Logit-Based Losses Limit the Effectiveness of Feature Knowledge Distillation

Nicholas Cooper NICK.COOPER@COLORADO.EDU
Lijun Chen LIJUN.CHEN@COLORADO.EDU
Sailesh Dwivedy SAILESH.DWIVEDY@COLORADO.EDU
Danna Gurari DANNA.GURARI@COLORADO.EDU
Department of Computer Science, University of Colorado Boulder

Editors: List of editors' names

Abstract

Knowledge distillation (KD) methods transfer the knowledge of a parameter-heavy teacher model to a light-weight student model. The status quo for feature KD methods is to utilize loss functions based on logits (i.e., pre-softmax class scores) and intermediate layer features (i.e., latent representations). Unlike previous approaches, we propose a feature KD framework for training the student's backbone using feature-based losses *exclusively* (i.e., without logit-based losses such as cross entropy). Leveraging recent discoveries about the geometry of latent representations, we introduce a *knowledge quality metric* for identifying which teacher layers provide the most effective knowledge for distillation. Experiments on three image classification datasets with four diverse student-teacher pairs, spanning convolutional neural networks and vision transformers, demonstrate our KD method achieves state-of-the-art performance, delivering top-1 accuracy boosts of up to 15% over standard approaches. We publicly share our code to facilitate future work at this GitHub page.

1. Introduction

Knowledge distillation (KD) is a popular approach for model compression which infuses "dark knowledge" from a parameter-heavy teacher model into a more compact student model. Fundamental to KD is the question: *What knowledge should be transferred*? Prior work proposes two answers. *First*, the pioneering KD paper (Hinton et al., 2015) standardized that a student should be trained to mimic the teacher's softened class probability distribution (i.e., temperature-scaled softmax). This is accomplished by regularizing cross entropy (CE) loss with the KL divergence between the softened logits of the two models, making this method solely reliant on logit-based supervision. *Second*, FitNets (Romero et al., 2015) pioneered the popular trend of using additional loss terms to guide the student to also mimic the teacher's intermediate layer features (i.e., latent representations), by minimizing the distance between their corresponding feature maps. We refer to these two techniques as *vanilla knowledge distillation* (VKD) and *feature knowledge distillation* (FKD), respectively.

Our work is motivated by the hypothesis that *the performance of FKD methods is compromised by training student backbones with logit-based losses.* This is inspired by the observation that there are limits to the information logit-losses can transfer. Feature-based losses are computed in very high dimensional spaces, enabling them to capture richer information about a teacher's representations. Logit-based losses, in contrast, are computed

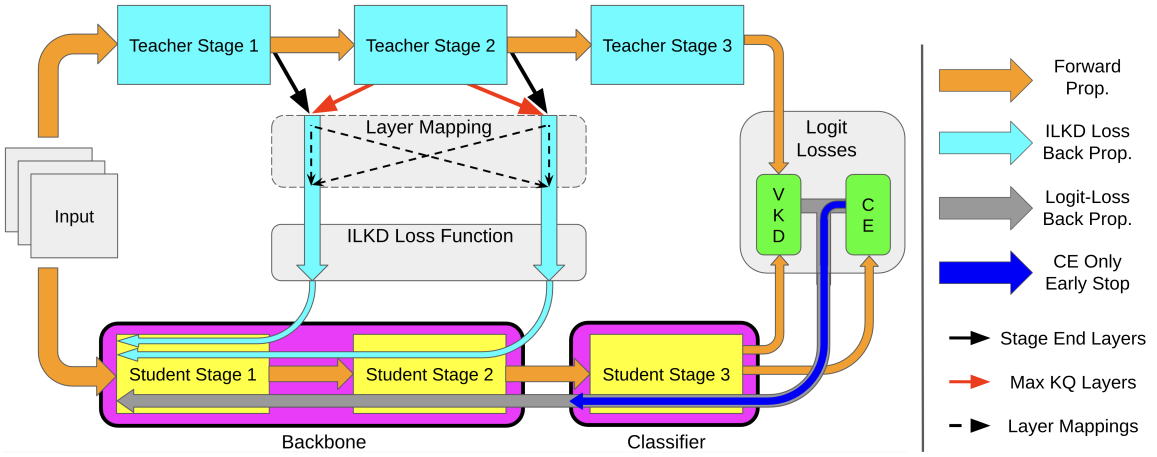

Figure 1: Illustration of our feature knowledge distillation framework and its two key distinctions from prior work. First, while the status quo is to back-propagate logit-based losses through the student backbone (→), our method back-propagates only CE through just the classifier (→). Second, while the default strategy is to select teacher layers to distill from the end of each 'stage' (→), we introduce a metric for principledly selecting the layers with the highest knowledge quality which often occur within a single stage (→).

in lower dimensional spaces, limiting the level of detailed knowledge they can transfer. We suspect that such lower-dimensional losses dilute the potential effectiveness of FKD methods.

Accordingly, we propose a new FKD framework for training student backbones *only* with feature-based losses (i.e., no logit-based loss terms), as illustrated in **Figure 1**. Here, the *student backbone* refers to the layers which receive information from the teacher's intermediate layers. Our key insight is that this loss recipe often fails when standard techniques are used to select intermediate teacher layers (**Figure 4**). We introduce a novel, geometry-aware layer selection metric to select intermediate teacher layers with the highest *knowledge quality* (KQ) for distillation. Experiments with three image classification datasets and four diverse student-teacher pairs, spanning convolutional neural networks (CNNs) and vision transformers (ViT), demonstrate that our proposed approach outperforms existing KD baselines by up to 15% top-1 accuracy. Our fine-grained analyses reveal that (1) logit-based losses prevent the student from fully benefiting from distillation and (2) our teacher layer selection method is necessary to guarantee a performance boost when excluding logit-based losses.

2. Related Work

Feature Knowledge Distillation (FKD). FKD involves three intertwined steps: (1) teacher *layer selection*, (2) teacher-to-student *layer mapping*, and (3) teacher-to-student *dimensional translation* to match the teacher's higher dimensional latent space to that of the student. While much research focuses on the second step (Chen et al., 2021; Zhu et al., 2024; Zhu and Wang, 2021) and third step (Romero et al., 2015; Zagoruyko and Komodakis,

2017; Park et al., 2019; Tung and Mori, 2019; Guo et al., 2023), the initial step of teacher layer selection has received little attention. Most works default to selecting the final layer from each teacher model's *stage*, defined as layers preceding a pooling operation. We instead introduce the first automated metric for teacher layer selection, which identifies layers with the highest knowledge quality.

Distillation Without Logit Losses. Several works have hinted at potential benefits of training the student backbone without logit losses. For example, the classic FitNets (Romero et al., 2015) excluded logit losses during the first part of a two-stage process, but then introduced them in the second phase. More recently, (Lin et al., 2022) showed that removing the VKD loss term does not significantly affect performance. Most similar to our work is (Zhu and Wang, 2021), which demonstrated that FKD methods suffer from mis-alignment between gradients of the logit and feature losses. However, they addressed this issue by dynamically disabling the feature loss rather than removing logit losses. Complementing these works, we instead propose distilling student backbones *without any* logit losses.

Geometry of Latent Representations. Recent observations about the latent geometry of image classification models inspired the design of our metric for measuring the knowledge quality of teacher layers. In particular, growing evidence shows that models process data in two distinct stages: *extraction* and *compression* (Masarczyk et al., 2023). The notion of *compression* first emerged with Deep Neural Collapse (DNC) (Papyan et al., 2020), which demonstrated that models learn to *compress* same-class points towards the class mean in the final layer representations, and was later extended to latent representations (Rangamani et al., 2023). The notion of *extraction* emerged from observations about the *intrinsic dimension* of latent representations (Ansuini et al., 2019; Brown et al., 2022), showing that image classification models process data by *expanding representation dimensionality* in their early layers (e.g., the first 70% of layers) before *compressing representation dimensionality* in later layers (e.g., the final 30% of layers). Our proposed metric is based on three geometric properties of a teacher's feature representations[1]. Experiments reveal that the *transition layers* between extraction and compression exhibit the highest 'knowledge quality' and yield the best KD performance.

3. Methods

3.1. Basic Notation and Background

We consider teacher and student models as sequences of parameterized functions $f(x) = f_l \circ ...f_2 \circ f_1(x) : \mathbb{R}^d \to \mathbb{R}^C$, where d is the dimensionality of the input, C the number of class labels, and l the number of layers. The training dataset (X, Y) consists of a set $X \in \mathbb{R}^{N \times d}$ of N d-dimensional inputs and a set $Y \in \{1, 2, ..., C\}^N$ of ground truth labels.

We use the term *representation* to refer to the output from a sub-sequence of layers. For example, the representations from the i^{th} layer, denoted by R_i, is the set:

$$R_i = \{f_i \circ f_{i-1} \circ ...f_2(x) \circ f_1(x) \mid x \in X\} \subset \mathbb{R}^{d_i}$$

where d_i denotes the *ambient dimension* of the i^{th} layer's representations, e.g., a linear layer with 100 neurons has an ambient dimension of 100. Note that we do not consider

1. While prior work (Theodorakopoulos and Tsourounis, 2023) explored latent feature geometry to improve FKD, they focused on the *dimensional translation* step rather than teacher *layer selection*.

architecture-specific tensor structure (e.g., a c-channel $h \times w$ feature map is interpreted as a "flattened" vector in $\mathbb{R}^{chw}$). The final layer's representations, $R_l \subset \mathbb{R}^C$, are called *logits*.

For layer selection, we denote the index sets of the selected teacher and student layers by $L^T \subseteq \{1, 2, ..., l^T\}$ and $L^S \subseteq \{1, 2, ..., l^S\}$, where l^T and l^S denote the total number of teacher and student layers respectively. The mapping between the teacher and student layers is then defined by a *mapping matrix* $A \in \mathbb{R}^{|L^T| \times |L^S|}$ that assigns weights to the losses computed between each of the teacher-student layer pairs. These weights can be dynamically learned (Chen et al., 2021) or adjusted via handcrafted rules (Zhu and Wang, 2021).

FKD learning involves two types of losses. First is the classic VKD loss (Hinton et al., 2015), computed as follows:

$$\mathcal{L}_{KL}(R_n^S, R_n^T, t) = t^2 KL\left[softmax\left(\frac{R_n^S}{t}\right), softmax\left(\frac{R_n^T}{t}\right)\right] \tag{1}$$

where $KL[\cdot, \cdot]$ denotes the KL divergence, $softmax(\cdot)$ the softmax function, t the temperature, and t^2 a balancing constant that counteracts the decay of KL when logits are softened. Second is a loss computed between pairs of teacher-student intermediate layers, resulting in the following total loss:

$$\mathcal{L}_{FKD} = \mathcal{L}_{CE} + \mathcal{L}_{KL} + \sum_{i \in L^T} \sum_{j \in L^S} A_{ij} \mathcal{L}_F(R_i^T, R_j^S) \tag{2}$$

where $\mathcal{L}_{CE}$ denotes cross-entropy loss and $\mathcal{L}_F$ denotes the feature loss computed after aligning the teacher and student ambient dimensions.

