Student-Teacher Learning from Clean Inputs to Noisy Inputs

Guanzhe Hong, Zhiyuan Mao, Xiaojun Lin, Stanley H. Chan
School of Electrical and Computer Engineering, Purdue University, West Lafayette, Indiana USA
{hong288, mao114, linx, stanchan}@purdue.edu

Abstract

Feature-based student-teacher learning, a training method that encourages the student's hidden features to mimic those of the teacher network, is empirically successful in transferring the knowledge from a pre-trained teacher network to the student network. Furthermore, recent empirical results demonstrate that, the teacher's features can boost the student network's generalization even when the student's input sample is corrupted by noise. However, there is a lack of theoretical insights into why and when this method of transferring knowledge can be successful between such heterogeneous tasks. We analyze this method theoretically using deep linear networks, and experimentally using nonlinear networks. We identify three vital factors to the success of the method: (1) whether the student is trained to zero training loss; (2) how knowledgeable the teacher is on the clean-input problem; (3) how the teacher decomposes its knowledge in its hidden features. Lack of proper control in any of the three factors leads to failure of the student-teacher learning method.

1. Introduction

1.1. What is student-teacher learning?

Student-teacher learning is a form of supervised learning that uses a well-trained *teacher* network to train a *student* network for various low-level and high-level vision tasks. Inspired by the knowledge distillation work of Hinton et al. [14], Romero et al. [24] started a major line of experimental work demonstrating the utility of feature-based student-teacher training [1, 7, 9, 13, 15, 17, 19, 20, 26–28, 31–33].

Figure 1 shows an illustration of the scheme. Suppose that we want to perform classification (or regression) where the input image is corrupted by noise. In student-teacher learning, the teacher is a model trained to classify *clean* images. We assume that the teacher's prediction quality is acceptable, and the features extracted by the teacher are meaningful. However, the teacher cannot handle noisy images because it has never seen one before. Student-teacher learning says that, given a pair of clean-noisy input, we can train a student by forcing the *noisy* features extracted by the student to be similar to those *clean* features extracted by the teacher, via a loss term known as the student-teacher loss. In some sense, the training scheme forces the student network to adjust its weights so that the features are "denoised". During testing, we drop the teacher and use the student for inference.

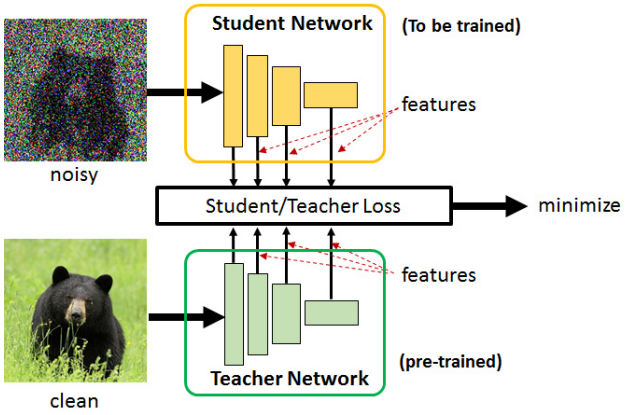

Figure 1. The student-teacher training loss, computed by measuring the difference between the hidden features of the student and the teacher networks. During training, the input signal to the student network is the noisy version of the teacher's.

The success of student-teacher learning from clean inputs to corrupted inputs has been demonstrated in recent papers, including classification with noisy input [9], low-light denoising [7], and image dehazing [15]. However, on the theory side, there is very little analysis of why and when the hidden features of the teacher can boost the generalization power of the student. Most of the explanations in the experimental papers boil down to stating that the hidden features contain rich and abstract information about the task which the teacher solves, which could be difficult for the student network to discover on its own.

In this paper, we provide the first insights into the mechanism of feature-based student teacher learning from clean inputs to noisy inputs, for classification and regression

tasks. The questions we ask are: *When will student-teacher learning succeed? When will it fail? What are the contributing factors? What is the generalization capability of the student?*

The main results of our theoretical and experimental findings can be summarized in the three points below:

- The student should **not** be trained to **zero training loss**.
- A **knowledgeable** teacher is generally preferred, but there are limitations.
- **Well-decomposed** knowledge leads to better knowledge transfer.

To verify these findings, we prove several theoretical results, including showing how missing one or more of those can lead to failure, by studying deep linear networks. We experimentally verify these findings by studying wide nonlinear networks.

1.2. Related works

Most of the existing papers on feature-based student-teacher learning are experimental in nature and include little analysis. As there already are two comprehensive review papers on these works [10, 30], we do not attempt to provide another one here, but instead list a few representative uses of the learning method: homogeneous-task knowledge transfer techniques, which include general-purpose model compression [1, 13, 17, 19, 28, 33], compression of object detection models [31], performance improvement on small datasets [33]; heterogeneous-task knowledge transfer techniques similar to that depicted in Figure 1: the student's input is usually corrupted by noise [9, 27], blur [15], noise with motion [7], etc.

On the theory front, there are three papers that are most related to our work. The first is [29], which formulates student-teacher learning in the framework of "privileged information". We found two limitations of the work: first, it only focuses on student-teacher learning using kernel classifiers and not neural networks; second, it does not clearly identify and elaborate on the factors that lead to the success and failure cases of student-teacher learning. The second paper of interest is [23], which focuses on the *training dynamics* of student-teacher learning, while our work focuses on the *generalization* performance of the trained student. The third one is [22]. Aside from studying *target*-based (instead of feature-based) student-teacher learning for deep linear networks, there are two additional differences between their work and ours: they only focus on the case that the teacher and student's tasks are identical, while we assume the student faces noisy inputs, moreover, some of their messages appear opposite to ours, e.g. from their results, early stopping the student network is not necessary, and might, in fact, harm the student's generalization performance, while we claim the opposite.

