You Don't Need Aligned Representations: Knowledge Distillation via Random Prototype Spaces

Hamza Etcibasi [1,2] **Ramazan Gokberk Cinbis** [1,3]
[1]Department of Computer Engineering, Middle East Technical University, Ankara, Türkiye
[2]ASELSAN Inc., Ankara, Türkiye
[3]ROMER Robotics and AI Center, Middle East Technical University, Ankara, Türkiye
hamza.etcibasi@metu.edu.tr haetcibasi@aselsan.com gcinbis@ceng.metu.edu.tr

Abstract

A widely held assumption in knowledge distillation is that teacher and student representations must occupy a compatible or explicitly aligned feature space. We propose RPKD (**R**andom **P**rototype **K**nowledge **D**istillation), which discards this assumption by comparing both networks against a shared bank of randomly initialized, frozen prototype vectors. Alignment is therefore carried out in an external reference space rather than in the teacher's native feature coordinates: the teacher side is projected by fixed random maps that carry no trainable parameters, and only the student adapts. Two branches handle the transfer: a logit branch aligning global prototype similarity distributions and a feature branch matching spatially-resolved prototype activation maps. Both operate within the same fixed prototype space, which can be interpreted as a random feature embedding that approximately preserves inter-sample similarity structure, so the prototype bank and the distillation objective stay unchanged across teacher–student pairs while only the input projections are instantiated per architecture. In low-category settings, decoupling the prototype vocabulary from the task label space yields richer supervision than the label space alone, an advantage absent from logit-based methods, whose supervisory signal collapses as class count shrinks. This advantage is especially relevant in real-world scenarios where the label space is typically constrained. Extensive experiments demonstrate the effectiveness of RPKD, achieving a maximum gain of $+\mathbf{7.98}\%$ over OFA on CIFAR-100, $+\mathbf{12.44}\%$ on ImageNet-100, and $+\mathbf{1.28}\%$ on ImageNet-1K. Source code is available at `https://github.com/HamzaEtcibasi/RPKD`.

1 Introduction

Knowledge distillation (KD) [1, 2] transfers the predictive behavior of a large *teacher* network into a compact *student*, serving as a practical compression technique alongside pruning, quantization, and low-rank decomposition in standard training pipelines. Hinton et al. [1] align student and teacher output distributions via temperature-scaled softmax logits, whose soft targets encode inter-class similarity structure (the so-called *dark knowledge*) that hard labels discard.

The literature has since diverged into two directions. *Logit-based methods* [3–9] revisit output-level objectives, showing that iterative, collaborative, or standardized soft-target losses can match feature-level transfer with far less overhead. *Feature-based methods* [10–23] transfer intermediate representations—hints, attention maps, activation boundaries, relational structures, and contrastive embeddings—providing supervisory signal richer than soft labels alone. The strongest results come from combining both [24, 25].

Nearly all these methods assume teacher and student belong to the same architectural family. This assumption is frequently violated in practice. Modern visual recognition spans convolutional net-

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

works [26–28], vision transformers [29–31], and MLP-based models [32, 33], each with different inductive biases, feature dimensionalities, and spatial representations. Distilling across these boundaries is not a niche scenario: an organization deploying a large pretrained ViT often has no same-family teacher for its target CNN student, making cross-architecture KD the only path short of retraining from scratch. The obstacle is geometric. Heterogeneous feature spaces differ in dimensionality, spatial structure, and statistical properties, so standard feature-based KD is inapplicable without alignment modules that add parameters and risk introducing misalignment rather than resolving it.

Existing methods address this through *learned* alignment [34, 25, 35, 36]. OFA [25] maps student features into the class logit space via learned stage-wise projectors; subsequent work refines this direction through redundancy suppression [35] or representation fusion before transfer [36]. Despite their effectiveness, these approaches share a structural limitation: alignment modules are architecture-specific and pair-dependent, and methods tied to the class logit space suffer supervisory signal collapse as the number of task classes shrinks.

We ask a more fundamental question: *does the shared alignment space itself need to be learned, or can a fixed random space suffice?* We propose **RPKD** (**R**andom **P**rototype **K**nowledge **D**istillation), which transfers knowledge through a shared set of randomly initialized, *frozen* prototype vectors entirely decoupled from class semantics. Both teacher and student are projected into this fixed random prototype space independently: the teacher via frozen random linear projectors that require no training, the student via a learnable MLP that adapts to match the teacher's *fixed* prototype similarity patterns. Because the shared space and all teacher-side projections are fixed at initialization, the student faces a stable, well-defined alignment target throughout training with no joint teacher–student optimization required, and the shared space has fixed dimension M regardless of the number of task classes, avoiding the supervisory collapse that plagues class-logit-based methods in low-category settings. Distillation proceeds through two complementary branches: a *logit branch* that aligns global prototype similarity distributions, and a *feature branch* that matches spatially-resolved prototype activation maps, jointly transferring semantic and structural knowledge within the same random space.

Crucially, the entire teacher-side construction reduces to just two lightweight frozen random linear projectors—one for global features and one for spatial features—requiring no training whatsoever; only the student's MLP adapter is optimized. This minimal footprint makes RPKD immediately applicable to a new teacher–student pair: neither the objective nor the prototype space has to be redesigned, and the teacher side needs no training at all. A notable practical advantage of this design emerges in low-category settings: because the prototype vocabulary is decoupled from the number of task classes, it provides consistently rich supervisory signal even when few class labels are available, a condition common in domain-specific and real-world deployment scenarios. Our contributions are:

- **RPKD**: a cross-architecture distillation framework using a frozen random prototype space, in which the shared space and all teacher-side projections are fixed at initialization and no design choice depends on the number of task classes.
- **Dual-path objective**: a logit branch over global prototype similarities and a feature branch over spatial activation maps, transferring both semantic and structural knowledge within a single shared space.
- **Low-category robustness**: an empirical and theoretical motivation for why prototype spaces provide richer, more stable supervision than the task label space when class counts are small.
- **State-of-the-art results**: up to $+7.98\%$ over OFA on CIFAR-100, up to $+12.44\%$ on ImageNet-100, and $+1.28\%$ on ImageNet-1K, with consistent gains across CNN and ViT architectural families and few-class subsets.

2 Related work

2.1 Vision architecture families

Convolutional neural networks (CNNs). Convolutional neural networks [26–28] have long dominated visual recognition through local connectivity and weight sharing. Residual connections [26] enable very deep networks, depthwise-separable convolutions [27] yield highly efficient variants, and ConvNeXt [28] modernizes the CNN design space with ideas borrowed from transformers. CNNs remain the architecture of choice for resource-constrained deployment thanks to decades of hardware optimization.

Vision transformers (ViTs). The self-attention mechanism, originally proposed for natural language processing [37], was adapted to vision by the Vision Transformer (ViT) [29], which splits an image into fixed-size non-overlapping patches, linearly embeds each patch into a token, and processes the resulting sequence with a standard transformer encoder. This formulation showed that a pure self-attention backbone matches or surpasses CNNs at scale. DeiT [31] addresses data efficiency through a dedicated distillation token, while Swin Transformer [30] introduces hierarchical windowed attention that recovers spatial locality and supports dense prediction. Beyond classification, ViTs have become the backbone of choice across detection, segmentation, generation, and vision-language tasks [38–42], yet their compute overhead motivates their use as teachers rather than deployed students.

Multi-layer perceptrons (MLPs). MLP-Mixer [32] and ResMLP [33] demonstrate competitive image classification using only dense matrix multiplications, replacing both convolutions and self-attention with alternating token-mixing and channel-mixing layers. These models represent a third architectural paradigm with distinct inductive biases, further enriching the heterogeneous landscape in which cross-architecture distillation must operate.