3.2. Proposed Method

We now formalize the two complementary innovations of our proposed FKD framework, illustrated in **Figure 1**.

First, we modify the standard loss recipe in **Equation 2** by removing *all* logit losses from the student backbone during training. Here "backbone" is defined as the set of layers $\{f_i^S \mid i \leq l_{final}^S\}$, where $l_{final}^S := max(L^S)$ denotes the last (deepest) student layer selected for distillation. We drop the $\mathcal{L}_{KL}$ term completely, and stop back-propagation of $\mathcal{L}_{CE}$ at l_{final}^S. This means the student backbone is trained to minimize *only* the feature-based loss $\mathcal{L}_F$, while the student classifier is trained to minimize *only* the cross-entropy loss $\mathcal{L}_{CE}$.

Second, for teacher layer selection (i.e., L^T), we choose the top-k layers with the highest knowledge quality from our metric, which we denote by $\mathcal{Q}$. This metric combines three geometric properties of layer representations as follows:

$$\mathcal{Q}(R) := \mathcal{S}(R) + \sqrt{\mathcal{I}(R)\mathcal{E}(R)} \tag{3}$$

with the components being measures of *separation* ($\mathcal{S}$), *information* ($\mathcal{I}$), and *efficiency* ($\mathcal{E}$). These are computed from the average within-class dot product (avgDPW), between-class dot product (avgDPB), minimum within-class dot product (minDPW), between-class distance (minDistB), and average norm (avgNorm). Due to space limits, we provide the full definitions in the appendix.

Separation measures the extent to which a representation distinguishes classes by using the average within-class cosine similarity and between-class cosine similarity:

$$\mathcal{S}(R) = avgDPW(R) - avgDPB(R) \tag{4}$$

Selecting layers with higher scores ensures the teacher layers can convey rich information about class labels to the student. As we will empirically show, separation increases during the compression stage, which is consistent with prior findings (Rangamani et al., 2023).

Information measures the richness of a layer's "dark knowledge" by considering its embedding dimension and within-class variation. Our use of the *embedding dimension* aligns with prior work (Brown et al., 2022), which found that peak intrinsic dimension (ID)—i.e., the maximum ID achieved across all layers—is correlated with better model performance. While in practice it can be difficult to measure the intrinsic dimension—specifically, the *actual* dimension of the latent feature manifold—from samples (Navarro et al., 2017; Chavez et al., 2001; Facco et al., 2017), the embedding dimension serves as a reasonable proxy that is relatively easy to compute (e.g., with principal component analysis). We also incorporate within-class variance since the ground truth labels lack such information. Formally, we combine the minimum within-class similarity (a proxy for within-class variance), and the average class-wise normalized *SVD Entropy* (a measure of embedding dimension) as follows:

$$\mathcal{I}(R) = [1 - minDPW(R)]\, avgSVDE(R) \tag{5}$$

where $avgSVDE(\cdot)$ is defined as:

$$avgSVDE(R) = \frac{1}{C}\sum_{c=1}^{C}\frac{H(\bar{\sigma_{\mathbf{c}}})}{ln(N_c)}$$

where $H(\bar{\sigma}_c) = -\sum_{i=1}^{d_c}\bar{\sigma}_{ci}ln(\bar{\sigma}_{ci})$ is the Shannon Entropy of the normalized singular values $(\bar{\sigma}_c)$ of the class covariance matrix and d_c is the embedding dimension of class c estimated by PCA [2]. $H(\bar{\sigma}_c)$ attains a maximum value of $ln(N_c)$ when all N_c data points are equidistant from the mean and mutually pair-wise orthogonal, so we normalize it to get values between 0 and 1. Intuitively, $avgSVDE$ measures the dimensionality of the representations weighted by how evenly their variance is distributed across dimensions. We will show experimentally that models increase $\mathcal{I}$ during the extraction layers and then decrease $\mathcal{I}$ in order to increase $\mathcal{S}$ during the compression layers.

Efficiency measures how large the representations are relative to how large they must be to allow their information $(\mathcal{I})$. This is motivated by our empirical observation that representations of large norm often lead to student training instability, which we suspect results from the corresponding increase in the magnitude of $\mathcal{L}_F$. To formalize this, we consider a hyper-spherical packing problem. Suppose there are N data points and a minimum distance of ε is required between any two of them, i.e., $||r_i - r_j||_2 \geq \varepsilon, \forall i \neq j$. Then, for ReLU-family (e.g., ReLU, Leaky ReLU, GELU) activated D-dimensional Euclidean space, the radius of the smallest hypersphere that can accommodate such points can be approximated by:

$$r_{min}(\varepsilon, d) \approx \; 2K\varepsilon \tag{6}$$

with $K = \left(\frac{N}{\pi}\right)^{\frac{1}{D-1}}$. **Equation 6** describes how dimension and radius affect the number of available "ε-rooms" on a hypersphere's surface. We set $\varepsilon = minDistB(R)$, then define the packing efficiency as the ratio between the empirical norm and the estimate of the smallest required norm:

2. We use the number of principal components required to account for 95% of the variance as an estimate of embedding dimension.

$$\mathcal{E}(R) = \frac{2KminDistB(R)}{avgNorm(R)}. \tag{7}$$

Consistent with $\mathcal{I}$, we use a PCA estimate of the global embedding dimension for D. $\mathcal{E}$ signifies when representations are size efficient, with lower values indicating the norm is unnecessarily inflated. This value facilitates selecting teacher layers which will improve the student's ability to converge during training. As we will show in the experiments, $\mathcal{E}$ resembles $\mathcal{I}$ in that it increases during the extraction layers, then decreases during compression.

We make several *implementation* choices for our proposed KD method. We follow existing work (Chen et al., 2021, 2022) and set $|L^T| = |L^S| = 4$ with a simple one-to-one layer mapping. We only consider representations of the training dataset X when computing $\mathcal{Q}$ and selecting L^T. We restrict the definition of "layer" to minimal sets of functions ending with non-linear activation, since nonlinearities influence the geometric properties of R_l[3]. For dimensional translation, we construct a single layer projector. It consists of a pooling layer for spatial alignment, followed by a convolution layer with minimal filters and no bias. We form it directly from the projectors used in (Chen et al., 2021) but reduce the number of layers to 1. In principle, this formulation of $\mathcal{L}_{IL}$ is very similar to (Romero et al., 2015). We use a 2D version for CNNs, and a 1D version for transformers. They are identical up to the order of the parameter tensors. We constructed our KQ metric from $\mathcal{S}, \mathcal{I}$, and $\mathcal{E}$ as shown in **Equation 3** following preliminary empirical analysis that showed this combination of the three components led to the best results.

4. Experiments

We now describe our experiments validating improvements from our proposed approach and the importance of its different design choices.

Model Architectures. To demonstrate that our method is applicable to different architectures, we experiment with convolutional neural networks (CNNs) and vision transformers (ViTs). For *teachers*, we chose VGG19 (Simonyan and Zisserman, 2015), ResNet34 (He et al., 2016), and ViT_B (Dosovitskiy et al., 2021) to examine the efficacy of our $\mathcal{Q}$ metric on a "vanilla" CNN, a residual CNN, and an attention-based architecture. For *students*, we chose VGG11, MobileNetV2 (Sandler et al., 2018), ResNet9, and ViT_ET (extra tiny) to cover the same three classes of architecture.

Training Protocol. Models are trained for 50 epochs with the Adam optimizer (Kingma and Ba, 2015) and a single cycle learning rate schedule (Smith and Topin, 2017). We train without data augmentation so knowledge quality is measured in a reproducible manner. Results with data augmentation are provided in the appendices, and reinforce our findings.

Datasets. We use three datasets commonly employed for knowledge distillation experiments. Two are CIFAR10 and 100 datasets (Krizhevsky et al., 2009), which are composed of 60,000 32x32 color images from 10 and 100 classes, respectively. The third one is Tiny ImageNet (Le and Yang, 2015) that consists of 110,000 64x64 color images from 200 classes.

Evaluation Metrics. We evaluate with top-1 accuracy and average—over all relevant baseline distillation methods—relative improvement (ARI) (Tian et al., 2020). ARI indicates the benefit of KD_1 *relative to* KD_2:

3. E.g., ReLU restricts all points to the non-negative orthant, the relative volume of which vanishes in high dimension as $\frac{1}{2^d}$.

$$ARI(KD_1, KD_2) = \frac{Acc_{KD_1} - Acc_{KD_2}}{Acc_{KD_2} - Acc_{Baseline}}$$

where $Acc_{Baseline}$ denotes the accuracy of the student model without any distillation. For all experiments, we report mean and standard deviations from three, randomly initialized runs.

4.1. Analysis: Where Does Peak KQ Occur?

Common trends for the knowledge quality curves are exemplified in **Figure 2** for ResNet34 and ViT_B on CIFAR100. The components of $\mathcal{Q}$—$\mathcal{S}, \mathcal{I}$, and $\mathcal{E}$—exhibit clear extraction and compression phases in both models, reinforcing prior work's findings (Masarczyk et al., 2023). In the final layers, separation increases at the cost of information and efficiency, a trend that generalizes across datasets (see appendicies). These findings also show $\mathcal{Q}$ peaks at different relative depths for the models with ResNet34 exhibiting knowledge quality decay in the final layers. This suggests that ViT_B contains better knowledge at the logits layer, which we validate experimentally.

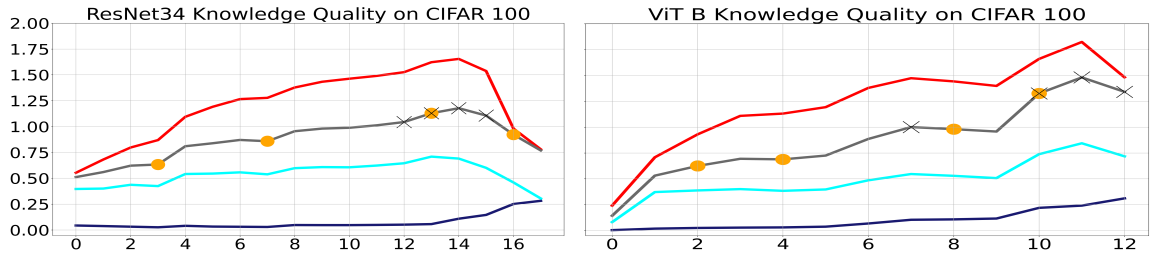

Figure 2: Per-layer knowledge quality analysis of ResNet34 (left) and ViT_B (right) on CIFAR100. X-axes: layer indices. Y-axis: $\mathcal{S}$ (dark blue), $\mathcal{I}$ (light blue), $\mathcal{E}$ (red), $\mathcal{Q}$ (gray). Orange circles indicate standard layer selections and gray Xs indicate maximal knowledge quality layers.

