Winning the Lottery Once and For All: Towards Pruning Neural Networks at Initialization

David S. Hippocampus*
Department of Computer Science
Cranberry-Lemon University
Pittsburgh, PA 15213
`hippo@cs.cranberry-lemon.edu`

Abstract

Lottery Ticket Hypothesis (LTH) posits the existence of *winning tickets*, i.e., sparse subnetworks within randomly initialized dense neural networks that are capable of achieving test accuracy comparable to the original, unpruned counterpart when trained from scratch, with an optimal learning rate and in a similar training budget. Despite this promising conjecture, recent studies (Liu et al., 2018; Frankle et al., 2020a; Su et al., 2020; Ma et al., 2021a) have cast doubt on the feasibility of identifying such winning tickets at initialization, particularly in large-scale settings. They suggest that in such expansive environments, winning tickets exclusively and only emerge during the early phase of training. This observation, according to Frankle and Carbin (2018), contradicts the core tenet of LTH as these winning tickets do not truly *win the initialization lottery*. In light of recent findings, we address a critical question: **If winning tickets can only be obtained during early iterations, does the initial training phase of a neural network encode vital knowledge, which we refer to as lottery-ticket information, that can be utilized to generate winning tickets at initialization, especially in large-scale scenarios?**

We affirmatively answer this question by introducing a novel premise, Knowledge Distillation-based Lottery Ticket Search (KD-LTS). Our framework harnesses latent response, feature, and relation-based lottery-ticket information from an ensemble of teacher networks, employing a series of deterministic approximations to address an intractable Mixed Integer Optimization problem. This enables us to consistently *win the initialization lottery* in complex settings, identifying winning tickets right from the initialization point at unprecedented sparsity levels - achieving as low as 5% for VGG-16 and 30% for ResNet-20, and accomplishing this 19 times faster than Iterative Magnitude Pruning (IMP). Remarkably, without bells and whistles, even winning tickets identified early in the training process using our technique - consistently yield a performance gain of 2% for VGG-16 and 1.5% for ResNet-20 across various levels of sparsity, thereby surpassing existing methods. Furthermore, we assert that our work is the first in the literature to obtain state-of-the-art results in winning tickets over various experimental settings while successfully passing all sanity checks referenced in prior work (Ma et al., 2021a; Frankle et al., 2020a; Su et al., 2020), thereby reinstating the possibility for pruning at initialization.

1 Introduction

In the ever-evolving landscape of machine learning, the demand for integrating the intelligence of Deep Neural Networks (DNNs) into diverse applications has surged. Recent breakthroughs in DNNs have facilitated remarkable advancements in challenging domains such as, computer vision (Krizhevsky et al., 2012; Taigman

*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

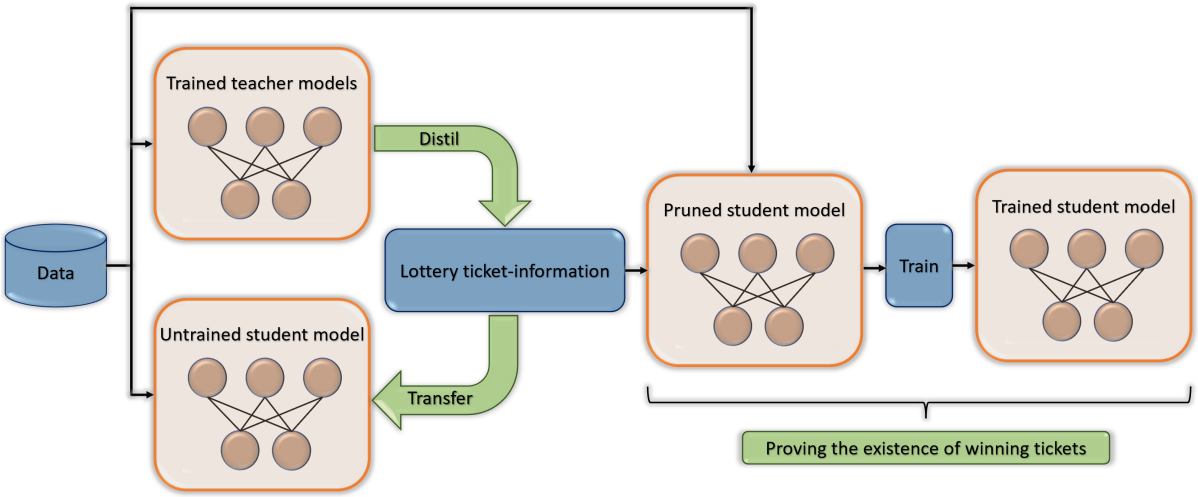

Figure 1: Illustration of our Knowledge Distillation inspired Lottery Ticket Search (KD-LTS).

et al., 2014; Xiao et al., 2018) - enabling feats such as real-time facial recognition and autonomous vehicle navigation; reinforcement learning (Silver et al., 2016; Mnih et al., 2015) - mastering complex games and outperforming human experts; speech recognition (Graves et al., 2013; Hannun et al., 2014) - bringing us virtual assistants; and in machine translation (Sutskever et al., 2014) - breaking language barriers, among others. However, the pursuit of high predictive performance has often come at a steep price to pay: excessive overparameterization (Simonyan and Zisserman, 2014). This tendency towards deploying increasingly larger and more complex networks, while sometimes necessary for achieving top-tier results, has raised legitimate concerns. Consider the immense computational and storage costs - often needing billions of parameters; the environmental impact - training models like BERT can have a carbon footprint comparable to a trans America flight for one person; and the financial cost - training advanced models such as XLNet is estimated to be around US $61,000 (Erion et al., 2022; Verdecchia et al., 2023; Heikkilä, 2022; Peng and Sarazen, 2019). This issue has impeded practical deployment of DNNs in resource-constrained real-world environments, such as on-device inference in mobile phones, and similar edge devices (Zhu and Gupta, 2017).

Post vs pre-training pruning. To tackle this challenge, network pruning has emerged as a promising technique (Han et al., 2015; Hu et al., 2016; Mariet and Sra, 2015; Dong et al., 2017; Yang et al., 2017; Louizos et al., 2017), allowing DNNs to be significantly pruned during inference without sacrificing test accuracy. For example, pruning has led to a 5x reduction in the number of parameters in Convolutional Neural Networks Li et al. (2016), and it has also achieved a 20.79% reduction in inference time for ResNet-50, which is now down to 17.22 milliseconds (Gong et al., 2022), all while maintaining a minimal drop in accuracy, often less than 1%. Expanding the application of pruning beyond mere inference efficiency improvements, researchers are additionally exploring early pruning in neural networks to reduce training costs (Strubell et al., 2019). However, this strategy can result in less trainable architectures with decreased performance compared to the original network, suggesting that a certain degree of overparameterization might still be necessary for successful training (Livni et al., 2014).

Pruning at initialization: missing the mark? Towards early-stage pruning. Contrary to conventional notion that sparse networks can only be obtained through post-training pruning, LTH (Frankle and Carbin, 2018) postulated a paradigm-shift by unveiling high-performing, sparse subnetworks - called winning tickets - right at initialization. Employing the Iterative Magnitude Pruning (IMP) (Han et al., 2015) technique with resetting mechanism, Frankle and Carbin (2018) successfully uncovered these winning tickets in Conv-2, Conv-4, and Conv-6 architectures trained on MNIST and CIFAR-10, achieving high levels of sparsity (80-90% or more). However, these promising results were predominantly limited to simpler settings. The practical success of LTH has been profound, demonstrating wide applicability across various domains (Evci et al., 2022; Yu et al., 2019; Kalibhat et al., 2021) and inspiring a multitude of follow-up studies to identify winning tickets at initialization, (Lee et al., 2018a; de Jorge et al., 2020; Wang et al., 2020; Sreenivasan et al., 2022;

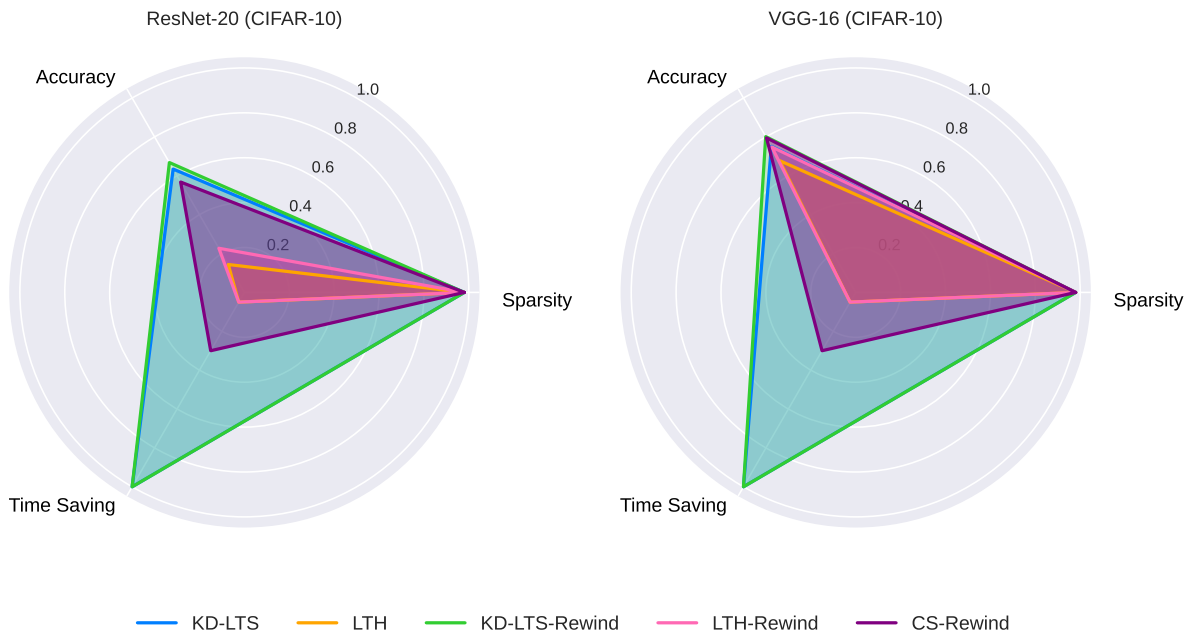

Figure 2: Comparative evaluation of KD-LTS' winning ticket performance in VGG-16 and ResNet-20 architectures on the CIFAR-10 dataset, extracted at initialization and early epoch contrasted with LTH, LTH-Rewind, and CS. Note: Scale of the accuracy axis has been adjusted to $[0.7, 1]$.

Alizadeh et al., 2022; Tanaka et al., 2020). Despite practical success, recent studies (Gale et al., 2019; Frankle et al., 2020c; Savarese et al., 2019; Frankle et al., 2020a) highlight the limitations of these pruning techniques, including IMP, in more demanding large-scale settings. Specifically, for complex architectures like VGGs and ResNets trained on CIFAR-10/CIFAR-100 and ImageNet, these methods struggle to identify winning tickets with significant or meaningful sparsity levels. As illustrated in Figure 3, these methods only manage to extract winning tickets at trivial sparsities - $\sim 20\%$ or even less - in larger networks such as VGG-16, ResNet-18, and ResNet-20, marking a stark contrast to the performance seen in smaller-scale scenarios.

Building on the critiques outlined by recent research, investigations (Su et al., 2020; Frankle et al., 2020b; Sreenivasan et al., 2022; Ma et al., 2021a; Frankle et al., 2020a) into existing pruning-at-initialization techniques have further underscored their limitations through baseline tests that incorporate both sensitivity and stability analyses. These studies demonstrate that randomly sampled sparse subnetworks, when adjusted for carefully chosen layer-wise sparsity, frequently outperform those identified by prevailing pruning-at-initialization methods. Furthermore, these subnetworks exhibit significant vulnerability to perturbations introduced by Stochastic Gradient Descent (SGD) noise, shedding light on their comparative underperformance. For an in-depth examination and insights into sensitivity and stability analyses, readers are encouraged to refer to Sections 6 and 6, respectively.