2.2 Knowledge distillation

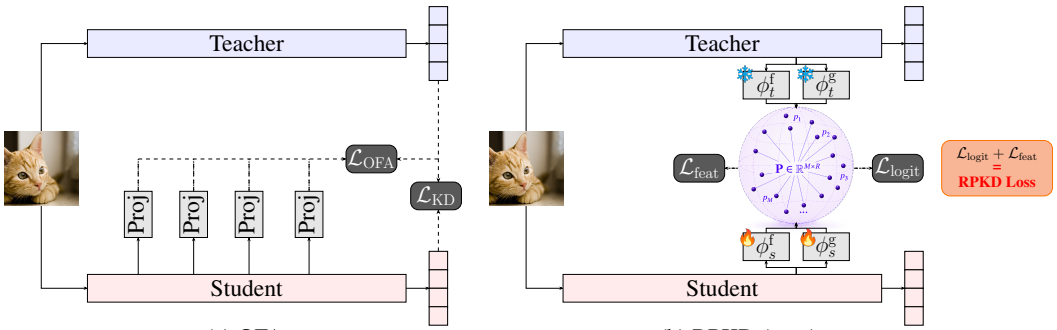
Logit-based distillation. Hinton et al. [1] showed that temperature-scaled output distributions carry rich inter-class similarity structure as soft targets. Born-Again Networks [3] demonstrate that iterative self-distillation steadily improves accuracy; Deep Mutual Learning [4] and densely guided distillation [6] extend this to online and multi-teacher settings. More recently, DIST [7] replaces exact distribution matching with correlation-based objectives; DKD [8] decouples the KL loss into target and non-target terms; and LSKD [9] introduces logit standardization to remove scale sensitivity. Because output distributions are defined over a shared class vocabulary regardless of architecture, logit-based methods are inherently architecture-agnostic; their limitation is that the supervisory signal is weaker than that of feature-level methods. Combined approaches [24] that stack both objectives achieve the strongest results, but reintroduce the feature alignment challenge when teacher and student architectures differ.

Feature-based distillation. Feature-based methods transfer knowledge through intermediate representations beyond the final output. FitNet [10] introduced hint-based learning by matching compressed teacher activations; subsequent work diversified the transferred signal to attention maps [11], flow-of-solution matrices [12], activation boundaries [13], pairwise relational structures [17, 18, 16], neuron selectivity patterns [15], contrastive embeddings [19], variational mutual information [20], and transformer-aware feature transfer [21]. OFD [14] provides a principled analysis of feature distillation objectives, while SimKD [22] bypasses alignment by reusing the teacher's classifier directly. Applying these methods across heterogeneous architectures, however, demands careful design: differing feature dimensionalities, spatial structures, and inductive biases make direct intermediate feature matching ill-posed, and naive adaptation risks introducing misalignment rather than resolving it.

Cross-architecture distillation. Early works explored cross-architecture transfer in limited settings, such as CNN-to-ViT distillation [31, 43] and data-efficient locality guidance [44], but lacked a unified formulation. More recently, OFA [25] established a general paradigm by introducing learned projectors that map heterogeneous features into a shared logit space with stage-wise supervision, enabling systematic evaluation across CNNs, ViTs, and MLPs. Subsequent work further improves this direction by simplifying alignment through redundancy suppression [35] or explicitly fusing heterogeneous representations before transfer [36]. These methods are architecture-agnostic in the sense that they apply to heterogeneous pairs, but in all of them the shared space is *learned* and anchored to one of the two networks. RSD [35], for instance, maps the student into the teacher's embedding space and suppresses redundancy within a $D_t \times D_t$ correlation matrix, so both the adapter output and the comparison space change whenever the teacher does ($D_t = 192$ for DeiT-T versus 768 for Swin-T); OFA instead anchors to the class logit space, whose width is set by the task. RPKD differs in where the shared space comes from rather than in what it spans. Both networks are projected into a prototype space that is random, frozen at initialization, and sized independently of the task label space, so the alignment target is fully determined before training begins and is reused unchanged across arbitrary model pairs, with only the student's input projection learned.

3 Method

We propose **RPKD**, a knowledge distillation framework that transfers information through a shared *random prototype space* consisting of randomly initialized, class-agnostic reference directions fixed in a common embedding space, eliminating the need for *teacher-space-defined* feature alignment between heterogeneous architectures: neither network's native coordinate system serves as the target that the other must be mapped into. Both teacher and student are projected into this space independently, and distillation is performed by matching their similarity structures within it.



(a) OFA (b) RPKD (ours)

Figure 1: **Method comparison: OFA vs. RPKD. (a) OFA** places four learned Proj modules at intermediate student stages, each mapping student features into the shared *class logit space* (C-dimensional) for stage-wise alignment with the teacher's output distribution; projectors are trained jointly and are architecture-specific. **(b) RPKD** transfers knowledge through a *frozen random* prototype matrix $\mathbf{P} \in \mathbb{R}^{M \times R}$, fixed at initialization and shared by both networks. The teacher projects into $\mathbf{P}$ via frozen random linear maps $\phi_t^{\text{g}}, \phi_t^{\text{f}}$ (❄ frozen), requiring no teacher-side updates. The student adapts via learnable MLP projectors $\phi_s^{\text{g}}, \phi_s^{\text{f}}$ (🔥 learnable) to match the teacher's prototype similarity patterns. Two complementary branches are optimized alongside the task cross-entropy: $\mathcal{L}_{\text{logit}}$ (global prototype similarity) and $\mathcal{L}_{\text{feat}}$ (spatial prototype activation maps). Unlike OFA, whose projectors target a space whose width is set by the task, RPKD keeps the prototype bank and both losses unchanged across pairs; only the two input projections are sized to each network's feature width.

Random prototype space. We define a prototype matrix $\mathbf{P} \in \mathbb{R}^{M \times R}$, where M is the number of prototypes and R is the shared embedding dimension. Rows of $\mathbf{P}$ are sampled independently from a standard Gaussian and L2-normalized; $\mathbf{P}$ is fixed after initialization and contains no learnable parameters, removing any dependency on class semantics or architecture-specific inductive biases.

To project representations into this space, we use separate mappings for teacher and student. The teacher uses frozen random linear projections $\phi_t^{\text{g}}, \phi_t^{\text{f}} : \mathbb{R}^{D_t} \to \mathbb{R}^R$ (where D_t is the teacher feature dimension), while the student uses learnable two-layer MLP projections $\phi_s^{\text{g}}, \phi_s^{\text{f}} : \mathbb{R}^{D_s} \to \mathbb{R}^R$ with GELU activations. This asymmetry gives the student a fixed, stable target: the teacher's prototype similarity distribution does not shift during training.

Architecture-agnostic feature routing. We use each network's native classification feature as the global representation: global average pooling for CNNs, the CLS token for ViT-style models [29, 31], and mean pooling over output tokens for models without a dedicated CLS token (e.g., Swin Transformer [30], MLP-Mixer [32], ResMLP [33]). Spatial representations are the corresponding spatial feature maps or token sequences from the final network stage, reshaped to a common $B \times N \times D$ format (B: batch size, N: number of spatial positions or tokens). Both branches operate on *final-stage* features by default; Section 4 ablates this choice and confirms that including earlier stages degrades performance, since shallow features carry low-level spatial statistics that are difficult to reconcile across heterogeneous architectures through prototype similarity alone.

Distillation in prototype space. RPKD transfers knowledge through two complementary objectives.

Logit branch. Let $\mathbf{f}_g^{(x)} \in \mathbb{R}^{D_x}$ be the global feature of model $x \in \{t, s\}$. We project it to $\mathbb{R}^R$, L2-normalize, and compute its cosine similarity to each prototype:

$$\mathbf{z}^{(x)} = \mathrm{norm}\Big(\phi_x^{\mathrm{g}}(\mathbf{f}_g^{(x)})\Big) \in \mathbb{R}^R, \qquad \mathbf{l}^{(x)} = \mathbf{z}^{(x)}\mathbf{P}^\top \in \mathbb{R}^M. \tag{1}$$

We refer to $\mathbf{l}^{(x)}$ as *prototype logits*: despite the name, its M entries are similarities to the prototypes, not scores over the C task classes, and this branch never enters the class space. That is precisely what keeps the width of the transferred signal fixed at M regardless of C. We match the teacher and student prototype-logit distributions with a temperature-scaled KL divergence, applying the per-sample logit standardization of LSKD [9] to each side beforehand:

$$\mathcal{L}_{\mathrm{logit}} = T^2 \cdot \mathrm{KL}\Big(\sigma\Big(\tfrac{\tilde{\mathbf{l}}^{(s)}}{T}\Big) \;\Big\|\; \sigma\Big(\tfrac{\tilde{\mathbf{l}}^{(t)}}{T}\Big)\Big), \tag{2}$$

where $\tilde{\mathbf{l}}^{(x)} = (\mathbf{l}^{(x)} - \mu_x)/s_x$, with μ_x and s_x the mean and standard deviation of the M entries of $\mathbf{l}^{(x)}$ for that sample, $T > 0$ the distillation temperature, and $\sigma(\cdot)$ the softmax. Standardizing each side separately removes the scale and spread mismatch between the teacher's and the student's similarity vectors, so the KL term compares the shape of the two prototype-logit distributions rather than their magnitude.