4.2. Distillation Results

Experiment 1: Ours vs. Baselines. We compare our FKD method to seven baselines: Vanilla KD (Van. KD (Hinton et al., 2015)), Logit Standardization (Van. KD Std. (Sun et al., 2024)), Base FKD (a modernization of FitNets (Romero et al., 2015)), Base FKD + FC (base FKD with fully connected layer mapping), Similarity Preserving (Sim. Pres. (Tung and Mori, 2019)), SemCKD ((Chen et al., 2021)), and Reused Teacher Classifier (SimKD Chen et al. (2022)). These represent top-performing logit and feature KD strategies. For the baselines, we use the widely accepted standard of selecting teacher layers at the end of each model stage which occur next to pooling operations. For ViT, which lacks pooling operations, we select layers mimicking the average relative depths chosen for the CNNs. For baselines which use $\mathcal{L}_{KL}$, we adopt common practice (Chen et al., 2021; Tung and Mori, 2019) and set $t = 4$. Results are shown in **Figure 3**.

Overall, our method considerably outperforms all baselines. Moreover, we observe a positive correlation between dataset difficulty and the performance gains achieved by our

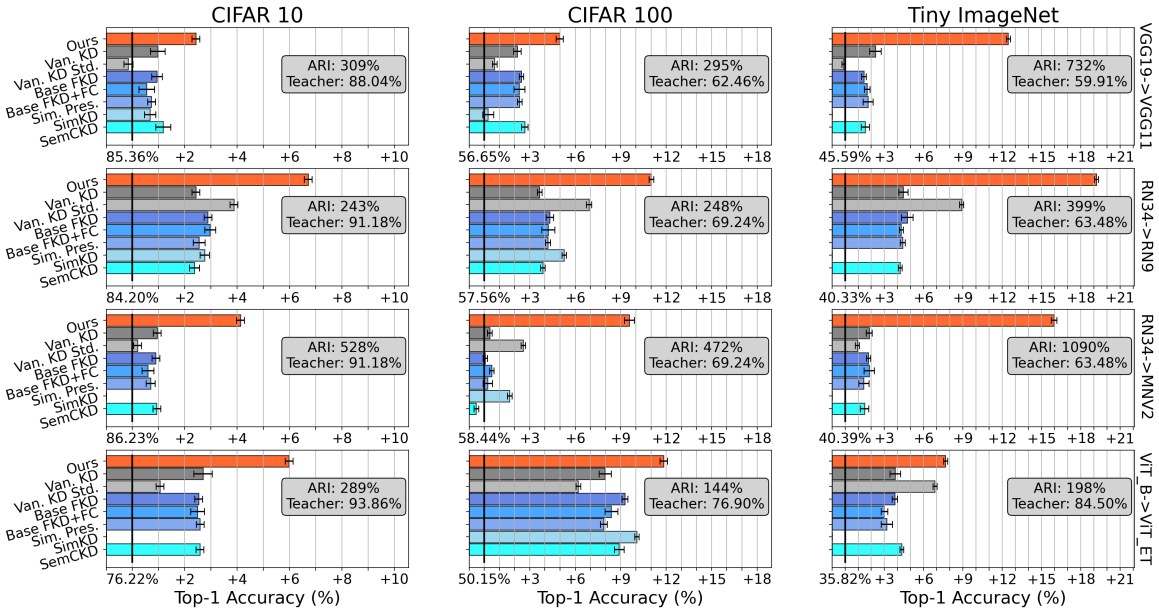

Figure 3: Performance of proposed method and baselines. Vertical black lines denote baseline student performance and the end of each bar shows standard deviation values from three runs. Configurations which failed to converge are not plotted. ARI denotes the mean ARI from our method to all baselines.

method, with the highest ARI scores obtained on Tiny ImageNet. This suggests that the benefit of training the student's backbone without logit losses increases as the classification task becomes less easily separable. We suspect this is because harder datasets result in teacher representations with higher intrinsic dimensionality, thereby offering richer knowledge for the student to learn from.

Examining the influence of model architecture, our method achieves similar absolute benefits but smaller ARIs on transformers than CNNs. We attribute this difference to the way CNNs and ViTs modify knowledge quality across their layers, as shown in **Figure 2**. ViT_B increases $\mathcal{Q}$ in an almost monotone fashion across its layers, whereas CNNs exhibit degrading knowledge quality in their final layers. As predicted by this analysis, logit loss based methods perform better (by $\sim 2x$) on this model pair because ViT contains higher quality knowledge at its final layers than the CNNs.

Experiment 2: When Does $\mathcal{Q}$-Based Layer Selection Matter? We next test both our method and the Base FKD baseline (which are identical up to layer selection and loss recipe) with all combinations of layer selection and loss recipe (holding dimensional translation constant). Specifically, for our method, we added both the CE and KL logit losses ("Ours+LL") and just the CE loss ("Ours+CE"). For Base FKD, we removed the KL loss ("Base FKD-KL") and all logit losses ("Base FKD-LL"). Results are in **Figure 4**. We found our layer selection method is critical, as the student failed to converge during training for 6 of 12 cases using standard layer selection. When logit losses *are* used, our layer selection

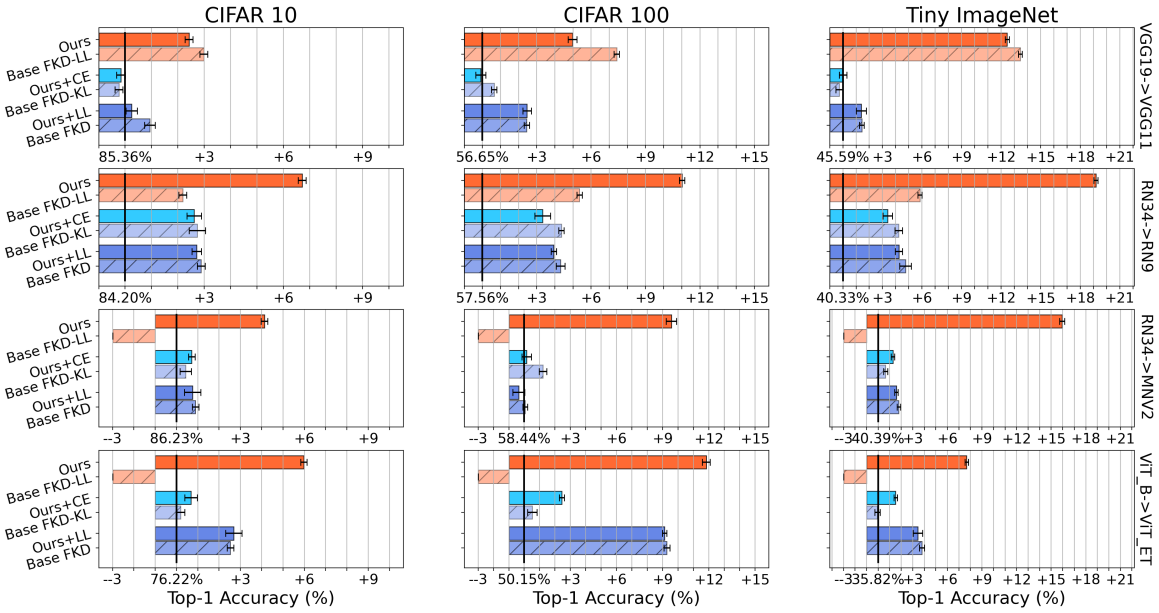

Figure 4: Performance of different teacher layer selection methods when paired with three loss recipes: our loss recipe (Orange), CE loss used in backbone (Light blue), and both CE and KL loss used in backbone (Dark blue). Configurations which failed to converge are clipped to -3 for improved legibility. Ours and standard layer selection are indicated by solid and striped bars, respectively.

method has a smaller benefit suggesting they safeguard against poor quality teacher layers, despite preventing a student from fully learning a teacher's knowledge.

Experiment 3: Ablation on Knowledge Quality Metric. Finally, we evaluate the importance of each term in our KD metric by choosing layers based on each term independently (i.e., $\mathcal{S}$, $\mathcal{I}$, $\mathcal{E}$) as well as the $\sqrt{\mathcal{I}\mathcal{E}}$ term. The *only* strategy which *always* successfully converged during training is our method with all three metrics: $\mathcal{Q}$. The student did not converge in 4 of 12 cases when using only $\mathcal{S}$, $\mathcal{I}$, or $\sqrt{\mathcal{I}\mathcal{E}}$ and in 5 cases when using only $\mathcal{E}$.

5. Conclusion

We have demonstrated that our geometry-aware novel teacher layer selection method enables training the student backbone without any logit-based losses which achieves state-of-the-art KD performance. Directions for future work include: (1) exploring how the $\mathcal{S}$, $\mathcal{I}$, and $\mathcal{E}$ measures relate to model generalization outside of KD, (2) studying how student layer selection influences distillation, and (3) investigating if other tasks benefit from similar loss recipe refinements.

Acknowledgments

We thank François Meyer and James Meiss for insightful discussions.

References

Alessio Ansuini, Alessandro Laio, Jakob H. Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In *NeurIPS*, 2019.

Bradley C.A. Brown, Jordan Juravsky, Anthony L. Caterini, and Gabriel Loaiza-Ganem. Relating regularization and generalization through the intrinsic dimension of activations. In *Workshop on Optimization for Machine Learning*, 2022.

Edgar Chavez, Gonzalo Navarro, Ricardo Baeza-Yates, and Jose Luis Marroquin. Searching in metric spaces. *ACM Computing Surveys*, 33:273–321, 2001.

Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *AAAI*, pages 7028–7036, 2021.

Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen. Knowledge distillation with the reused teacher classifier. In *CVPR*, 2022.

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. In *ICLR*, 2021.

Elena Facco, Maria d'Errico, Alex Rodriguez, and Alessandro Laio. Estimating the intrinsic dimension of datasets by a minimal neighborhood information. In *Nature Sci Rep*, 2017.

Yi Guo, Yiqian He, Xiaoyang Li, Haotong Qin, Van Tung Pham, Yang Zhang, and Shouda Liu. Rdimkd: Generic distillation paradigm by dimensionality reduction, 2023.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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

D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 and cifar100 datasets. 2009. URL https://www.cs.toronto.edu/kriz/cifar.html.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015. URL https://api.semanticscholar.org/CorpusID:16664790.

Sihao Lin, Hongwei Xie, Bing Wang, Kaicheng Yu, Xiaojun Chang, Xiaodan Liang, and Gang Wang. Knowledge distillation via the target-aware transformer. In *CVPR*, pages 10915–10924, 2022.

