Representations*, 2016.

[14] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4133–4141, 2017.

[15] Baoyun Peng, Xi Li, Yifan Wu, Yizhou Fan, Bo Wang, Qi Tian, and Jun Liang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5007–5016, 2019.

[16] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation, 2022.

[17] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019.

[19] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization, 2019.

[20] Zhiyuan Fang, Jianfeng Wang, Lijuan Wang, Lei Zhang, Yezhou Yang, and Zicheng Liu. Seed: Self-supervised distillation for visual representation, 2021.

[21] Nikolaos Giakoumoglou and Tania Stathaki. Relational Representation Distillation, 2024. URL https://arxiv.org/abs/2407.12073.

[22] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning, 2019.

[23] Junnan Li, Pan Zhou, Caiming Xiong, and Steven C. H. Hoi. Prototypical contrastive learning of unsupervised representations, 2021.

[24] Chun-Hsiao Yeh, Cheng-Yao Hong, Yen-Chi Hsu, Tyng-Luh Liu, Yubei Chen, and Yann LeCun. Decoupled contrastive learning, 2022.

[25] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[26] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation, 2022. URL https://arxiv.org/abs/2203.08679.

[27] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective, 2021. URL https://arxiv.org/abs/2102.00650.

[28] Yulei Niu, Long Chen, Chang Zhou, and Hanwang Zhang. Respecting transfer gap in knowledge distillation, 2022. URL https://arxiv.org/abs/2210.12787.

[29] Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation, 2024. URL https://arxiv.org/abs/2403.01427.

[30] Zheng Li, Xiang Li, Lingfeng Yang, Borui Zhao, Renjie Song, Lei Luo, Jun Li, and Jian Yang. Curriculum temperature for knowledge distillation, 2022. URL https://arxiv.org/abs/2211.16231.

[31] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher, 2022. URL https://arxiv.org/abs/2205.10536.

[32] Kaixiang Zheng and En-Hui Yang. Knowledge distillation based on transformed teacher matching, 2024. URL https://arxiv.org/abs/2402.11148.

[33] Jing Yang, Brais Martinez, Adrian Bulat, and Georgios Tzimiropoulos. Knowledge distillation via softmax regression representation learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=ZzwDy_wiWv.

[34] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[35] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1365–1374, 2019.

[36] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018.

[37] Jangho Kim, Seongwon Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems*, pages 2760–2769, 2018.

[38] Byeongho Heo, Minsik Lee, Seong Joon Yun, Jin Young Choi, and In So Kweon. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3779–3787, 2019.

[39] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. In *Advances in Neural Information Processing Systems*, pages 185–195, 2017.

[40] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[41] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation, 2019. URL https://arxiv.org/abs/1904.01866.

[42] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review, 2021. URL https://arxiv.org/abs/2104.09044.

[43] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier, 2022. URL https://arxiv.org/abs/2203.14001.

[44] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration, 2021. URL https://arxiv.org/abs/2012.03236.

[45] Xiaolong Liu, Lujun Li, Chao Li, and Anbang Yao. Norm: Knowledge distillation via n-to-one representation matching, 2023. URL https://arxiv.org/abs/2305.13803.

[46] Dongyang Liu, Meina Kan, Shiguang Shan, and Xilin Chen. Function-consistent feature distillation, 2023. URL https://arxiv.org/abs/2304.11832.

[47] Ziyao Guo, Haonan Yan, Hui Li, and Xiaodong Lin. Class attention transfer based knowledge distillation, 2023. URL https://arxiv.org/abs/2304.12777.

[48] Liqun Chen, Dong Wang, Zhe Gan, Jingjing Liu, Ricardo Henao, and Lawrence Carin. Wasserstein contrastive representation distillation, 2021. URL https://arxiv.org/abs/2012.08674.

[49] Jinguo Zhu, Shixiang Tang, Dapeng Chen, Shijie Yu, Yakun Liu, Aijun Yang, Mingzhe Rong, and Xiaohua Wang. Complementary relation contrastive distillation, 2021. URL https://arxiv.org/abs/2103.16367.

[50] Bobby He and Mete Ozay. Feature kernel distillation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=tBIQEvApZK5.

[51] Roy Miles, Adrian Lopez Rodriguez, and Krystian Mikolajczyk. Information theoretic representation distillation, 2022. URL https://arxiv.org/abs/2112.00459.

[52] Zhirong Wu, Yuanjun Xiong, Stella Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance-level discrimination, 2018.

[53] Mang Ye, Xu Zhang, Pong C. Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature, 2019.

[54] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, 2005. doi: 10.1109/CVPR.2005.202.

[55] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*, volume 2, page 1735–1742. IEEE, 2006. doi: 10.1109/cvpr.2006.100. URL http://dx.doi.org/10.1109/CVPR.2006.100.

[56] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020.

[57] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020.

[58] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

[59] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning, 2020.

[60] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers, 2021.

[61] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021.

[62] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2022.

[63] Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms, 2020.

[64] Nicolas Michel, Maorong Wang, Ling Xiao, and Toshihiko Yamasaki. Rethinking momentum knowledge distillation in online continual learning, 2024. URL https://arxiv.org/abs/2309.02870.

[65] Arun Mishra and Debbie Marr. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In *International Conference on Learning Representations*, 2017.

[66] Li Shen and Marios Savvides. Amalgamating knowledge towards comprehensive classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1687–1696, 2020.

[67] 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.