Feature branch. Let $\mathbf{F}^{(x)} \in \mathbb{R}^{B\times N\times D_x}$ be the spatial token matrix (N spatial positions or patch tokens, each of dimension D_x). We project each token to $\mathbb{R}^R$, L2-normalize, and compute per-token prototype similarities:

$$\mathbf{G}^{(x)} = \mathrm{norm}\Big(\phi_x^{\mathrm{f}}(\mathbf{F}^{(x)})\Big) \in \mathbb{R}^{B\times N\times R}, \qquad \mathbf{A}^{(x)} = \mathbf{G}^{(x)}\mathbf{P}^\top \in \mathbb{R}^{B\times N\times M}. \tag{3}$$

Each of the M columns of $\mathbf{A}^{(x)}$ is reshaped into a spatial map and resized to 7×7, yielding prototype activation maps $\mathbf{a}_{b,m}^{(x)} \in \mathbb{R}^{49}$ for sample b and prototype m. Every map is then standardized over its own spatial extent, $\hat{\mathbf{a}}_{b,m}^{(x)} = (\mathbf{a}_{b,m}^{(x)} - \mu_{b,m}^{(x)})/(s_{b,m}^{(x)} + \epsilon)$ with μ and s the mean and standard deviation over the 49 positions, and teacher and student maps are aligned via a cosine similarity loss:

$$\mathcal{L}_{\mathrm{feat}} = \frac{1}{BM}\sum_{b=1}^{B}\sum_{m=1}^{M}\Big(1 - \cos\Big(\hat{\mathbf{a}}_{b,m}^{(t)},\, \hat{\mathbf{a}}_{b,m}^{(s)}\Big)\Big). \tag{4}$$

Because the cosine is scale invariant, the division only conditions the computation numerically; it is the centring that changes the objective, making each per-map term a Pearson correlation between the teacher's and the student's activation pattern rather than a raw cosine.[1]

Training objective. The overall loss combines cross-entropy with both distillation branches:

$$\mathcal{L} = \lambda_{\mathrm{CE}}\,\mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{logit}}\,\mathcal{L}_{\mathrm{logit}} + \lambda_{\mathrm{feat}}\,\mathcal{L}_{\mathrm{feat}}. \tag{5}$$

Gradients are applied only to the student encoder and its projection heads. The teacher network, prototype matrix, and teacher projectors remain fixed throughout training.

Discussion. RPKD replaces teacher-space-defined feature alignment with similarity matching in a shared random space that belongs to neither network. Original feature spaces cannot serve as the shared medium: they differ in dimensionality ($D_t \neq D_s$), inductive bias (local vs. global vs. token-mixing), and distributional properties, making direct similarity comparison ill-posed. A random projection space resolves this by providing an external reference frame independent of both architectures: by the Johnson–Lindenstrauss lemma [45, 46], a random linear map approximately preserves pairwise inner products between unit-norm vectors with high probability, so the similarity structure meaningful to the teacher survives the projection. Concretely, for any two L2-normalized projected features $\mathbf{z}_1, \mathbf{z}_2 \in \mathbb{R}^R$ and prototype rows $\mathbf{p}_m$ drawn uniformly from the unit sphere,

$$\mathbb{E}[(\mathbf{z}_1 \cdot \mathbf{p}_m)(\mathbf{z}_2 \cdot \mathbf{p}_m)] \;\propto\; \mathbf{z}_1 \cdot \mathbf{z}_2; \tag{6}$$

by concentration, $\frac{1}{M}\,\mathbf{l}^{(t)}(x_1) \cdot \mathbf{l}^{(t)}(x_2) \approx \mathbf{z}^{(t)}(x_1) \cdot \mathbf{z}^{(t)}(x_2)$ holds simultaneously for all input pairs with high probability as M grows. As a consequence, the inter-sample similarity structure of the teacher's representations is faithfully reflected in the random prototype space: two inputs that are semantically close in the teacher's feature space produce similar prototype activation vectors. The student, by aligning its activation distribution to the teacher's, implicitly learns to reproduce this structural geometry without direct access to the teacher's feature space, which is what decouples distillation from architectural constraints.

[1] For Swin students we compare the maps without this standardization, which otherwise destabilizes their training.

4 Experiments

We evaluate RPKD on CIFAR-100 cross-architecture benchmarks spanning CNN, ViT, and MLP architectural families, on full-scale ImageNet-1K, and on ImageNet subsets designed specifically to stress the low-category regime discussed in Section 3.

Setup. CIFAR-100 [47] experiments cover 10 teacher–student pairs across two architectural families. *CNN-based students* (ResNet18 [26], MobileNetV2 [27]) are paired with transformer and MLP teachers (Swin-T [30], ViT-S [29], Mixer-B/16 [32]). *ViT-based students* (DeiT-T [31], Swin-P[2]) are paired with CNN and MLP teachers (ConvNeXt-T [28], Mixer-B/16). All teacher models follow the pretrained checkpoints provided by OFA [25] for fair comparison and are frozen during distillation. All images are resized to 224×224. For CIFAR-100, students are trained for 300 epochs using SGD with a batch size of 128, an initial learning rate of 0.05, and weight decay of 2×10^{-3}. For ImageNet-100 and ImageNet-1K, CNN students are trained for 100 epochs using AdamW with a batch size of 64, an initial learning rate of 3×10^{-3}, weight decay of 3×10^{-2}, and 3 warmup epochs. ViT-based students on ImageNet-100 use AdamW with a batch size of 128, a learning rate of 5×10^{-4}, weight decay of 0.05, and 20 warmup epochs. All settings use a cosine learning rate schedule. RPKD uses $M = 500$ randomly initialized frozen prototypes of dimension $R = 256$ (ablated in Table 7), distillation temperature $T = 4.0$, and loss coefficients selected following common practice in prior KD works and validated via a small grid search. Teacher projectors are initialized with weight standard deviation $\sigma = 0.02$. All experiments are implemented in PyTorch [48] and run on four NVIDIA H100 GPUs.

Baselines. We compare against two groups of methods. *Hint-based methods* (FitNet [10], CC [16], RKD [17], and CRD [19]) transfer intermediate features or relational structures; for cross-architecture pairs they are equipped with lightweight linear adapters following the protocol of OFA [25]. *Logit-based methods* (KD [1], DKD [8], and DIST [7]) operate on output distributions and require no feature alignment. *Cross-architecture methods* are the closest competitors: OFA [25] uses a jointly optimized universal feature bridge, FBT [36] fuses heterogeneous representations before transfer, and RSD [35] simplifies alignment through redundancy suppression. For FBT and RSD we report the numbers published by their authors; benchmarks not covered by the original paper are omitted from the corresponding table. Neither OFA nor RSD reports ImageNet-100 results, so for that benchmark we ran both ourselves from the official implementations over three seeds (RSD with a single fixed configuration, $\lambda = 0.1$, $\kappa = 0.01$, $\gamma = 2$), and FBT is omitted because its implementation is not public. All other baselines and training hyperparameters follow those of OFA [25] for a fair comparison. In Tables 1, 2, and 3, these methods and RPKD are grouped under **CAKD** (Cross-Architecture Knowledge Distillation), denoting methods designed to operate across heterogeneous architectural families. RPKD results are reported as mean $\pm$ std over three independent runs on all three benchmarks; the ImageNet-100 OFA and RSD entries are likewise three-seed means, and all remaining baseline numbers are the single-run figures published by their authors.

