# LIT: Learned Intermediate Representation Training for Model Compression

**Animesh Koratana** [*1] **Daniel Kang** [*1] **Peter Bailis** [1] **Matei Zaharia** [1]

## Abstract

Researchers have proposed a range of model compression techniques to reduce the computational and memory footprint of deep neural networks (DNNs). In this work, we introduce **L**earned **I**ntermediate representation **T**raining (LIT), a novel model compression technique that outperforms a range of recent model compression techniques by leveraging the highly repetitive structure of modern DNNs (e.g., ResNet). LIT uses a teacher DNN to train a student DNN of reduced depth by leveraging two key ideas: 1) LIT directly compares intermediate representations of the teacher and student model and 2) LIT uses the intermediate representation from the teacher model's previous block as input to the current student block during training, improving stability of intermediate representations in the student network. We show that LIT can substantially reduce network size without loss in accuracy on a range of DNN architectures and datasets. For example, LIT can compress ResNet on CIFAR10 by 3.4× outperforming network slimming and FitNets. Furthermore, LIT can compress, by depth, ResNeXt 5.5× on CIFAR10 (image classification), VDCNN by 1.7× on Amazon Reviews (sentiment analysis), and StarGAN by 1.8× on CelebA (style transfer, i.e., GANs).

## 1. Introduction

Modern deep networks have improved in accuracy at the cost of higher computational overhead (Ioffe & Szegedy, 2015; He et al., 2016). In response, researchers have proposed many compression techniques to reduce this computational overhead at inference time, which broadly fall into two categories. The first category are student/teacher methods—introduced in knowledge distillation (KD) (Hinton et al., 2014) and further extended (Romero et al., 2015; Kim & Rush, 2016; Furlanello et al., 2018)—in which a smaller student model learns from a large teacher model (e.g., by matching the logits of the teacher and student models). The second category is deep compression (Han et al., 2016b; Zhu et al., 2017; Li et al., 2017; Hubara et al., 2017), in which parts of a model are pruned or quantized to reduce the number of weights and/or the computational footprint. In this work, we refer to this class of methods as "deep compression" and methods to reduce model size more generally as "model compression."

We focus on student/teacher methods since the student model can be designed to leverage existing hardware, e.g., hardware that is optimized for dense matrix multiplication. In contrast, deep compression typically requires new hardware (such as sparse multipliers) for inference improvements (Han et al., 2016a). Additionally, weight pruning methods achieve lower compression ratios on modern DNNs. The majority of compression on older DNNs comes from compressing fully connected layers: ResNet can only be pruned by ~1.5× (Liu et al., 2019) compared to ~10× for older networks (Han et al., 2016b).

In this work, we introduce **L**earned **I**ntermediate representation **T**raining (LIT), a student/teacher compression technique that outperforms a range of model compression techniques on modern networks *without* requiring changes in hardware. LIT targets highly structured, modern networks that consist of repetitive blocks (i.e., groups of layers) that can be scaled up/down for accuracy/speed trade-offs. For example, ResNets have standard configurations from 20 to hundreds of layers (He et al., 2016) across four blocks. LIT leverages two key ideas to directly reduce the depth of student networks as we illustrate in Figure 1.[1] First, LIT directly penalizes deviations of one or more student and teacher intermediate representations (IRs, i.e., the output from a hidden layer). Second, LIT uses the IR from the previous block in the *teacher* model as input to the current student block during training; each student block is effectively trained in isolation to match the corresponding (deeper) block in the teacher. This improves stability of

[*]Equal contribution [1]Stanford University, DAWN Project. Correspondence to: Daniel Kang <ddkang@stanford.edu>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. 

[1]LIT is similar to FitNets (Romero et al., 2015) but targets modern networks. The key difference is using the teacher model's intermediates as input to the student model. We discuss further differences in Section 2.

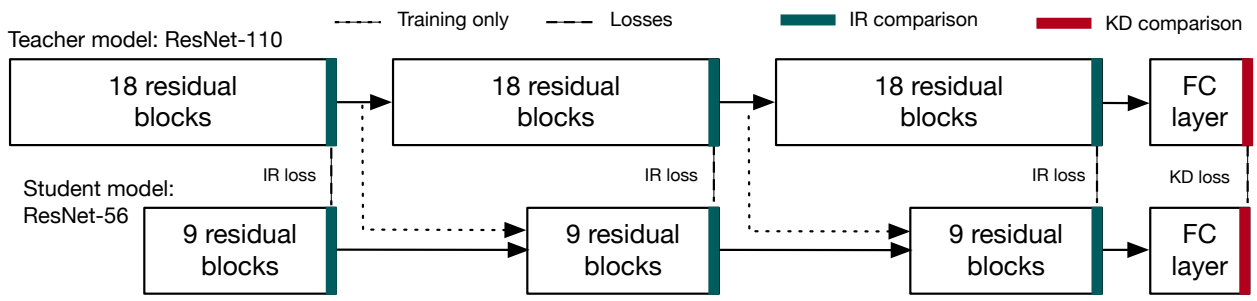



*Figure 1.* A schematic of LIT. The teacher model's intermediate representations are used as input to the student model's blocks during training, except for the first block. Specifically, denoting the blocks $S_1, ..., S_4$ for the student and $T_1, ..., T_4$ for the teacher, $S_2(T_1(x))$ is compared against $T_2(T_1(x))$ in training and similarly for deeper parts of the network. $S_1(x)$ and $T_1(x)$ are directly compared. LIT additionally compares $S(x)$ and $T(x)$ through the KD loss. The teacher model is not updated in training. See Section 3 for full details. Best viewed in color.

student IRs deep in the network. For example, consider compressing a ResNet-56 from a ResNet-110 (Figure 1), each of which have four blocks. The IR loss is applied to the output of each block, and the teacher model's IRs are used as input to the student blocks.