1.3. Scope and limitations

We acknowledge that, due to the varieties and use cases of student-teacher learning, a single paper cannot analyze them all. In this paper, we focus on the case depicted in Figure 1: the teacher and student network have identical architecture, no transform is applied to their features, and they solve the same type of task except that the student's input is the noisy version of the teacher's. We do *not* study the learning method in other situations, such as model compression. Moreover, our focus is on the generalization performance of the student, not its training dynamics.

2. Background

We first introduce the notations that shall be used throughout this paper. We denote the clean training samples $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N_s} \subset \mathbb{R}^{d_x} \times \mathbb{R}^{d_y}$, and the noise vectors $\{\boldsymbol{\epsilon}_i\}_{i=1}^{N_s} \subset \mathbb{R}^{d_x}$. For matrix $\boldsymbol{M}$, we use $[\boldsymbol{M}]_{i,j}$ to denote the (i,j) entry of $\boldsymbol{M}$, and $[\boldsymbol{M}]_{i,:}$ and $[\boldsymbol{M}]_{:,j}$ to denote the i-th row and j-th column of $\boldsymbol{M}$. For convenience, we define matrices $[\boldsymbol{X}]_{:,i} = \boldsymbol{x}_i$, $[\boldsymbol{X}_\epsilon]_{:,i} = \boldsymbol{x}_i + \boldsymbol{\epsilon}_i$, and $[\boldsymbol{Y}]_{:,i} = \boldsymbol{y}_i$.

We write an L-layer neural network $\boldsymbol{f}(\boldsymbol{W}_1, ..., \boldsymbol{W}_L; \cdot) : \mathbb{R}^{d_x} \to \mathbb{R}^{d_y}$ as (for simplicity we skip the bias terms):

$$\boldsymbol{f}(\boldsymbol{W}_1, ..., \boldsymbol{W}_L; \boldsymbol{x}) = \sigma(\boldsymbol{W}_L \sigma(\boldsymbol{W}_{L-1} ... \sigma(\boldsymbol{W}_1 \boldsymbol{x}) ...)$$

where the $\boldsymbol{W}_i$'s are the weights, and $\sigma(\cdot)$ is the activation function. In this model, if $\sigma(\cdot)$ is the identity function, we have a deep linear network as a special case. Deep linear networks have been used in the theoretical literature of neural networks [2–4, 18, 22, 25], as they are often more analytically tractable than their nonlinear counterparts, and help provide insights on the mechanisms of the nonlinear networks. We denote $\boldsymbol{W}_{\mathbf{L}} = \prod_{i=1}^{L} \boldsymbol{W}_i$.

While we will demonstrate numerical results for L-layer linear (and nonlinear) networks, to make our theoretical analysis tractable, we make the following assumptions:

Assumptions for theoretical results in this paper:

- Assumption 1: The student and the teacher share the *same* 2-layer architecture: shallow ($L = 2$), fully-connected, and the dimension of the single hidden layer is m.
- Assumption 2: Noise is only applied to the *input* of the student, the targets are always noiseless.

In terms of the training losses of the student network, we

denote $\widehat{\mathcal{L}}_{\text{base}}(\boldsymbol{W}_1, \boldsymbol{W}_2)$ as the *base training loss*:

$$\widehat{\mathcal{L}}_{\text{base}}(\boldsymbol{W}_1, \boldsymbol{W}_2) = \sum_{i=1}^{N_s} \ell(\boldsymbol{f}(\boldsymbol{W}_1, \boldsymbol{W}_2; \underbrace{\boldsymbol{x}_i + \boldsymbol{\epsilon}_i}_{\text{noisy input}}), \underbrace{\boldsymbol{y}_i}_{\text{clean label}}) \tag{1}$$

where $\ell : \mathbb{R}^{d_y} \times \mathbb{R}^{d_y} \to \mathbb{R}_{\geq 0}$ can be, for instance, the square loss. Moreover, we define the *student-teacher (ST) training loss* as follows:

$$\widehat{\mathcal{L}}_{\text{st}}(\boldsymbol{W}_1, \boldsymbol{W}_2) = \underbrace{\widehat{\mathcal{L}}_{\text{base}}(\boldsymbol{W}_1, \boldsymbol{W}_2)}_{\text{base training loss}} + \underbrace{\lambda \sum_{i=1}^{N_s} \|\sigma(\boldsymbol{W}_1(\boldsymbol{x}_i + \boldsymbol{\epsilon}_i)) - \sigma(\widetilde{\boldsymbol{W}}_1 \boldsymbol{x}_i)\|_2^2}_{\text{feature difference loss}} \tag{2}$$

where the $\widetilde{\boldsymbol{W}}_i$'s are the weights of the teacher network. The feature difference loss for 2-layer networks can be easily generalized to deeper networks: for every $h \in \{1, ..., L-1\}$, sum the ℓ_2 difference between the hidden features of layer h from the student and teacher networks.

During testing, we evaluate the student network's generalization performance using the *base testing loss*:

$$\mathcal{L}_{\text{test}}(\boldsymbol{W}_1, \boldsymbol{W}_2) \stackrel{\text{def}}{=} \mathbb{E}_{\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\epsilon}} \left[\ell(\boldsymbol{f}(\boldsymbol{W}_1, \boldsymbol{W}_2; \boldsymbol{x} + \boldsymbol{\epsilon}), \boldsymbol{y}) \right]. \tag{3}$$

Unlike many existing experimental works, we do not apply any additional operation to the hidden features of the student and teacher networks. We choose the particular student-teacher loss because we wish to study this training method in its simplest form. Furthermore, this form of student-teacher loss is close to the ones used in [7, 9].

3. Message I: Do not train student to zero loss

The student-teacher loss (2) can be viewed as the base loss (1) regularized by the feature difference loss [9, 24]. A natural question then arises: since we are already regularizing the base loss, shall we train the overall student-teacher loss to zero so that we have the optimal student-teacher solution? The answer is *no*. The main results are stated as follows.

Message I: Do not train the student to zero training loss.