[68] Wei Liu, Andrew Rabinovich, and Alexander C Berg. Learning efficient single-stage pedestrian detectors by asymptotic localization fitting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 618–634, 2018.

[69] Gongfan Chen, Yuting Wang, Jiajun Xu, Zhe Du, Qionghai Dai, Shiyang Geng, and Tao Mei. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017.

[70] Alex Krizhevsky. Learning multiple layers of features from tiny images, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[71] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. URL https://api.semanticscholar.org/CorpusID:57246310.

[72] Adam Coates and Andrew Y. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 921–928, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

[73] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[74] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[75] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017.

[76] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015.

[77] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.

[78] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018.

[79] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[80] Xueqing Deng, Dawei Sun, Shawn Newsam, and Peng Wang. Distpro: Searching a fast knowledge distillation process via meta optimization, 2022. URL https://arxiv.org/abs/2204.05547.

[81] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 116–131, 2018.

[82] Roy Miles and Krystian Mikolajczyk. Understanding the role of the projector in knowledge distillation, 2024. URL https://arxiv.org/abs/2303.11098.

[83] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

Appendix

A Algorithm

Algorithm 1 provides the pseudo-code of DCD.

Algorithm 1 Pseudocode of DCD in a PyTorch-like style.

```
# f_s, f_t: student and teacher networks
# t_s, t_t: temperature for student and teacher
# tau: learnable temperature parameter
# b: learnable bias parameter
# max_tau: maximum temperature value

for x in loader: # load a minibatch x with N samples
    s = f_s.forward(x) # student embeddings: NxC
    s = normalize(s, dim=1) # L2 normalization

    with torch.no_grad(): # no gradients for teacher
        t = f_t.forward(x) # teacher embeddings: NxC
        t = normalize(t, dim=1) # L2 normalization

    # bound temperature
    tau = tau.exp().clamp(0, max_tau)

    # contrastive loss
    logits = torch.mm(f_s, f_t.t()) * tau + b
    labels = torch.arange(N) # diagonal labels
    contrastive_loss = cross_entropy(logits, labels)

    # consistent loss
    p1 = log_softmax(logits, dim=1)
    p2 = softmax(logits, dim=0)
    consistent_loss = kl_div(p1, p2)

    # total loss
    loss = contrastive_loss + 0.5 * consistent_loss

    # SGD update: student network and parameters
    loss.backward()
    update(f_s.params)
    update(tau) # update temperature
    update(b) # update bias
```

`mm`: matrix multiplication; `softmax`: exponential normalization across dimensions; `log_softmax`: logarithm of softmax; `cross_entropy`: negative log likelihood loss; `kl_div`: Kullback-Leibler divergence between distributions

B Implementation Details

We implement DCD in PyTorch following the protocol of [16][2]. This protocol has become a common standard and is widely used among many papers to demonstrate their knowledge distillation methods [1; 12; 13; 35; 15; 40; 34; 36; 38; 37; 14; 39; 16]. This approach is consistent with the standard practice in the field, allowing for fair comparison across methods. For visualization, we follow the implementation of [29][3]. For efficiency we use the MDistiller framework [26][4].

B.1 Baseline Methods

We compare our approach to the following state-of-the-art methods from the literature: (1) Knowledge Distillation (KD) [1]; (2) FitNets: Hints for Thin Deep Nets [12]; (3) Attention Transfer (AT) [13]; (4) Similarity-Preserving Knowledge Distillation (SP) [35]; (5) Correlation Congruence (CC) [15]; (6) Variational Information Distillation for Knowledge Transfer (VID) [40]; (7) Relational Knowledge Distillation (RKD) [34]; (8) Learning Deep Representations with Probabilistic Knowledge Transfer (PKT) [36]; (9) Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons (AB) [38]; (10) Paraphrasing Complex Network: Network Compression via Factor Transfer (FT) [37]; (11) A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning (FSP) [14]; (12) Like What You Like: Knowledge

[2] Available at: https://github.com/HobbitLong/RepDistiller.
[3] Available at: https://github.com/sunshangquan/logit-standardization-KD.
[4] Available at: https://github.com/megvii-research/mdistiller

Distill via Neuron Selectivity Transfer (NST) [39]; (13) Contrastive Representation Distillation (CRD) [16]; (14) A Comprehensive Overhaul of Feature Distillation (OFD) [41]; (15) Rethinking Soft Labels for Knowledge Distillation: A Bias-Variance Tradeoff Perspective (WSLD) [27]; (16) Respecting Transfer Gap in Knowledge Distillation (IPWD) [28]; (17) Knowledge Distillation via Softmax Regression Representation Learning (SRRL) [33]; (18) Cross-Layer Distillation with Semantic Calibration (SemCKD) [44]; (19) Distilling Knowledge via Knowledge Review (ReviewKD) [42]; (20) Knowledge Distillation with the Reused Teacher Classifier (SimKD) [43]; (21) Searching A Fast Knowledge Distillation Process via Meta Optimization (DistPro) [80]; (22) Knowledge Distillation via N-to-One Representation Matching (NORM) [45]; (23) Wasserstein Contrastive Representation Distillation (WCoRD) [48]; (24) Complementary Relation Contrastive Distillation (CRCD) [49]; (25) Feature Kernel Distillation (FKD) [50]; (26) Information Theoretic Representation (ITRD) [51]; (27) Test-Time Memory Distillation (TTM) [32]; (28) Weighted Test-Time Memory Distillation (WTTM) [32]; (29) Decoupled Knowledge Distillation (DKD) [26]; (30) Feature Correlation-based Feature Distillation (FCFD) [46]; (31) Category-aligned Knowledge Distillation (CAT-KD) [47]; (32) Distillation with Instance-wise Selective Teaching (DIST) [31]; (33) Cross-layer Transfer Knowledge Distillation (CTKD) [30].