Wojciech Masarczyk, Mateusz Ostaszewski, Ehsan Imani, Razvan Pascanu, Piotr Milos, and Tomasz Trzcinski. The tunnel effect: Building data representations in deep neural networks. In *NIPS*, 2023.

Gonzalo Navarro, Rodrigo Paredes, Nora Reyes, and Cristian Bustos. An empirical evaluation of intrinsic dimension estimators. In *Information Systems*, pages 206–218, 2017.

Vardan Papyan, X. Y. Han, and David L. Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. In *PNAS*, 2020.

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, page 3967–3976, 2019.

Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso Poggio. Feature learning in deep classifiers through intermediate neural collapse. In *ICML*, 2023.

A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio. Fitnets: Hints for thin deep nets. In *ICLR*, 2015.

M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Raffaello Seri. Asymptotic distributions of covering and separation measures on the hypersphere. In *Discrete and Computational Geometry*, 2023.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2017.

Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *CVPR*, 2024.

Ilias Theodorakopoulos and Dimitrios Tsourounis. A geometric perspective on feature-based distillation. *Studies in Computational Intelligence*, pages 33–63, 2023.

Y. Tian, D. Krishnan, and P. Isola. Contrastive representation distillation. In *ICLR*, 2020.

F. Tung and G. Mori. Similarity-preserving knowledge distillation. In *CVPR*, page 1365–1374, 2019.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

Honglin Zhu, Ning Jiang, Jialiang Tang, and Xinlei Huang. Knowledge distillation via information matching. In *Neural Information Processing, ICONIP.*, pages 405–417, 2024.

Yichen Zhu and Yi Wang. Student customized knowledge distillation: Bridging the gap between student and teacher. In *ICCV*, pages 5057, 5066, 2021.

Appendix A. Theoretical Appendix

A.1. Definitions

Below are the complete definitions of the average within-class dot product (avgDPW), between-class dot product (avgDPB), minimum within-class dot product (minDPW), between-class distance (minDistB), and average norm (avgNorm):

$$avgDPW(R) = \frac{1}{C}\sum_{c=1}^{C}\sum_{(i,j)\in N_c^2}\frac{\langle r_i \cdot r_j\rangle}{|N_c|^2 - |N_c|}$$

$$avgDPB(R) = \frac{2}{C^2 - C}\sum_{c=1}^{C}\sum_{c'=c+1}^{C}\sum_{i\in N_c}\sum_{j\in N_{c'}}\frac{\langle r_i \cdot r_j\rangle}{|N_c||N_{c'}|}$$

$$minDPW(R) = \frac{1}{C}\sum_{c=1}^{C}\min_{(i,j)\in N_c^2}|\langle r_i \cdot r_j\rangle|$$

$$minDistB(R) = \frac{2}{C^2 - C}\sum_{c=1}^{C}\sum_{c'=c+1}^{C}\min_{i\in N_c, j\in N_{c'}}\|r_i - r_j\|_2$$

$$avgNorm(R) = \frac{1}{N}\sum_{i=1}^{N}\|r_i\|_2$$

where $\langle r_i \cdot r_i\rangle = \frac{r_i \cdot r_j}{\|r_i\|_2\|r_j\|_2}$ is the normalized dot product (i.e., cosine similarity), N_c is the subset of indices corresponding to class c, and $(i,j) \in N_c^2$ denotes a distinct pair of such indices.

A.2. Hyperspherical Packing

We provide here the proof of the result on hyperspherical packing that is used to define efficiency $\mathcal{E}$. Recall the ReLU activation function:

$$ReLU(x) = \max\{x, 0\}.$$

The D-dimensional ReLU activated space is thus the non-negative orthant $\mathbb{R}_+^D$.

Theorem: Given a set of N distinct points p_i in D-dimensional ReLU activated space:

$$P = \{p_1,\ p_2,\ p_3, \cdots,\ p_N\}$$

and a minimum distance $d_{min} > 0$ between these points, the smallest radius of the D-dimensional hypersphere S^{D-1} centered at the origin that can accommodate the set P of points can be approximated by:

$$r_{min} \approx 2d_{min}\left(\frac{N}{\pi}\right)^{\frac{1}{D-1}}.$$

Proof: Note that the surface area of hypersphere S^{D-1} of radius r is given by:

$$Surf(S^{D-1}(r)) = r^{D-1}\frac{2\pi^{\frac{D+1}{2}}}{\Gamma(\frac{D+1}{2})}$$

where $\Gamma(\cdot)$ is the gamma function. We approximate the problem by finding how many $(D-1)$-balls of radius d_{min} fit in the surface area. This is a good approximation when the number of points

N is large; e.g., $N \geq 50,000$ for CIFAR10/100 and Tiny ImageNet. For a proper treatment of the asymptotic behavior (with respect to N) of this problem, see (Seri, 2023). The volume of a $(D-1)$-ball is given by:

$$Vol(B^{D-1}(d_{min})) = d_{min}^{D-1} \frac{\pi^{\frac{D-1}{2}}}{\Gamma(\frac{D+1}{2})}.$$

The positive orthant only accounts for a factor of 2^{-D} of the hypersphere. So, the number of $(D-1)$-balls of radius d_{min} that can fit on the non-negative orthant portion of the hypersphere of radius r is approximated by:

$$\begin{aligned} N &\approx \frac{2^{-D} Surf(S^{D-1}(r))}{Vol(B^{D-1}(d_{min}))} \\ &= 2^{-D} \frac{r^{D-1}}{d_{min}^{D-1}} \frac{2\pi^{\frac{D+1}{2}}}{\pi^{\frac{D-1}{2}}} \\ &= \pi \left(\frac{r}{2d_{min}} \right)^{D-1}, \end{aligned}$$

from which we obtain:

$$r \approx 2d_{min} \left(\frac{N}{\pi} \right)^{\frac{1}{D-1}}.$$

Remark: We consider ReLU because of its popularity and simple geometric interpretation and analytic properties. Similar results can be obtained for other activation functions such as Leaky ReLU and GeLU.

A.3. Intrinsic and Embedding Dimensions

To facilitate understanding of the important geometric concept of intrinsic dimension, we provide a visualization of the differences between the ambient, intrinsic, and embedding dimensions in **Figure 5**. Pictured there is 3-D Euclidean space with two embedded sub-manifolds; the circle and the plane. Because the circle has a "true" dimension of 1, its ID is 1. Intuitively, this is because at any point on the circle, there are only two possible directions to move; counter-clockwise and clockwise. However, because we cannot draw circles in 1-D Euclidean space, the embedding dimension of the circle is 2.

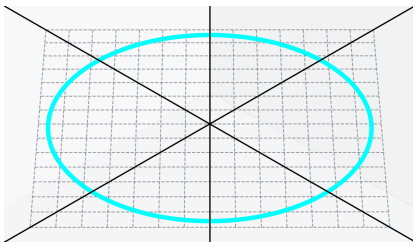

Figure 5: Relationship between the intrinsic, embedding, and ambient dimensions (ID, ED, AD). The blue circle has ID 1, because it is a 1-dimensional manifold. However, it has ED 2 (dashed lines), because it cannot exist in R^n, when $n < 2$. Yet, the circle is drawn in AD 3. Generally, $ID \leq ED \leq AD$.

Visualizing what an increase in embedding dimension might look like can be non-obvious at first glance. To see how embedding dimension can increase via processing in non-linear neural networks, consider **Figure 6**.

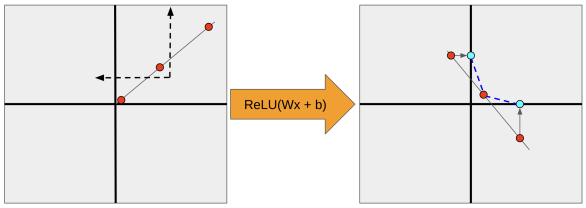

Figure 6: Example of increased embedding dimension as a result of ReLU activation.

Model	MaxLR
VGG19	0.005
VGG11	0.005
ResNet34	0.0075
ResNet9	0.0075
MobileNetV2	0.0075
ViT_B	0.0001
ViT_ET	0.001

Table 1: Maximum learning rates for each architecture. Kept fixed for all datasets and all KD configurations.

Starting on the left, 3 points on a line in $\mathbb{R}^2$ (red dots) with embedding dimension 1. Then, a new basis is defined by a linear layer (dashed lines) and the red dots are reinterpreted in this new basis (right). ReLU activation is applied (blue dots), increasing the embedding dimension to 2 (dashed blue lines). Intuitively, the data are "bent" around a mold of shape defined by the activation function. The position of this mold is determined by the learned change of basis.

Appendix B. Technical Appendix

B.1. Model Architecture Details

ResNet9. ResNet18 consists of 2 residual blocks (He et al., 2016) per stage, meaning there are 2 blocks for each spatial dimension of the representations. ResNet9 is formed simply by reducing this to 1 residual block per stage. All other layers, such as the initial convolution prior to the residual blocks, remain unaltered.

ViT_ET. For the "extra tiny" vision transformer, we follow the structure of ViT_B (Dosovitskiy et al., 2021) but reduce the number of layers, hidden dimension, and number of attention heads. Recall that ViT_B has 12 layers with 12 attention heads, and a hidden dimension of 768. ViT_ET has 8 layers with 6 attention heads, and a hidden dimension of 192. This results in a total of $\sim 3M$ parameters, which is comparable to MobileNetV2 (Sandler et al., 2018) at $2.5M$ parameters. For additional context, VGG11 (Simonyan and Zisserman, 2015) has $28M$ and ResNet9 has $5M$.