In addressing the challenges highlighted by stability and sensitivity analyses, Frankle et al. (2020a) explored modifications to the 'IMP with resetting technique' to better suit large-scale settings. Their modified approach successfully identified winning tickets while passing both stability and sensitivity tests. However, this advancement compromised the core premise of the LTH: the identification of winning tickets strictly at the network's initialization. Their revised methodology requires pre-training the dense network for a few initial epochs, followed by rewinding the weights to these early epochs rather than resetting them to the original initialization point. This departure from pruning at initialization necessitates substantial hardware and software resources for pre-training the dense network for at least a few initial epochs.

Given the paramount importance of identifying winning tickets at initialization to significantly simplify training costs for practical real-world deployments, the current state of affairs regarding pruning at initialization in large-scale settings brings us to a critical juncture: **Do winning tickets with non-trivial sparsity exist**

at initialization in complex settings? Moreover, can these tickets not only demonstrate tangible accuracy advantages compared to the original, dense networks but also meet the standards of stability and sensitivity analyses to ascertain their credibility? To date, to the best of our knowledge, these two questions remain open problems.

Knowledge distillation-based Lottery Ticket Search: Towards pruning at initialization. In this study, we delve into these open questions by establishing a novel link between the valuable information encoded within neural networks during their various training phases and the identification of winning tickets at initialization - a connection that has not been explored before.

Key Insight

Knowledge Distillation-aided Lottery Ticket Search (KD-LTS) efficiently distills a rich tapestry of critical response, feature, and relation-based heterogeneous information - referred to as lottery-ticket information - from an ensemble of teacher networks during their initial training stages, in addition to leveraging the information contained in data labels, to identify subnetworks within a dense randomly initialized student network. This pioneering approach facilitates the task of identifying winning tickets - a feat that existing pruning-at-initialization techniques, which rely on traditional model training solely with labels, could not accomplish.

In brief, the KD-LTS framework is structured around three key components. **Phase-1** involves precise identification of diverse lottery-ticket information acorss an array of teacher networks in their early training stages. In **Phase-2**, KD-LTS introduces a series of deterministic relaxations to a Mixed Integer Optimization problem, a crucial step to effectively train binary masks within a randomly initialized dense student network. This step facilitates the efficient transfer of identified lottery-ticket information from the teacher models. The final component, **Phase-3**, incorporates a unique custom-designed scoring function, wrapped around a Straight-Through Estimator (STE), a freezing mechanism, and a regularization term. This combination effectively orchestrates control of sparsity, playing a pivotal role in striking an optimal balance between sparsity and performance. Ultimately, these three components synergize to equip subnetworks at initialization with the essential lottery-ticket information, thus qualifying them as winning tickets. Detailed information about the KD-LTS methodology can be found in Section 4, and an illustrative representation of this framework is provided in Figure 14. The complete code is publicly available at `https://github.com/gap48/Winning-the-lottery`.

Contributions. We summarize our contributions as follows:

(i) **Revitalizing pruning at initialization:** We introduce KD-LTS, a novel methodology that harnesses knowledge distillation to resurrect the potential for pruning at initialization. As highlighted in Figure 2, KD-LTS outperform state-of-the-art techniques, demonstrating Pareto-optimal performance in terms of sparsity, accuracy, and computational efficiency. Through extensive experiments and comprehensive evaluation of training settings and quality analyses, we formalize a strict definition of LTH in Section 6.3 to standardize and catalyze research on pruning at initialization, and rigorously validate KD-LTS winning tickets' compliance with this definition, thus ensuring their reliability and effectiveness.

(ii) **Empirical validation above and beyond overall performance:** KD-LTS framework facilitates the identification of subnetworks that exhibit significant improvements over existing benchmarks, including significant gains in test accuracy ($\uparrow$ 2%), sparsity ($\uparrow$ 40%) of winning tickets at initialization, test accuracy ($\uparrow$ 1%), sparsity ($\uparrow$ 5%) of winning tickets at early epoch, and supermasks ($\uparrow$ 3%). Additionally, we observed nontrivial enhancements in mCE, ROC-AUC, adversarial robustness, ECE, NLL, Stability, Fidelity, and Transferability ($\uparrow$ 1-5%), thereby meeting the ever-evolving and contrasting demands in applications of DNNs. Please refer to Sections 5, C, and D.1 for more details. Furthermore, we uncover a surprising and non-intuitive interplay between specialized characteristics (e.g., adversarial robustness) of teacher models and the properties of extracted winning tickets, as presented in Section A. This suggests the potential for strategically embedding such traits beyond mere improvement in performance regarding that trait.

(iii) **Practical utility of KD-LTS:** Advancing beyond the utility of KD-LTS as merely a research exploration into the dynamics of pruning at initialization, this study broadens the scope, showcasing its effectiveness

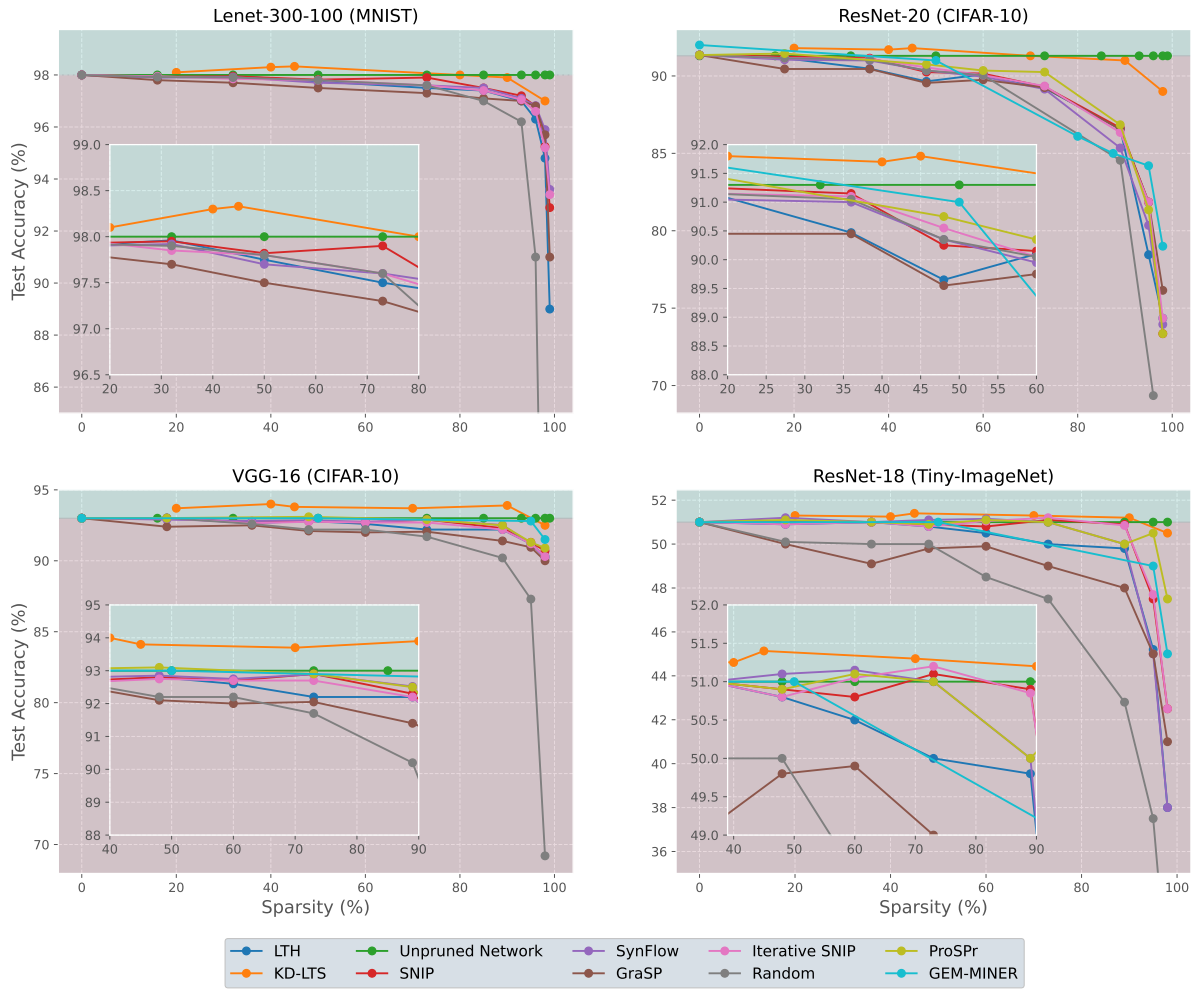

Figure 3: Comparative assessment of KD-LTS' winning ticket performance across diverse networks and datasets, juxtaposed against leading methods for pruning at initialization, including LTH, SNIP, Iterative SNIP, GraSP, SynFlow, ProsPr, and GEM-MINER.

as a practical tool for the extraction of winning tickets. Comprehensive ablation studies on the key components of KD-LTS, as illustrated in Section B, illuminate the technique's capacity to identify winning tickets with significant sparsity by utilizing just a single teacher model trained for merely three epochs, using only 70% of the training time and 70% of the full dataset, and leveraging a bare minimum response-based knowledge category.

(iv) **Transforming knowledge distillation with KD-LTS:** KD-LTS not only challenges conventional approaches to pre-training pruning but also knowledge distillation, offering innovative new perspectives within both fields. Diverging from traditional KD methods that transfer knowledge from large teacher models to smaller students through weight training, KD-LTS leverages large student networks and employs a novel mask-training technique for knowledge transfer. As highlighted in Section D.1, student networks pruned using KD-LTS demonstrate significant performance improvements and a substantial reduction in parameter counts compared to small, dense student networks trained through standard KD methods.

2 Related Works

Pruning after initialization. Studies (Zhang et al., 2021a; Du et al., 2018; Neyshabur et al., 2018; Allen-Zhu et al., 2019) have underscored the requisite of overparameterization in the performance of DNNs.

To tackle the challenges associated with overparameterization, a range of model compression techniques have been developed. These approaches include weight factorization (Denton et al., 2014), weight sharing (Wu et al., 2018), quantization (Hubara et al., 2017), network architecture search (Zoph and Le, 2016), dynamic inference (Wang et al., 2018), and notably, pruning (Han et al., 2015) and knowledge distillation (Hinton et al., 2015). Conventional pruning methods typically employ saliency criteria such as weight magnitude (Han et al., 2015), gradients (Molchanov et al., 2016; 2019), Taylor expansion (Molchanov et al., 2016), movement analysis (Sanh et al., 2020), and second-order derivatives (LeCun et al., 1990) for the selective removal of weights in neural networks. This study represents a notable shift away from traditional methods, transcending beyond the limitations of ad-hoc criteria and fixed pruning intervals typically associated with conventional approaches. We implement a series of deterministic optimization problems within a framework that enables the continuous recalibration of pruned weights. This methodology not only facilitates the potential recovery of pruned weights but also offers a more dynamic and adaptive approach to network sparsification.

Knowledge distillation. Alongside pruning, knowledge distillation has emerged as another potent strategy to counter overparameterization, offering significant advancements in model compression since its initial conceptualization by Ba and Caruana (2014) and Hinton et al. (2015). Knowledge distillation has been extensively studied, with some notable strategies including the utilization of ensemble diversity (Malinin et al., 2019), Jacobian information (Srinivas and Fleuret, 2018), relational knowledge (Park et al., 2019), attention maps (Srinivas and Fleuret, 2018), mutual information (Passalis et al., 2020), sequential transfer (Furlanello et al., 2018), and online/self-distillation (Anil et al., 2018; Lee et al., 2018b). Its applications span diverse domains such as computer vision (Kim and Rush, 2016; Papernot et al., 2016; Yu et al., 2017), sequence modeling (Kim and Rush, 2016), semi-supervised learning (Tarvainen and Valpola, 2017), hyperparameter search (Liu et al., 2023), training efficiency improvement(Blakeney et al., 2022), and multi-modal learning (Gupta et al., 2016), among others. In this paper, we extend the applications of knowledge distillation to an unexplored domain - pruning at initialization. The selection of knowledge types plays a crucial role in shaping the learning process of the student model. Reflecting on the extensive knowledge categories in the literature as previously discussed, the exploration has been particularly focused on stitching together the most effective knowledge sources, which is referred to as lottery-ticket information, pertinent to our application - securing high quality winning tickets right from the initialization stage.