B.2 Datasets

We take five widely researched datasets: (1) CIFAR-100 [70] is a standard benchmark for knowledge distillation and contains 50,000 training images of size 32×32 with 500 images per class and 10,000 test images. (2) ImageNet ILSVRC-2012 [71], which is more challenging that CIFAR, and includes 1.2 million images from 1,000 classes for training and 50,000 for validation. (3) STL-10 [72] consists of a training set of 5,000 labeled images from 10 classes, and a test set of 8,000 images. (4) Tiny ImageNet [71] has 200 classes, each with 500 training images and 50 validation images. (5) MS-COCO [73] is an 80-category general object detection dataset. The `train2017` split contains 118,000 images, and the `val2017` split contains 5,000 images.

B.3 Network Architectures

We use the following network architectures as described in [16]: (1) Wide Residual Network (WRN) [75], where WRN-d-w represents a wide ResNet with depth d and width factor w; (2) ResNet (rn) [74], where rn-d represents a CIFAR-style ResNet with 3 groups of basic blocks having 16, 32, and 64 channels, respectively, and rn-8 $\times 4$ and rn-32 $\times 4$ indicate a 4-times wider network with 64, 128, and 256 channels; (3) ResNet (RN) [74], where RN-d represents an ImageNet-style ResNet with Bottleneck blocks and more channels; (4) MobileNet-v2 (MN-v2) [77], using a width multiplier of 0.5 in our experiments; (5) VGG [76], where the VGG network used is adapted from its original ImageNet counterpart; and (6) ShuffleNet-v1 (SN-v1) [78] and ShuffleNet-v2 (SN-v2) [81], which are adapted for efficient training with input sizes of 32×32. To ensure dimensional consistency and preserve relational information during feature alignment, both teacher and student features are passed through lightweight *projection heads*—two-layer MLPs (512 hidden, 128 output)—that nonlinearly project intermediate representations into a shared embedding space for computing relational similarity distributions, while being discarded after training to avoid any additional inference cost. This projection head facilitates knowledge transfer by implicitly encoding relational information from previous samples [82].

B.4 Optimization

We closely follow the protocol of [16]. All methods evaluated in our experiments use SGD with 0.9 Nesterov momentum. For CIFAR-100, we initialize the learning rate as 0.05, and decay it by 0.1 every 30 epochs after the first 150 epochs until the last 240 epoch. For MobileNet-v2, ShuffleNet-v1, and ShuffleNet-v2, we use a learning rate of 0.01 as this learning rate is optimal for these models in a grid search, while 0.05 is optimal for other models. The batch size is set to 64 for CIFAR-100, and the weight decay is set to 5×10^{-4}. For ImageNet[5], the initial learning rate is set to 0.1 and then divided by 10 at the 30th, 60th, and 90th epochs of the total 120 training epochs. The mini-batch size is set to 256, and the weight decay is set to 1×10^{-4}. All results are reported as means over five trials, except for the results on ImageNet, which are reported in a single trial. Our implementation for MS-COCO follows the settings in [26]. We use the two-stage method Faster R-CNN [4] with Feature Pyramid Network (FPN) [5] as the detection framework. We

[5]Available at: https://www.image-net.org/.

evaluate three teacher–student settings: ResNet-101 to ResNet-18, ResNet-101 to ResNet-50, and ResNet-50 to MobileNet-V2 [77]. All students are trained with the 1× scheduler (schedulers and task-specific loss weights follow Detectron2 [83]). We take one RTX 6000 GPU to train the model on CIFAR-100 and four L40 GPUs on ImageNet and MS-COCO.

C Extended Results

C.1 CIFAR-100

Table 6 and Table 7 provide a comprehensive overview of the top-1 accuracies of student networks trained with various state-of-the-art distillation techniques across a wide range of teacher-student architectural combinations. Our method benefits from its simplicity, as it has no trainable parameters, and the only hyperparameters involved are the loss coefficients.

C.2 Inter-class Correlations

Figure 5 compares the correlation matrix differences between teacher (WRN-40-2) and student (WRN-40-1) logits on CIFAR-100. Our method achieves better alignment of correlation structures compared to models trained without distillation or with alternative methods.

C.3 t-SNE

Figures 6 and 7 presents t-SNE visualizations of embeddings from the teacher (WRN-40-2) and student (WRN-40-1) networks on CIFAR-100. Specifically, Figure 6 shows t-SNE visualization focused on just the first 20 classes for clearer interpretation, and Figure 7 has the same t-SNE visualizations but for all 100 classes of CIFAR-100.

D Broader Impact

The presented research should be categorized as research in the field of knowledge distillation. The primary goal is to reduce computational demands, thereby lowering the energy requirements of AI systems and contributing to more sustainable technology deployment. However, this technique also harbors risks, notably the potential to perpetuate existing biases present in teacher models. Such biases could have profound ethical implications, as in sensitive applications. Furthermore, the versatility of the algorithms developed here enables their application across a broad spectrum of vision-related tasks, but this versatility also introduces the dual-use dilemma, where the technology might yield both beneficial and adverse impacts. Thus, careful consideration is needed when deploying these methods to ensure they align with ethical guidelines and promote fairness in AI applications.