B.2. Optimization Details

We used the Adam (Kingma and Ba, 2015) optimizer to train all models. Weight decay was set to 0.01, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the numerical stabilization term was set to $1e-8$. We trained all models for 50 epochs. We used the One Cycle learning rate scheduler (Smith and Topin, 2017),

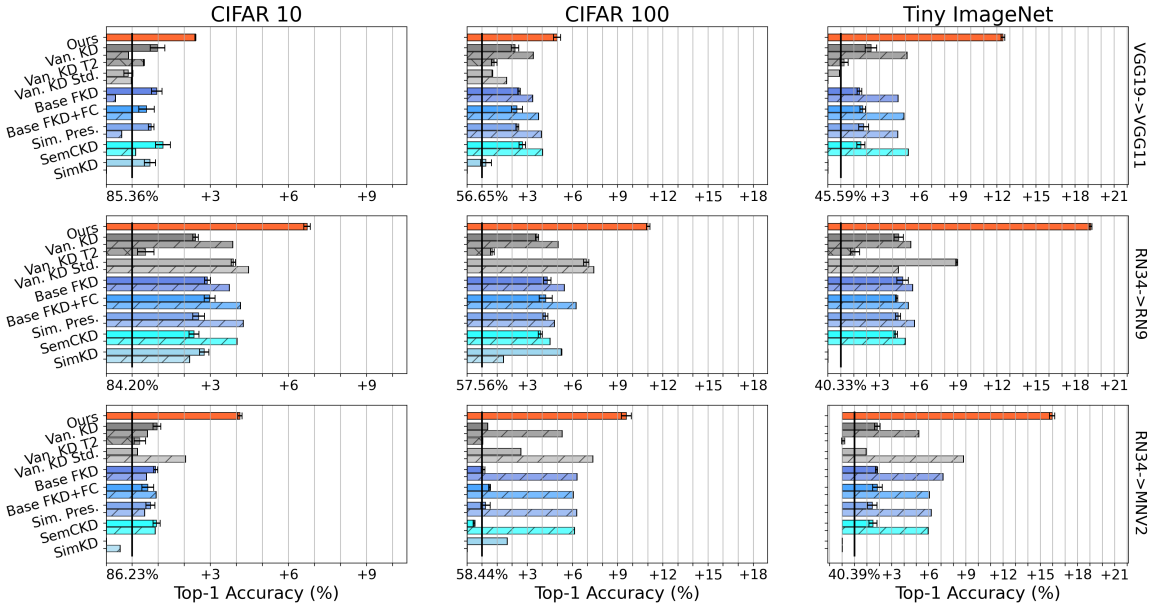

Figure 7: Training protocol validation. Our method is trained for 50 epochs. Baselines are given $\sim 5\times$ training time of 240 epochs (striped bars). VKD-T2 denotes Vanilla KD w/ $T = 2$. Configurations which failed to converge are not plotted.

with its suggested technique for choosing the maximum learning rate; i.e., training baseline models (without any distillation) with a range of different maximum learning rates for 25 epochs to select the optimal value. This was done for each architecture, and then the result was fixed for all experiments. See **Table 1** for the specific values.

The other parameters of the scheduler are as follows: 30% of the iterations were spent increasing the learning rate (η), at the start of training $\eta = \frac{MaxLR}{25}$, and at the end $\eta = \frac{MaxLR}{10000}$. We used cosine annealing, with base and peak momentums set to 0.85 and 0.95, respectively.

B.3. Extra Validation Experiments

Is 50 Epochs Long Enough? We now justify our training approach (**Section B.2**) by showing our findings from experiment 1 of the main paper are consistent even when leveraging other training strategies. Specifically, we follow prior work (Chen et al., 2021; Zhu et al., 2024; Chen et al., 2022) by also training student models for 240 epochs using stochastic gradient descent (SGD). This analysis is intended to address potential concerns that our findings are due to baseline methods not properly converging. We call this the "standard" optimization scheme. We use momentum of 0.9 following a multi-step learning rate scheduler, where the learning rate was stepped down by a factor of 10 at epochs 150, 180, and 210. For ResNet9 and MobileNetV2, the initial learning rate was 0.05, while for VGG11 it was set to 0.01 because 0.05 resulted in failure to converge. We omit the vision transformer model pair (ViT_B $\rightarrow$ ViT_ET) from this experiment because we observed significant performance degradation ($\sim -10\%$) in the student model when using this optimization scheme.

Results are shown in **Figure 7**, with updated ARI values provided in **Figure 8**. Despite being provided with almost $5\times$ more training time, none of the baseline techniques surpass the performance of our method presented in the main paper. Instead, we still observe the same conclusions as articulated in the main paper: (1) our method achieves superior performance, (2) its best results

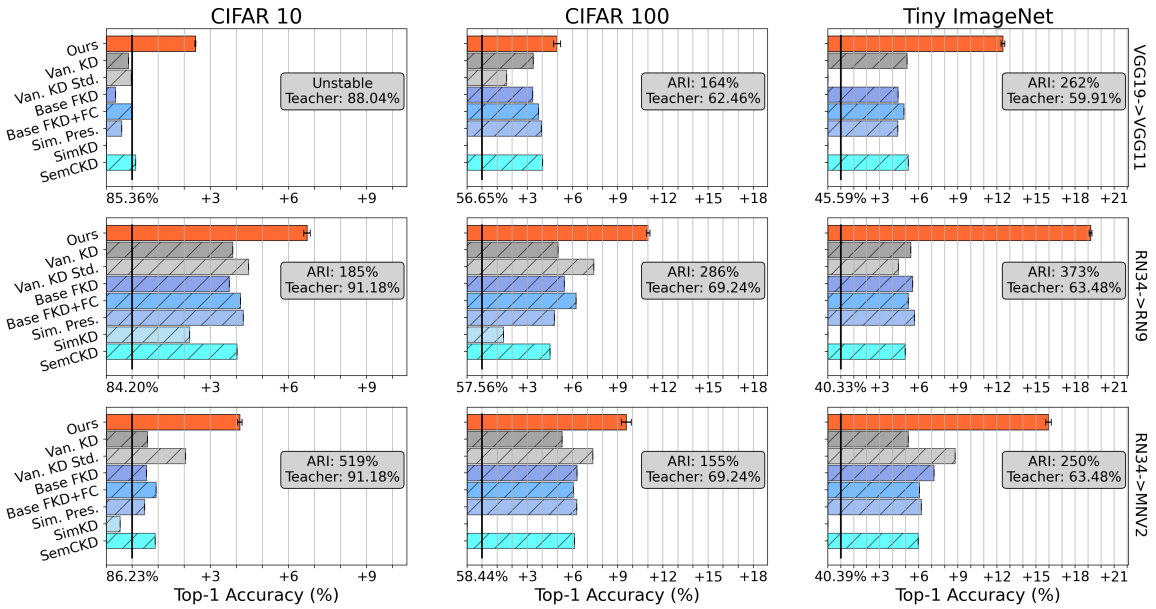

Figure 8: Training protocol validation ARIs. Our method is trained for 50 epochs. Baselines are given $\sim 5\times$ training time of 240 epochs (striped bars). Unstable ARI indicates unrepresentative values due to poor baseline performance. Configurations which failed to converge are not plotted.

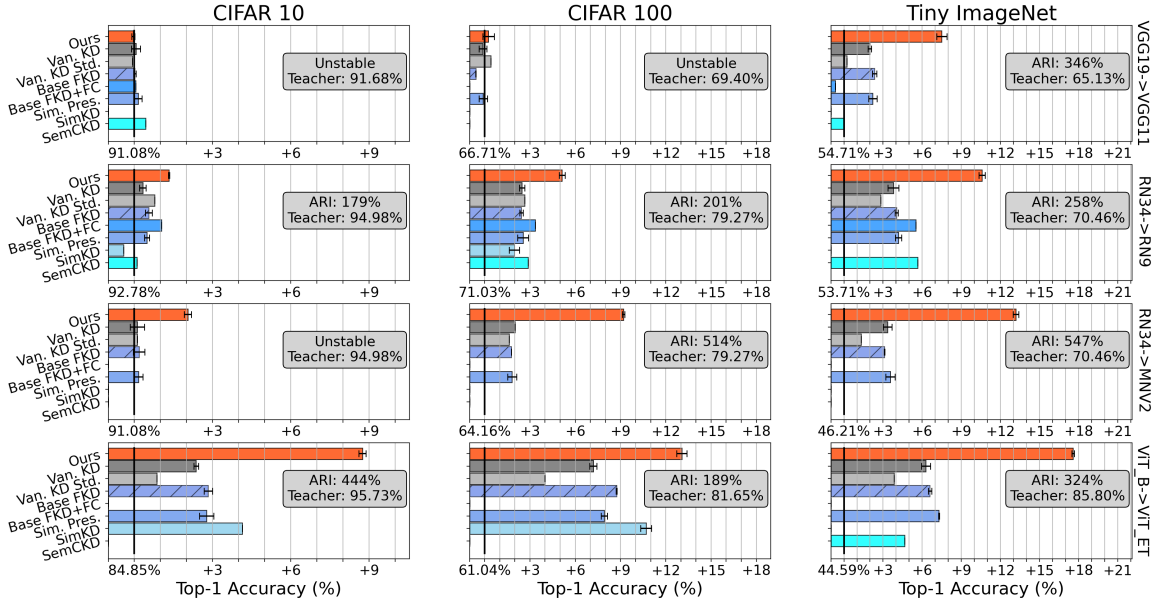

Figure 9: Data Augmentation validation results. Configurations which failed to converge are not plotted. Unstable ARI indicates unrepresentative values due to proximity to baseline student.

are on Tiny ImageNet, and (3) its worst results are on the vision transformer model pair. See **Section C.1** training curves of the student models. Additionally, our method often surpasses the (non-distillation) student's accuracy after only $10-30$ epochs. This is distinctly different behavior from all baseline methods, which appear to demand significantly increased computational budget.

Is $T = 4$ Too High? We also investigated if an overly high temperature could be the cause of poor baseline performance. This is inspired in part by our observation during baseline training that the $\mathcal{L}_{KL}$ loss term was sometimes larger than the cross-entropy loss. We re-trained VKD with a reduced temperature of 2. This configuration is denoted by "VKD-T2". Results are shown in **Figure 7**. We observe that the temperature choice of $T = 4$ is superior to $T = 2$, with the reduced temperature resulting in considerable performance degradation in most cells.

Data Augmentation Experiments. As discussed in the main paper, we did not use data augmentation in the main experiments to ensure reproducible knowledge quality computation. To validate our method's effectiveness when trained with data augmentation, we re-trained all configurations with a standard data augmentation recipe of random horizontal flips (with 50% probability) and random crops from zero padded images. We used a padding size of 4 on all sides of the images. **Figure 9** contains the results.

Once again, we re-observe the same conclusions articulated in the main paper. That is, our method outperforms all the baselines, and achieves its best (relative) results on Tiny Imagenet and worst results on the ViT model pair. In fact, our method achieves a new best recorded result for MobileNetV2 on CIFAR100 at 73.36% accuracy.

Appendix C. Complete Results

C.1. Training Curves

We provide accuracy versus epoch curves for all model pairs presented in Experiment 1 of the main paper. Horizontal black lines are drawn at the baseline student's peak accuracy. For legibility, we plot a representative subset of baselines against the original student (without any distillation) and our method. Vertical lines are drawn at the first epoch where the student surpasses the baseline (no KD) accuracy. Results are shown in (**Figures 10, 11, 12, and 13**).