- Section 3.1: If the deep linear network is over-parametrized $N_s < d_x$, training the student until zero training loss using (2) will return a solution close to the base one (Theorem 1). Similar conclusion holds for $N_s \geq d_x$ (Theorem 2).
- Section 3.2: An early-stopped student trained with (2) has better test error than one trained to convergence.

3.1. Theoretical insights from linear networks

To prove the theoretical results in this sub-section, we assume that $\sigma(\cdot)$ is the identity function, and the base training and testing loss are the MSE loss, i.e. $\ell(\widehat{\boldsymbol{y}}, \boldsymbol{y}) = \|\widehat{\boldsymbol{y}} - \boldsymbol{y}\|_2^2$. We explicitly characterize how close the solutions of the MSE and S/T losses are.

Theorem 1 *Let $L = 2$. Suppose the student's sample amount $N_s < d_x$, $\{\boldsymbol{x}\}_{i=1}^{N_s}$ and $\{\boldsymbol{\epsilon}\}_{i=1}^{N_s}$ are sampled independently from continuous distributions, and the optimizer is gradient flow. Denote $\boldsymbol{W}_i^{base}(t)$ and $\boldsymbol{W}_i^{st}(t)$ as the weights for the student network trained with the base loss (1) and the student-teacher loss (2), respectively.*

Assume that the following statements are true:

i There exists some $\delta > 0$ such that $\|\boldsymbol{W}_i^{base}(0)\|_F \leq \delta$ and $\|\boldsymbol{W}_i^{st}(0)\|_F \leq \delta$ for all i;

ii The teacher network minimizes the training loss for clean data $\sum_{i=1}^{N_s} \ell(\boldsymbol{f}(\widetilde{\boldsymbol{W}}_1, \widetilde{\boldsymbol{W}}_2); \boldsymbol{x}_i)$;

iii Gradient flow successfully converges to a global minimizer for both the MSE- and ST-trained networks

With mild assumptions on the initialized weights and the gradient flow dynamics induced by the two losses, and with δ sufficiently small, the following is true almost surely:

$$\lim_{t \to \infty} \|\boldsymbol{W}_{\boldsymbol{L}}^{base}(t) - \boldsymbol{W}_{\boldsymbol{L}}^{st}(t)\|_F \leq C\delta \tag{4}$$

for some constant C that is independent of δ.

Proof. See supplementary materials.

The implication of the theorem is the following. When we initialize the student's weights with small norms, which is a standard practice [8, 12], and if the teacher satisfies several mild assumptions, then the final solution reached by the MSE- and the student-teacher-induced gradient flow are very close to each other. In other words, *using student-teacher training does not help if we train to zero loss*.

We elaborate on some of the assumptions. The assumption $N_s < d_x$ causes the optimization problem to be underdetermined, leading to nonunique global minima to the base and student-teacher problems. Thus, we need to consider solutions that the gradient flow optimizer chooses. Assumption (iii) simplifies our analysis and is similar to the one made in [4]. It helps us to focus on the end result of the training rather than the dynamics.

We observe similar phenomenon when $N_s \geq d_x$, albeit with stricter assumptions on the two networks.

Theorem 2 *Suppose $N_s \geq d_x$. Assume that $L = 2$, span $\left(\{\boldsymbol{x}_i + \boldsymbol{\epsilon}_i\}_{i=1}^{N_s}\right) = \mathbb{R}^{d_x}$, the teacher network can perfectly interpolate the clean training samples, and the dimension of the hidden space m is no less than*

rank$(\boldsymbol{Y}\boldsymbol{X}_{\boldsymbol{\epsilon}}^T(\boldsymbol{X}_{\boldsymbol{\epsilon}}\boldsymbol{X}_{\boldsymbol{\epsilon}}^T)^{-1})$. *Then the global minimizers of MSE and S/T satisfy:*

$$\boldsymbol{W}_{\boldsymbol{L}}^{base} = \boldsymbol{W}_{\boldsymbol{L}}^{st} = \boldsymbol{Y}\boldsymbol{X}_{\boldsymbol{\epsilon}}^T(\boldsymbol{X}_{\boldsymbol{\epsilon}}\boldsymbol{X}_{\boldsymbol{\epsilon}}^T)^{-1} \tag{5}$$

Proof. See supplementary materials.

Theorem 2 tells us that when the teacher network has zero training error on the clean-input task, and the student possesses sufficient capacity, MSE and S/T learning produce exactly the same student network. Additionally, as proven in the supplementary materials, very similar versions of the current and previous theorem hold even if the teacher's activation function is *not* the identity. It can be any function.

The two theorems show that, even though the feature difference loss in (2) can be viewed as a regularizer, it is important to add other regularizers or use early stopping so that (2) can provide benefit to the student.

3.2. Experimental evidence

Since the theoretical analysis has provided justifications to the linear networks, in this sub-section, we conduct a numerical experiment on nonlinear networks to strengthen our claims.

Choices of teacher and student. We consider a teacher and a student that both are shallow and wide fully-connected ReLU networks with hidden dimension $m = 20,000$, input dimension $d_x = 500$, and output dimension $d_y = 1$. We assume that the teacher network is the ground truth here, and the teacher's layers are set by the Xavier Normal initialization in PyTorch, i.e. each entry of $\widetilde{\boldsymbol{W}}_1$ is sampled from $\mathcal{N}(0, 2/(d_x + m))$, and each entry of $\widetilde{\boldsymbol{W}}_2$ is sampled from $\mathcal{N}(0, 2/(d_y + m))$. The clean input data $\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and the noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \sigma_\epsilon^2 \boldsymbol{I})$, with $\sigma_\epsilon = 0.5$. The loss $\ell(\cdot, \cdot)$ is the square loss, so the learning task is MSE regression. All networks are optimized with batch gradient descent.