E Reproducibility Statement

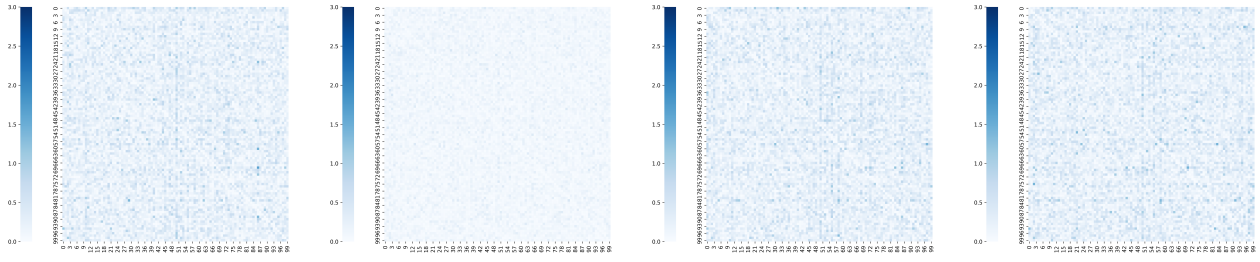
Our code is publicly available at `https://github.com/giakoumoglou/rrd`. The implementation builds on the RepDistiller [16] codebase, a widely adopted standard for knowledge distillation evaluation, and follows its experimental protocol to ensure fair comparison across all baselines. Pseudocode is provided in Algorithm 1 of Section A and all hyperparameters are reported in Section B.

Table 6 Results on CIFAR-100 (same architecture). Test top-1 accuracy (%) of student networks for same-architecture distillation. ↑ and ↓ denote performance relative to KD. Results for our method are averaged over five independent runs. Abbreviations are defined in Section B.1.