CIFAR-100 cross-architecture results. Table 1 reports top-1 accuracy across all 10 pairs. RPKD achieves a maximum gain of $+\mathbf{7.98}\%$ over OFA.

Among CNN-based students, RPKD achieves 85.15% (Swin-T $\rightarrow$ ResNet18) and 84.90% (Swin-T $\rightarrow$ MobileNetV2), corresponding to gains of $+4.61\%$ and $+3.92\%$ over OFA. The fragility of direct feature matching under heterogeneous architectures is illustrated by FitNet, which collapses to 24.06% on ConvNeXt-T $\rightarrow$ Swin-P when feature spaces are incompatible.

Among ViT-based students, RPKD leads on ConvNeXt-T $\rightarrow$ DeiT-T (83.74%, $+7.98$ pp over OFA and $+1.28$ pp over RSD [35]), where the CNN-to-ViT structural gap is greatest. On the remaining three ViT-student pairs, RSD leads; all three involve an MLP-Mixer teacher or the Swin-P student, where token-mixing weakens the spatial structure that the RPKD feature branch relies on.

ImageNet-100 cross-architecture results. Table 2 extends the evaluation to 12 teacher–student pairs on ImageNet-100. RPKD ranks first on 8 of 12 pairs and exceeds OFA on 11 of 12, improving over the student baseline on every pair; the four pairs it does not lead are all won by RSD [35]. The

[2]Swin-P (Swin-Pico) and Swin-N (Swin-Nano) are compact modified variants of Swin-T [30] introduced by OFA [25]; we follow their architecture definitions for fair comparison.

Table 1: Cross-architecture KD results on CIFAR-100. Best results in **bold**, second best underlined.

Teacher	Student	From Scratch		Hint-based				Logits-based			CAKD			
		T.	S.	FitNet	CC	RKD	CRD	KD	DKD	DIST	OFA	FBT	RSD	RPKD
CNN-based students														
Swin-T	ResNet18	89.26	74.01	78.87	74.19	74.11	77.63	78.74	80.26	77.75	80.54	81.61	83.92	**85.15**$_{\pm 0.18}$
ViT-S	ResNet18	92.04	74.01	77.71	74.26	73.72	76.60	77.26	78.10	76.49	80.15	81.93	81.50	**84.78**$_{\pm 0.20}$
Mixer-B/16	ResNet18	87.29	74.01	77.15	74.26	73.75	76.42	77.79	78.67	76.36	79.39	**81.90**	81.85	81.82$_{\pm 0.21}$
Swin-T	MobileNetV2	89.26	73.68	74.28	71.19	69.00	79.80	74.68	71.07	72.89	80.98	81.28	83.68	**84.90**$_{\pm 0.04}$
ViT-S	MobileNetV2	92.04	73.68	73.54	70.67	68.46	78.14	72.77	69.80	72.54	78.45	82.10	81.68	**84.17**$_{\pm 0.11}$
Mixer-B/16	MobileNetV2	87.29	73.68	73.78	70.73	68.95	78.15	73.33	70.20	73.26	78.78	80.83	81.74	**82.43**$_{\pm 0.11}$
ViT-based students														
ConvNeXt-T	DeiT-T	88.41	68.00	60.78	68.01	69.79	65.94	72.99	74.60	73.55	75.76	79.57	82.46	**83.74**$_{\pm 0.25}$
Mixer-B/16	DeiT-T	87.29	68.00	71.05	68.13	69.89	65.35	71.36	73.44	71.67	73.90	74.40	**78.50**	76.66$_{\pm 0.05}$
ConvNeXt-T	Swin-P	88.41	72.63	24.06	72.63	71.73	67.09	76.44	76.80	76.41	78.32	80.73	**82.21**	79.98$_{\pm 0.26}$
Mixer-B/16	Swin-P	87.29	72.63	75.20	73.32	70.82	67.03	75.93	76.39	75.85	78.93	78.44	**81.28**	75.50$_{\pm 0.07}$

Table 2: Cross-architecture KD results on ImageNet-100. OFA, RSD and RPKD are our own runs, each averaged over three seeds; standard deviations are shown for RPKD. The remaining baselines are single-run. Best results in **bold**, second best underlined.

Teacher	Student	From Scratch		Hint-based				Logits-based			CAKD		
		T.	S.	FitNet	CC	RKD	CRD	KD	DKD	DIST	OFA	RSD	RPKD
CNN-based students													
DeiT-T	ResNet18	89.72	81.58	85.84	81.70	83.14	84.54	84.32	85.50	84.52	85.46	87.14	**87.87**$_{\pm 0.24}$
Swin-T	ResNet18	94.48	81.58	81.94	81.68	82.78	83.44	82.76	84.70	83.42	84.67	87.25	**87.86**$_{\pm 0.33}$
Mixer-B/16	ResNet18	90.42	81.58	82.58	81.98	81.90	83.88	82.84	84.88	83.38	84.19	**85.85**	85.37$_{\pm 0.20}$
DeiT-T	MobileNetV2	89.72	83.51	86.94	84.24	83.94	85.06	86.14	86.44	85.38	85.99	87.01	**87.35**$_{\pm 0.08}$
Swin-T	MobileNetV2	94.48	83.51	85.20	83.84	82.94	84.92	84.96	86.18	84.80	85.24	**87.17**	86.95$_{\pm 0.18}$
Mixer-B/16	MobileNetV2	90.42	83.51	84.78	84.06	83.28	84.12	83.90	85.62	84.62	85.06	**86.06**	85.07$_{\pm 0.39}$
ViT-based students													
ResNet50	DeiT-T	93.52	63.88	64.62	63.68	64.62	56.94	65.76	66.24	65.34	68.56	72.18	**81.00**$_{\pm 0.12}$
ConvNeXt-T	DeiT-T	94.76	63.88	63.38	64.24	63.64	58.78	66.08	63.42	64.82	67.38	67.93	**79.14**$_{\pm 0.26}$
Mixer-B/16	DeiT-T	90.42	63.88	65.96	64.66	65.22	55.30	68.12	66.88	67.52	69.75	69.21	**70.20**$_{\pm 0.04}$
ResNet50	Swin-N	93.52	78.78	79.16	78.90	75.30	75.24	80.94	80.62	80.50	80.38	84.31	**84.57**$_{\pm 0.16}$
ConvNeXt-T	Swin-N	94.76	78.78	78.90	78.82	73.66	75.44	79.90	79.86	80.60	79.93	83.09	**83.15**$_{\pm 0.21}$
Mixer-B/16	Swin-N	90.42	78.78	79.52	78.76	74.02	74.62	81.28	81.40	80.90	81.19	**82.57**	79.92$_{\pm 0.13}$

gains are most pronounced for ViT-based students paired with CNN teachers: RPKD reaches 81.00% on ResNet50 $\rightarrow$ DeiT-T ($+12.44\%$ over OFA) and 79.14% on ConvNeXt-T $\rightarrow$ DeiT-T ($+11.76\%$ over OFA), suggesting that prototype-based transfer succeeds precisely by avoiding model-specific structural assumptions, relying instead on class-similarity geometry that is approximately preserved under random projection regardless of architectural boundaries. CNN-based students show consistent but more moderate improvements, with RPKD outperforming OFA on all six CNN-student pairs. The single exception is Mixer-B/16 $\rightarrow$ Swin-N, where RSD (82.57%) outperforms RPKD (79.92%), consistent with the analogous Mixer-B/16 $\rightarrow$ Swin-P result on CIFAR-100.

MLP-Mixer teacher behavior. When MLP-Mixer acts as teacher, RPKD underperforms relative to CNN and ViT teacher settings and in several pairs fails to outperform OFA — a trend shared by all feature-based methods across both benchmarks. We attribute this to the token-mixing nature of MLP-Mixers: repeated spatial mixing weakens localized positional correspondence in the representations, undermining the spatial alignment that the feature branch relies on.