LIT has several key benefits. First, LIT improves inference performance without requiring hardware support for sparse computation or new numeric formats, as LIT directly reduces the depth of networks. Second, LIT is complementary to other forms of model compression, such as weight pruning. Third, LIT can selectively compress parts of networks by only comparing relevant blocks.

We perform an extensive set of experiments:

- We show that LIT can effectively compress a range of model architectures (ResNet, ResNeXt, VDCNN, StarGAN) on a range of datasets and tasks (CIFAR10, CIFAR100, Amazon Reviews, CelebA): empirically, LIT can reduce model sizes from 1.7× to 5.5× with no loss in accuracy. Throughout this paper, we describe reductions in model sizes by depth unless stated otherwise.

- We show that LIT outperforms a range of model compression methods (including recent pruning and distillation methods) on the standard ResNet.

- Recent work on Born Again networks (Furlanello et al., 2018) uses standard KD to train identical student and teacher models to higher accuracies (i.e., no compression). We show that the benefits of this procedure also apply to LIT student/teacher training, and LIT enables up to 0.64% higher accuracy than KD-based Born Again networks on the networks we consider.

- We show that every component of LIT is necessary for high performance.

- We show that LIT performs well across a range of hyperparameters and the optimal hyperparameters appear to be consistent within an architecture type and dataset.

## 2. Related Work

**Knowledge distillation.** Hinton et al. (2014); Bucilu et al. (2006) introduced knowledge distillation (KD) in which a teacher ensemble's or model's outputs are used to train a smaller student model. KD has inspired a variety of related methods, such as for cross-modal distillation or faster training (Gupta et al., 2016; Chen et al., 2016; Frosst & Hinton, 2017; Romero et al., 2015; Furlanello et al., 2018), but we focus on compression in this work. FitNets extends KD by regressing a student model's IR to a teacher model's IR, as the student models Romero et al. (2015) consider are thinner and deeper. Wang et al. (2018) extend FitNets by training networks iteratively using hints. In contrast, LIT uses the teacher IRs as input to the student model in training and directly penalizes deviations of the student model's IRs from the teacher model's IRs, which helps guide training for higher accuracy and improved inference performance. Furthermore, LIT directly reduces the depth of networks. In Born Again networks (Furlanello et al., 2018), the same network architecture is used as both the teacher and student in standard KD, resulting in higher accuracy. We show that LIT outperforms the Born Again procedure on ResNet, ResNeXt, and VDCNN.

**Deep compression.** In deep compression, parts of a network (weights, groups of weights, kernels, or filters (Mao et al., 2017)) are removed for efficient inference (Han et al., 2016b), and the weights of the network are quantized, hashed, or compressed (Hubara et al., 2016; Rastegari et al., 2016; Zhu et al., 2017; Hubara et al., 2017). These methods largely do not take advantage of a teacher model and typically require new hardware, which has features such as sparse arithmetic, for efficiency gains (Han et al., 2016a).

Methods that prune filters (Li et al., 2017) can result in speedups on existing hardware, but largely degrade accuracy. We show that LIT models can be pruned, and thus these methods can be used with LIT.

Additionally, deep compression does not perform as well on modern networks: Han et al. (2016b) compressed VGG by ~10×, but ResNet-110 can only be compressed ~1.6× (Li et al., 2017), compared to LIT's 3.7× compression. Modern networks are more difficult to compress largely due to sparser connections (e.g., more weights in the convolutional layers compared to the fully connected layers).

**Network architectures for fast inference.** Researchers have proposed network architectures, e.g., MobileNet (Howard et al., 2017), and new operations for fast inference, e.g., ShuffleNet (Zhang et al., 2017), on specific hardware. However, these architectures and operations are largely designed for power/resource-constrained mobile devices and sacrifice accuracy for low power. We focus on highly accurate, very deep networks in this work.

## 3. Methods

LIT uses an augmented loss function and training procedure to distill a teacher model into a student model. In its training procedure, LIT both 1) penalizes deviations of the student model's IRs from the teacher model's IRs (IR loss) and 2) uses the KD loss on the output of the entire student network. As LIT directly penalizes deviations in IRs, LIT requires that the teacher model and student model have outputs of the same size at some intermediate layer.

A key challenge in the LIT procedure is that the student network will not have meaningful IRs for a large part of the training, e.g., at the start of training when the weights are initialized randomly. To address this issue, LIT uses the teacher model's IRs as inputs to the student model (described below).

We describe the overall LIT procedure, the KD loss, how IRs are used in LIT, and our hyperparameter optimization method.

**Notation.** Throughout this section, we denote the teacher as $T$, the student as $S$, $x$ as the input, and $y$ as the true label.