Teacher Student	WRN-40-2 WRN-16-2	WRN-40-2 WRN-40-1	rn-56 rn-20	rn-110 rn-20	rn-110 rn-32	rn-32x4 rn-8x4	VGG-13 VGG-8
Teacher	75.61	75.61	72.34	74.31	74.31	79.42	74.64
Student	73.26	71.98	69.06	69.06	71.14	72.50	70.36
KD [1]	74.92	73.54	70.66	70.67	73.08	73.33	72.98
FitNet [12]	73.58 (↓)	72.24 (↓)	69.21 (↓)	68.99 (↓)	71.06 (↓)	73.50 (↑)	71.02 (↓)
AT [13]	74.08 (↓)	72.77 (↓)	70.55 (↓)	70.22 (↓)	72.31 (↓)	73.44 (↑)	71.43 (↓)
SP [35]	73.83 (↓)	72.43 (↓)	69.67 (↓)	70.04 (↓)	72.69 (↓)	72.94 (↓)	72.68 (↓)
CC [15]	73.56 (↓)	72.21 (↓)	69.63 (↓)	69.48 (↓)	71.48 (↓)	72.97 (↓)	70.81 (↓)
VID [40]	74.11 (↓)	73.30 (↓)	70.38 (↓)	70.16 (↓)	72.61 (↓)	73.09 (↓)	71.23 (↓)
RKD [34]	73.35 (↓)	72.22 (↓)	69.61 (↓)	69.25 (↓)	71.82 (↓)	71.90 (↓)	71.48 (↓)
PKT [36]	74.54 (↓)	73.45 (↓)	70.34 (↓)	70.25 (↓)	72.61 (↓)	73.64 (↑)	72.88 (↓)
AB [38]	72.50 (↓)	72.38 (↓)	69.47 (↓)	69.53 (↓)	70.98 (↓)	73.17 (↓)	70.94 (↓)
FT [37]	73.25 (↓)	71.59 (↓)	69.84 (↓)	70.22 (↓)	72.37 (↓)	72.86 (↓)	70.58 (↓)
FSP [14]	72.91 (↓)	n/a	69.95 (↓)	70.11 (↓)	71.89 (↓)	72.62 (↓)	70.33 (↓)
NST [39]	73.68 (↓)	72.24 (↓)	69.60 (↓)	69.53 (↓)	71.96 (↓)	73.30 (↓)	71.53 (↓)
CRD [16]	75.48 (↑)	74.14 (↑)	71.16 (↑)	71.46 (↑)	73.48 (↑)	75.51 (↑)	73.94 (↑)
CRD+KD [16]	75.64 (↑)	74.38 (↑)	71.63 (↑)	71.56 (↑)	73.75 (↑)	75.46 (↑)	74.29 (↑)
OFD [41]	75.24 (↑)	74.33 (↑)	70.38 (↓)	n/a	73.23 (↑)	74.95 (↑)	73.95 (↑)
WSLD [27]	n/a	73.74 (↑)	71.53 (↑)	n/a	73.36 (↑)	74.79 (↑)	n/a
IPWD [28]	n/a	74.64 (↑)	71.32 (↑)	n/a	73.91 (↑)	76.03 (↑)	n/a
SRRL [33]	n/a	74.64 (↑)	n/a	n/a	n/a	75.39 (↑)	n/a
SemCKD [44]	n/a	74.41 (↑)	n/a	n/a	n/a	76.23 (↑)	n/a
ReviewKD [42]	76.12 (↑)	75.09 (↑)	71.89 (↑)	n/a	73.89 (↑)	75.63 (↑)	74.84 (↑)
SimKD [43]	n/a	75.56 (↑)	n/a	n/a	n/a	78.08 (↑)	n/a
DistPro [80]	76.36 (↑)	n/a	72.03 (↑)	n/a	73.74 (↑)	n/a	n/a
NORM [45]	75.65 (↑)	74.82 (↑)	71.35 (↑)	71.55 (↑)	73.67 (↑)	76.49 (↑)	73.95 (↑)
NORM+KD [45]	76.26 (↑)	75.42 (↑)	71.61 (↑)	72.00 (↑)	74.95 (↑)	76.98 (↑)	74.46 (↑)
NORM+CRD [45]	76.02 (↑)	75.37 (↑)	71.51 (↑)	71.90 (↑)	73.81 (↑)	76.49 (↑)	73.58 (↑)
WCoRD [48]	75.88 (↑)	74.73 (↑)	71.56 (↑)	71.57 (↑)	73.81 (↑)	75.95 (↑)	74.55 (↑)
WCoRD+KD [48]	76.11 (↑)	74.72 (↑)	71.92 (↑)	71.88 (↑)	74.20 (↑)	76.15 (↑)	74.72 (↑)
CRCD [49]	76.67 (↑)	75.95 (↑)	73.21 (↑)	72.33 (↑)	74.98 (↑)	76.42 (↑)	74.97 (↑)
FKD [50]	n/a	n/a	n/a	n/a	n/a	75.57 (↑)	73.78 (↑)
ITRD (corr) [51]	75.85 (↑)	74.90 (↑)	71.45 (↑)	71.77 (↑)	74.02 (↑)	75.63 (↑)	74.70 (↑)
ITRD (corr+mi) [51]	76.12 (↑)	75.18 (↑)	71.47 (↑)	71.99 (↑)	74.26 (↑)	76.19 (↑)	74.93 (↑)
TTM [32]	76.23 (↑)	74.32 (↑)	71.83 (↑)	71.46 (↑)	73.97 (↑)	76.17 (↑)	74.33 (↑)
WTTM [32]	76.37 (↑)	74.58 (↑)	71.92 (↑)	71.67 (↑)	74.13 (↑)	76.06 (↑)	74.44 (↑)
WTTM+CRD [32]	76.61 (↑)	74.94 (↑)	72.20 (↑)	72.13 (↑)	74.52 (↑)	76.65 (↑)	74.71 (↑)
WTTM+ITRD [32]	76.65 (↑)	75.34 (↑)	72.16 (↑)	72.20 (↑)	74.36 (↑)	77.36 (↑)	75.13 (↑)
DKD [26]	76.24 (↑)	74.81 (↑)	71.97 (↑)	n/a	74.11 (↑)	76.32 (↑)	74.68 (↑)
FCFD [46]	76.34 (↑)	75.43 (↑)	71.68 (↑)	n/a	n/a	76.80 (↑)	74.86 (↑)
FCFD+KD [46]	76.43 (↑)	75.46 (↑)	71.96 (↑)	n/a	n/a	76.62 (↑)	75.22 (↑)
CAT-KD [47]	75.60 (↑)	74.82 (↑)	71.62 (↑)	n/a	73.62 (↑)	76.91 (↑)	74.65 (↑)
DIST [31]	n/a	74.73 (↑)	71.75 (↑)	n/a	n/a	76.31 (↑)	n/a
CTKD [30]	75.45 (↑)	73.93 (↑)	71.19 (↑)	70.99 (↑)	73.52 (↑)	n/a	73.52 (↑)
DCD (ours)	74.99 (↑)	73.69 (↑)	71.18 (↑)	71.00 (↑)	73.12 (↑)	74.23 (↑)	73.22 (↑)
DCD+KD (ours)	76.06 (↑)	74.76 (↑)	71.81 (↑)	72.03 (↑)	73.62 (↑)	75.09 (↑)	73.95 (↑)

Table 7 Results on CIFAR-100 (cross architecture). Test top-1 accuracy (%) for teacher-student pairs with different architectures. ↑ and ↓ denote performance relative to KD. Results for our method are averaged over five independent runs. Abbreviations are defined in Section B.1.