ImageNet-1K cross-architecture results. Table 3 reports full-scale ImageNet-1K results for four cross-architecture pairs. RPKD exceeds OFA on all four pairs and leads overall on DeiT-T $\rightarrow$ MobileNetV2 (72.67%, $+1.28\%$ over OFA and $+0.49\%$ over RSD [35]). On the remaining pairs RPKD stays within 0.32 pp of the better of FBT [36] and RSD, indicating that at full ImageNet-1K scale the three CAKD methods are broadly comparable. The larger gain on MobileNetV2 is consistent with

Table 3: Cross-architecture KD results on ImageNet-1K. Best results in **bold**, second best underlined.

Teacher	Student	From Scratch		Hint-based				Logits-based			CAKD			
		T.	S.	FitNet	CC	RKD	CRD	KD	DKD	DIST	OFA	FBT	RSD	RPKD
DeiT-T	ResNet18	72.17	69.75	70.44	69.77	69.47	69.25	70.22	69.39	70.64	71.34	71.22	**71.70**	$\underline{71.38}_{\pm 0.14}$
Swin-T	ResNet18	81.38	69.75	71.18	70.07	68.89	69.09	71.14	71.10	70.91	71.85	**72.21**	$\underline{72.13}$	$72.01_{\pm 0.12}$
DeiT-T	MobileNetV2	72.17	68.87	70.95	70.69	69.72	69.60	70.87	70.14	71.08	71.39	71.78	$\underline{72.18}$	$\mathbf{72.67}_{\pm 0.11}$
Swin-T	MobileNetV2	81.38	68.87	71.75	70.69	67.52	69.58	72.05	71.71	71.76	72.32	**72.54**	72.36	$\underline{72.40}_{\pm 0.21}$

Table 4: Standard CNN-to-CNN KD on ImageNet-1K. Best in **bold**, second best underlined.

Teacher	Student	T.	S.	KD	OFD	Review	CRD	DKD	DIST	OFA	FBT	RSD	RPKD
ResNet34	ResNet18	73.31	69.75	70.66	70.81	71.61	71.17	71.70	72.07	72.10	**72.29**	72.18	$\underline{72.23}_{\pm 0.13}$
ResNet50	MobileNetV1	76.16	68.58	68.58	71.25	72.56	71.37	72.05	73.24	73.24	$\underline{73.45}$	73.08	$\mathbf{74.20}_{\pm 0.09}$

the CIFAR-100 trend: the narrow channel bottlenecks of inverted-residual architectures [27] produce low-rank features that prototype-space supervision handles better than direct learned alignment.

Homogeneous architecture results. Table 4 evaluates RPKD in the standard same-family setting on ImageNet-1K, where learned alignment methods operate at their best. On ResNet34 $\rightarrow$ ResNet18, RPKD reaches 72.23%, ahead of OFA (72.10%) and RSD [35] (72.18%) and behind FBT [36] (72.29%); the three leading methods span only 0.11 pp, less than RPKD's own run-to-run variation (± 0.13), so this pair is better read as a tie among recent methods than as a ranking. On ResNet50 $\rightarrow$ MobileNetV1 the margin is unambiguous: RPKD achieves 74.20%, surpassing every prior method including FBT (73.45%) by $+0.75$ pp. Prototype-space distillation therefore generalizes to standard CNN-to-CNN settings, matching learned-alignment methods where they are strongest and exceeding them when the student is narrow.

Low-category advantage. A distinctive property of RPKD is that the prototype vocabulary is decoupled from the number of task classes: the student always receives $M = 500$-dimensional supervisory signal regardless of dataset size, whereas logit-based soft targets collapse toward near-uniform distributions as the class count decreases. Table 5 tests this on ImageNet subsets of 10, 20, 50 and 100 classes (DeiT-T $\rightarrow$ ResNet18), formed by taking the first 10/20/50/100 classes of ImageNet-100 [49] ordered alphabetically by synset name. At IN-10, both hint-based CRD (88.60) and logit-based DIST (89.80) fall *below* the student trained from scratch (90.80), while RPKD reaches 95.40, a $+4.60\%$ gain over the student and $+2.80\%$ over OFA; the advantage holds at every subset size, with $+3.00\%$ over OFA on IN-20 and $+2.41\%$ on IN-100. The scaling is what the design predicts: as the class count drops from 100 to 10, KD's gain over the student shrinks from $+2.74$ to $+0.60$ pp and OFA's from $+3.88$ to $+1.80$ pp, while RPKD's stays between $+6.29$ and $+4.60$ pp, so its advantage over OFA widens from roughly 1.5–$1.6\times$ at 50–100 classes to 2.3–$2.6\times$ at 10–20. Each branch is individually competitive — $\text{RPKD}_{\text{feat}}$ ranks second at IN-10 and IN-20, $\text{RPKD}_{\text{logit}}$ at IN-50 and IN-100 — yet the full method outperforms both at every size, confirming that the two objectives are complementary.

Table 5: Top-1 accuracy on ImageNet subsets of varying class count (T: DeiT-T, S: ResNet18). IN-10/20/50 use the first 10/20/50 ImageNet-100 [49] classes and are single run; in the IN-100 row the OFA and RPKD entries repeat the three-seed means of Table 2. Best in **bold**, second best underlined.

Subset	From Scratch		Hint-based				Logits-based				Ours		
	T.	S.	FitNet	CC	RKD	CRD	KD	DKD	DIST	OFA	$\text{RPKD}_{\text{logit}}$	$\text{RPKD}_{\text{feat}}$	RPKD
IN-10	97.80	90.80	93.20	91.80	92.80	88.60	91.40	91.80	89.80	92.60	93.80	$\underline{95.20}$	**95.40**
IN-20	97.40	89.80	92.80	90.00	90.20	88.10	90.80	92.20	90.60	92.20	92.30	$\underline{94.80}$	**95.20**
IN-50	93.28	84.76	88.80	83.60	85.84	86.12	87.20	88.32	86.68	88.48	$\underline{88.88}$	88.80	**90.44**
IN-100	89.72	81.58	85.84	81.70	83.14	84.54	84.32	85.50	84.52	85.46	$\underline{86.22}$	86.16	**87.87**

Table 6: Ablation of feature extraction stage on IN-100 (DeiT-T $\rightarrow$ ResNet18). Stages 1–4 denote depth from shallow to deep; {4} is the default. Single run, except {4} (three-seed mean, cf. Table 2).

Stages used	{1, 2, 3, 4}	{2, 3, 4}	{3, 4}	{4} (default)
Top-1 Acc.	86.30	86.80	87.65	**87.87**

Table 7: Ablation of prototype count M (left, $R=256$) and embedding dimension R (right, $M=500$) on IN-100. Mean $\pm$ std over 3 seeds, using an independent seed set from Table 2. Defaults in **bold**.

Prototype count M ($R=256$)			**Embedding dimension** R ($M=500$)		
M	DeiT-T $\rightarrow$ RN18	RN50 $\rightarrow$ DeiT-T	R	DeiT-T $\rightarrow$ RN18	RN50 $\rightarrow$ DeiT-T
125	$87.65_{\pm 0.13}$	$80.36_{\pm 0.10}$	64	$87.60_{\pm 0.09}$	$80.63_{\pm 0.45}$
250	$87.74_{\pm 0.16}$	$81.07_{\pm 0.15}$	128	$87.72_{\pm 0.14}$	$80.93_{\pm 0.17}$
500	$\mathbf{87.87_{\pm 0.24}}$	$\mathbf{81.20_{\pm 0.29}}$	**256**	$\mathbf{87.87_{\pm 0.24}}$	$81.20_{\pm 0.29}$
1000	$87.67_{\pm 0.32}$	$81.01_{\pm 0.46}$	512	$87.71_{\pm 0.19}$	$\mathbf{81.80_{\pm 0.27}}$
2000	$87.79_{\pm 0.28}$	$81.13_{\pm 0.06}$	1024	$87.63_{\pm 0.22}$	$80.53_{\pm 0.19}$
Range	0.22	0.84	*Range*	0.27	1.27

Ablation: feature extraction stage. Table 6 ablates the network stages used for feature extraction, evaluated on IN-100 (DeiT-T $\rightarrow$ ResNet18). Each network is divided into four stages of increasing depth, and distillation is progressively restricted to later stages only. Performance rises monotonically as earlier stages are removed: using all four stages yields 86.30%, while the final stage alone achieves 87.87%. Early-stage features encode low-level spatial statistics that are hard to reconcile across heterogeneous architectures through prototype matching; final-stage features (as in all other experiments) carry high-level semantics that align more readily in the shared prototype space.