Formally, $T, S : \mathbb{R}^{n_0} \to \mathbb{R}^{n_k}$. We assume the teacher network can be decomposed as $T(x) = T_k(T_{k-1}(\cdots T_1(x)))$ (i.e., into sub-networks) and similarly for the student network. We assume the intermediates have the same dimension, formally, $T_i, S_i : \mathbb{R}^{n_{i-1}} \to \mathbb{R}^{n_i}$. We denote $\tilde{T} = T(x)$ and similarly for the student network and intermediates.

**LIT.** In LIT, we combine the KD ($\mathcal{L}_{\text{KD},\alpha}$) and IR loss ($\mathcal{L}_{\text{IR}}$). We show that combining the losses results in smaller models for a fixed accuracy in Section 4. Specifically, for teacher $T$ and student $S$ the full LIT loss is:

$$\beta \cdot \mathcal{L}_{\text{KD},\alpha}(T, S; y) + (1 - \beta) \cdot \mathcal{L}_{\text{IR}}(T_{1,\ldots,k}, S_{1,\ldots,k}) \quad (1)$$

with $\alpha, \beta \in [0, 1]$ ($\alpha$ is described below, $\beta$ is an interpolation parameter). In some cases, we use $\beta = 0$, i.e., we only use the IR loss (e.g., for GANs).

As the IRs have matching dimensions, LIT also allows parts of the teacher model to be copied directly into the student model. For example, for ResNets, we copy the teacher's first convolution (before the skip connections) and fully connected layer to the student model. LIT can also be used to compress specific parts of a model, as we do with StarGAN's generator (Choi et al., 2018).

Finally, after we train the student model with combined loss, we fine-tune the student model with the original loss. Namely, we use KD loss for classification and the generator loss for GANs.

**Knowledge distillation loss.** In KD, a (typically larger) teacher model or ensemble is used to train a (typically smaller) student model. Specifically, the KL-divergence between the probabilities of the student and teacher model is minimized, in addition to the standard cross-entropy loss.

Formally, denote $q_i^\tau = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)}$ for the teacher model, where $z_i$ are the inputs to the softmax and $\tau$ is a hyperparmeter that "softens" the distribution (Hinton et al., 2014). Denote $p_i^\tau$ to be the corresponding quantity for the student model.

Then, the full KD loss is:

$$\mathcal{L}_{\text{KD},\alpha}(p^\tau, q^\tau; y) = \alpha \cdot H(y, p^\tau) + (1-\alpha) \cdot H(p^\tau, q^\tau) \quad (2)$$

where $y$ is the true label, $H$ is the cross-entropy loss, and $\alpha$ is the interpolation parameter.

Hinton et al. (2014) sets $\alpha = 0.5$, but we show that the choice of $\alpha$ can affect performance in Section 4.4.

**Training via intermediate representations.** In LIT, we logically divide the student and teacher networks into $k$ subnetworks such that the input and output dimensions match for the corresponding sub-networks (an example is shown in Figure 1).

Denote the loss on the IR loss $l$ (e.g., L2 loss). The full intermediate loss (given the set of splits) is:

$$\mathcal{L}_{\text{IR}}(T_{1,\ldots,k}, S_{1,\ldots,k}) := l(\tilde{S_1}, \tilde{T_1}) + \textstyle\sum_{i=2}^{k} l(S_i(\tilde{T}_{i-1}), \tilde{T_i}). \quad (3)$$

Concretely, consider a ResNet-110 as the teacher and a ResNet-56 as the student, each with three "stages", i.e.,

| Dataset | Task | Models |
|---|---|---|
| CIFAR10 | Image classification | ResNet, ResNeXt |
| CIFAR100 | Image classification | ResNet, ResNeXt |
| Amazon Reviews | Sentiment analysis (full, polarity) | VDCNN |
| CelebA | Image-to-image translation | StarGAN |

*Table 1.* List of datasets, tasks, and models for standard tasks that we compress with LIT.

| Method | ResNet | Prune % | Test acc. gain (absolute) | LIT's relative improvement (rel. error reduction, size) |
|---|---|---|---|---|
| LIT | ResNet-110 | **73.1%** | **0.24%** | |
| Filter pruning (Li et al., 2017) | ResNet-110 | 32.4% | -0.23% | 6.5%, 2.5× |
| | ResNet-110 | 2.3% | 0.02% | 3.2%, 3.6× |
| Pruned retraining (Liu et al., 2019) | ResNet-164 | 60% | -0.33% | |
| | ResNet-164 | 40% | -0.14% | |
| Network slimming (Liu et al., 2017) | ResNet-164 | 35.2% | 0.15% | |
| Channel pruning (He et al., 2017) | ResNet-50 | 50% | -1.0% | |
| LIT | ResNet-110 | **73.1%** | **0.24%** | |
| KD via ONE (Zhu et al., 2018) | ResNet-110 | 73.1% | -0.43% | 9%, 1× |
| FitNets (Romero et al., 2015) | ResNet-110 | 61.9% | 0.02% | 3.2%, 1.4× |
| KD (Hinton et al., 2014) | ResNet-110 | 61.9% | -0.05% | 4.1%, 1.4× |

*Table 2.* Comparison of LIT and pruning (top) and knowledge distillation (bottom) methods for ResNets on CIFAR10. A higher pruned percentage and a higher accuracy gain is better. LIT outperforms all surveyed alternative methods by up to 2.5×. The pruning numbers were taken directly from the corresponding citations. KD via ONE was taken from the corresponding paper and we ran experiments with KD and FitNets with ResNet-110. While not all pruning methods used the base ResNet-110 architecture, we believe that similar results will hold.

layers in the network with downsampling, and an L2 intermediate loss. Here, the first teacher ResNet "stage" is $T_1$, etc. and the L2 deviation from the feature maps, across all the downsampling feature maps, is the full intermediate loss. A schematic is shown in Figure 1.