Experimental setting. The goal of the experiment is to demonstrate the benefit of early stopping to the trained student's testing error. We first randomly sample $\{\boldsymbol{x}_i + \boldsymbol{\epsilon}_i\}_{i=1}^{N_s}$ and compute the $\{y_i\}_{i=1}^{N_s}$. To train the student network using (2), we carry out parameter sweep over λ in (2), and for each λ used, we record that student's best testing error during training and at the end of training. Note that all of these trained students use Xavier normal initialization with the same random seed and the same training samples. We found that the best test error always occurs *during* training, i.e. early stopping is necessary. Out of all the early-stopped networks trained with different λ's, we pick out the one that has the best early-stopped test error, and plot this error on the "Early-Stopped" curve, and that network's error at the end of training on the "Zero Training Loss" curve. Finally, for comparison purposes, for all the N_s's we choose, we also train student networks using the base loss (1), with the same samples and initialization, and early-stopped for optimal generalization.

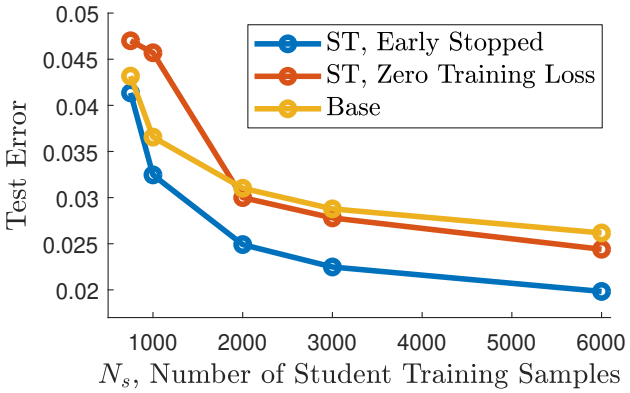

Figure 2. Testing error of student networks trained with the student-teacher loss (2), with and without early stopping, and student network trained with the base loss (1). The figure shows that early stopping is necessary for student-teacher learning to have significant improvement over baseline learning.

Conclusion. The experimental results are depicted in Figure 2. The horizontal axis is N_s, i.e. the amount of noisy training samples available to the student, and the vertical axis is the test error of the trained student. Indeed, the early-stopped students trained with (2) can outperform both the "zero-training-loss" student and the baseline student, which supports the necessity of early-stopping the student.

4. Message II: Use a knowledgeable teacher

In this section, we shift our attention to the teacher. We consider the following questions: How knowledgeable should the teacher be (1) if we want student-teacher learning to generalize better than using the base learning? (2) if the input data becomes noisier so that more help from the teacher is needed? To quantify the level of a teacher's "knowledge", we use the number of training samples seen by the teacher as a proxy. The intuition is that if the teacher sees more (clean) samples, it should be more knowledgeable.

Message II: For any teacher pre-trained with a finite amount of data, there exists an operating regime for the student-teacher learning to be effective. The regime depends on the number of training samples available to the teacher and student. Generally, a more knowledgeable teacher is preferred.

- Section 4.2: If more training samples are available to the students, the teacher needs to be more knowledgeable for the student-teacher learning to be effective.

- Section 4.3: If the student's task becomes more difficult (i.e. the noise level is higher), the teacher needs to be more knowledgeable in order to help the student.

4.1. Experimental setting

We conduct several experiments using deep nonlinear convolutional networks to verify the message. Before we dive into the details, we define a few notations, as shown in Table 1.

N_s	number of noisy samples/class for student
N_t	number of clean samples/class for teacher
$\boldsymbol{f}_{\text{t}}(N_t)$	teacher trained with N_t clean samples
$\boldsymbol{f}_{\text{st}}(N_t, N_s)$	student trained with N_s noisy samples and $\boldsymbol{f}_t(N_t)$ using student-teacher loss (2)
$\boldsymbol{f}_{\text{base}}(N_s)$	same as $\boldsymbol{f}_{\text{st}}$ but trained using base loss (1)
$E_{\text{t}}(N_t)$	testing error for $\boldsymbol{f}_{\text{t}}(N_t)$
$E_{\text{st}}(N_t, N_s)$	testing error for $\boldsymbol{f}_{\text{st}}(N_t, N_s)$
$E_{\text{base}}(N_s)$	testing error for $\boldsymbol{f}_{\text{base}}(N_s)$

Table 1. Notations for Section 4.

The goal of this experiment is to show the regime where student-teacher learning is beneficial. To this end, we aim to visualize the equation

$$E_{\text{st}}(N_t, N_s) \leq (1-\delta)E_{\text{base}}(N_s), \tag{6}$$

for some hyper-parameter $\delta > 0$. Given noise level σ_ϵ, this equation depends on how knowledgeable the teachers is (based on N_t), and how many samples the student can see (N_s).

For this experiment, we consider a classification problem on CIFAR10 dataset. We use ResNet-18 as the backbone for both student and teacher networks. The feature-difference loss is applied to the output of each stage of the network, and we fix the hyper-parameter λ in (2) to 0.001 for all training instances, as it already yields good testing error. Optimization-wise, both the student and the teacher networks are trained with SGD optimizer from scratch for 300 epochs, and the learning rate is set to 0.01 initially and is divided by 10 after every 100 epochs. To make early stopping possible, we allocate 2000 images from the testing set to form a validation set. The best model on the validation set from the 300 epochs is saved.

To minimize the random effect during the training process, we do not use any dropout or data augmentation. We also make sure that the networks with the same training sample amount (N_s or N_t) are trained with the *same* subset of images. Each model is trained 5 times with different random seeds, and the average performance is reported.