Teacher Student	VGG-13 MN-v2	RN-50 MN-v2	RN-50 VGG-8	RN-32x4 SN-v1	RN-32x4 SN-v2	WRN-40-2 SN-v1
Teacher	74.64	79.34	79.34	79.42	79.42	75.61
Student	64.60	64.60	70.36	70.5	71.82	70.5
KD [1]	67.37	67.35	73.81	74.07	74.45	74.83
FitNet [12]	64.14 (↓)	63.16 (↓)	70.69 (↓)	73.59 (↓)	73.54 (↓)	73.73 (↓)
AT [13]	59.40 (↓)	58.58 (↓)	71.84 (↓)	71.73 (↓)	72.73 (↓)	73.32 (↓)
SP [35]	66.30 (↓)	68.08 (↑)	73.34 (↓)	73.48 (↓)	74.56 (↑)	74.52 (↓)
CC [15]	64.86 (↓)	65.43 (↓)	70.25 (↓)	71.14 (↓)	71.29 (↓)	71.38 (↓)
VID [40]	65.56 (↓)	67.57 (↑)	70.30 (↓)	73.38 (↓)	73.40 (↓)	73.61 (↓)
RKD [34]	64.52 (↓)	64.43 (↓)	71.50 (↓)	72.28 (↓)	73.21 (↓)	72.21 (↓)
PKT [36]	67.13 (↓)	66.52 (↓)	73.01 (↓)	74.10 (↑)	74.69 (↑)	73.89 (↓)
AB [38]	66.06 (↓)	67.20 (↓)	70.65 (↓)	73.55 (↓)	74.31 (↓)	73.34 (↓)
FT [37]	61.78 (↓)	60.99 (↓)	70.29 (↓)	71.75 (↓)	72.50 (↓)	72.03 (↓)
NST [39]	58.16 (↓)	64.96 (↓)	71.28 (↓)	74.12 (↑)	74.68 (↑)	76.09 (↑)
CRD [16]	69.73 (↑)	69.11 (↑)	74.3 (↑)	75.11 (↑)	75.65 (↑)	76.05 (↑)
CRD+KD [16]	69.94 (↑)	69.54 (↑)	74.58 (↑)	75.12 (↑)	76.05 (↑)	76.27 (↑)
OFD [41]	69.48 (↑)	69.04 (↑)	n/a	75.98 (↑)	76.82 (↑)	75.85 (↑)
WSLD [27]	n/a	68.79 (↑)	73.80 (↓)	75.09 (↑)	n/a	75.23 (↑)
IPWD [28]	n/a	70.25 (↑)	74.97 (↑)	76.03 (↑)	n/a	76.44 (↑)
SRRL [33]	n/a	n/a	n/a	75.18 (↑)	n/a	n/a
SemCKD [44]	n/a	n/a	n/a	n/a	77.62 (↑)	n/a
ReviewKD [42]	70.37 (↑)	69.89 (↑)	n/a	77.45 (↑)	77.78 (↑)	77.14 (↑)
SimKD [43]	n/a	n/a	n/a	77.18 (↑)	n/a	n/a
DistPro [80]	n/a	n/a	n/a	77.18 (↑)	77.54 (↑)	77.24 (↑)
NORM [45]	68.94 (↑)	70.56 (↑)	75.17 (↑)	77.42 (↑)	78.07 (↑)	77.06 (↑)
NORM+KD [45]	69.38 (↑)	71.17 (↑)	75.67 (↑)	77.79 (↑)	78.32 (↑)	77.63 (↑)
NORM+CRD [45]	69.17 (↑)	71.08 (↑)	75.51 (↑)	77.50 (↑)	77.96 (↑)	77.09 (↑)
WCoRD [48]	69.47 (↑)	70.45 (↑)	74.86 (↑)	75.40 (↑)	75.96 (↑)	76.32 (↑)
WCoRD+KD [48]	70.02 (↑)	70.12 (↑)	74.68 (↑)	75.77 (↑)	76.48 (↑)	76.68 (↑)
CRCD [49]	n/a	n/a	n/a	n/a	n/a	n/a
FKD [50]	n/a	n/a	74.61 (↑)	75 (↑)	n/a	n/a
ITRD (corr) [51]	69.97 (↑)	71.41 (↑)	75.71 (↑)	76.8 (↑)	77.27 (↑)	77.35 (↑)
ITRD (corr+mi) [51]	70.39 (↑)	71.34 (↑)	75.49 (↑)	76.91 (↑)	77.40 (↑)	77.09 (↑)
TTM [32]	68.98 (↑)	69.24 (↑)	74.87 (↑)	74.18 (↑)	76.57 (↑)	75.39 (↑)
WTTM [32]	69.16 (↑)	69.59 (↑)	74.82 (↑)	74.37 (↑)	76.55 (↑)	75.42 (↑)
WTTM+CRD [32]	70.30 (↑)	70.84 (↑)	75.30 (↑)	75.82 (↑)	77.04 (↑)	76.86 (↑)
WTTM+ITRD [32]	70.70 (↑)	71.56 (↑)	76.00 (↑)	77.03 (↑)	77.68 (↑)	77.44 (↑)
DKD [26]	69.71 (↑)	70.35 (↑)	n/a	76.45 (↑)	77.07 (↑)	76.70 (↑)
FCFD [46]	70.67 (↑)	71.07 (↑)	n/a	78.12 (↑)	78.20 (↑)	77.81 (↑)
FCFD+KD [46]	70.65 (↑)	71.00 (↑)	n/a	78.12 (↑)	78.18 (↑)	77.99 (↑)
CAT-KD [47]	69.13 (↑)	71.36 (↑)	n/a	78.26 (↑)	78.41 (↑)	77.35 (↑)
DIST [31]	n/a	68.66 (↑)	n/a	76.34 (↑)	77.35 (↑)	n/a
CTKD [30]	68.46 (↑)	68.47 (↑)	n/a	74.78 (↑)	75.31 (↑)	75.78 (↑)
DCD (ours)	68.35 (↑)	67.39 (↑)	73.85 (↑)	74.26 (↑)	75.26 (↑)	74.98 (↑)
DCD+KD (ours)	69.77 (↑)	70.03 (↑)	74.08 (↑)	76.01 (↑)	76.95 (↑)	76.51 (↑)