This procedure has two key decisions: 1) where to logically split the teacher and student models and 2) the choice of IR loss. We discuss these settings in the hyperparameter optimization below.

**Hyperparameter optimization.** LIT inherits two hyperparameters from KD and introduces one more: $\tau$ (the temperature in KD), $\alpha$ (the interpolation parameter in KD), and $\beta$ (the interpolation parameter in LIT), along with an intermediate representation loss and split points. In this work, we only consider adding the IR loss between natural split points, e.g., when a downsampling occurs in a convolutional network, as we have found it to work well in practice. We have additionally found that L2 loss works well in practice, so we use the L2 loss for all experiments unless otherwise noted (Section 4.4).

We have found that iteratively setting $\tau$, then $\alpha$, then $\beta$ to work well in practice. We have found that the same hyperparameters work well for a given student/teacher structure and dataset (e.g., ResNet teacher and ResNet student). Thus, we use the same set of hyperparameters for a given student/teacher structure and dataset (e.g., we use the same hyperparameters for a teacher/student of ResNet-110/ResNet-20 and ResNet-110/ResNet-32). To set the hyperparameters for a given structure, we first set $\tau$ using a small student model, then $\alpha$ for the fixed $\tau$, then $\beta$ for the fixed $\alpha$ and $\tau$ (all on the validation set).

## 4. Experiments

We evaluated LIT's efficacy at compressing models on a range of tasks and models, including image classification, sentiment analysis, and image-to-image translation (GAN). Throughout, we useid student and teacher networks with the same broad architecture (e.g., ResNet to ResNet). We considered ResNet (He et al., 2016), ResNeXt (Xie et al., 2017), VDCNN (Conneau et al., 2017), and StarGAN (Choi et al., 2018) (Table 1). We used standard architecture depths, widths, and learning rate schedules (described in the Appendix). Code for LIT is provided at `http://github.com/stanford-futuredata/lit-code`.

We show that:

- LIT is effective at compressing modern networks across a range of tasks and outperforms alternate methods of model compression by up to 2.5× (Section 4.1, Table 2).

- LIT is complementary to standard weight pruning (Han et al., 2015) (Section 4.2).
- All components of LIT contribute to high compression (Section 4.3).
- LIT performs well across a range of hyperparameters and the optimal hyperparameters appear to be consistent for a given model architecture (Section 4.4).

### 4.1. LIT Significantly Compresses Models

**LIT is effective at compressing a range of datasets and models.** We ran LIT on a variety of models and datasets for image classification and sentiment analysis (Table 1). We additionally performed KD and hint training on these datasets and models. We selected the hyperparameters sequentially (Section 3).

Figure 2 shows the results for ResNet and ResNeXt for CIFAR10 and CIFAR100, and Figure 3 shows the results for VDCNN on Amazon Reviews (full, polarity). LIT can compress models by up to 5.5× (CIFAR10, ResNeXt 110 to 20) on image classification and up to 1.7× on sentiment analysis (Amazon Reviews, VDCNN 29 to 17) with no loss in accuracy. LIT outperforms KD and hint training on all settings. Additionally, LIT outperforms the recently proposed Born Again procedure in which the same architecture is used as both the student and teacher model (Furlanello et al., 2018) (i.e., only for improved accuracy, not for compression).

We also found that in some cases, KD degrades the accuracy of student models when the teacher model is the same architecture (ResNeXt-110 on CIFAR100, VDCNN-29 on Amazon Reviews polarity). This corroborates prior observations in Mishra & Marr (2018).

**Comparison to alternative methods.** In this work we focus on *modern* networks, which are significantly harder to compress. Older networks, such as VGG, have a disproportionate amount of weights in the fully connected (FC) layers. FC layers are significantly easier to compress: Han et al. (2016b) achieves ~10× compression on FC layers, but only a 1.14× compression rate for convolutional layers.

We show recent compression methods on ResNets in Table 2 (top). As shown, the compression ratios are significantly worse on modern networks (e.g., ResNet) compared to older networks (e.g., VGG). Additionally, LIT outperforms these pruning methods.

We also compare LIT against various forms of knowledge distillation in Table 2 (bottom). LIT also outperforms these methods of compressing models.

**LIT can reduce group cardinality.** While LIT requires the size of at least one IR to be the same width between the teacher and student model, several classes of models have an internal width or group *cardinality*. For example, ResNeXt (Xie et al., 2017) has a "grouped convolution," which is equivalent to several convolutions with the same input (for details, see Figure 3 in Xie et al. (2017)). The width of the network is not affected by the group size, so LIT is oblivious to the group size.