Overall assessment of winning tickets. LTH impacted research on pruning at initialization, leading to the development of techniques such as SNIP (Lee et al., 2018a), GraSP (Wang et al., 2020), SynFlow (Tanaka et al., 2020), Iterative SNIP (Verdenius et al., 2020; de Jorge et al., 2020), ProsPr (Alizadeh et al., 2022), and GEM-MINER (Sreenivasan et al., 2022). This milieu of studies on extracting subnetworks has fostered the practical applicability of LTH in image classification (Evci et al., 2022; You et al., 2019; Savarese et al., 2019), object detection (Girish et al., 2021), natural language processing (Yu et al., 2019; Prasanna et al., 2020; Chen et al., 2021a), generative adversarial networks (Kalibhat et al., 2021), graph neural networks (Chen et al., 2021a), reinforcement Learning (Yu et al., 2019), lifelong learning (Chen et al., 2020), and the introduction of supermasks (Zhou et al., 2019). In addition, this has stimulated the assessment of subnetworks in various contexts, such as transferability across datasets, optimizers, and architectures (Chen et al., 2021b; Morcos et al., 2019), enhancing fairness (Hooker et al., 2019; Paganini, 2020), improving robustness (Gui et al., 2019; Ye et al., 2019; Chen et al., 2022), addressing privacy concerns (Wang et al., 2019), and providing tools for uncertainty estimation (Venkatesh et al., 2020). Nonetheless, the majority of evaluations have predominantly focused on tickets derived from the early training phase. Our work extends the horizon of LTH by identifying supermasks that surpass existing methods (Sreenivasan et al., 2022; Ramanujan et al., 2020; Savarese et al., 2019; Zhou et al., 2019), and showcasing that KD-LTS' winning tickets, identified at initialization, can meet or even exceed the benchmarks across a wide range of assessments typically associated with post-initialization pruning or dense network training.

3 Notation

We consider a C-class classification problem, where $\mathcal{X} \subseteq \mathbb{R}^d$ represents the feature space and $\mathbb{D} = \{(x_i, y_i)\}_{i=1}^{I}$ is the dataset. Every sample x_i is associated with a class c drawn from set $\{1, 2, \ldots, C\}$ and y_i is a one-hot vector which has a value of 1 in the c-th component to represent ground-truth label c, and a value of 0 in all other components. Our objective is to identify winning tickets in a neural network that models the mapping

Table 1: Overview of key attributes in initialization and early phase pruning methods.

Pruning method	Stability	Sensitivity	Prunes at init.	Wins the Jackpot	Cost
IMP	✗	✗	✓	✗	2850 epochs
SNIP	✗	✗	✓	✗	1 epoch
GraSP	✗	✗	✓	✗	1 epoch
LTH-Rewind	✓	✓	✗	✗	3000 epochs
Synflow	✗	✗	✓	✗	3 epochs
ProsPr	✗	✗	✓	✗	200 epochs
Gem-Miner	✗	✓	✓	✗	220 epochs
CS	✓	✓	✗	✗	360 epochs
KD-LTS	✓	✓	✓	✓	150 epochs

of input features to corresponding labels, denoted as $f_L^{\mathrm{s}}(x; m \odot \theta) : \mathcal{X} \to \mathbb{R}^C$. The parameters of this network are represented as $\theta \in \mathbb{R}^P$. Here, 's' represents the student model, L represents the penultimate layer in the set of layers $\{1, 2, \ldots, L\}$, m is a trainable binary mask, i.e., $m \in \{0, 1\}^P$ and $\odot$ refers to element-wise multiplication. We denote a vector of all-ones as $\mathbf{1}$. In the context of a dense network, where $m = \mathbf{1}$, the network is represented as $f_L^{\mathrm{s}}(x; \theta)$. We refer to a teacher model as $f_L^{\mathrm{t}_k}$, where $k \in \{1, 2, \ldots, K\}$ represents the index of the teacher model in the ensemble. To identify a specific element i in the feature map associated with layer l of the neural network, we use the notation f_{il}. The training process spans a total of E epochs. For each epoch $e \in \{1, 2, \ldots, E\}$ the associated parameters are denoted as $\theta^{(e)}$, and the corresponding binary masks are represented as $m^{(e)}$. Sparsity S of the pruned network is calculated as $S = 1 - \frac{\|m\|_0}{P}$.

4 Knowledge Distillation-based Lottery Ticket Search (KD-LTS)

Definition 4.1 (Subnetwork)**.** For a dense network $f_L(x; \theta) : \mathcal{X} \to \mathbb{R}^C$, a subnetwork is denoted as $f_L(x; m \odot \theta)$ and is defined by a binary mask $m \in \{0, 1\}^P$. Here, a parameter component θ_u is retained if $m_u = 1$ and removed otherwise. For any given configuration of m, the effective parameter space of the resultant subnetwork is characterized by $\|m\|_0$ parameters.

KD-LTS leverages the rich history of knowledge distillation, including: • **Response-based** soft target Knowledge Distillation (KD) (Hinton et al., 2015) • **Feature-based** Mean Squared Error (MSE) loss (Ba and Caruana, 2014), and first-order Jacobian information (Czarnecki et al., 2017) • **Relational-based** attention maps (Zagoruyko and Komodakis, 2016), Contrastive Representation Distillation (CRD) framework (Tian et al., 2019), critical period-aware mutual information between features (PKT) Passalis et al. (2020), (RKD) Park et al. (2019), to leverage sample and feature diversity, the solution of the flow procedure (FSP) (Yim et al., 2017) to exploit layer diversity, and diversity within ensembles through Adaptive Ensemble Knowledge Distillation (AE-KD) (Malinin et al., 2019). The objective is to meticulously identify key latent information imprinted within a trained teacher model during the training process, carefully selecting them according to specific applications. Given the inherrent drawback, which is the requisite of training the network in the weight resetting technique, we then distill these attributes, in addition to learning from one-hot hard labels as output targets, into a densely connected, random student network. This approach enables us to craft a subnetwork within the student network that mirrors the essential characteristics of the trained teacher model - the very attributes that make this trained phase of the teacher model contain winning tickets, a phase that weight resetting specifically capitalizes on.

4.1 First Stop: Distilling Knowledge

The subsequent section details our KD-LTS framework and its multifaceted approach, which combines ground-truth labels, response-based, feature-based, and relational-based learning components to extract winning tickets.

Learning from ground-truth label. The student network $f_L(x, (m \odot \theta))$ outputs a probabilistic class posterior $\mathcal{P}(z|x; (m \odot \theta))$ for a given sample x over class c, which is formulated as follows:

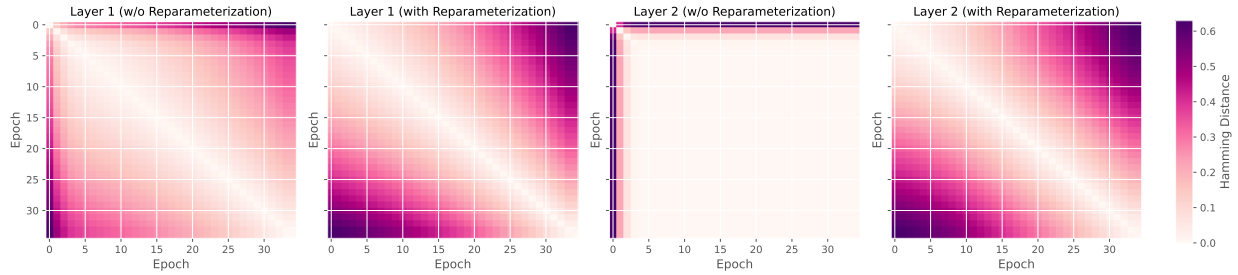

Figure 4: Hamming distance of masks across epochs within a KD-LTS framework for a LeNet-300-100/MNIST experiment, with and without batch normalized mask reparameterization.

$$\mathcal{P}(z|x;(m\odot\theta)) = \mathrm{Cat}\left(\frac{\mathrm{e}^{f_L(x;\theta)}}{\sum_{c=1}^{C}\mathrm{e}^{f_{cL}(x;\theta)}}\right).$$

In this expression, z represents the logits or unnormalized log probabilities generated by the network, $\mathrm{e}^{(\cdot)}$ denotes element-wise exponentiation and 'Cat' signifies Categorical distribution. The conventional objective function for training a multi-class classification model is the cross-entropy measurement between the predicted and ground-truth label distributions, defined as:

$$L_{\mathrm{CE}}(x;(m\odot\theta)) = \frac{1}{I}\sum_{i=1}^{I} y_i^{\mathsf{T}}\log\mathcal{P}(z_i|x_i;(m\odot\theta)).$$

Here, T denotes the transpose operator. L_{CE} guides the network towards predicting the correct class label through maximum likelihood estimation.

Response-based distillation. Our response-based distillation aligns the logits of the student network with those of the teacher network, softened by a high temperature T. This softening, achieved by higher values of T, results in smoother distributions. The distillation loss function, accounting for the network response, is expressed as:

$$L_{\mathrm{KD}}(x;(m\odot\theta)) = \frac{1}{IK}\sum_{i=1}^{I}\sum_{k=1}^{K}\nu_k\left(\frac{\mathrm{e}^{\frac{f_L^{\mathrm{t}_k}(x_i;\theta)}{T}}}{\sum_{c=1}^{C}\mathrm{e}^{\frac{f_{cL}^{\mathrm{t}_k}(x_i;\theta)}{T}}}\right)^{\mathsf{T}}\log\mathcal{P}(z_i|x_i;(m\odot\theta)).$$

In this formulation, ν_k is the weight assigned to the kth teacher, and $\log(\cdot)$ is applied element-wise. L_{KD} minimizes the Kullback-Leibler (KL) divergence loss to effectively transfer nuanced, class-discriminative information from the teacher to the student.

Feature-based distillation. Inspired by Czarnecki et al. (2017), our feature-based distillation loss for aligning internal representations goes beyond conventional considerations of target information by incorporating Jacobian information alongside target information. In the pursuit of extracting winning tickets, the feature-based loss, denoted as L_{FD} utilizing Sobolev training, is formulated as follows:

$$L_{\mathrm{FD}} = \frac{1}{IK}\sum_{i=1}^{I}\sum_{k=1}^{K}\nu_k\left\|\frac{\nabla_\theta f_L^{\mathrm{t}_k}(x_i;\theta)}{\left\|\nabla_\theta f_L^{\mathrm{t}_k}(x_i;\theta)\right\|_2} - \frac{\nabla_{\theta^{(0)}} f_L^{\mathrm{s}}(x_i;(m\odot\theta^{(0)}))}{\left\|\nabla_{\theta^{(0)}} f_L^{\mathrm{s}}(x_i;(m\odot\theta^{(0)}))\right\|_2}\right\|_2^2.$$

Here, $\nabla_\theta f_L^{\mathrm{t}}(x;\theta)$ and $\nabla_{\theta^{(0)}} f_L^{\mathrm{s}}(x;(m\odot\theta^{(0)}))$ represent the first-order Jacobian matrices of the final layer outputs of the teacher and student networks with respect to their parameters, respectively. By employing the

Algorithm 1: KD-LTS

Initialization: Initialize $s^{(0)}$, penalty λ, iterations T, α, γ, parameter $\theta^{(0)}$, multiplication factor μ, $\hat{c}$ and $\widetilde{c}$

for epoch e in E **do**

  for iteration n in N **do**

    Solve for L_{M}:

$$\min_{s^{(e)}} L_{\mathrm{M}}\left(x; r\left(\sigma\left(\beta^{(e)}\widetilde{s}\right)\right) \odot \theta^{(0)} \odot q\right) + \lambda \left\|\sigma\left(\beta^{(e)}\widetilde{s}\right)\right\|_1 .$$

    /*** STE ***/

    $p_{\text{sorted}} \leftarrow \text{Sort}\left(\sigma\left(\beta^{(e)}\widetilde{s}\right)\right)$.