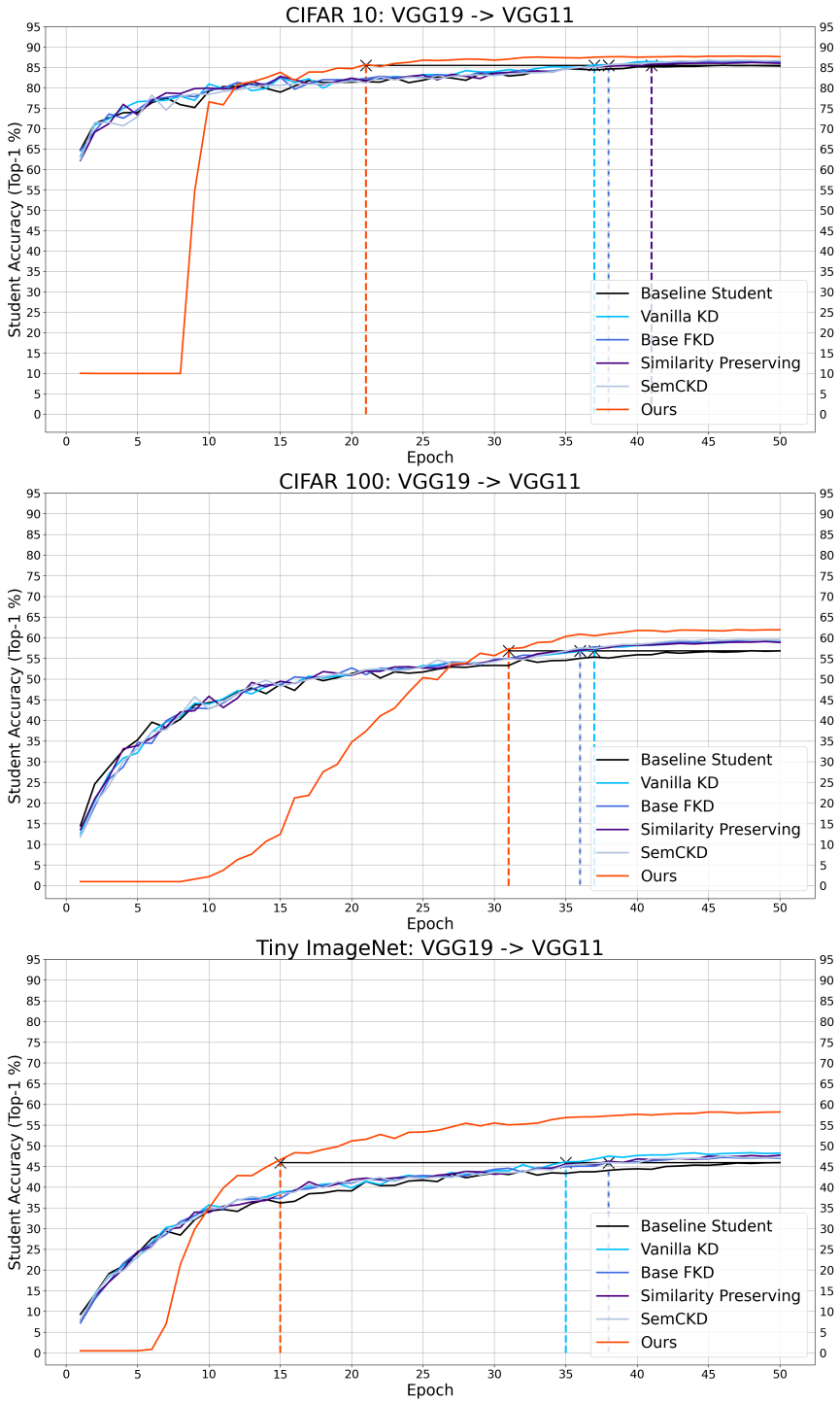

Figure 10: VGG19 → VGG11 Student Training Curves. X-axis: epochs. Y-axis: top-1 accuracy. Our method shown in orange. Vertical lines indicate when each distillation method surpasses the baseline student's accuracy.

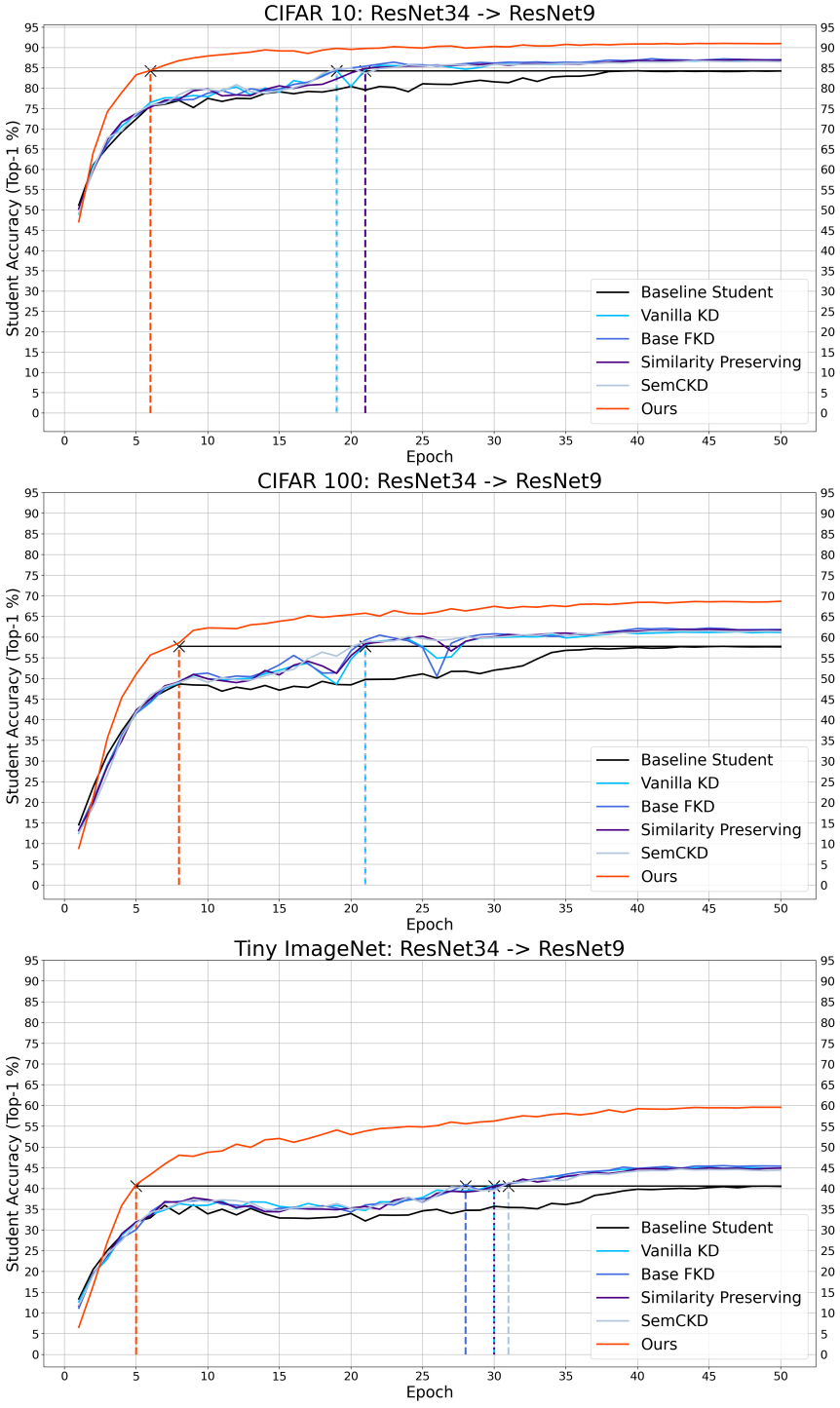

Figure 11: ResNet34 → ResNet9 Training Curves. X-axis: epochs. Y-axis: top-1 accuracy. Our method shown in orange. Vertical lines indicate when each distillation method surpasses the baseline student's accuracy.

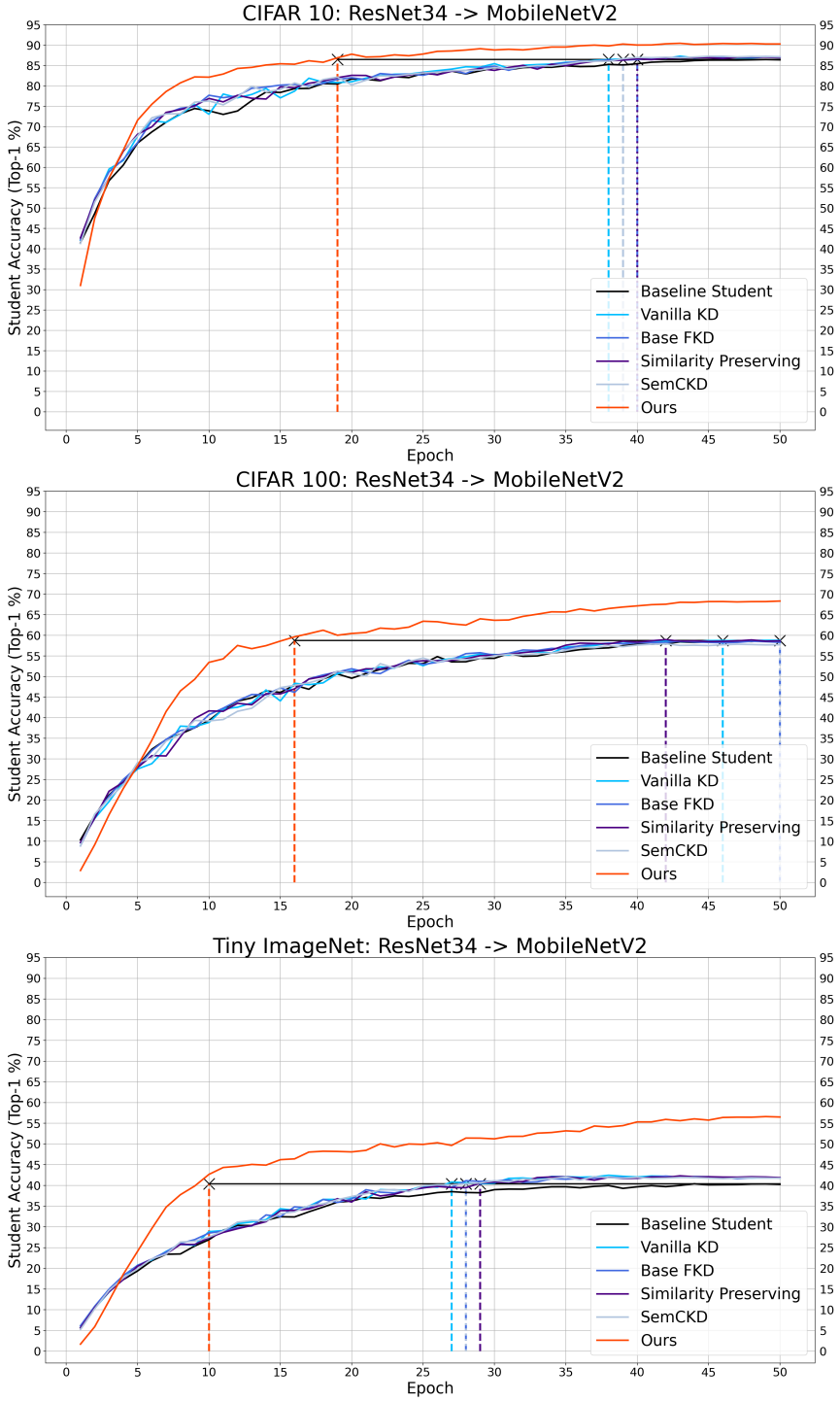

Figure 12: ResNet34 → MobileNetV2 Training Curves. X-axis: epochs. Y-axis: top-1 accuracy. Our method shown in orange. Vertical lines indicate when each distillation method surpasses the baseline student's accuracy.