We show that LIT can reduce the group cardinality for ResNeXt. We trained student ResNeXts with cardinality 16 (instead of the default 32) from a ResNeXt-110 (cardinality 32) on CIFAR10. Figure 4 illustrates the results. As before, LIT outperforms KD and hint training in this setting.

| Model | Inception score (higher is better) | FID score (lower is better) |
|---|---|---|
| Teacher (18) | 3.49 | 6.43 |
| LIT student (10) | **3.56** | **5.84** |
| L2 student (10) | 3.46 | 6.47 |
| From scratch (10) | 3.37 | 6.56 |
| Rand init (10) | 2.63 | 94.00 |
| Rand init (18) | 2.45 | 151.43 |

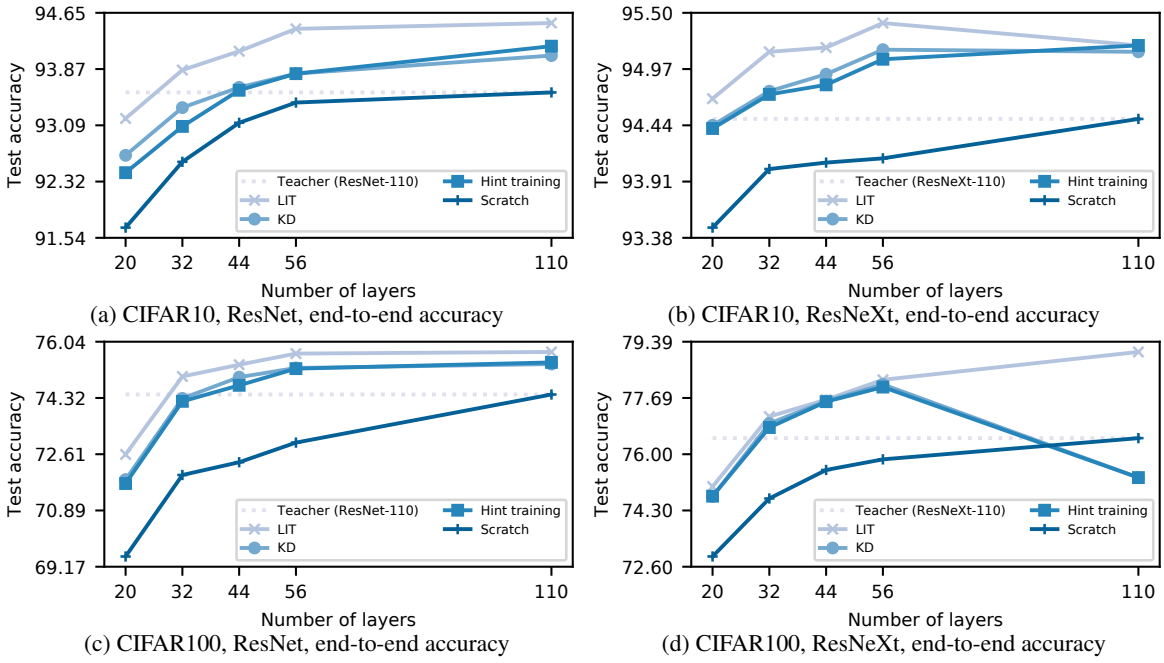
*Table 3.* Inception (Salimans et al., 2016) and FID (Heusel et al., 2017) scores for different versions of StarGAN. The LIT model achieves the best scores despite having fewer layers than the teacher. Numbers in parentheses are the number of layers.

**LIT can compress GANs.** We compressed StarGAN's generator (Choi et al., 2018) using the LIT procedure with $\beta = 0$ (i.e., only using the intermediate representation loss) and with an analogous procedure to KD with the L2 loss as a baseline. The original StarGAN has 18 total convolutional layers (including transposed convolutional layers), with 12 of the layers in the residual blocks (for a total of six residual blocks). We compressed the six residual blocks to two residual blocks (i.e., 12 to four layers) while keeping the rest of the layers fixed. The remaining layers for the teacher model were copied to the student model and fine-tuned. The discriminator remained fixed.

As shown in Table 3, LIT outperforms all baselines in inception (Salimans et al., 2016) and FID (Heusel et al., 2017) scores. Additionally, as shown in Figure 5, the student model appears to perceptually outperform both the teacher model and equivalent model trained from scratch, suggesting LIT can both compress GANs and serve as a form of regularization.

### 4.2. LIT is Complementary to Pruning

Weight pruning is a key technique in deep compression in which parts of a network are set to zero, which reduces the number of weights and, on specialized hardware, reduces the computational footprint of networks (Han et al., 2015). To see whether LIT models are amenable to pruning, we