    $p_{\text{bottom}} \leftarrow$ Bottom-$(1 - e^{-\hat{c}e})$ fraction of p_{sorted}.

    $r \leftarrow r \odot 1_{p_i \notin p_{\text{bottom}}}$.

    /*** STE ***/

  end

  $\beta^{(e)} \leftarrow \mu\beta^{(e)}$.

  if $\text{mod}(e, V) = 0$ **then**

    $I_2 \leftarrow \{i : q_i = 1\}$.

    $w_{\text{sorted}} \leftarrow \text{Sort}\left(\sigma\left(\left(\beta^{(e)}\widetilde{s}\right) \odot \theta^{(0)}\right) \in I_2\right)$.

    $w_{\text{bottom}} \leftarrow$ Bottom-$(1 - e^{-\widetilde{c}e})$ fraction of w_{sorted}.

    $q \leftarrow r \odot 1_{q_i \notin w_{\text{bottom}}}$.

  end

end

Fix the masks and output $f_L^{\mathrm{s}}(x; (H(\beta^{(E)}s^{(E)}) \odot \theta^{(0)}))$.

Solve $\min_{\theta} L_{\mathrm{CE}}\left(x; (H(\beta^{(E)}s^{(E)}) \odot \theta)\right)$ to prove winning tickets.

Sobolev loss function, we encourage the student network's parameters to replicate the gradient flow of the trained teacher network. As a result, the parameters of the subnetwork identified within the student model emulate a learning pattern similar to that of the teacher network, thereby fostering a stable learning platform post-extraction.

Relational-based distillation. Drawing upon and significantly adding to the work by Yim et al. (2017) on transferring layer diversity, our relational-based loss, denoted as L_{RD}, captures and aligns the diversity in feature representations between teacher and student networks to craft winning tickets. The relational-based loss is formulated as follows:

$$L_{\mathrm{RD}}(x; (m \odot \theta)) = \frac{1}{IK} \sum_{i=1}^{I} \sum_{k=1}^{K} \sum_{l=1}^{L} \nu_k D\left(\mathcal{P}_{(i,j)}^{(\mathrm{t},l)}, \mathcal{P}_{(i,j)}^{(\mathrm{s},l)}\right).$$

where the probability distributions $\mathcal{P}_{(i,j)}^{(\mathrm{t},l)}$ and $\mathcal{P}_{(i,j)}^{(\mathrm{s},l)}$ are defined as follows:

$$\mathcal{P}_{(i,j)}^{(\mathrm{t},l)} = \frac{\mathrm{Ker}(f_{i(l-1)}^{\mathrm{t}}, f_{jl}^{\mathrm{t}})}{\sum_{i',j',i' \neq j'} \mathrm{Ker}(f_{i'(l-1)}^{\mathrm{t}}, f_{j'l}^{\mathrm{t}})}, \quad \mathcal{P}_{(i,j)}^{(\mathrm{s},l)} = \frac{\mathrm{Ker}(f_{i(l-1)}^{\mathrm{s}}, f_{jl}^{\mathrm{s}})}{\sum_{i',j',i' \neq j'} \mathrm{Ker}(f_{i'(l-1)}^{\mathrm{s}}, f_{j'l}^{\mathrm{s}})}.$$

These equations define the pairwise similarity probabilities for features between layers $l-1$ and l within the teacher and student networks, normalized by the sum of all pairwise kernel similarities at the same layer, excluding self-similarity (where $i' \neq j'$), thereby capturing the flow of information within each network. The choice of kernel function $\mathrm{Ker}(\cdot)$, which evaluates the similarity between feature vectors, is crucial in this computation. We systematically experiment and adopt an angle-wise metric: $\mathrm{Ker}(a, b) = \frac{1}{2}\left(\frac{a^{\intercal}b}{\|a\|_2\|b\|_2} + 1\right)$ aimed at measuring cosine similarity to penalize structural differences in feature relations and enhance the

quality of transfer. The divergence $D(\cdot,\cdot)$ between the teacher's and student's relational distributions is evaluated using the Jeffreys divergence, a symmetric version of the KL divergence, to ensure a balanced knowledge transfer process:

$$L_{\mathrm{RD}}(x;(m\odot\theta)) = \frac{1}{KI(I-1)}\sum_{i,j,i\neq j}\sum_{k=1}^{K}\sum_{l=1}^{L}\nu_k\left(\mathcal{P}_{(i,j)}^{(\mathrm{t}_k,l)}-\mathcal{P}_{(i,j)}^{(\mathrm{s},l)}\right)\left(\log\mathcal{P}_{(i,j)}^{(\mathrm{t}_k,l)}-\log\mathcal{P}_{(i,j)}^{(\mathrm{s},l)}\right)$$

Through this approach, we facilitate the transfer of the teacher's internal feature relations and efficient learning patterns onto the student network. The final optimization objective of our knowledge distillation framework to holistically capture the complexities of the teacher-student knowledge transfer process:

$$L_{\mathrm{M}}(x;(m\odot\theta)) = \alpha L_{\mathrm{CE}}(x;(m\odot\theta)) + T^2(1-\alpha)L_{\mathrm{KD}}(x;(m\odot\theta)) + \gamma L_{\mathrm{FD}}(x;(m\odot\theta)) + \gamma L_{\mathrm{RD}}(x;(m\odot\theta)).$$

The weight coefficients, α and $(1-\alpha)$ integrate the cross-entropy loss L_{CE} and the response-based knowledge distillation loss L_{KD}, respectively. Since gradient magnitudes from soft targets $\mathrm{e}^{\frac{f_L(x;\theta)}{T}}$ are scaled by $1/T^2$, L_{KD} is multiplied by a factor of T^2 to ensure balanced contributions from L_{CE} and L_{KD} during training.

Supervision scheme for different facets of distillation. KD-LTS employs a dynamic weighting scheme for the hyperparameter γ. This scheme prioritizes aligning the intermediate layer information flow during the initial training phase, gradually shifting focus towards the task-specific knowledge in the teacher's final layer. The weight γ is calculated as: $\gamma^{(e)} = \gamma_{\mathrm{init}}\omega^e$ where e is the current training epoch, ω is a decay factor and γ_{init} is the initial weight. Following Passalis et al. (2020), this scheme facilitates the minimization of information flow divergence and the effective use of target labels. By establishing connections between intermediate layers early on, it helps the student network learn essential representational features before information plasticity diminishes.

4.2 Second Stop: Picking the Winning Ticket

Expanding on the knowledge distillation phase, the goal in this second stop is to identify subnetworks that capture the essence of lottery-ticket information derived from an ensemble of teacher models. To achieve this, we formulate and solve the following Mixed-Integer Optimization (MIO) problem:

$$\begin{aligned}\min \quad & L_{\mathrm{M}}(x; m\odot\theta^{(0)}) + \lambda\|m\|_0\\ \text{subject to} \quad & m\in\{0,1\}^P\end{aligned}$$

where $\lambda \geq 0$ is the regularization coefficient.

Continuous approximation for tractability. Following Savarese et al. (2019), we tackle the tractability issue of MIO by re-parameterizing the mask m as a function of a new variable $s\in\mathcal{R}^P$, and introducing a continuous approximation using a series of continuous sigmoid functions indexed by $\beta\in[1,\infty)$. This leads to the reparameterization $m=\sigma(\beta s)$. Specifically, for $\beta=1, \sigma(\beta s)=\sigma(s)$ and for $\lim_{\beta\to\infty}\sigma(\beta s)=H(s)$, where H(s) is the Heaviside function. Such an approximation approach facilitates a sequence of continuous optimization problems, thus enabling effective learning of masks. The objective for each value of β is given by:

$$L_{\mathrm{M}}\left(x;\sigma\left(\beta s\right)\odot\theta^{(0)}\right) + \lambda\|\sigma\left(\beta s\right)\|_1.$$

This strategy represents a fully deterministic approximation, sidestepping the biases and noise typically associated with stochastic approximation methodsLouizos et al. (2017); Srinivas et al. (2017).

We implement an exponential scheduling strategy for β, denoted as $\mu = (\beta^{(E)})^{\frac{1}{E}}$ for each training epoch e, ranging from 1 to E, to progressively increase β from 1 to $\beta^{(E)}$. $\beta^{(E)}$ should be chosen sufficiently large to ensure the binarization of the learned mask, such that $\sigma(\beta^{(E)} s^{(E)})$ closely approximates the Heaviside function $H(s^{(E)})$ by the conclusion of training.

Gradient Normalization for Seamless Learning. Nevertheless, this reformulation revealed a significant challenge: as the parameter β increases, it induces an early selection of weights for pruning within the training process. To counter this saturation in the mask learning process, we implement l_∞ normalization of the gradients for s across all layers. This innovative approach mitigates the learning plateau often observed in previous studies Savarese et al. (2019), thereby encouraging a more dynamic and seamless progression of mask parameter learning. This adjustment not only improves the conditioning of the optimization problem but also accelerates convergence. Additionally, it simplifies the selection of hyperparameters, such as learning rates and β. The magnitude of gradient updates tends to vary significantly across layers, leading to varying saturation rates in the learning of masks as β is increased. By employing normalized gradient method, the learn at comparable rates, preventing premature saturation in layers that would otherwise dominate due to larger gradient updates, promoting a smoother optimization landscape.

4.3 On the Path to Winning Tickets: Final Stop - Controlling Sparsity

It is important to note that the standard regularization approach, specifically the term $\|\sigma(\beta\widetilde{s})\|_1$, effectively encourages sparsity in the model. However, this technique lacks the precision needed to exert fine control over the sparsity level, particularly when aiming to prune a predetermined fraction of parameters

In tackling this inherent challenge of sparsity control, our approach diverges from traditional constraint-based optimization methods. These methods not only are not designed for precise, fine-grained control over sparsity levels, and they also tend to compromise training accuracy, especially in the context of pruning at initialization. It demands a strategic approach to retain a predefined fraction of weights while avoiding undesired layer disconnections. Such disconnections, known as layer collapse – a phenomenon highlighted in (Tanaka et al., 2020) – could potentially render the network untrainable, thereby necessitating meticulous balance in our sparsity management strategy while fostering learning and not compromising accuracy.

Custom designed scoring function. We leverage ideas from Edge-Popup (EP) and GEM-MINER to control the sparsity of winning tickets. A scoring function, akin to EP, associates a score with each parameter. During the forward pass, the top parameters are selected based on their scores, and during the backward pass, scores are updated using STE. The scoring function, distinct from EP, is elegantly designed to accommodate global pruning and gradual network pruning across epochs. KD-LTS employs a form of backpropagation to control sparsity while optimizing performance, which is achieved by employing a scoring function. It leverages STE and involves the following steps,

1) Sorting the scores: $p_{\text{sorted}} \leftarrow \text{Sort}\left(\sigma\left(\beta\widetilde{s}\right)\right)$.

2) Selecting the bottom fraction of scores: $p_{\text{bottom}} \leftarrow \text{Bottom-}(1 - e^{-\hat{c}e})$ fraction of p_{sorted}.

3) Defining the scoring function: $r \leftarrow r \odot \mathbf{1} p_i \notin p\text{bottom}$.

4) Setting the effective parameters: $\theta = \theta \odot r(\sigma\left(\beta\widetilde{s}\right))$.

To effectively control network sparsity, we employ a strategically designed exponential function, $1 - e^{-\hat{c}e}$, where $\hat{c} = \frac{1}{E} \ln \frac{1}{1-S}$. This function is pivotal in determining the specific percentile of parameters to be pruned at each epoch e, thereby ensuring that the desired final sparsity level S is attained by the targeted epoch E.