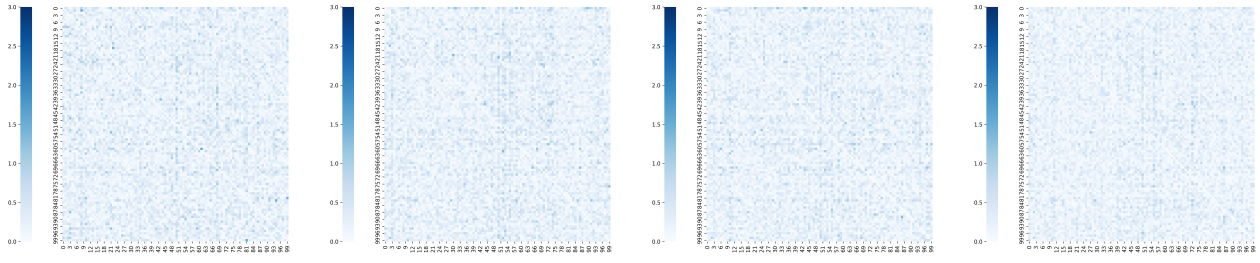


(a) Vanilla
Mean: 0.24, Max: 1.66

(b) KD [1]
Mean: 0.09, Max: 0.49

(c) FitNets [12]
Mean: 0.25, Max: 1.38

(d) AT [13]
Mean: 0.26, Max: 1.43

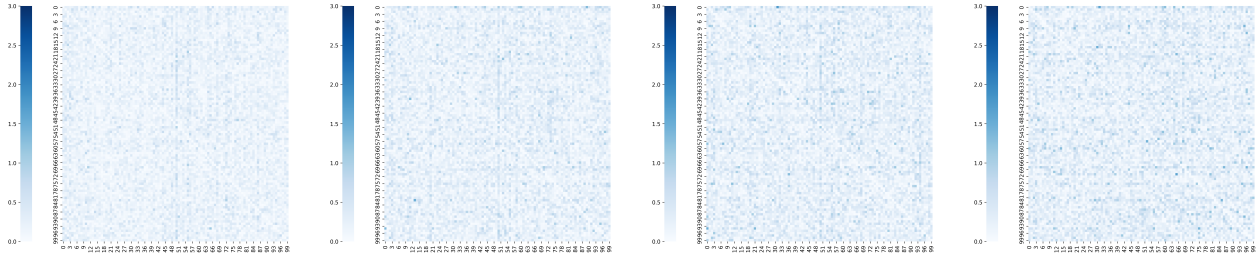


(e) SP [35]
Mean: 0.25, Max: 1.57

(f) CC [15]
Mean: 0.234, Max: 1.37

(g) VID [40]
Mean: 0.23, Max: 1.46

(h) RKD [34]
Mean: 0.21, Max: 1.24

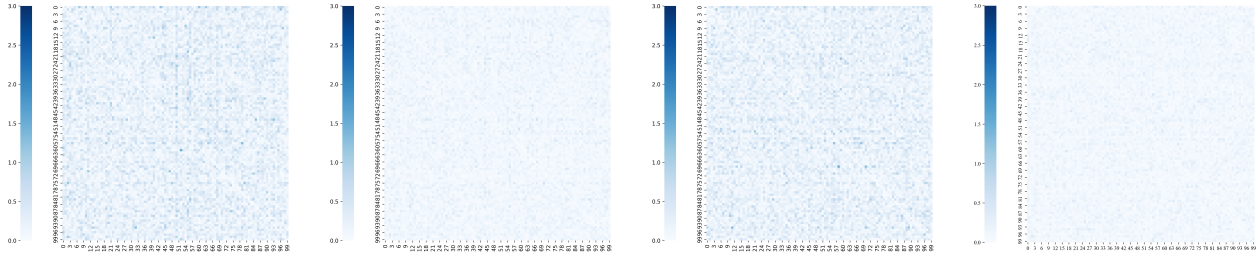


(i) PKT [36]
Mean: 0.19, Max: 1.14

(j) AB [38]
Mean: 0.24, Max: 1.49

(k) FT [37]
Mean: 0.26, Max: 1.76

(l) NST [39]
Mean: 0.28, Max: 1.65



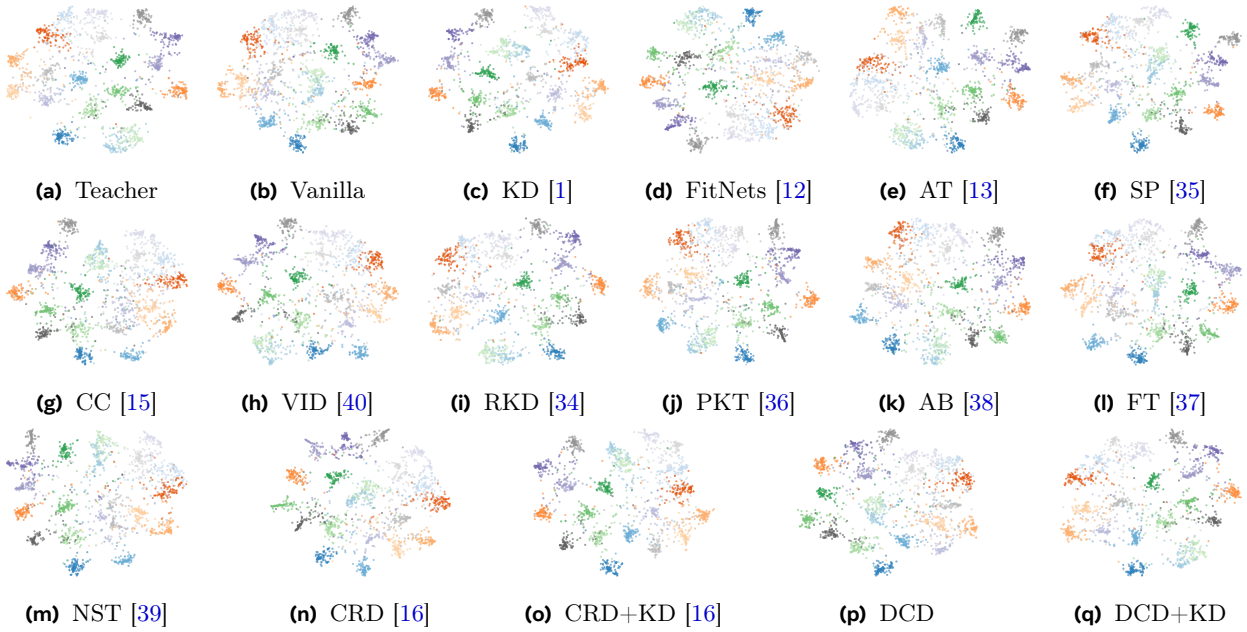
(m) CRD [16]
Mean: 0.23, Max: 1.56

(n) CRD+KD [16]
Mean: 0.10, Max: 0.57

(o) DCD
Mean: 0.18, Max: 0.99

(p) DCD+KD
Mean: 0.07, Max: 0.55

Figure 5 Logit correlation analysis. Matrix of the average logit difference between teacher and student outputs on CIFAR-100 (lower values indicate higher similarity). Results are based on a WRN-40-2 teacher and a WRN-40-1 student. The abbreviations are listed in Section B.1.

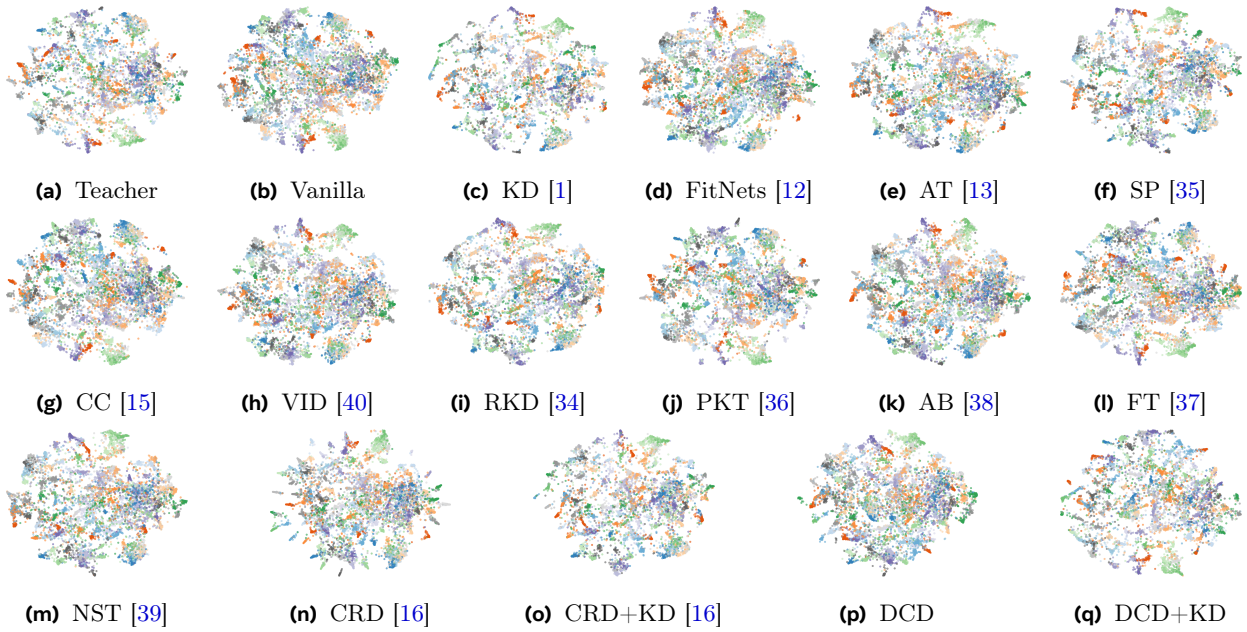