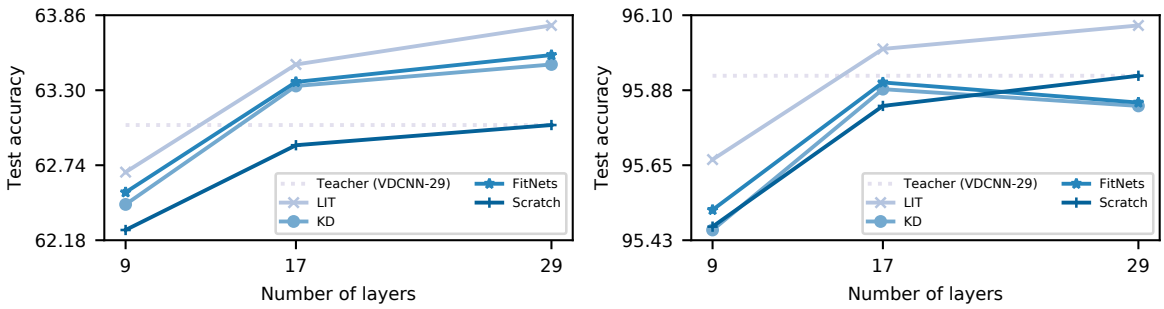

(a) CIFAR10, ResNet, end-to-end accuracy

(b) CIFAR10, ResNeXt, end-to-end accuracy

(c) CIFAR100, ResNet, end-to-end accuracy

(d) CIFAR100, ResNeXt, end-to-end accuracy

*Figure 2.* The accuracy of ResNet and ResNeXt trained from scratch, trained via KD, and trained via LIT for CIFAR10/100. The teacher model was ResNet-110 and ResNeXt-110 respectively. As shown, LIT outperforms KD for every student model. Identical student and teacher architectures correspond to born again networks, which LIT also outperforms. In some cases, KD can reduce the accuracy of the student model, as reported in Mishra & Marr (2018).




(a) Amazon full, VDCNN, end-to-end accuracy

(b) Amazon polarity, VDCNN, end-to-end accuracy

*Figure 3.* The accuracy of VDCNN on Amazon reviews (full and polarity) trained from scratch, trained via KD, and trained via LIT.

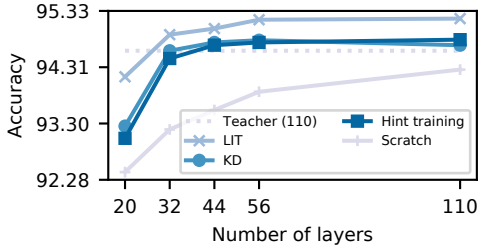



*Figure 4.* ResNeXt student models with cardinality 16 trained from a ResNeXt-110 with cardinality 32 on CIFAR10. We show that LIT can reduce the cardinality and that LIT outperforms KD.

pruned ResNets trained via LIT. We additionally pruned ResNets trained from scratch. All experiments were done on CIFAR10 using a standard pruning procedure (Han et al., 2015).

As shown in Figure 6, LIT models outperform standard pruning for accuracy at a given model size. Additionally, LIT models can be pruned, although less than their trained-from-scratch counterparts. However, LIT models are more accurate and are thus likely learning more meaningful representations. Thus, we expect LIT models to be more difficult to prune, as each weight is more important.

| Type | Accuracy |
|---|---|
| LIT | **93.25%** |
| KD | 92.75% |
| One IR, teacher input | 92.74% |
| One IR, no teacher input (FitNets) | 92.68% |
| LIT (splits between blocks) | 91.15% |
| Multiple IRs, no teacher input | 90.42% |

| Type | Accuracy |
|---|---|
| LIT | **94.72%** |
| KD | 94.42% |
| One IR, teacher input | 94.21% |
| One IR, no teacher input (FitNets) | 94.18% |
| LIT (splits between blocks) | 92.69% |
| Multiple IRs, no teacher input | 91.27% |

*Table 4.* Ablation study of LIT. We performed LIT, KD, and four modifications of LIT. The second block was used for single IR experiments. As shown, LIT outperforms KD and the modifications, while all the modifications underperform standard KD. **Left:** ResNet, **Right:** ResNeXt.

| Model | Loss | Accuracy |
|---|---|---|
| ResNet | L2 | $93.20 \pm 0.04$ |
| ResNet | L1 | $93.19 \pm 0.05$ |
| ResNet | Smoothed L1 | $93.02 \pm 0.06$ |

| Model | Loss | Accuracy |
|---|---|---|
| ResNeXt | L2 | $94.63 \pm 0.07$ |
| ResNeXt | L1 | $94.62 \pm 0.07$ |
| ResNeXt | Smoothed L1 | $93.86 \pm 0.08$ |

*Table 5.* Effect of intermediate representation loss on student model accuracy. L2 and L1 do not significantly differ, but smoothed L1 degrades accuracy. Average of three runs on CIFAR10.

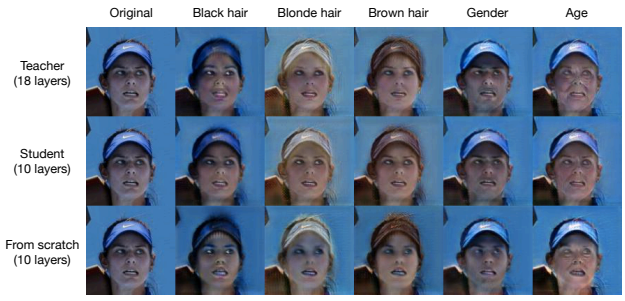



*Figure 5.* Selected images from the teacher (six residual blocks), student (two residual blocks), and trained from scratch (two residual blocks) StarGANs. As shown in columns two and four, LIT can visually to improve GAN performance while significantly compressing models. We show a randomly selected set of images in the Appendix. Best viewed in color.

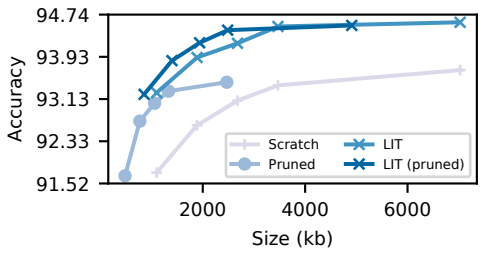



*Figure 6.* The trade-off curves of size vs accuracy of various ResNets pruned on CIFAR10. LIT outperforms standard pruning (Han et al., 2015).