The function $r(\cdot)$ serves as an element-wise scoring function. The subnetwork $f_L^{\text{s}}\left(x; r(\sigma\left(\beta^{(e)}\widetilde{s}^{(e)}\right)\right) \odot \theta^{(0)}\right)$ symbolizes the resultant supermask at epoch e. Uniquely, the scoring function prioritizes the retention of parameters that rank within the top percentile based on their absolute magnitudes. This assessment, conducted across the entire network, effectively establishes our approach as a global pruning strategy. Note that pruning decisions are made at each iteration, considering the network as a whole.

KD-LTS at each iteration involves sampling a batch of training data and executing backpropagation based on the loss calculated from the effective weights. During the forward pass, the effective network utilized is indeed

a subnetwork of the original network, which is due to the selective activation of parameters based on the scoring function. Since the scoring function $r(s)$ is inherently non-differentiable, we utilize STE to facilitate backpropagation through the scoring function as if it were an identity function. This application of STE serves as an effective workaround, ensuring a seamless computational process despite the non-differentiable nature of the scoring function.

Diverging from the methodologies employed in Edge-Popup (EP) and GEM-MINER, where the scoring function is directly applied to mask parameters, KD-LTS framework applies scoring function on a continuous sigmoid function of re-parameterized mask parameters. This methodology, detailed in Algorithm 1, promotes a smoother and more refined process for mask learning and sparsity control. A critical observation in our research is the importance of considering the impact of weight parameters during score calculation. This consideration is vital to ensure that the pruned subnetworks are optimally configured for further training, thereby facilitating the identification of winning tickets.

Iterative freezing and regularization. Alongside the scoring function, our framework integrates an iterative freezing mechanism. This mechanism sets the effective parameters as $\theta = \theta \odot r(\sigma(\beta \widetilde{s}_i)) \odot q$. While the scoring function is actively applied in each training iteration, the freezing operation is distinctively implemented at regular intervals, specifically every V epochs. Differing from the scoring function's dynamic approach, the freezing mechanism plays a critical role in stabilizing the network's learning process. It achieves this by blocking the gradients to pruned scores, thereby effectively preventing their influence on further training iterations. We have observed that selectively freezing certain parameters plays a pivotal role in the process of permitting each subnetwork to learn progressively at every iteration, leveraging the learning dynamics of the unpruned network. Such a strategy is instrumental in cultivating a pruned subnetwork wherein the weight parameters can undergo training that is as effective and smooth as that experienced by the original dense network in its early training stages.

Despite the integration of both a scoring function, encompassed within a Straight-Through Estimation (STE) operator and a freezing function, the backpropagated gradient still effectively flows through the reparameterized sigmoid function. This allows for nuanced identification of weights, even amidst the discontinuities introduced by the scoring and freezing operations. Moreover, the gradual pruning process embedded within the scoring operation facilitates finer-grained decisions regarding parameter retention.

This carefully crafted framework, encompassing both innovative sparsity control and parameter retention strategies, paves the way for the identification of effective winning ticket architectures in lareg-scale neural networks. The complete pseudocode detailing the KD-LTS process is presented in Algorithm 1.

Notably, Allen-Zhu and Li (2020) discovered that an ensemble of independently trained neural networks can provably enhance test accuracy, and this improvement can be distilled into a single student model by training it to match the ensemble's output - a feat that traditional training methods, which rely solely on true labels, could not achieve. Building upon this foundation, we extend this unique characteristic of knowledge distillation further into another novel domain, i.e., network pruning. By transferring heterogeneous information from an ensemble of teacher networks into a single student network, we identify sparse, trainable subnetworks at initialization that outperform the original, densely connected student network - a feat that existing pruning-at-initialization techniques, relying solely on true labels, have not accomplished.

5 Experimental Observations

We conducted our experiments using PyTorch on NVIDIA A100 GPUs, spanning a diverse set of standard image classification architectures: VGG-16, LeNet-300-100, ResNet-18, ResNet-20, ResNet-14, and a 6-layer convolutional network. The conducted trials used popular datasets: MNIST, CIFAR-10, CIFAR-100, and Tiny-ImageNet. Optimal training hyperparameters were employed to ensure standard and comparability across experiments in the literature, as illustrated in Tables 4, 5, 6 and 7. For thorough benchmarking, the approach was compared to state-of-the-art pruning methods: SNIP, GraSP, SynFlow, ProsPr, GEM-MINER, LTH, LTH-Rewind, CS, SS, Edge-Popup, Smart Ratio, and Iterative SNIP. Comprehensive details regarding the experimental setup, including data pre-processing, network architectures, implementation of comparative

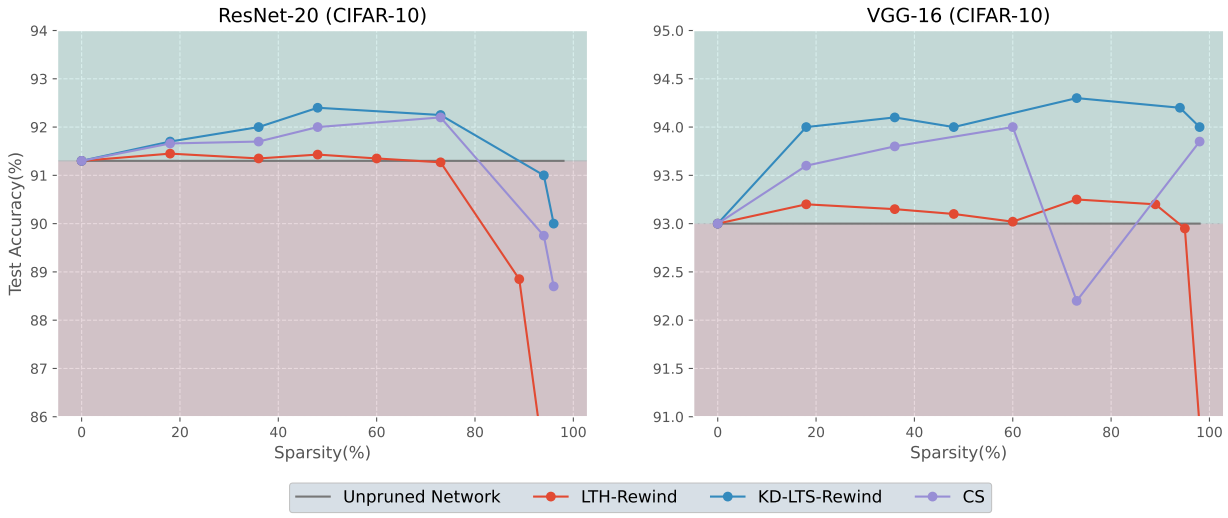

Figure 5: Comparative evaluation of KD-LTS winning ticket performance in VGG-16 and ResNet-20 architectures on the CIFAR-10 dataset, contrasted with prominent early epoch pruning methods, namely LTH-Rewind and CS.

techniques, optimizer settings, and configurations for the knowledge distillation framework, are provided in Section G of the appendix.

Assessment criteria and metrics. Three core metrics form the foundation of our analysis: • **Accuracy:** Measures post-training precision of subnetworks, evaluating the pruning process's impact on performance preservation or enhancement. • **Parameter efficiency:** Indicates the pruned network's ability to maintain performance with significantly fewer parameters. Expressed as the percentage of weights pruned, this reflects the network's compactness. • **Search cost:** Quantifies the computational effort to find pruned subnetworks, measured in epochs needed to reach a desired sparsity. This reflects the method's practical efficiency.

This section compares the performance of KD-LTS winning tickets in sparse, large-scale settings, both at initialization and early training stages. In this assessment, special attention is given to two distinct types of subnetworks that demonstrate the efficacy of KD-LTS as an efficient pruning method: • **Sparsest winning ticket:** Refers to the most aggressively pruned subnetwork that, when trained independently from an early iteration, matches or exceeds the performance of the fully-trained dense network. • **Best performing winning ticket:** Denotes the subnetwork, that when trained in isolation, achieves the highest performance across all pruned networks, regardless of sparsity. These comparisons showcases KD-LTS's ability to maximize parameter efficiency, maintain accuracy, and minimize search costs, showcasing its real-world potential. Table 2 presents results for both sparsest and best-performing winning tickets across various pruning methods.

5.1 Winning Tickets That Really Win the Jackpot and Beyond

Winning tickets at initialization. To fully understand the potential of KD-LTS and initialization pruning, a comprehensive comparitive analysis must extend beyond just test performance against existing initialization methods. This analysis should evaluate search efficiency, parameter reduction, and accelerated learning against both dense networks and early-phase pruning methods. Such a comparison, we believe, would facilitate a research shift from early-stage to initialization pruning.

Figures 3, 23, and 6 provide crucial insights into these capabilities for KD-LTS winning tickets. Figures 3 and 23 illustrate the sparsity-accuracy trade-off of KD-LTS's initialization-derived winning tickets. They benchmark KD-LTS against established early pruning and pruning-at-initialization techniques across various architectures and datasets (VGG-16/CIFAR-10, LeNet-300-100/MNIST, ResNet-18/Tiny-ImageNet, ResNet-20/CIFAR-10, and ResNet-20/CIFAR-100). Figure 6 demonstrates the superior learning efficiency of KD-LTS winning tickets across training epochs at various sparsity levels, benchmarked against dense networks.

Table 2: Comparative evaluation of winning tickets extracted at initialization across diverse experiments: The table uses an ✗ symbol to indicate the absence of sparsest winning tickets, signifying a technique's failure to extract winning tickets at sparsity levels under 20%. 'T.A.' represents Test Accuracy.

Criteria	Method	VGG-16		ResNet-20		LeNet-300-100		ResNet-18	
		T.A. (%)	Sparsity (%)	T.A. (%)	Sparsity (%)	T.A. (%)	Sparsity (%)	T.A. (%)	Sparsity (%)
Sparsest winning ticket	LTH	93	20	✗	20	98	20	51	36
	SNIP	93	20	✗	20	98	20	51	73
	GraSP	✗	✗	✗	✗	✗	✗	✗	✗
	ProsPr	93	20	91.3	20	N/A	N/A	51	60
	Random	93	20	✗	✗	✗	✗	✗	✗
	SynFlow	93	20	✗	✗	✗	✗	51	73
	GEM-MINER	93	90	91.3	25	N/A	N/A	51	80
	Iterative SNIP	93	20	✗	✗	98	20	51	60
	KD-LTS	93	95	91.3	55	98	80	51	92
Best performing winning ticket	LTH	93	20	91.15	20	98	32	51	36
	SNIP	93	20	91.25	20	97.96	32	51	73
	GraSP	92.5	20	90.45	20	97.8	20	50	20
	ProsPr	93.1	48	91.45	20	N/A	N/A	51	60
	Random	93	20	91.15	20	97.92	20	50.1	20
	SynFlow	92.9	20	91.05	20	97.92	32	51.2	20
	GEM-MINER	93	55	91.3	20	N/A	N/A	51.2	55
	Iterative SNIP	93	20	91.15	20	98	20	51.2	73
	KD-LTS	94	40	91.8	40	98.33	45	51.4	45

The following summary highlights key findings on KD-LTS winning tickets at initialization, emphasizing advantages in test performance, parameter efficiency, and learning efficiency over previous benchmarks:

Key Findings

• **Parameter efficiency**: KD-LTS significantly outperforms previous state-of-the-art results in sparsity across various models and datasets: VGG-16/CIFAR-10: 95% sparsity achieved (vs. previous 90%), ResNet-20/CIFAR-10: 55% sparsity (vs. previous 25%), LeNet-300-100/MNIST: 80% sparsity (vs. previous 20%) and ResNet-18/Tiny-ImageNet: 92% sparsity (vs. previous 80%). • **Test efficiency**: Our method demonstrates superior test accuracy compared to existing techniques: VGG-16/CIFAR-10: 94% (vs. previous 93.1%), ResNet-20/CIFAR-10: 91.8% (vs. previous 91.4%), LeNet-300-100/MNIST: 98.33% (vs. previous 98%) and ResNet-18/Tiny-ImageNet: 51.4% (vs. previous 51.2%). • **Learning efficiency**: Subnetworks identified by KD-LTS exhibit significantly accelerated learning, converging on an average 1.2 times faster than the dense network, indicating a substantial improvement in training speed and efficiency.