(a) Teacher **(b)** Vanilla **(c)** KD [1] **(d)** FitNets [12] **(e)** AT [13] **(f)** SP [35]

(g) CC [15] **(h)** VID [40] **(i)** RKD [34] **(j)** PKT [36] **(k)** AB [38] **(l)** FT [37]

(m) NST [39] **(n)** CRD [16] **(o)** CRD+KD [16] **(p)** DCD **(q)** DCD+KD

Figure 6 t-SNE visualization of embedding spaces. Comparison of feature distributions from the teacher and student networks for the first 20 classes of CIFAR-100. We use a WRN-40-2 teacher and a WRN-40-1 student for the visualization. The abbreviations are listed in Section B.1.



(a) Teacher **(b)** Vanilla **(c)** KD [1] **(d)** FitNets [12] **(e)** AT [13] **(f)** SP [35]

(g) CC [15] **(h)** VID [40] **(i)** RKD [34] **(j)** PKT [36] **(k)** AB [38] **(l)** FT [37]

(m) NST [39] **(n)** CRD [16] **(o)** CRD+KD [16] **(p)** DCD **(q)** DCD+KD

Figure 7 t-SNE visualization of embedding spaces. Comparison of feature distributions from the teacher and student networks for all 100 classes of CIFAR-100. We use a WRN-40-2 teacher and a WRN-40-1 student for the visualization. The abbreviations are listed in Section B.1.