4.2. Operating regime of student-teacher learning

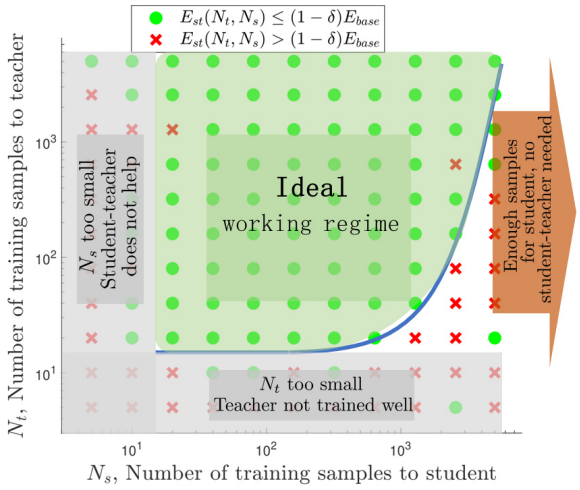

Figure 3. Operating regime of student-teacher learning. Green circles • represent the actual numerical experiment where $E_{\text{st}} \leq (1-\delta)E_{\text{base}}$, and red crosses × represent $E_{\text{st}} > (1-\delta)E_{\text{base}}$. We highlight regions where students-teacher learning can be benefited. If N_t is too small, N_s is too small, or N_s is too large, there is little benefit of student-teacher learning.

Understanding the operating regime can be broken down into two sub-questions:

(1a) Is there a range of N_s such that regardless of how big N_t is, student-teacher learning simply cannot beat base learning?

(1b) Away from the regime in (1a), as N_s varies, how should N_t, the teacher's training sample quantity, change such that student-teacher learning can outperform base learning?

Generation of Figure 3. The answers to the above questions can be obtained from Figure 3. The figure's x-axis is N_s and y-axis is N_t. Parameter-wise, the data in the figure is generated by varying N_s and N_t, while keeping σ_ϵ fixed to 0.5. Procedure-wise, we first select two sets, $\mathcal{N}_t$ and $\mathcal{N}_s \subset \mathbb{N}$. For every $N_t \in \mathcal{N}_t$, we train a teacher network $\boldsymbol{f}_{\text{t}}(N_t)$, early-stopped to have the best testing error on the clean-input task. Then for each $N_s \in \mathcal{N}_s$ and each $\boldsymbol{f}_{\text{t}}(N_t)$, we train a student network $\boldsymbol{f}_{\text{st}}(N_t, N_s)$ using the student-teacher loss (2), and train a $\boldsymbol{f}_{\text{base}}(N_s)$ with the base loss (1). The above experiment is repeated over different N_t's. Now, we fix $\delta = 0.02$, and compare $E_{\text{st}}(N_t, N_s)$ against $E_{\text{base}}(N_s)$ over all the pairs of N_t and N_s. If $E_{\text{st}}(N_t, N_s) \leq (1-\delta)E_{\text{base}}(N_s)$, we mark the position (N_t, N_s) with a green dot in the figure, otherwise, we mark it with a red cross. For clearer visualization, we use color blocks to emphasize the important regions in the figure.

Answering question (1a). In Figure 3, we see that when N_s is too small, the region is filled with red crosses, i.e. student-teacher learning cannot outperform baseline learning regardless of what N_t is. Intuitively speaking, when N_s is too small, it simply is impossible for the student to extract any meaningful pattern from its training data, regardless of how good the teacher's features are.

Answering question (1b). Figure 3 shows that, as N_s increases, the lower boundary of the green region keeps moving upward, which means that N_t must also increase for student-teacher learning to beat the baseline. This phenomenon is also intuitive to understand: as the student sees more and more training samples, its ability to capture the target-relevant information in the noisy input would also grow, so it should also have higher demand on how much target-relevant hidden-feature information the teacher provides about the clean input.

4.3. The influence of student's task difficulty

Another related question is the following:

(2) How knowledgeable should the teacher be when the student needs to handle a difficult task, so that student-teacher learning is effective?

To answer the above question, we conduct the following experiment. We fix $N_s = 320$ and $\delta = 0.04$, increase σ_ϵ from 0.1 to 0.5 by steps of 0.1, and observe how N_t needs to change in order for $E_{\text{st}}(N_t, N_s) \leq (1-\delta)E_{\text{base}}(N_s)$ to be maintained. The result is shown in Table 2. Note that the N_t's vary by steps of 100.

Interpreting Table 2. It can be seen that as σ_ϵ increases, N_t must also increase in order for $E_{\text{st}}(N_t, N_s) \leq (1-\delta)E_{\text{base}}(N_s)$. Intuitively speaking, as the noise in the student's training input samples becomes heavier, it becomes harder for the student to extract target-relevant patterns from the input, as the noise obscures the clean patterns. This in turn means that the teacher needs to give the student information of greater clarity in order to help the student, and this boils down to an increase in N_t.

σ_ϵ	0.1	0.2	0.3	0.4	0.5
N_t	200	300	500	700	800

Table 2. Minimum training samples N_t required at each σ_ϵ level.

4.4. Summary

The experimental results above suggest a few important observations. Firstly, a large N_t (i.e., a more knowledgeable teacher) is generally beneficial. Secondly, if N_s is too small or too large, the student-teacher offers little benefit. Thirdly, a larger σ_ϵ generally demands a more knowledgeable teacher.

5. Message III: Well-decomposed knowledge leads to better knowledge transfer

In Section 4, we observed that when N_t is large, student-teacher learning usually outperforms the baseline learning. However, the following question remains unanswered: *Does a good teacher only mean someone with a low testing error?* Intuitively we would think yes, because low-test-error means that the teacher performs well on its own task. However, having a low testing error does not mean that the teacher can *teach*. In fact, student-teacher learning benefits from a "divide-and-conquer" strategy. If the knowledge can be decomposed into smaller pieces, student-teacher learning tends to perform better.

Message III: Student-teacher learning improves if the teacher can decompose knowledge into smaller pieces.

- Section 5.2: If the teacher's hidden features have sufficiently low complexity, then it is easy for the student to mimic the teacher's features, hence resulting in low test error on the noisy task (Theorem 3);
- Section 5.3: When N_s is not too small, a similar phenomenon happens for nonlinear networks.

5.1. Theoretical setting

We first need to settle on a way to quantify how decomposed the knowledge is. Since the concept of "knowledge" itself is vague, we acknowledge that any definition surrounding its decomposition would have some degree of arbitrariness.