Supplementary benefit: Enhancing performance through early-phase extraction. Our method represents a multifaceted approach to identifying sparse networks. Emphasizing the aspect of performance, particularly in terms of accuracy, we showcase a complementary advantage: the winning tickets, identified not only at initialization but also during early training epochs using KD-LTS, exhibit state-of-the-art performance. While extraction of winning tickets at initialization follows the protocol outlined in Algorithm 1, a slight adjustment is employed for extracting winning tickets at the early training phase: In this variant, the student model's weights and biases are trained up to the third epoch, mirroring the training settings used for subnetworks post-extraction.

The results presented in Table 3 and Figure 23 aim to showcase the trade-off between resource conservation (in terms of storage and computational costs saved by extracting winning tickets at initialization) and the achievement of performance gains (in terms of accuracy and sparsity achieved by extracting winning tickets at the third epoch) across experiments. This equilibrium demonstrates the versatility of our method, highlighting its utility in various applications with diverse requirements and specifications, across the trade-off continuum mentioned. For further insights into early-training extraction, Figures 5 and 2 provide detailed comparisons of KD-LTS's performance in ResNet-20/CIFAR-10 and VGG-16/CIFAR-10 settings. Figure 5 focuses on the sparsity/accuracy efficiency, while Figure 2 emphasizes search cost efficiency.

Table 3: Comparative evaluation of winning tickets extracted at the third epoch across diverse experiments.

Criteria	Method	VGG-16		ResNet-20	
		Test Accuracy (%)	Sparsity (%)	Test Accuracy (%)	Sparsity (%)
Sparsest winning ticket	CS	93	97	91.3	80
	LTH-Rewind	93	89	91.3	73
	KD-LTS	93	98	91.3	90
Best performing winning ticket	CS	94	60	92.2	73
	LTH-Rewind	93.2	73	91.45	20
	KD-LTS	94.2	94	92.4	48

Key Findings

• **Parameter efficiency**: KD-LTS significantly outperforms LTH-Rewind in terms of sparsest winning tickets extracted at early epochs: VGG-16/CIFAR-10: 98% sparsity (vs. LTH-Rewind 89%) and ResNet-20/CIFAR-10: 90% sparsity (vs. LTH-Rewind 73%). • **Test efficiency**: The best-performing subnetworks achieve superior test accuracy compared to LTH-Rewind: VGG-16/CIFAR-10: 94.2% (vs. LTH-Rewind 93.2%) and ResNet-20/CIFAR-10: 92.4% (vs. LTH-Rewind 91.45%). • **Search efficiency**: This approach is significantly faster in identifying high-performance, sparse winning tickets: 3-4x faster than Continuous Sparsification (CS), 19x faster than LTH-Rewind. • **Initialization vs early epoch pruning**: Winning tickets extracted from early epoch using KD-LTS demonstrate improved performance over initialization-derived counterparts: Test accuracy increase of 0.6-0.9% and sparsity reduction of 8-35%. • **Comprehensive analysis of training and optimization landscape**: Subnetworks identified by our technique at initialization display learning patterns akin to those identified at early epochs by current methods. This similarity substantiates their superior performance and positions KD-LTS as a viable alternative to early epoch pruning.

Assessment of the training landscape. We present an in-depth analysis on the intricate dynamics of KD-LTS' subnetworks extracted at initialization against those extracted in early epochs, and with other pruning at initialization methods, highlighting the superior performance of KD-LTS in fostering stable training and enhanced generalization. In Figure 7, we illustrate the evolution of ResNet-20 during the initial training phase, with a specific focus on the first 2112 iterations (equivalent to 6 epochs). Our first line of examination includes a range of pivotal telemetrics, encompassing average weight magnitude, the proportion of weights that change sign after initialization, gradient magnitude, L_2 distance measuring the deviation of weights from their initial to current values, and the cosine similarity between initial and current weights. The research by Frankle et al. (2020a) points out that large learning rates or complex experimental settings can disrupt the optimization process by excessively perturbing weights, resulting in reduced stability to SGD noise. This phenomenon is identified as a principal reason why existing initialization pruning techniques struggle to discover winning tickets at the onset of training and in contrast why early-phase pruning methods tend to be successful in this regard. The comprehensive telemetric assessment highlight the analogous training behavior of KD-LTS' subnetworks extracted at initialization to those identified in early epochs by current methods, thereby underlining their transformative potential.

Examination across the optimization landscape. Second line of exploration plotted in Figure 8 includes the concept of gradient predictiveness, as elucidated in Santurkar et al. (2018), this metric refers to the L_2 change in gradient when moved along the gradient direction for different learning rates. In the context of KD-LTS , the findings showcase a smoother gradient predictiveness akin to the characteristics observed in subnetworks identified through early pruning techniques. The smoother gradient predictiveness suggests a more predictable and stable gradient evolution in the training trajectory. Utilizing PyHessian, we delve into the Hessian eigen spectral density, examining the Hessian properties of subnetworks extracted using KD-LTS in comparison to standard methods. The results demonstrate a notable capacity to evade saddle points, similar to networks identified by early pruning methods. By measuring the variation in loss at each training step, illustrated as a shaded region in our graphs, we observe how the loss fluctuates across iterations. This metric not only indicates enhanced Lipschitz interpretability akin to early pruning methods. Following the insights from (Jastrzębski et al., 2017; Sankar et al., 2021), a reduced value of Hessian trace correlates with a broader optimum width, indicative of flatter minima and consequently, enhanced generalization capabilities. In an effort to substantiate the superior generalization of KD-LTS subnetworks, we employ the 3D loss

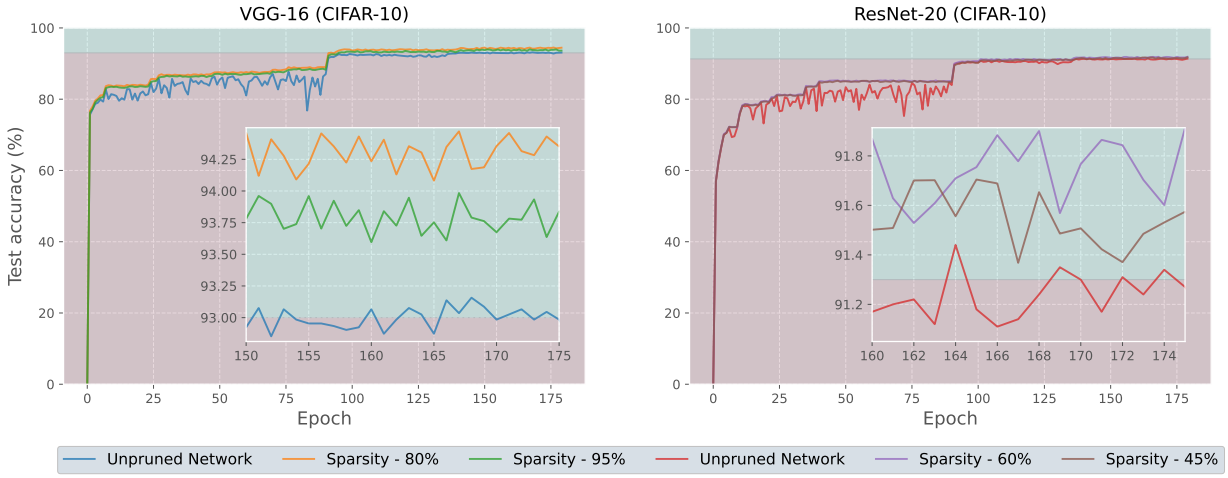

Figure 6: Learning efficiency of KD-LTS' winning tickets juxtaposed with the dense network.

landscape visualization technique, using tools from (Li et al., 2018) and the Paraview application. This visualization at the optimum, presented in Figure 9 **figure and this citation is too far away!**, provides a compelling representation of our subnetworks' generalization strengths. Complementing this, Figures 29, 30 and 31 extends our analysis to additional networks.

Through these findings, our modest hope is to reinvigorate the concept of pruning at initialization and shift the research focus from extracting winning tickets from early epochs towards initialization, thereby opening up prospects for a significant decrease in both computational and storage complexities right from the start of training.

6 Validating the Quality of Winning Tickets

This study undertakes a comprehensive re-evaluation of LTH and addresses limitations of existing pruning-at-initialization techniques, introducing the KD-LTS framework. The findings of Frankle and Carbin (2018) revealed that the subnetworks extracted using prevailing pruning-at-initialization techniques are often susceptible to SGD noise and various random perturbations. This susceptibility was characterized in terms of the stability (Frankle et al., 2020a) and sensitivity (Frankle et al., 2020b) of the subnetworks. It should be noted that these assessments were critical in ascertaining why prevalent pruning-at-initialization techniques, including IMP, fall short in effectively identifying winning tickets and have led to a shift in the focus of pruning towards the early epochs. It's essential that research efforts refocusing the pruning techniques toward the initialization stage should involve subjecting the extracted subnetworks to established tests and quality assessments. Consequently, our investigation extends beyond merely showcasing the superior performance of KD-LTS identified winning tickets; it also encompasses a rigorous validation of their credibility. We begin by providing formal definitions of stability and sensitivity analyses, followed by articulating a strict definition of LTH to promote standardization in works on LTH. Our results demonstrate that KD-LTS subnetworks align with this exacting definition of LTH, which we hope offers a promising path to revitalize pruning at initialization.

Stability analysis. Stability analysis in a subnetwork, specifically $f_L(x; m^{(E)} \odot \theta^{(0)})$ with initial weights $\theta^{(0)}$, involves assessing its response to SGD noise. This is achieved by creating two copies of the network and training each with different samples of SGD noise, such as through varying data order and data augmentation techniques. The resulting networks, $f_L(x; m^{(E)} \odot \theta^{(E)'})$ and $f_L(x; m^{(E)} \odot \theta^{(E)''})$, are then analyzed for stability. The test error of a network is denoted by $\varepsilon(f_L(x; m \odot \theta))$. For a given point ζ within the range $[0, 1]$, the error $\varepsilon_\zeta(f_L(x; m^{(E)} \odot \theta^{(E)'}), f_L(x; m^{(E)} \odot \theta^{(E)''}))$ is computed by interpolating between $\theta^{(E)'}$ and $\theta^{(E)''}$, thus forming a network $f_L(x; m^{(E)} \odot (\zeta\theta^{(E)'} + (1-\zeta)\theta^{(E)''}))$ along the linear path connecting the resulting networks $f_L(x; m^{(E)} \odot \theta^{(E)'})$ and $f_L(x; m^{(E)} \odot \theta^{(E)''})$.