Ablation: prototype count and dimension. Table 7 ablates the prototype count M and embedding dimension R on IN-100 over three independent seeds, for a CNN student (DeiT-T $\rightarrow$ ResNet18) and a ViT student (ResNet50 $\rightarrow$ DeiT-T). Across a $16\times$ sweep of either hyperparameter the total range is 0.22 and 0.27 pp for the CNN student—within its training noise (± 0.24 pp)—and 0.84 and 1.27 pp for the ViT student, whose only notable preference is $R=512$. RPKD retains its margin over OFA at every setting ($\geq +2.14$ pp and $\geq +11.80$ pp respectively), so the defaults were fixed once and reused across all 28 pairs without per-pair tuning. The plateau around $M=500$ is consistent with the Johnson–Lindenstrauss argument: for batch size $n=128$ and distortion $\varepsilon=0.1$, a sufficient threshold is $M \gtrsim \ln(128)/0.1^2 \approx 485$, placing our default near this bound; beyond it, additional prototypes yield diminishing returns. Appendix C further shows that varying only the random draw of $\mathbf{P}$ moves accuracy no more than varying the training seed.

Ablation: frozen vs. learned prototypes. Table 8 tests the design choice at the core of RPKD: whether the prototype bank $\mathbf{P}$ should be frozen at initialization or learned jointly with the student projection heads. We run four architecturally diverse IN-100 pairs at $M=R=500$ (so that exact orthogonality is available for the Frozen Orthogonal variant) over three independent seeds, holding all other components and hyperparameters fixed. Every frozen variant outperforms the learned prototype on every pair, by $+0.57$ pp (DeiT-T $\rightarrow$ RN18), $+2.86$ pp (ConvNeXt-T $\rightarrow$ Swin-N), $+3.84$ pp (RN50 $\rightarrow$ DeiT-T), and $+0.89$ pp (Swin-T $\rightarrow$ MBv2): a fixed random geometry consistently beats an optimized, data-adapted one, and the margin grows for heterogeneous pairs. Among frozen variants no single structure dominates (maximum pairwise difference 0.54 pp), which indicates that what matters is a *fixed* reference geometry rather than the specific basis. A learned $\mathbf{P}$ co-evolves with the student projection heads and therefore presents a moving target; because both $\mathbf{P}$ and the teacher projections are frozen, the student instead aligns against a completely static basis throughout training.

Ablation: branch contributions. Appendix A isolates the two branches on six ImageNet-100 and five ImageNet-1K pairs. Each branch independently yields substantial gains over the CE-only baseline, and their relative importance is architecture-dependent: for CNN $\rightarrow$ ViT distillation the spatial branch $\mathcal{L}_{\text{feat}}$ does most of the work in bridging the representation gap, whereas for MLP-Mixer teachers, where token-mixing weakens spatial structure, the global branch drives performance.

Table 8: Frozen vs. learned prototype ablation on ImageNet-100 (mean ± std, 3 seeds; $M = R = 500$). Frozen Gaussian uses ℓ_2-normalized random rows, Frozen Orthogonal a QR-decomposed square matrix, and Frozen DCT the cosine frequency basis of $\mathbb{R}^{500}$. Bold = best per column.

Prototype	DeiT-T → RN18	ConvNeXt-T → Swin-N	RN50 → DeiT-T	Swin-T → MBv2
Frozen Gaussian	$87.78_{\pm 0.29}$	$\mathbf{83.90_{\pm 0.24}}$	$80.93_{\pm 0.38}$	$\mathbf{87.10_{\pm 0.16}}$
Frozen Orthogonal	$\mathbf{87.86_{\pm 0.27}}$	$83.47_{\pm 0.12}$	$80.67_{\pm 0.36}$	$87.09_{\pm 0.01}$
Frozen DCT	$87.71_{\pm 0.06}$	$83.73_{\pm 0.18}$	$\mathbf{81.21_{\pm 0.29}}$	$87.01_{\pm 0.30}$
Learned (Gaussian-init)	$87.29_{\pm 0.10}$	$81.04_{\pm 0.03}$	$77.37_{\pm 0.38}$	$86.21_{\pm 0.15}$

Combined, the two are complementary: on ImageNet-1K full RPKD strictly outperforms both individual branches on all five pairs.

Limitations. RPKD's performance is robust to the prototype count M and embedding dimension R (total range ≤ 0.27 pp across $16\times$ sweeps), and the distillation temperature is fixed at $T = 4.0$ throughout, but the loss coefficients require a small grid search; adaptive weighting schemes could reduce this tuning burden. Its effectiveness is also bounded by teacher quality: a poorly calibrated or biased teacher transfers suboptimal representations. While ImageNet-100 provides 12-pair coverage across CNN and ViT families, full-scale ImageNet-1K evaluation is restricted to four CNN-based student pairs owing to computational constraints. RPKD has been evaluated exclusively on image classification; its applicability to other recognition problems such as object detection and semantic segmentation requires further study. Finally, RPKD's effectiveness degrades when the teacher is an MLP-Mixer, as the token-mixing nature of such architectures weakens the spatial correspondence that the feature branch relies on.

5 Conclusion

We presented **RPKD**, a knowledge distillation framework built on the observation that effective cross-architecture knowledge transfer does not require learned feature alignment. By projecting both teacher and student into a shared random prototype space that is fixed at initialization and reused without modification across all teacher–student pairs, RPKD removes the need for a learned alignment space, for joint teacher–student optimization, and for any pair-specific change to the objective. Knowledge is transferred through two complementary objectives: a logit branch that aligns global prototype similarity distributions and a feature branch that matches spatially-resolved activation maps, jointly encoding semantic and structural information within the same frozen space.

Empirical results. On CIFAR-100, across 10 cross-architecture pairs spanning convolutional, transformer, and MLP families, RPKD improves over training from scratch on every pair, exceeding OFA by up to $+\mathbf{7.98}\%$ and reaching 85.15% on Swin-T → ResNet18. On ImageNet-100 it ranks first on 8 of 12 pairs and exceeds OFA on 11 of 12, with a maximum gain of $+\mathbf{12.44}\%$ over OFA. On ImageNet-1K it exceeds OFA by up to $+1.28\%$ on cross-architecture pairs, ties the leading recent methods on ResNet34 → ResNet18 (72.23%, within run-to-run variation), and reaches 74.20% on ResNet50 → MobileNetV1, ahead of all prior methods by at least $+0.75$ pp. The low-category advantage is pronounced: on 10- and 20-class subsets RPKD surpasses OFA by $+2.80\%$ and $+3.00\%$, while hint- and logit-based methods degrade as the label space shrinks.

Implications. That a random, frozen prototype space suffices as a distillation medium challenges the prevailing assumption that cross-architecture alignment must be learned: the information bottleneck of a random projection is enough to align teacher and student similarity structures across large architectural gaps. The entire teacher side reduces to two frozen random matrices with no trainable parameters, yet matches or outperforms learned alignment while training faster and with no more peak memory (Appendix B). We hope RPKD encourages further study of structure-preserving random projections as knowledge transfer media.