Unit of knowledge — how neurons are grouped. We adopt the following definition of *units of knowledge* in the hidden layer. For linear networks, the unit is any hidden neuron with weight that has sparsity level of 1, i.e. only one of its entries is nonzero. This choice fits the intuition of the simplest linear transform possible, and is compatible with the popular LASSO regression model. We shall further elaborate on this in section 5.2.

For ReLU networks, we treat any hidden ReLU neuron as one unit of knowledge. When outputs from more ReLU neurons are linearly combined together, we treat them as larger units of knowledge as they form more complex piecewise linear functions. This observation is further supported on wide fully-connected ReLU networks. If such a network was trained with gradient descent and initialized with standard schemes, such as the Xavier Normal initialization, the hidden neurons' weights would be close to their random initialization [5, 16]. Therefore, given a group of these neurons, as long as the group is not too large, their weights are unlikely to be col-linear, so linearly combining the outputs of them indeed create more complex functions.

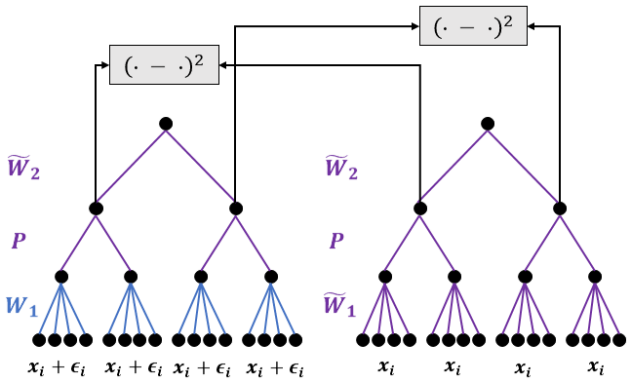

Figure 4. An illustration of the simplified student-teacher loss (7). Here, $d_x = 4$, $m = 4$, and $g = 2$. Notice that the square difference is taken between the pooled features of the student and teacher networks.

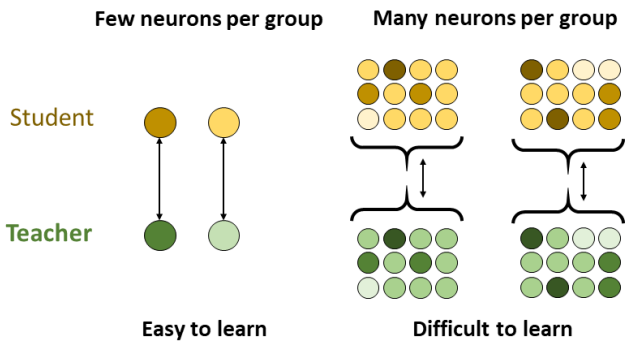

Figure 5. Small-g (left) vs. large-g (right) student-teacher learning.

Additional assumptions. To provide a concrete theoretical result, we make several additional assumptions:

i We assume that the teacher network has **zero test error**. This is the best-case-scenario in Section 4.

ii We focus on the *simplified student-teacher training loss*, defined as follows:

$$\widehat{\mathcal{L}}_{\text{st}}^{\text{simp}}(\boldsymbol{W}_1) = \sum_{i=1}^{N_s} \left\| \boldsymbol{P}\left[\sigma(\boldsymbol{W}_1(\boldsymbol{x}_i + \boldsymbol{\epsilon}_i)) - \sigma(\widetilde{\boldsymbol{W}}_1 \boldsymbol{x}_i)\right] \right\|_2^2 \quad (7)$$

An illustration of the above loss is shown in Figure 4. The base loss $\widehat{\mathcal{L}}_{\text{base}}(\boldsymbol{W}_1, \boldsymbol{W}_2)$, which provides target information, is not present here. The matrix $\boldsymbol{P} \in \mathbb{R}^{(m/g)\times m}$, where $g \in \mathbb{N}$ is a divisor of m, and $\boldsymbol{P}_{i,j} = 1$ if $j \in \{ig, ..., (i+1)g\}$, and zero everywhere else. Multiplication with $\boldsymbol{P}$ essentially *sums every g neurons' output*, similar to how average pooling works in convolutional neural networks. We treat g as a proxy of how decomposed the teacher's features are: the larger it is, the less decomposed the features are.

iii We fix $\boldsymbol{W}_2 = \widetilde{\boldsymbol{W}}_2$, i.e. the second layer of the student is fixed to be identical to the teacher's, and only $\boldsymbol{W}_1$ is trainable. At inference, the student computes $\widetilde{\boldsymbol{W}}_2 \boldsymbol{P} \sigma(\boldsymbol{W}_1(\boldsymbol{x} + \boldsymbol{\epsilon}))$, and teacher computes $\widetilde{\boldsymbol{W}}_2 \boldsymbol{P} \sigma(\widetilde{\boldsymbol{W}}_1 \boldsymbol{x})$.

iv We assume that the entries in the noise vectors $\boldsymbol{\epsilon}$ are all zero-mean Gaussian random variables with variance σ_ϵ^2.

5.2. Theoretical analysis via LASSO

We formulate the knowledge decomposition analysis via LASSO, because it offers the most natural (and clean) analytical results. We use the identity for the activation function $\sigma(\cdot)$. For simplicity, we use $d_y = 1$, and use the square loss for $\ell(\cdot, \cdot)$. Thus, our learning problem reduces to linear regression. Following the suggestion of Section 3, we impose an ℓ_1-regularization onto the student so that it becomes a LASSO.