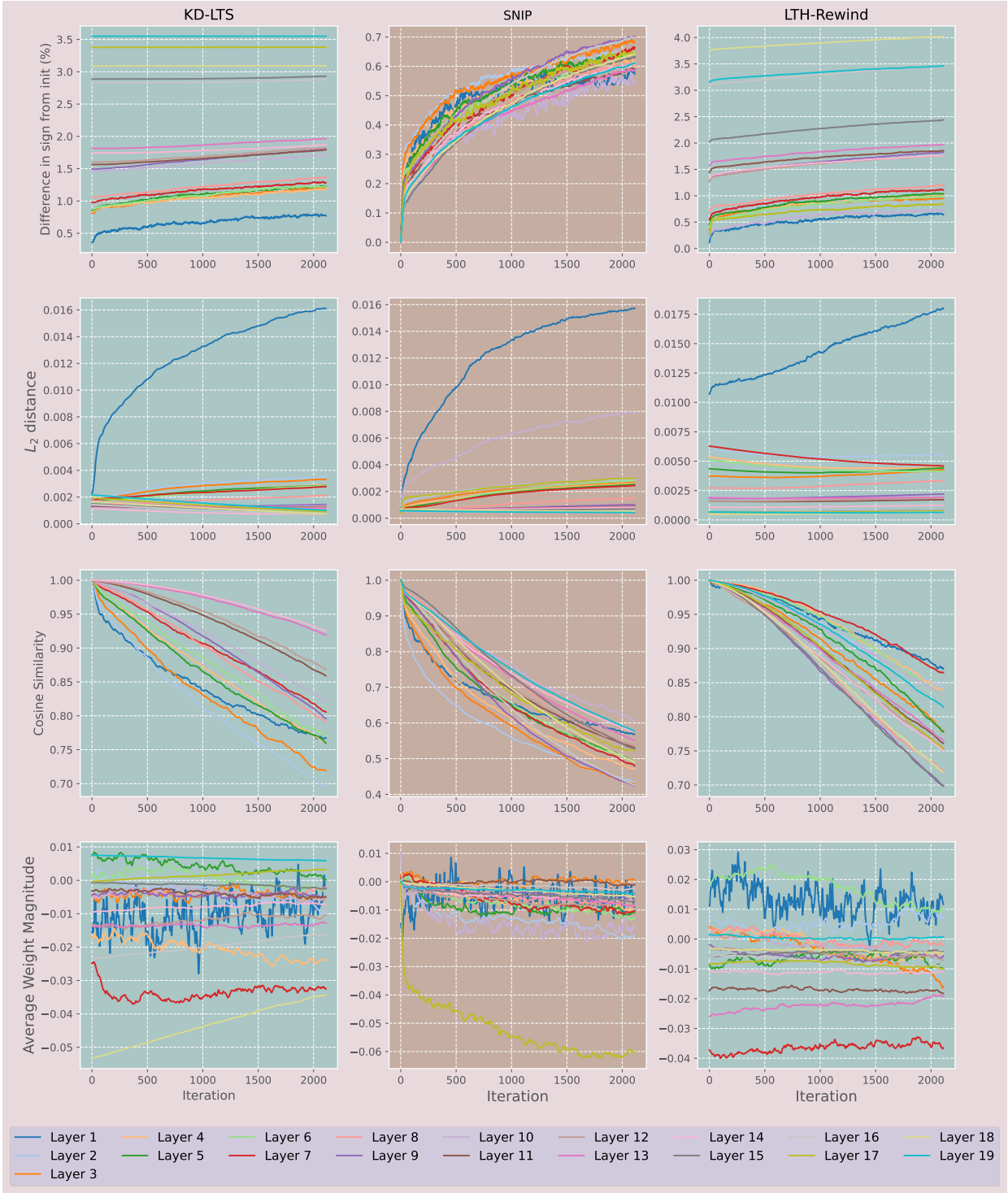

Figure 7: Training dynamics of weight parameters in ResNet-20 subnetworks on CIFAR-10 across the first 2112 iterations, for subnetworks extracted using KD-LTS, SNIP, and LTH-Rewind.

The maximum error observed during this interpolation, denoted as ε_{sup}, and the mean error, denoted as $\bar{\varepsilon}$, are key metrics to assessing stability. With this background, we define stability in the following terms:

Definition 6.1 (Stability)**.** A subnetwork, denoted as $f_L(x; m^{(E)} \odot \theta^{(0)})$, is deemed 'stable' with respect to SGD noise if it exhibits linear mode connectivity between the trained instances, $f_L(x; m^{(E)} \odot \theta^{(E)'})$

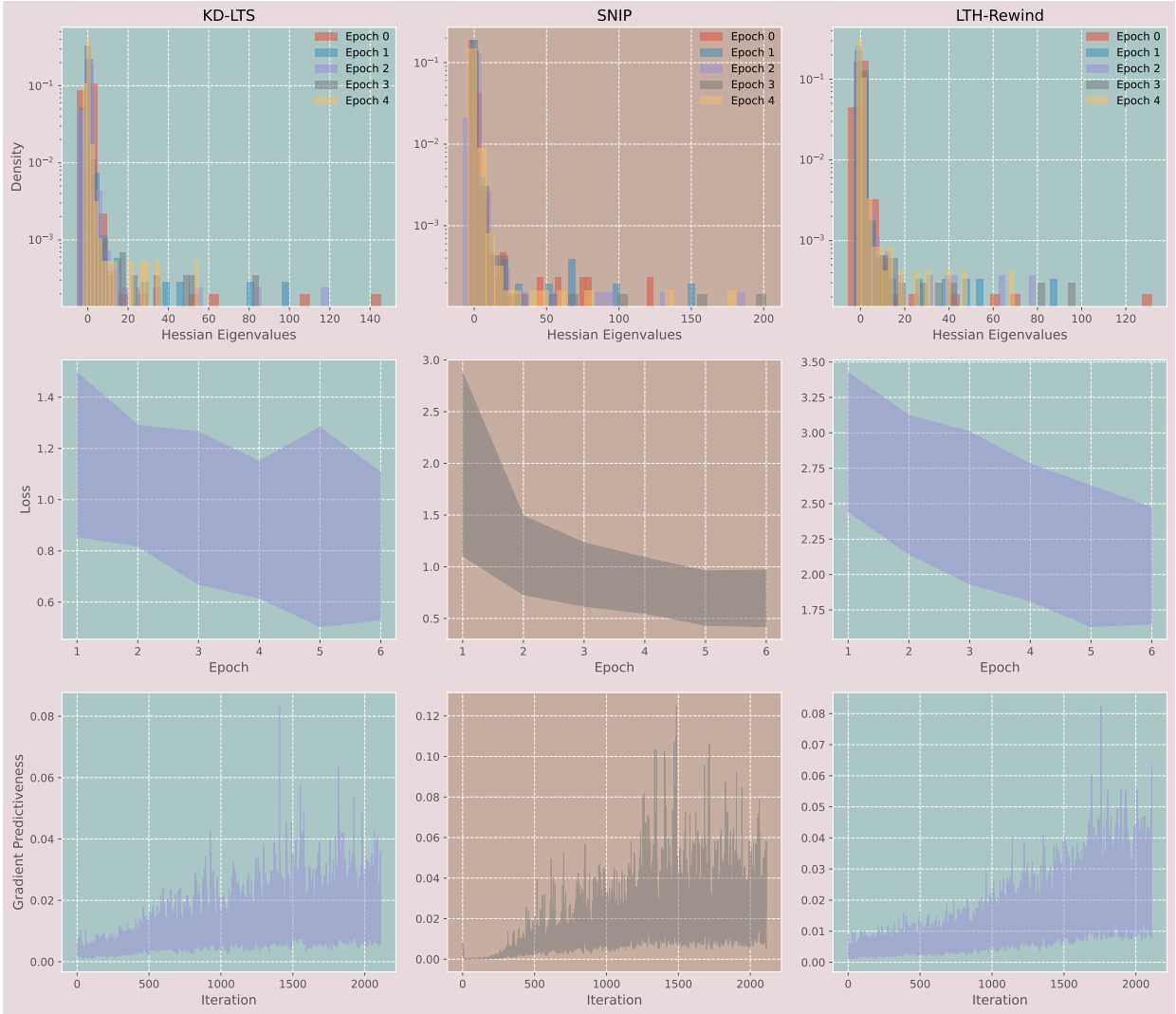

Figure 8: Observation of the optimization landscape through the lens of key Metrics in ResNet-20 subentworks trained on CIFAR-10 across the first six epochs.

and $f_L(x; m^{(E)} \odot \theta^{(E)''})$. Stability is characterized by the difference between the maximum error measure, $\varepsilon_{\text{sup}}(f_L(x; m^{(E)} \odot \theta^{(E)'}), f_L(x; m^{(E)} \odot \theta^{(E)''}))$, and the mean error measure, $\overline{\varepsilon}(f_L(x; m^{(E)} \odot \theta^{(E)'}), f_L(x; m^{(E)} \odot \theta^{(E)''}))$, observed along the linear path connecting the trained instances and is given by

$$\varepsilon_{\text{sup}}(f_L(x; m^{(E)} \odot \theta^{(E)'}), f_L(x; m^{(E)} \odot \theta^{(E)''})) - \overline{\varepsilon}(f_L(x; m^{(E)} \odot \theta^{(E)'}), f_L(x; m^{(E)} \odot \theta^{(E)''})) \leq \Delta,$$

with Δ representing an acceptable threshold of deviation. A network is classified as 'stable' when this difference is within the threshold, indicating minimal deviation due to SGD noise. Conversely, a substantial deviation classifies the subnetwork as 'unstable', demonstrating its susceptibility to SGD noise. An allowance for a tolerance margin is included to account for noise and error variations in the analysis.

Sensitivity analysis. Sensitivity analysis encompasses a range of perturbation studies, each designed to assess the response of the subnetworks identified by various early pruning and initialization pruning methods under various altered conditions. Given a subnetwork $f_L(x; m^{(E)} \odot \theta^{(0)})$ with initial weight parameters $\theta^{(0)}$ and trained masks $m^{(E)}$, the perturbation procedures include:

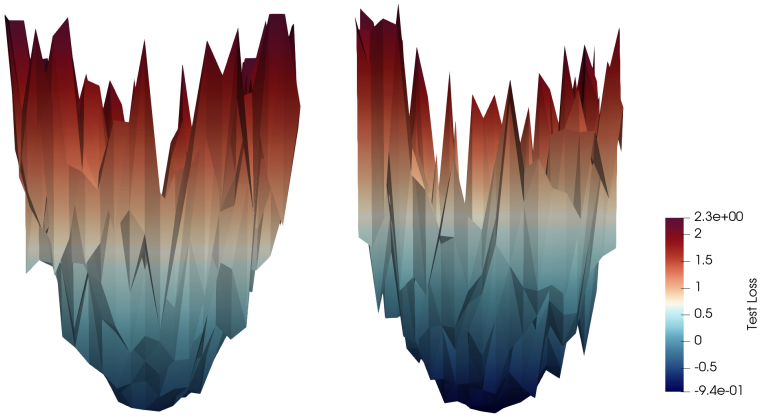

Figure 9: Visualization of the loss landscape at the optimum for ResNet-20 subnetworks, trained on CIFAR-10, with subnetworks extracted via KD-LTS (left) and LTH-Rewind (right).

- **Random reinitialization**: Random reinitialization refers to the procedure in which $\theta^{(0)}$, are reset randomly, resulting in a new set of initial weights $\theta^{(0)'}$. These reinitialized weights are then combined with $m^{(E)}$, to form a randomly reinitialized subnetwork, expressed as $f_L(x; m^{(E)} \odot \theta^{(0)'})$. The test accuracy achieved by training this randomly reinitialized network is denoted as A_{RR}.

- **Inversion**: Inversion represents the operation of sign inversion of these weights, followed by the application of $m^{(E)}$, yields a perturbed subnetwork. This subnetwork, featuring sign-inverted weights, is represented as $f_L(x; m^{(E)} \odot -\theta^{(0)})$. The test accuracy of this subnetwork, post-training, is denoted as A_{I}.

- **Random shuffling**: Random Shuffling is a procedure that entails the random permutation of $m^{(E)}$ to produce a shuffled variant, denoted as $m^{(E)'}$. These shuffled masks are then applied to $\theta^{(0)}$ to form a subnetwork, expressed as $f_L(x; m^{(E)'} \odot \theta^{(0)})$. The test accuracy achieved by training this subnetwork with the randomly shuffled masks is designated as A_{RS}.

- **Small-dense training**: Small-Dense Training refers to the process of training a compact and densely connected neural network, denoted as $f_L(x; \theta_{\text{SD}})$. This network is characterized by reduced width, while maintaining the same depth as $f_L(x; m^{(E)} \odot \theta^{(0)})$, and is initialized with weight parameters θ_{SD}. The test accuracy achieved following the training of this compact network is represented as A_{SD}.