### 4.3. Impact of Training Techniques

We investigated if every component of LIT is necessary for high compression. Recall that LIT uses block-wise training with the teacher IRs as input to the student model. To test our hypothesis that all components were necessary, we performed an ablation study with the following variations of LIT:

- Standard KD.
- Matching a single IR, with no input from the teacher (i.e., standard hint training/FitNets).
- A single IR with teacher input.
- IRs between residual blocks.
- Multiple IRs with no teacher input.

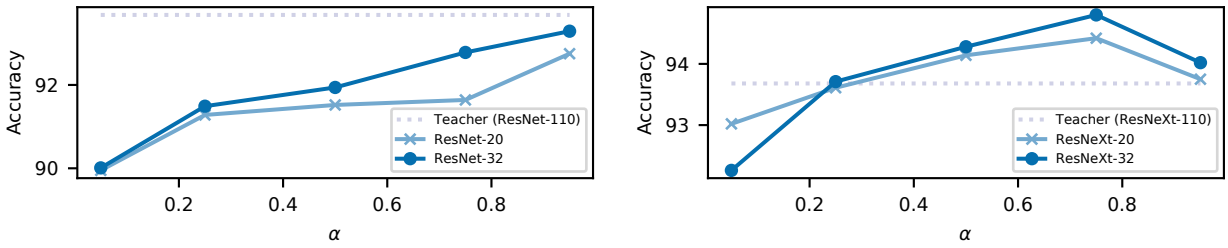
We performed these variations on a teacher model of ResNet-110 and a student model of ResNet-20 on CIFAR10 and similarly for ResNeXt. We used the second block for the single IR experiments.

As shown in Table 4, none of the four variants are as effective as LIT or KD. Thus, we see that LIT's block-wise training is critical for high accuracy compression.

### 4.4. Sensitivity Analysis of Hyperparameters

We perform a hyperparameter analysis on the IR penalty, hyperparameter analysis, and mixed precision. We show that LIT performs well across a range of hyperparameters and that the optimal hyperparameters appear to be consistent for a given architecture type.

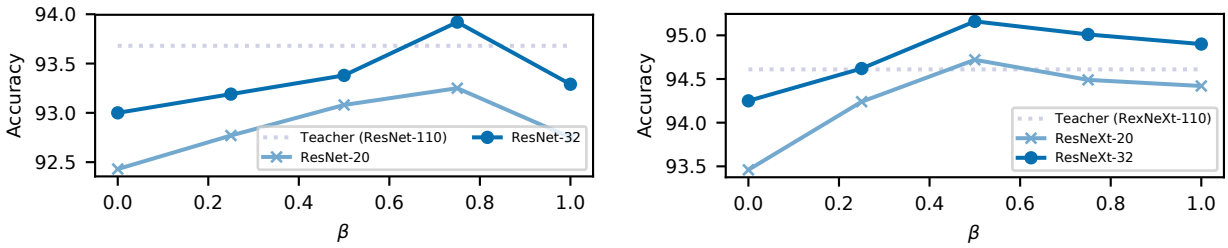
**Intermediate loss penalty.** To see the affect of the intermediate loss penalty, we performed LIT from a teacher model of ResNet-110 to a student of ResNet-20 with the L1, L2, and smoothed L1 loss (all on CIFAR10). The results are shown in Table 5. As shown, L2 and L1 do not significantly differ ($p = 0.78$), but smoothed L1 degrades accuracy ($p = 0.02$).




(a) CIFAR10, ResNet, $\alpha$ sensitivity

(b) CIFAR10, ResNeXt, $\alpha$ sensitivity

*Figure 7.* The accuracy of student models as $\alpha$ (KD's interpolation factor for the cross-entropy and logit loss) varies for ResNet and ResNeXt on CIFAR10. The optimal $\alpha$ varies by model type.




(a) CIFAR10, ResNet, $\beta$ sensitivity

(b) CIFAR10, ResNeXt, $\beta$ sensitivity

*Figure 8.* The accuracy of student models as $\beta$ (LIT's interpolation factor between KD loss and IR loss) varies for ResNet and ResNeXt on CIFAR10. As shown, LIT outperforms training only via KD ($\beta = 1$) and only via intermediate representations ($\beta = 0$). The optimal $\beta$ appears to be lower (i.e., closer to only using the intermediate representation loss) for more accurate models; we hypothesize that more accurate models learn more informative intermediate representations, which helps the students learn better.

***α* and *β*.** Recall that $\alpha$ is the weighting parameter in KD and $\beta$ is the relative weight of KD vs the intermediate representation loss (Section 3).

To see the effect of $\alpha$, a KD hyperparameter, we varied $\alpha$ between 0 and 1 for ResNet and ResNeXt on CIFAR10 and CIFAR100. As shown in Figure 7, $\alpha$ can significantly affect accuracy. Thus, we searched for $\alpha$ as opposed to using a static policy of 0.5 as in Hinton et al. (2014).

We additionally varied $\beta$ between 0 and 1 for ResNet and ResNeXt on CIFAR10. As shown in Figure 8, the optimal $\beta$ varies between architectures but appears to be consistent within the same meta-architecture and dataset.