Theorem 3 *Assume assumptions (i)-(iv) in Section 5.1, and consider the following conditions:*

- *The ground truth is a linear model characterized by the vector $\boldsymbol{\beta}^* \in \mathbb{R}^{d_x}$, and without loss of generality, only the first s entries are nonzero.*
- *The hidden dimension of the networks m is equal to the number of non-zeros s.*
- *The weights of the teacher satisfy $[\widetilde{\boldsymbol{W}}_2]_i = 1$ for all $i = 1, ..., s/g$; $[\widetilde{\boldsymbol{W}}_1]_{i,i} = \beta_i^*$ for $i = 1, ..., s$, and the remaining entries are all zeros. Essentially, the s/g groups of pooled teacher neurons in (7) each has g distinct entries from $\boldsymbol{\beta}^*$.*
- *The number of samples satisfies [1] $N_s \in \widetilde{\Omega}\left(g^2 \log(d_x)\right)$.*
- *The samples $\{\boldsymbol{x}\}_{i=1}^{N_s}$ and some of the parameters above satisfy certain technical conditions (for LASSO analysis).*

Then, with high probability, the student network which minimizes (7) achieves mean square test error

$$\mathbb{E}\left[\left(\widetilde{\boldsymbol{W}}_2 \boldsymbol{P} \boldsymbol{W}_1(\boldsymbol{x} + \boldsymbol{\epsilon}) - \boldsymbol{\beta}^{*T}\boldsymbol{x}\right)^2\right] = \widetilde{\mathcal{O}}^{[1]}\left(\frac{\sigma_\epsilon^2 \|\boldsymbol{\beta}^*\|_2^2}{1+\sigma_\epsilon^2}\right). \quad (8)$$

Proof. See supplementary materials.

Interpreting the theorem. Note that, when g is small, the above error can be quite close to the optimal test error $\sigma_\epsilon^2 \|\boldsymbol{\beta}^*\|_F^2/(1+\sigma_\epsilon^2)$, shown in the supplementary notes.

[1] We hide constants coming from the technical LASSO analysis with $\widetilde{}$ on top of $\mathcal{O}$ and Ω.

More importantly, the required sample amount N_s is independent of s, the "complexity" of the ground truth linear model. In contrast, if we only use the targets to train the student, standard LASSO literature suggests that N_s should at least be $\Omega(s)$ to achieve nontrivial generalization [6,11,21]. Thus, by decomposing the teacher's knowledge into simple units, student-teacher learning can succeed with much fewer samples than base learning. See the supplementary notes for experimental demonstrations of students trained by (7) outperforming those trained with targets by a significant margin.

Besides the fact that the teacher has zero testing error, the key reason behind this effective learning is the "divide-and-conquer" strategy adopted by (7). This idea is roughly illustrated in Figure 5. Imagine that each small disk represents a hidden neuron of a network, and the left and right sides represent two ways of teaching the student. The left is essentially giving the student neurons simple pieces of information *one at a time*, while the right floods the student neurons complex information pooled from many teacher neurons *all at once*. The left side clearly represents a better way of teaching, and corresponds to a choice of small g.

Now, let us consider the more precise example in Figure 4, in which $d_x = 4$, $m = 4$, $g = 2$, and suppose $s = d_x$. If we use the base loss (1) to train the student, the student can only see $\boldsymbol{\beta}^{*T}\boldsymbol{x}$, i.e. the action of every element in $\boldsymbol{\beta}^* = (\beta_1^*, ..., \beta_4^*)$ on $\boldsymbol{x}$ *all at once*. On the other hand, as stated in the third bullet point of Theorem 3, for every $i \in \{1, ..., s\}$, the i^{th} hidden neuron $[\widetilde{\boldsymbol{W}}_1]_{i,:}$ of $\widetilde{\boldsymbol{W}}_1$ encodes exactly the i^{th} entry in $\boldsymbol{\beta}^*$, so the first group of the student neurons sees the action of $(\beta_1^*, \beta_2^*, 0, 0)$ on $\boldsymbol{x}$, and the second group sees the action of $(0, 0, \beta_3^*, \beta_4^*)$ on $\boldsymbol{x}$. In other words, the two groups of student neurons each observes response to the input $\boldsymbol{x}$ created by a 2-sparse subset of $\boldsymbol{\beta}^*$. Due to the lower sparsity in such responses, with the help of LASSO, the student neurons can learn more easily.

On a more abstract level, the above theorem suggests an important angle of studying student-teacher learning: the "simpler" the hidden features of the teacher are, the more likely it is for the student to benefit from the teacher's features.

5.3. Numerical evidence

We verify our claims using a nonlinear network.

Network setting. The networks are shallow and fully-connected, with $m = 20,000$, and the activation function $\sigma(\cdot)$ is the ReLU function. We define $\ell(\cdot, \cdot)$ to be the square loss. All student networks are initialized with the Xavier Normal initialization, and optimized with SGD.

Experiment setting. The clean input signal $\boldsymbol{x} \in \mathbb{R}^{500}$ has the distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$, and the noise has distribution $\mathcal{N}(\boldsymbol{0}, 0.09\boldsymbol{I})$. We assume that the ground truth network is identical to the teacher network. As a result, during testing, we simply compute $\mathbb{E}[(\widetilde{\boldsymbol{W}}_2\boldsymbol{P}\sigma(\boldsymbol{W}_1(\boldsymbol{x} + \boldsymbol{\epsilon})) - \widetilde{\boldsymbol{W}}_2\boldsymbol{P}\sigma(\widetilde{\boldsymbol{W}}_1\boldsymbol{x}))^2]$. To construct the teacher network, we set $\widetilde{\boldsymbol{W}}_1$ with Xavier Normal initialization, and we set $[\widetilde{\boldsymbol{W}}_2]_i = 1$ for all $i \in \{1, ..., m/2g\}$, and $[\widetilde{\boldsymbol{W}}_2]_i = -1$ for all $i \in \{m/2g+1, ..., m/g\}$. Notice that, for any g such that m/g is divisible by 2, the overall function $\widetilde{\boldsymbol{W}}_2\boldsymbol{P}\sigma(\widetilde{\boldsymbol{W}}_1\cdot)$ remains the same, i.e. regardless of what g is, a network trained with the base loss (1) remains the same.