Acknowledgments and Disclosure of Funding

We are grateful to Dr. Mert Bülent Sarıyıldız for his valuable guidance on the knowledge distillation literature during the early stages of this work. The numerical calculations reported in this paper were performed on MareNostrum 5, operated by the Barcelona Supercomputing Center (BSC-CNS), through the TURMETUGC project supported by TÜBİTAK. This study is supported in part by the ADEP project with grant no. ADEP-312-2024-11525. G. Cinbis is supported by the "Young Scientist Awards Program (BAGEP)" of Science Academy, Türkiye. The authors declare no competing interests.

References

[1] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[2] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

[3] Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International conference on machine learning*, pages 1607–1616. PMLR, 2018.

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

[5] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1345–1354, 2019.

[6] Wonchul Son, Jaemin Na, Junyong Choi, and Wonjun Hwang. Densely guided knowledge distillation using multiple teacher assistants. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9395–9404, 2021.

[7] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge distillation from a stronger teacher. *Advances in neural information processing systems*, 35:33716–33727, 2022.

[8] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022.

[9] Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15731–15740, 2024.

[10] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

[11] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[12] 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.

[13] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3779–3787, 2019.

[14] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1921–1930, 2019.

[15] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

[16] Baoyun Peng, Xiao Jin, Jiaheng Liu, Dongsheng Li, Yichao Wu, Yu Liu, Shunfeng Zhou, and Zhaoning Zhang. Correlation congruence for knowledge distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5007–5016, 2019.

[17] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3967–3976, 2019.

[18] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7096–7104, 2019.

[19] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[20] Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9163–9171, 2019.

[21] Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and Gang Wang. Knowledge distillation via the target-aware transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10915–10924, 2022.

[22] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11933–11942, 2022.

[23] Linfeng Zhang, Yukang Shi, Zuoqiang Shi, Kaisheng Ma, and Chenglong Bao. Task-oriented feature distillation. *Advances in Neural Information Processing Systems*, 33:14759–14771, 2020.

[24] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5008–5017, 2021.

[25] Zhiwei Hao, Jianyuan Guo, Kai Han, Yehui Tang, Han Hu, Yunhe Wang, and Chang Xu. One-for-all: Bridge the gap between heterogeneous architectures in knowledge distillation. *Advances in Neural Information Processing Systems*, 36:79570–79582, 2023.

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

[27] 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.

[28] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[29] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[30] 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. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[31] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[32] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34: 24261–24272, 2021.

[33] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, et al. Resmlp: Feedforward networks for image classification with data-efficient training. *IEEE transactions on pattern analysis and machine intelligence*, 45(4):5314–5321, 2022.

[34] Yufan Liu, Jiajiong Cao, Bing Li, Weiming Hu, Jingting Ding, and Liang Li. Cross-architecture knowledge distillation. In *Proceedings of the Asian conference on computer vision*, pages 3396–3411, 2022.

[35] Weijia Zhang, Yuehao Liu, Wu Ran, and Chao Ma. Cross-architecture distillation made simple with redundancy suppression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23256–23266, 2025.

[36] Guopeng Li, Qiang Wang, Ke Yan, Shouhong Ding, Yuan Gao, and Gui-Song Xia. Fuse before transfer: Knowledge fusion for heterogeneous distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3445–3454, 2025.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[38] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[39] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34:12077–12090, 2021.

[40] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023.

[41] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[42] 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.

[43] Xianing Chen, Qiong Cao, Yujie Zhong, Jing Zhang, Shenghua Gao, and Dacheng Tao. Dearkd: data-efficient early knowledge distillation for vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12052–12062, 2022.

[44] Kehan Li, Runyi Yu, Zhennan Wang, Li Yuan, Guoli Song, and Jie Chen. Locality guidance for improving vision transformers on tiny datasets. In *European Conference on Computer Vision*, pages 110–127. Springer, 2022.

[45] William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[46] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

[47] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[48] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

[49] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020.

[50] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction state that RPKD transfers knowledge via a shared random prototype space without teacher-space-defined feature alignment, and every quantitative claim ($+7.98\%$ over OFA on CIFAR-100, $+12.44\%$ on ImageNet-100, $+1.28\%$ on ImageNet-1K) is directly supported by the corresponding cells of Tables 1–4. Where RPKD does not lead, the text says so explicitly (the Mixer-B/16 pairs in Section 4, and the ResNet34 $\rightarrow$ ResNet18 tie in Table 4).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Limitations paragraph in Section 4 discusses five limitations: the loss coefficients still require a small grid search (while M and R are shown to be robust in Table 7 and the temperature is fixed throughout); performance is bounded by teacher quality; full-scale ImageNet-1K evaluation covers only four pairs owing to compute cost; evaluation is restricted to image classification, with dense prediction left as future work; and effectiveness degrades for MLP-Mixer teachers, which is also visible in Tables 1–2 and analysed in the MLP-Mixer teacher behavior paragraph.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper invokes the Johnson–Lindenstrauss lemma as informal theoretical motivation for the prototype space design, but does not present original theoretical results or formal proofs; all cited results are established in prior work [45, 46].

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The experimental setup paragraph in Section 4 reports all training details needed to reproduce our results, including optimizer, learning rate, batch size, weight decay, scheduling, and all RPKD-specific hyperparameters (M, R, T, σ). The method is fully described in Section 3.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.

(b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The full implementation, including the configuration files and training commands for every reported setting, is publicly available at `https://github.com/HamzaEtcibasi/RPKD`. Both datasets are standard public benchmarks; the repository documents the expected directory layout and the teacher checkpoints used.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: All training and evaluation details are reported in the Setup paragraph of Section 4, and the configuration files that fix the remaining details are in the released repository.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Every RPKD number in the four main result tables (Tables 1–4), in the prototype ablations (Tables 7–8) and in the ImageNet-100 block of Table 9 is reported as mean $\pm$ one standard deviation over three independent runs that differ only in the training seed; the captions state this. Table 11 reports the same statistic over three draws of the prototype bank with the training seed held fixed, which its caption makes explicit. On ImageNet-100 the OFA and RSD columns are likewise our own three-seed means, shown without their standard deviations for space; on CIFAR-100 and ImageNet-1K the baseline columns are the single-run figures published by their authors, so a comparison there is between a point estimate and a three-seed mean. The remaining cells are single-run and labelled as such: the ImageNet-1K block of Table 9, the subset results of Table 5, and the stage ablation of Table 6, all owing to the cost of full-scale training. Where a margin is smaller than our measured standard deviation we say so rather than claiming a win (Table 4 discussion).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The Setup paragraph of Section 4 states that all experiments are implemented in PyTorch and run on four NVIDIA H100 GPUs, and gives the epoch budget for each benchmark. Appendix B additionally reports per-iteration wall-clock step time and peak GPU memory for seven representative pairs, measured on a single H100 with AMP, alongside the corresponding figures for OFA. Preliminary and discarded runs, which are not reported in the paper, consumed additional compute on the same cluster.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.

- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work conducts image classification experiments on standard public benchmarks and conforms fully with the NeurIPS Code of Ethics; there are no privacy, fairness, or safety concerns.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: RPKD enables deployment of compact, efficient models by reducing the computational cost of cross-architecture knowledge distillation, with positive societal implications for resource-constrained environments. We do not foresee negative societal impacts as this is foundational model compression research.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This paper proposes a knowledge distillation algorithm with no risk of misuse; all experiments are conducted on open-source datasets and no safeguards are required.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: All datasets and pretrained checkpoints are properly cited. CIFAR-100 [47] is freely available for academic research. ImageNet [50] is used under its non-commercial research license; our use is strictly for academic, non-commercial purposes. Pretrained teacher checkpoints are sourced from the OFA repository [25] and used in accordance with their respective model licenses for non-commercial research.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: The one new asset is the RPKD implementation, released at `https://github.com/HamzaEtcibasi/RPKD` with documentation covering installation, the expected data layout, the teacher checkpoints, and the training command for each benchmark. No new datasets or model checkpoints are introduced.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.

- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were used only for writing assistance (grammar, spelling, and word choice) and played no role in the research methodology, experimental design, or results.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.

A Component Ablation

Table 9 reports a component-wise ablation isolating the contributions of the two distillation branches: the spatial prototype activation map loss $\mathcal{L}_{\text{feat}}$ and the global prototype similarity loss $\mathcal{L}_{\text{logit}}$. The cross-entropy task loss $\mathcal{L}_{\text{CE}}$ is active in all configurations.