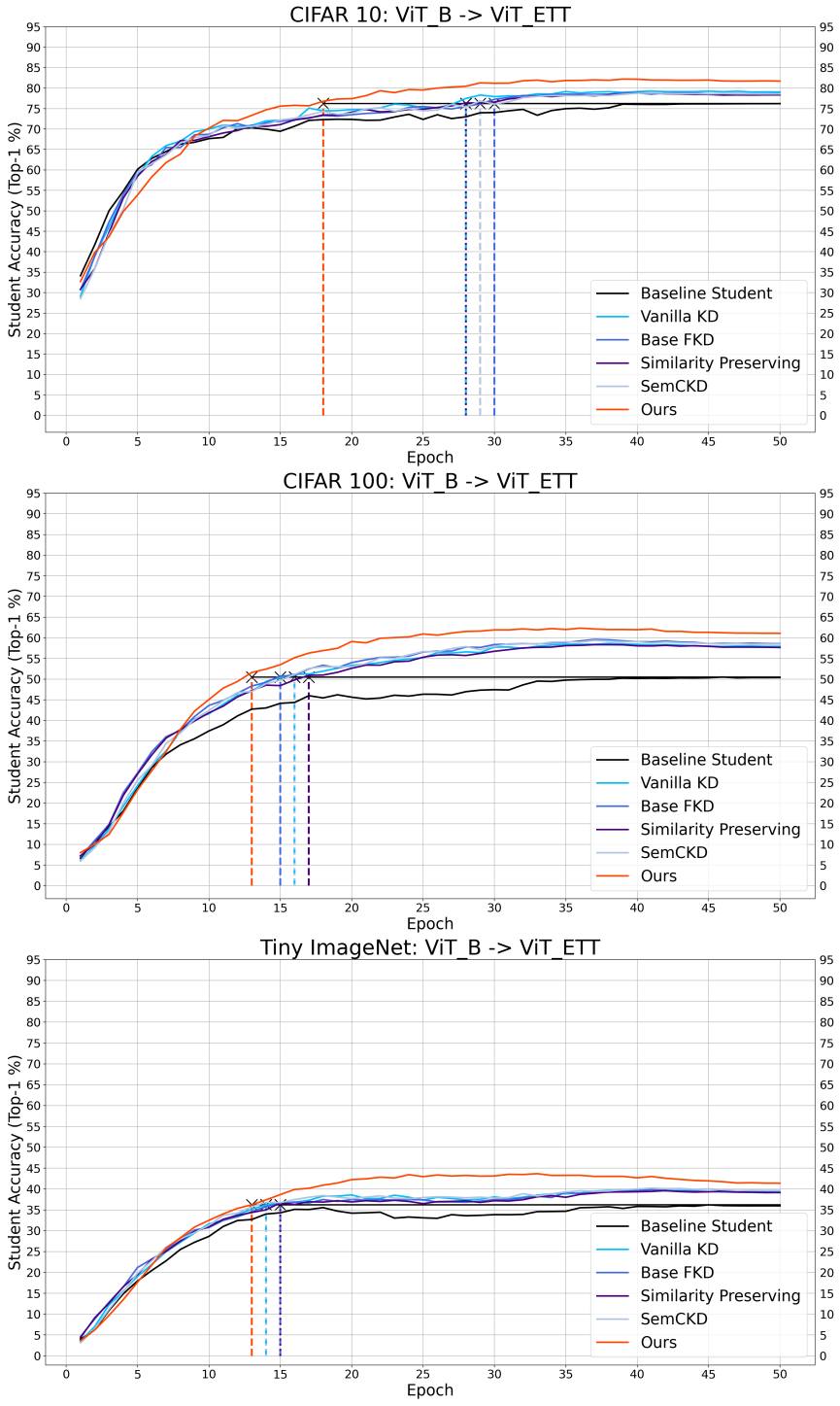

Figure 13: ViT_B → ViT_ET Training Curves. X-axis: epochs. Y-axis: top-1 accuracy. Our method shown in orange. Vertical lines indicate when each distillation method surpasses the baseline student's accuracy.

C.2. Knowledge Quality Plots

We present plots for all teachers and datasets. Per-layer curves of $\mathcal{S}, \mathcal{I}, \mathcal{E}$ are provided for all dataset-teacher combinations (**Figures 14 15 16**). As noted before, none of these curves were obtained with data augmentation to ensure reproducibility.

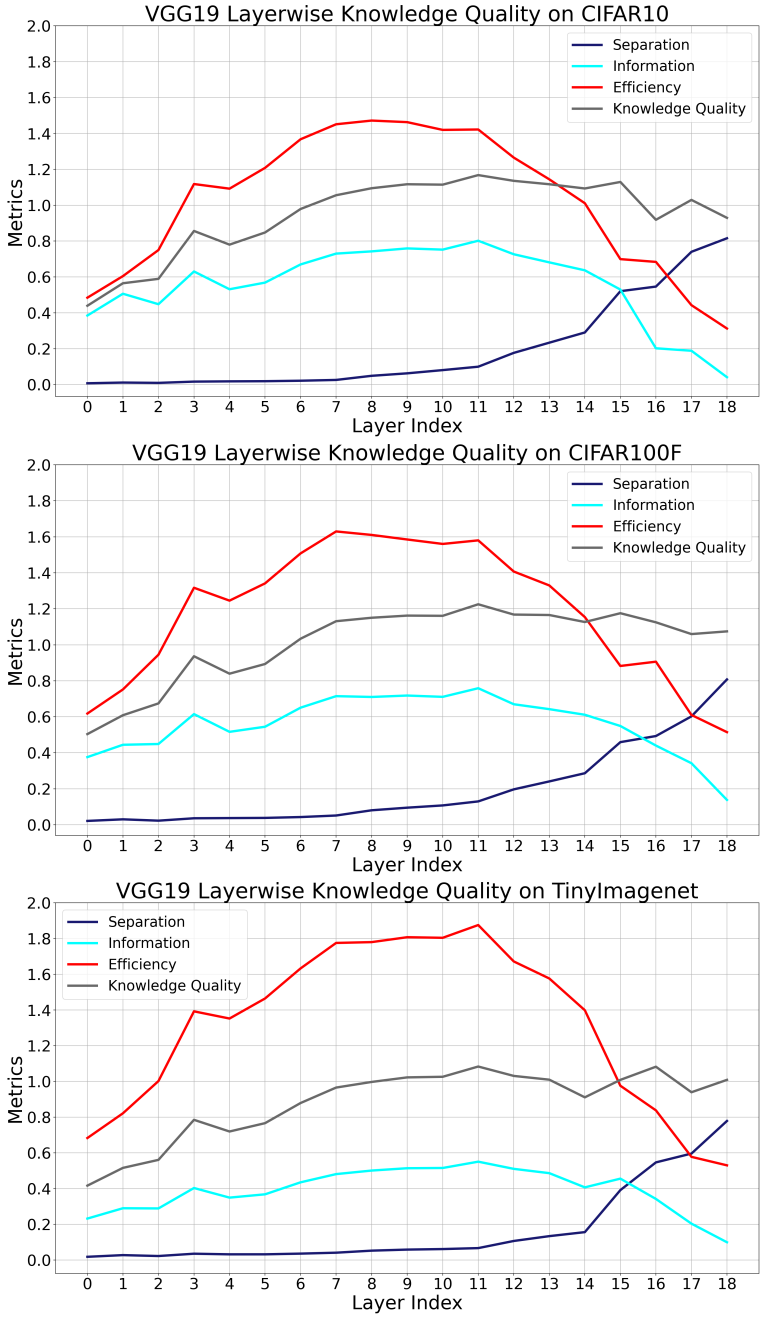

Figure 14: Breakdown of VGG19 Knowledge Quality. From top to bottom: CIFAR10, CIFAR100, Tiny ImageNet.

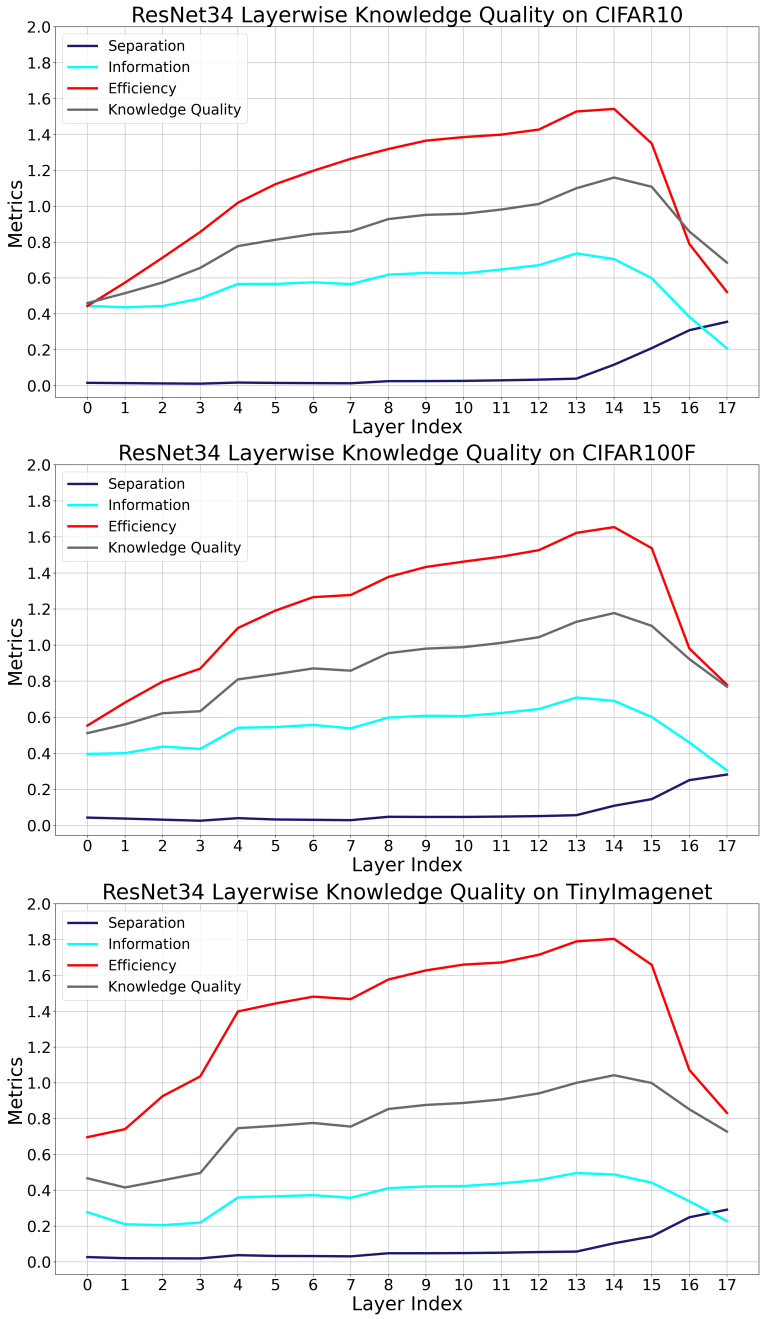

Figure 15: Breakdown of ResNet34 Knowledge Quality. From top to bottom: CIFAR10, CIFAR100, Tiny ImageNet.