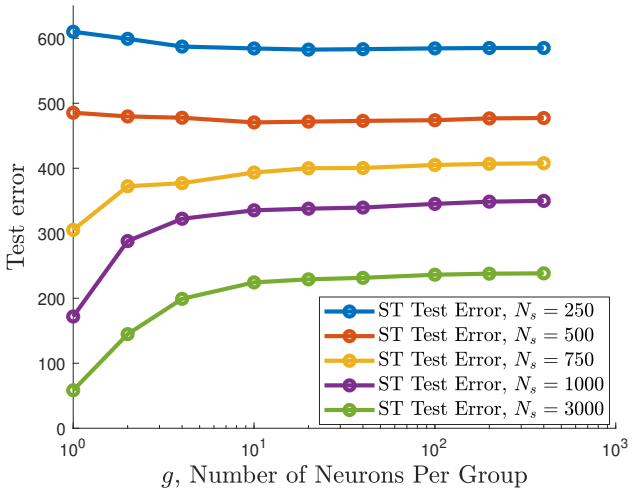

Figure 6. Here, $d_x = 500$ and $m = 20,000$. Test error vs. g, the number of neurons per group. From the figure it is clear that, as long as N_s is not too small, the *fewer* neurons per group, the *lower* the test error of the trained student network.

Interpreting the results. As shown in Figure 6, as long as N_s is not too small, the greater g is, the higher the test error of the student trained with (7). Intuitively speaking, an increase in g means that more teacher neurons are pooled in each of the s/g groups, so the piecewise-linear function formed by each of these groups is more complex. Therefore, it becomes more difficult for the student's hidden neurons to learn with limited samples.

6. Conclusion

This paper offers a systematic analysis of the mechanism of feature-based student-teacher learning. Specifically, the "when" and "why" of the success of student-teacher learning in terms of generalization were studied. Through theoretical and numerical analysis, three conclusions were reached: use early stopping, use a knowledgeable teacher, and make sure that the teacher can decompose its hidden features well. It is our hope that the analytical and experimental results could help systematize the design principles of student-teacher learning, and potentially inspire new learning protocols that better utilize the hidden features of the teacher network, or construct networks that are better at "teaching".

References

[1] Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 34(05):7350–7357, 2020. 1, 2

[2] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *arXiv preprint arXiv:1810.02281*, 2018. 2

[3] Sanjeev Arora, Nadav Cohen, and Elad Hazan. On the optimization of deep networks: Implicit acceleration by overparameterization. In *International Conference on Machine Learning (ICML)*, page 244–253, 2018. 2

[4] Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, page 7413–7424, 2019. 2, 3

[5] Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, Ruslan Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems (NIPS)*, volume 32, pages 8141–8150, 2019. 6

[6] Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011. 8

[7] Yiheng Chi, Abhiram Gnanasambandam, Vladlen Koltun, and Stanley H. Chann. Dynamic low-light imaging with quanta image sensors. In *16th European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3

[8] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS 2010*, volume 9, page 249–256, May 2010. 3

[9] Abhiram Gnanasambandam and Stanley H. Chan. Image classification in the dark using quanta image sensors. In *16th European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3

[10] Jianping Gou, Baosheng Yu, Stephen John Maybank, and Dacheng Tao. Knowledge distillation: A survey. *arXiv preprint arXiv:2006.05525*, 2020. 2

[11] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman & Hall/CRC, 2015. 8

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 8141–8150, 2015. 3

[13] 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 International Conference on Computer Vision (ICCV)*, page 1921–1930, 2019. 1, 2

[14] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 1

[15] Ming Hong, Yuan Xie, Cuihua Li, and Yanyun Qu. Distilling image dehazing with heterogeneous task imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3462–3471, 2020. 1, 2

[16] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, volume 31, pages 8571–8580, 2018. 6

[17] 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 International Conference on Computer Vision (ICCV)*, pages 1345–1354, 2019. 1, 2

[18] Kenji Kawaguchi. Deep learning without poor local minima. In *Advances In Neural Information Processing Systems*, page 586–594, 2016. 2

[19] Jangho Kim, Seonguk Park, and Nojun Kwak. Paraphrasing complex network: Network compression via factor transfer. In *Advances in Neural Information Processing Systems (NIPS)*, page 2760–2769, 2018. 1, 2

[20] Pengpeng Liu, Irwin King, Michael R. Lyu, and Jia Xu. Flow2stereo: Effective self-supervised learning of optical flow and stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[21] Nicolai Meinshausen and Bin Yu. Lasso-type recovery of sparse representations for high-dimensional data. Technical report, Departement of Statisics, UC Berkeley, 2006. 8

[22] Mary Phuong and Christoph Lampert. Towards understanding knowledge distillation. In *Proceedings of the 36th International Conference on Machine Learning, PMLR*, pages 97:5142–5151, 2019. 2

[23] Arman Rahbar, Ashkan Panahi, Chiranjib Bhattacharyya, Devdatt Dubhashi, and Morteza Haghir Chehreghani. On the unreasonable effectiveness of knowledge distillation: Analysis in the kernel regime. *arXiv preprint arXiv:2003.13438*, 2020. 2

[24] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 1, 3

[25] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014. 2

[26] Eli Schwartz, Alex Bronstein, and Raja Giryes. Isp distillation, 2021. 1

[27] Suraj Srinivas and Francois Fleuret. Knowledge transfer with jacobian matching. In *Proceedings of the 35th International Conference on Machine Learning, PMLR*, pages 80:4723–4731, 2018. 1, 2

[28] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, page 1365–1374, 2019. 1, 2

[29] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge trans-

fer. In *Journal of Machine Learning Research*, page 16(61):2023-2049, 2015. 2

[30] Lin Wang and Kuk-Jin Yoon. Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. *arXiv preprint arXiv:2004.05937*, 2020. 2

[31] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 4933–4942, 2019. 1, 2

[32] Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. In *arXiv preprint arXiv:2002.10957*, 2020. 1

[33] 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 (CVPR)*, page 4133–4141, 2017. 1, 2