Table 9: Component ablation (top-1 accuracy %). ImageNet-100 is mean $\pm$ std over 3 seeds; ImageNet-1K is single-run due to computational cost. Bold = best per row.

Teacher	Student	CE Only	$+\mathcal{L}_{\text{feat}}$ Only	$+\mathcal{L}_{\text{logit}}$ Only	RPKD (Full)
ImageNet-100					
DeiT-T	ResNet18	81.58	$86.46_{\pm 0.24}$	$86.89_{\pm 0.25}$	$\mathbf{87.87}_{\pm 0.24}$
Swin-T	ResNet18	81.58	$\mathbf{87.95}_{\pm 0.25}$	$82.93_{\pm 0.09}$	$87.86_{\pm 0.33}$
ConvNeXt-T	DeiT-T	63.88	$78.85_{\pm 0.17}$	$65.73_{\pm 0.67}$	$\mathbf{79.14}_{\pm 0.26}$
ResNet50	Swin-N	78.78	$84.19_{\pm 0.29}$	$79.61_{\pm 0.57}$	$\mathbf{84.57}_{\pm 0.16}$
Mixer-B/16	MobileNetV2	83.51	$81.23_{\pm 0.04}$	$\mathbf{85.44}_{\pm 0.33}$	$85.07_{\pm 0.39}$
Mixer-B/16	DeiT-T	63.88	$68.57_{\pm 0.19}$	$68.93_{\pm 0.13}$	$\mathbf{70.20}_{\pm 0.04}$
ImageNet-1K					
DeiT-T	ResNet18	69.75	70.90	70.87	**71.38**
Swin-T	ResNet18	69.75	71.85	70.82	**72.01**
DeiT-T	MobileNetV2	68.87	71.90	72.21	**72.67**
Swin-T	MobileNetV2	68.87	71.56	71.39	**72.40**
ResNet50	MobileNetV1	68.58	73.80	73.87	**74.20**

Both branches independently yield substantial gains over the CE-only baseline, confirming that mapping to a fixed random prototype space provides effective supervision at both the spatial and the global level. The relative importance of each branch is architecture-dependent. For CNN $\rightarrow$ ViT distillation (e.g., ConvNeXt-T $\rightarrow$ DeiT-T), the spatial feature branch $\mathcal{L}_{\text{feat}}$ plays the primary role in bridging the representation gap, while $\mathcal{L}_{\text{logit}}$ alone degrades toward the CE-only baseline. For MLP-Mixer teachers, where token-mixing weakens spatial structure, the global logit branch drives performance; Mixer-B/16 $\rightarrow$ MobileNetV2 is the only pair where $\mathcal{L}_{\text{logit}}$ alone outperforms full RPKD, consistent with the spatial branch limitation for such architectures. When combined, the two branches provide complementary supervision: on ImageNet-1K, full RPKD strictly outperforms both individual branches across all five pairs.

B Training Efficiency Comparison

Table 10 compares RPKD and OFA in terms of trainable parameters added beyond the student backbone, additional forward FLOPs, wall-clock training step time, and peak GPU memory across seven representative pairs. RPKD uses two trainable student MLPs, frozen teacher projections, and a non-trainable prototype buffer (500×256 values); its trainable parameter count is C-independent, whereas OFA's final heads scale with the number of classes C.

RPKD achieves lower step time and equal or lower peak GPU memory on all seven pairs ($1.02\times$–$1.23\times$ faster). The advantage is most pronounced for transformer students: on Mixer-B/16 $\rightarrow$ Swin-N, OFA adds 403 GFLOPs per batch—exceeding the 341 GFLOPs of the Swin-N backbone itself—while RPKD adds only 16 GFLOPs (a $25\times$ reduction), yielding a $1.23\times$ speed advantage and 1.83 GB less peak GPU memory. For CNN students with wide final-stage features (MobileNetV2, $D_s = 1280$), RPKD incurs noticeable FLOP overhead because its student MLPs scale with D_s; both methods reach comparable peak memory on those pairs. Notably, on Mixer-B/16 $\rightarrow$ Swin-N, RPKD has lower overhead but also lower accuracy: this pair reflects the spatial branch limitation discussed in Section 4.

C Prototype Initialization Seed Sensitivity

Because the prototype bank $\mathbf{P}$ is drawn once at random and then frozen, a natural question is whether performance depends on which draw is used. Table 11 isolates this: we fix the training seed and

Table 10: Training efficiency comparison on ImageNet-100 and ImageNet-1K. Extra Params: trainable parameters added beyond the student backbone. Addl. GFLOPs: method-specific forward operations per batch ($B = 128$, 224×224; values in parentheses relative to student backbone FLOPs). Step: forward + backward + optimizer step time on a single H100 with AMP, measured over 5×100 iters. Mem: peak GPU memory. Bold = row winner per metric.

		Extra Params		**Addl. GFLOPs**		**Step (ms)**		**Mem (GB)**		**Top-1**	
Teacher	**Student**	RPKD	OFA	RPKD	OFA	RPKD	OFA	RPKD	OFA	RPKD	OFA
ImageNet-100											
DeiT-T	ResNet18	1.58M	**1.19M**	14G (3%)	**8G (2%)**	**38.4**	41.4	**1.84**	2.07	**87.87**	85.46
Swin-T	MobileNetV2	7.87M	**2.53M**	56G (73%)	**23G (30%)**	**86.0**	87.6	5.21	5.21	**86.95**	85.24
ResNet50	DeiT-T	**0.35M**	3.78M	**12G (4%)**	47G (17%)	**59.0**	64.9	**3.50**	3.81	**81.00**	68.56
Mixer-B/16	Swin-N	**1.58M**	26.34M	**16G (5%)**	403G (118%)	**98.8**	121.2	**5.22**	7.05	79.92	**81.19**
ImageNet-1K											
Swin-T	ResNet18	**1.58M**	4.39M	16G (3%)	**11G (2%)**	**61.7**	65.1	**1.91**	2.20	**72.01**	71.85
DeiT-T	MobileNetV2	7.87M	**7.14M**	54G (70%)	**24G (31%)**	**62.3**	64.2	**5.16**	5.19	**72.67**	71.39
ResNet34	ResNet18	**1.58M**	3.04M	15G (3%)	**8G (2%)**	**36.8**	40.0	**1.88**	2.15	**72.23**	72.10

vary only the seed used to sample $\mathbf{P}$, over three independent draws on five ImageNet-100 pairs. For reference, the rightmost column repeats the standard deviation obtained the other way round, with $\mathbf{P}$ fixed and the training seed varied.

Table 11: Prototype initialization seed sensitivity on ImageNet-100. Prototype-seed column: mean $\pm$ std over three draws of $\mathbf{P}$ with the training seed fixed. Training-seed column: std over three training seeds with $\mathbf{P}$ fixed.

Teacher	**Student**	**Prototype-seed mean $\pm$ std**	**Training-seed std**
DeiT-T	ResNet18	$87.87_{\pm 0.24}$	± 0.24
Swin-T	MobileNetV2	$86.91_{\pm 0.28}$	± 0.18
ResNet50	DeiT-T	$81.30_{\pm 0.35}$	± 0.12
ConvNeXt-T	Swin-N	$83.51_{\pm 0.13}$	± 0.21
Mixer-B/16	Swin-N	$79.94_{\pm 0.06}$	± 0.13
Average std		± 0.21	± 0.18

Prototype-seed variance (± 0.06–0.35 pp, average ± 0.21 pp) lies in the same range as ordinary training-seed variance (± 0.12–0.24 pp, average ± 0.18 pp). This is what the design predicts: once M is large enough, any random draw of $\mathbf{P}$ approximately preserves pairwise inter-sample geometry, so the particular realization of the prototype space is not an additional axis of sensitivity beyond the randomness already present in training. Because these runs fix the training seed rather than averaging over it, their means differ slightly from the corresponding entries of Table 2, which average over three training seeds.