**LIT works with mixed precision.** To confirm mixed precision training (Micikevicius et al., 2018) works with LIT, we ran LIT on ResNet and ResNeXt (the teacher had 110 layers and the student had 20 layers) on CIFAR10 with both fp32 and mixed precision training. ResNet-20 achieves $93.20 \pm 0.04\%$ and $93.17 \pm 0.07\%$ accuracy with fp32 and mixed precision training respectively. ResNeXt-20 achieves $94.63 \pm 0.07\%$ and $94.57 \pm 0.10\%$ accuracy with fp32 and mixed precision training respectively. As shown, mixed precision does not significantly change the results for ResNet or ResNeXt ($p = 0.5, 1.0$ respectively).

## 5. Conclusion

Researchers have proposed a range of model compression techniques, but many of them are not as effective on modern networks. We introduce LIT, a novel model compression technique that trains a student model from a teacher model's intermediate representations. LIT requires at least one intermediate layer of the student and teacher to match in width, which allows parts of the teacher model to be copied to the student model. By combining several such intermediate layers, LIT students learn a high quality representation of the teacher state without the associated depth. To overcome the lack of useful intermediate representations within the student model at the beginning of training, LIT uses the teacher's intermediate representations as input to the student model during training. We show that LIT can compress models up to 5.5× with no loss in accuracy on standard classification benchmark tasks and image-to-image translation (i.e., GAN generators), outperforming a range of recent pruning and compression techniques.

### Acknowledgments

This research was supported in part by affiliate members and other supporters of the Stanford DAWN project—Ant Financial, Facebook, Google, Infosys, Intel, Microsoft, NEC, SAP, Teradata, and VMware—as well as Toyota Research Institute, Keysight Technologies, Amazon Web Services,

Cisco, and the NSF under CAREER grant CNS-1651570. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

## References

Bucilu, C., Caruana, R., and Niculescu-Mizil, A. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 535–541. ACM, 2006.

Chen, T., Goodfellow, I., and Shlens, J. Net2net: Accelerating learning via knowledge transfer. *ICLR*, 2016.

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S., and Choo, J. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. *CVPR*, 2018.

Conneau, A., Schwenk, H., Barrault, L., and Lecun, Y. Very deep convolutional networks for text classification. *EACL*, 2017.

Frosst, N. and Hinton, G. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.

Furlanello, T., Lipton, Z. C., Tschannen, M., Itti, L., and Anandkumar, A. Born again neural networks. *ICML*, 2018.

Gupta, S., Hoffman, J., and Malik, J. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2827–2836, 2016.

Han, S., Pool, J., Tran, J., and Dally, W. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pp. 1135–1143, 2015.

Han, S., Liu, X., Mao, H., Pu, J., Pedram, A., Horowitz, M. A., and Dally, W. J. Eie: efficient inference engine on compressed deep neural network. In *Computer Architecture (ISCA), 2016 ACM/IEEE 43rd Annual International Symposium on*, pp. 243–254. IEEE, 2016a.

Han, S., Mao, H., and Dally, W. J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *ICLR*, 2016b.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, R. and McAuley, J. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pp. 507–517. International World Wide Web Conferences Steering Committee, 2016.

He, Y., Zhang, X., and Sun, J. Channel pruning for accelerating very deep neural networks. In *International Conference on Computer Vision (ICCV)*, volume 2, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pp. 6626–6637, 2017.

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *NIPS Deep Learning Workshop*, 2014.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Binarized neural networks. In *Advances in neural information processing systems*, pp. 4107–4115, 2016.

Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., and Bengio, Y. Quantized neural networks: Training neural networks with low precision weights and activations. *Journal of Machine Learning Research*, 18:187–1, 2017.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, 2015.

Kim, Y. and Rush, A. M. Sequence-level knowledge distillation. *EMNLP*, 2016.

Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. P. Pruning filters for efficient convnets. *ICLR*, 2017.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. Learning efficient convolutional networks through network slimming. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pp. 2755–2763. IEEE, 2017.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. Rethinking the value of network pruning. *ICLR*, 2019.

Mao, H., Han, S., Pool, J., Li, W., Liu, X., Wang, Y., and Dally, W. J. Exploring the granularity of sparsity in convolutional neural networks. *IEEE CVPRW*, 17, 2017.

Micikevicius, P., Narang, S., Alben, J., Diamos, G., Elsen, E., Garcia, D., Ginsburg, B., Houston, M., Kuchaev, O., Venkatesh, G., et al. Mixed precision training. *ICLR*, 2018.

Mishra, A. and Marr, D. Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. *ICLR*, 2018.

Rastegari, M., Ordonez, V., Redmon, J., and Farhadi, A. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pp. 525–542. Springer, 2016.

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

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pp. 2234–2242, 2016.

Wang, H., Zhao, H., Li, X., and Tan, X. Progressive blockwise knowledge distillation for neural network acceleration. In *IJCAI*, pp. 2769–2775, 2018.

Wasserman, L. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

Xie, S., Girshick, R., Dollár, P., Tu, Z., and He, K. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 5987–5995. IEEE, 2017.

Zhang, X., Zhou, X., Lin, M., and Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. *CoRR*, abs/1707.01083, 2017.

Zhu, C., Han, S., Mao, H., and Dally, W. J. Trained ternary quantization. *ICLR*, 2017.

Zhu, X., Gong, S., et al. Knowledge distillation by on-the-fly native ensemble. In *Advances in Neural Information Processing Systems*, pp. 7528–7538, 2018.