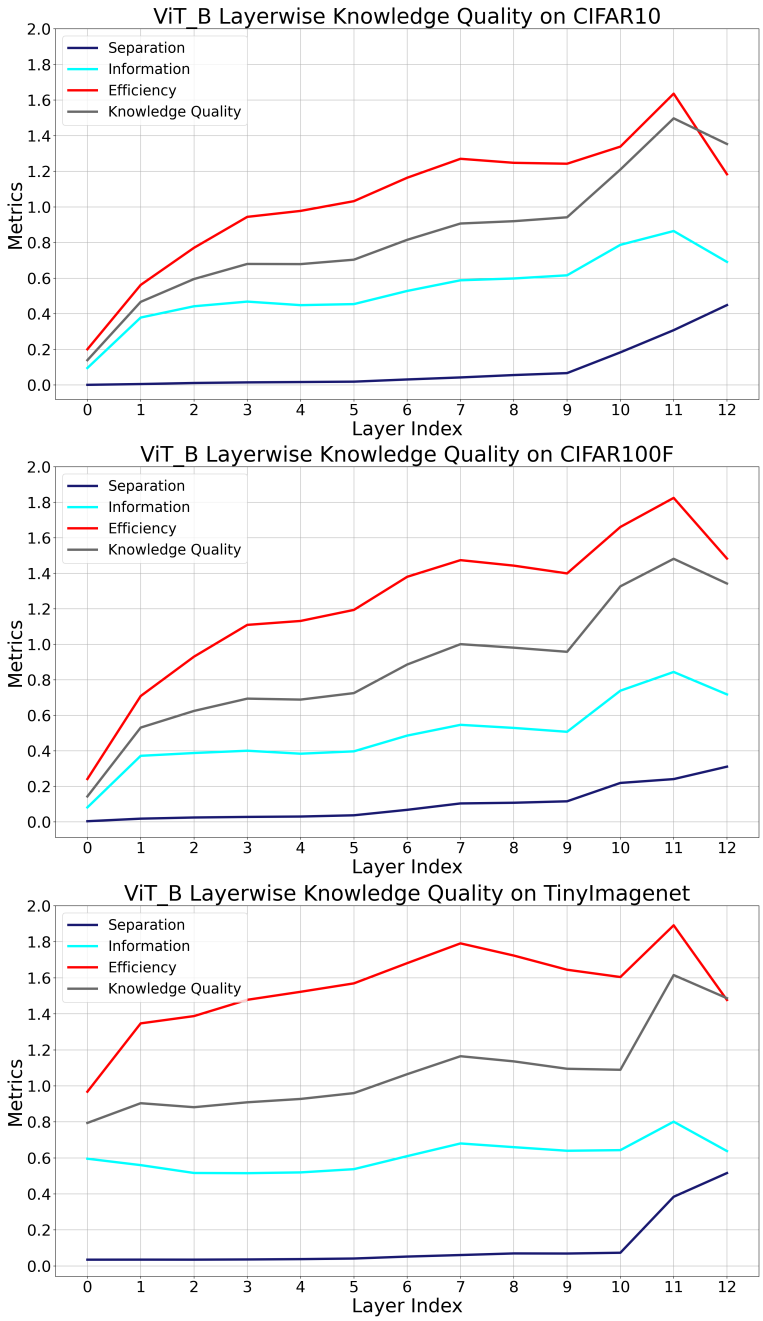

Figure 16: Breakdown of ViT_B Knowledge Quality. From top to bottom: CIFAR10, CIFAR100, Tiny ImageNet.

C.3. Teacher Layer Selections

C.3.1. Index Translation Tables

As discussed in the main paper, we consider only activated representations because of the geometric implications. This results in layer indices which may not have immediately obvious relationships to the traditional definition of "layer". Hence, we provide translation tables. They can be used to infer which "layers" (in the traditional sense) were selected based on the knowledge quality plots and other tables to come. Note that many of the layer types listed below end in non-linear activation; e.g., residual blocks and transformer layers. We do not subdivide either of these. Flatten layers are included when they occur in activated space. For the CNNs, we also indicate the stages each layer belongs to. **Tables 2 and 3** contain this information.

Layer Type	Stage	Index
ResNet34		
Conv2D	0	
BatchNorm2D	0	
ReLU	0	0
MaxPool2D	1	
BasicBlock	1	1
BasicBlock	1	2
BasicBlock	1	3
BasicBlock	2	4
BasicBlock	2	5
BasicBlock	2	6
BasicBlock	2	7
BasicBlock	3	8
BasicBlock	3	9
BasicBlock	3	10
BasicBlock	3	11
BasicBlock	3	12
BasicBlock	3	13
BasicBlock	4	14
BasicBlock	4	15
BasicBlock	4	16
AdaptAvgPool2D	5	
Flatten	5	17
Linear	5	

Layer Type	Index
ViT_B	
ViT Input	
Pos. Embedding	
Dropout	
EncoderBlock	0
EncoderBlock	1
EncoderBlock	2
EncoderBlock	3
EncoderBlock	4
EncoderBlock	5
EncoderBlock	6
EncoderBlock	7
EncoderBlock	8
EncoderBlock	9
EncoderBlock	10
EncoderBlock	11
LayerNorm	
Flatten	12
Linear	

Table 2: ResNet34 (left) and ViT_B (right) Layer Translation Tables. Stage indicates representation spatial resolution. Index denotes the layer indexing system throughout this paper.

Layer Type	Stage	Index
VGG19		
Conv2D	0	
BatchNorm2D	0	
ReLU	0	0
Conv2D	0	
BatchNorm2D	0	
ReLU	0	1
MaxPool2D	1	
Conv2D	1	
BatchNorm2D	1	
ReLU	1	2
Conv2D	1	
BatchNorm2D	1	
ReLU	1	3
MaxPool2D	2	
Conv2D	2	
BatchNorm2D	2	
ReLU	2	4
Conv2D	2	
BatchNorm2D	2	
ReLU	2	5
Conv2D	2	
BatchNorm2D	2	
ReLU	2	6
Conv2D	2	
BatchNorm2D	2	
ReLU	2	7
MaxPool2D	3	
Conv2D	3	
BatchNorm2D	3	
ReLU	3	8
Conv2D	3	
BatchNorm2D	3	
ReLU	3	9

Layer Type	Stage	Index
VGG19 Cont.		
Conv2D	3	
BatchNorm2D	3	
ReLU	3	10
Conv2D	3	
BatchNorm2D	3	
ReLU	3	11
MaxPool2D	4	
Conv2D	4	
BatchNorm2D	4	
ReLU	4	12
Conv2D	4	
BatchNorm2D	4	
ReLU	4	13
Conv2D	4	
BatchNorm2D	4	
ReLU	4	14
Conv2D	4	
BatchNorm2D	4	
ReLU	4	15
MaxPool2D	5	
AdaptAvgPool2D	5	
Flatten	5	16
Linear	5	
ReLU	5	17
Dropout	5	
Linear	5	
ReLU	5	18
Dropout	5	
Linear	5	

Table 3: VGG19 Layer Translation Table. Stage indicates representation spatial resolution. Index denotes the layer indexing system throughout this paper.

C.3.2. Table of Selected Indices

Table 4 contains all selected teacher layer indices L^T for all selection strategies used in the main paper.

L^T	CIFAR10	CIFAR100	Tiny ImageNet
VGG19			
Std	[3, 7, 11, 15]	[3, 7, 11, 15]	[3, 7, 11, 15]
Ours	[11, 12, 13, 15]	[11, 12, 13, 15]	[11, 12, 13, 16]
$\mathcal{Q} = \mathcal{S}$	[15, 16, 17, 18]	[15, 16, 17, 18]	[15, 16, 17, 18]
$\mathcal{Q} = \mathcal{I}$	[8, 9, 10, 11]	[8, 9, 10, 11]	[9, 10, 11, 12]
$\mathcal{Q} = \mathcal{E}$	[7, 8, 9, 10]	[7, 8, 9, 11]	[8, 9, 10, 11]
$\mathcal{Q} = \mathcal{IE}$	[7, 8, 9, 11]	[7, 8, 9, 11]	[8, 9, 10, 11]
ResNet34			
Std	[3, 7, 13, 16]	[3, 7, 13, 16]	[3, 7, 13, 16]
Ours	[12, 13, 14, 15]	[12, 13, 14, 15]	[12, 13, 14, 15]
$\mathcal{Q} = \mathcal{S}$	[14, 15, 16, 17]	[14, 15, 16, 17]	[14, 15, 16, 17]
$\mathcal{Q} = \mathcal{I}$	[11, 12, 13, 14]	[11, 12, 13, 14]	[11, 12, 13, 14]
$\mathcal{Q} = \mathcal{E}$	[11, 12, 13, 14]	[12, 13, 14, 15]	[11, 12, 13, 14]
$\mathcal{Q} = \mathcal{IE}$	[11, 12, 13, 14]	[11, 12, 13, 14]	[12, 13, 14, 15]
ViT_B			
Std	[2, 4, 8, 10]	[2, 4, 8, 10]	[2, 4, 8, 10]
Ours	[9, 10, 11, 12]	[7, 10, 11, 12]	[7, 8, 11, 12]
$\mathcal{Q} = \mathcal{S}$	[9, 10, 11, 12]	[9, 10, 11, 12]	[9, 10, 11, 12]
$\mathcal{Q} = \mathcal{I}$	[9, 10, 11, 12]	[7, 10, 11, 12]	[7, 8, 10, 11]
$\mathcal{Q} = \mathcal{E}$	[7, 8, 10, 11]	[7, 10, 11, 12]	[6, 7, 8, 11]
$\mathcal{Q} = \mathcal{IE}$	[9, 10, 11, 12]	[7, 10, 11, 12]	[7, 8, 9, 11]

Table 4: Indices resulting from all teacher layer selection strategies used in the main paper. Indices can be interpreted via. the layer translation tables.