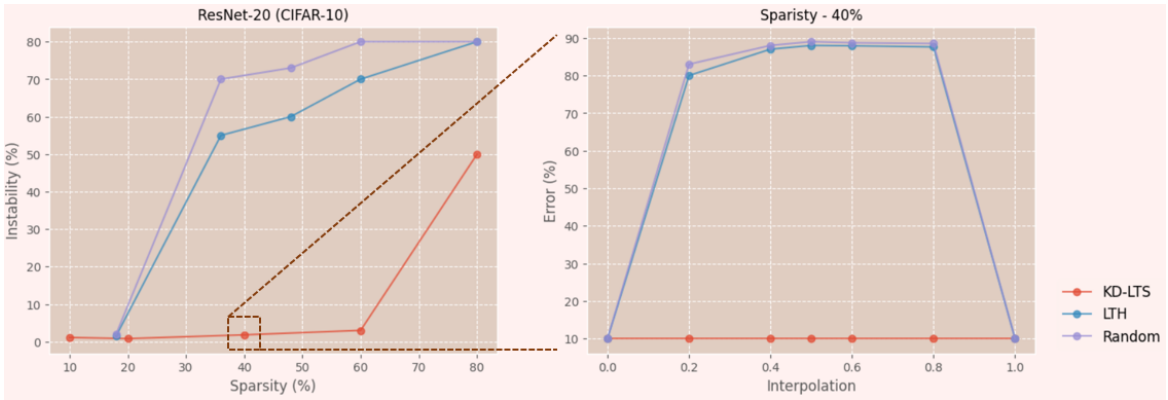

Figure 10: Assessing the indication of stability in LTH, random pruning, and KD-LTS at various sparsity levels in a ResNet-20 experiment on CIFAR-10, and evaluating linear mode connectivity, particularly at non-trivial sparsity levels.

Definition 6.2 (Sensitivity)**.** A subnetwork, denoted as $f_L(x; m^{(E)} \odot \theta^{(0)})$, is considered 'sensitive' with respect to the procedures of Random Reinitialization, Inversion, Random Shuffling, and Small-Dense Training if the following condition is satisfied:

$$\min(A_{\mathrm{RR}}, A_{\mathrm{I}}, A_{\mathrm{RS}}, A_{\mathrm{SD}}) < A_S$$

signifying that the subnetwork's learning capability is significantly affected by the perturbations. Here, A_S denotes the test accuracy of the trained unaltered subnetwork.

In addition to stability and sensitivity analysis, effective network training requires consideration of factors, such as appropriate learning rates and the duration of training epochs. For instance, selecting an optimal learning rate (e.g., 0.1 versus 0.01) combined with a sufficient number of training epochs (e.g., 160 versus 85) can markedly enhance test accuracy (e.g., 91.3% versus 89.0% in a ResNet-20/CIFAR-10 experiment). The literature (Frankle and Carbin, 2018; Liu et al., 2018) suggests that winning tickets often emerge under conditions of smaller learning rates and fewer epochs. As noted, inadequate training durations and suboptimal learning rates can lead to underperformance, thereby significantly undermining the potential for discovering any sparse network in the context of insufficient training. Having underscored these essential factors for assuring the quality of winning tickets, we now proceed to present the revised definition of the LTH.

Definition 6.3 (Lottery Ticket Hypothesis)**.** Consider a subnetwork $f_L(x; m^{(E)} \odot \theta^{(0)})$, with a non-trivial sparsity S, where $m^{(E)}$ is a binary mask obtained through a specific pruning algorithm applied to the initial parameters $\theta^{(0)}$. After training for E'-epochs, let A_{LT} denote the test accuracy achieved by $f_L(x; m^{(E)} \odot \theta^{(0)})$. Furthermore, let A_{D} represent the accuracy of the densely connected network $f_L(x; \theta^{(0)})$, trained with an optimal learning rate schedule over a sufficient duration of E epochs. The Lottery Ticket Hypothesis is then stated as follows: $\exists$ a subnetwork $f_L(x; m \odot \theta^{(0)})$, which when trained for $E' \leq E$ epochs, can achieve an accuracy A_{LT} such that $A_{\mathrm{LT}} \approx A_{\mathrm{D}}$, and $A_{\mathrm{LT}} > \max\{A_{\mathrm{RR}}, A_{\mathrm{I}}, A_{\mathrm{SD}}, A_{\mathrm{RS}}\}$, where "$>$" indicates a clear accuracy gap. Additionally, $\varepsilon_{\sup}(f_L(x; m^{(E)} \odot \theta^{(E)'}), f_L(x; m^{(E)} \odot \theta^{(E)''})) - \overline{\varepsilon}(f_L(x; m^{(E)} \odot \theta^{(E)'}), f_L(x; m^{(E)} \odot \theta^{(E)''})) \leq 2$. Such a subnetwork $f_L(x; m^{(E)} \odot \theta^{(0)})$, if identified, is referred to as a *winning ticket*.

Figure 10 illustrates the linear mode connectivity of KD-LTS winning tickets, verified across 30 interpolation points in a ResNet-20/CIFAR-10 setting. This analysis contrasts KD-LTS with subnetworks identified at initialization by LTH and random pruning methods. The confirmation of a linear path between trained instances of the KD-LTS identified subnetworks is consistent across experiments, as highlighted in Figure 27 for a VGG-16/CIFAR-10 setting. Furthermore, Figure 11 elucidates the sensitivity of these winning tickets to perturbations, such as random shuffling, inversion, and reinitialization, in contrast to subnetworks extracted by LTH, SNIP, GraSP, SynFlow, ProsPr, and GEM-MINER, spanning diverse networks and datasets.

Key Findings

- **Precise connection identification.** The insensitivity of existing initialization pruning methods to ablation studies, such as random shuffling, raises questions about the necessity of the fine-grained operations performed by these algorithms. This observation suggests that a more coarse-grained, per-layer approach to pruning might yield similar, if not superior, results. However, KD-LTS distinguishes itself with its precision by adeptly discerning not only the optimal percentage of weights to prune per layer but also by crucially identifying specific connections that are essential for forming a winning ticket.
- **Randomization ambiguity exclusion.** The response of KD-LTS subnetworks to perturbations, such as inversion and reinitialization, convincingly demonstrates that randomization cannot substitute for any critical stages in the KD-LTS process.
- **Assurance of search cost returns.** By adhering to an optimal training recipe and demonstrating the superior performance of KD-LTS winning tickets compared to small-dense networks, this methodology guarantees a positive return on the computational investment made in the pruning process. This establishes KD-LTS and knowledge distillation as guiding principles for a highly efficient, real-world sparse network solution, especially in resource-constrained environments.

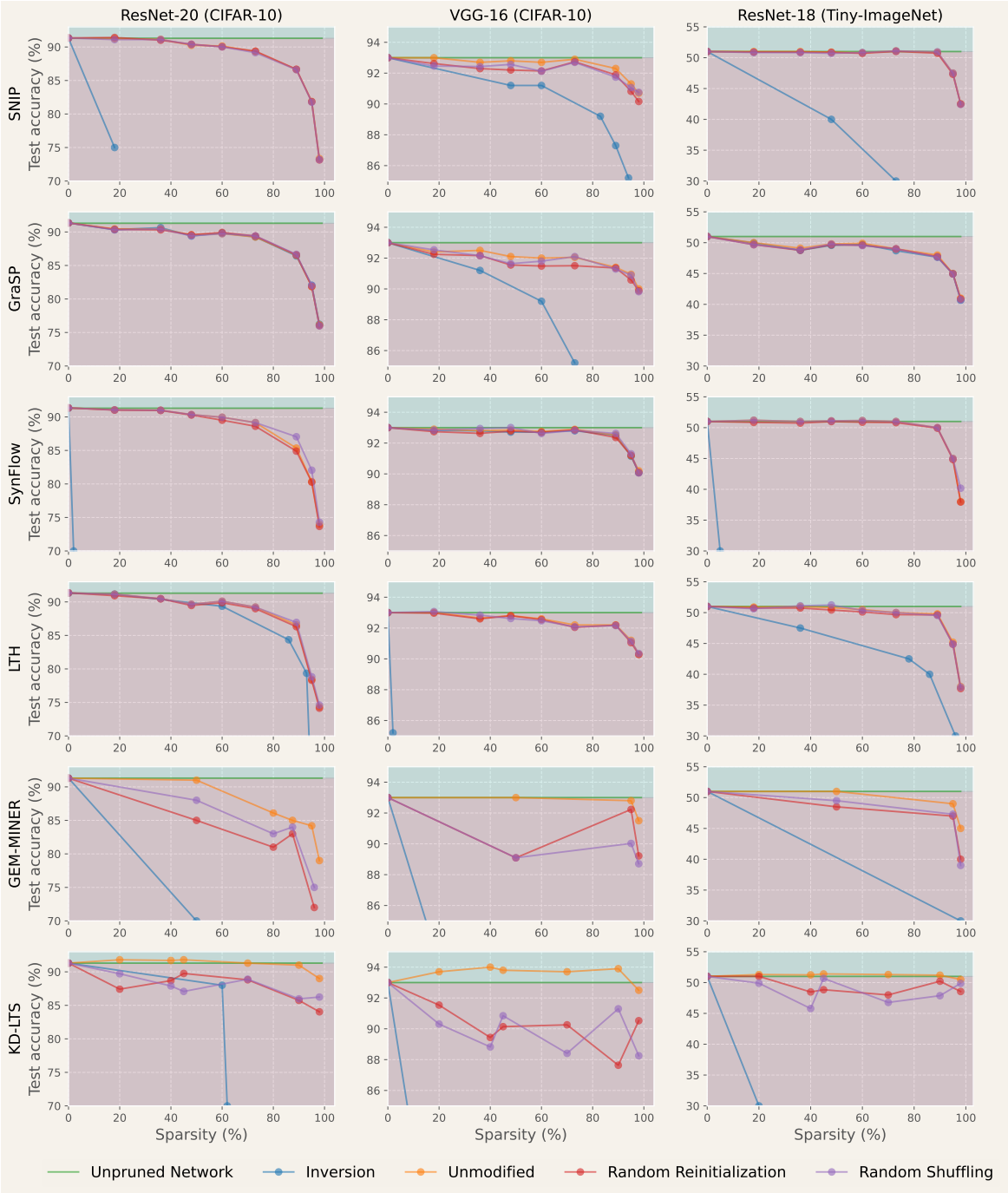

Figure 11: Perturbation studies on subnetworks extracted using LTH, SNIP, GraSP, SynFlow, GEM-MINER and KD-LTS at initialization across experiments.

7 Conclusion

The study challenges the conventional wisdom that highly efficient sparse subnetworks cannot be identified at initialization. Through the proposed KD-LTS technique, we demonstrate the feasibility of initiating pruning at the onset, particularly within demanding scenarios (deeper networks, larger datasets, and more rewarding training settings). By showcasing superior overall performance and rigorous verification of the quality of

KD-LTS winning tickets, we aim to catalyze this paradigm shift, which holds immense potential to significantly reduce training costs in terms of computation, storage, energy consumption, environmental impact, and financial expenditures. Additionally, we spotlight the potential of KD-LTS in embedding specialized traits and its capacity as a practical tool for pruning, leveraging the rich history of research in knowledge distillation.

References

Alessandro Achille, Matteo Rovere, and Stefano Soatto. Critical learning periods in deep neural networks. *arXiv preprint arXiv:1711.08856*, 2017.

Milad Alizadeh, Shyam A Tailor, Luisa M Zintgraf, Joost van Amersfoort, Sebastian Farquhar, Nicholas Donald Lane, and Yarin Gal. Prospect pruning: Finding trainable weights at initialization using meta-gradients. *arXiv preprint arXiv:2202.08132*, 2022.

Zeyuan Allen-Zhu and Yuanzhi Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32, 2019.

Rohan Anil, Gabriel Pereyra, Alexandre Passos, Robert Ormandi, George E Dahl, and Geoffrey E Hinton. Large scale distributed neural network training through online distillation. *arXiv preprint arXiv:1804.03235*, 2018.

Jimmy Ba and Rich Caruana. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27, 2014.

Cody Blakeney, Jessica Zosa Forde, Jonathan Frankle, Ziliang Zong, and Matthew L Leavitt. Reduce, reuse, recycle: Improving training efficiency with distillation. *arXiv preprint arXiv:2211.00683*, 2022.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Long live the lottery: The existence of winning tickets in lifelong learning. In *International Conference on Learning Representations*, 2020.

Tianlong Chen, Yongduo Sui, Xuxi Chen, Aston Zhang, and Zhangyang Wang. A unified lottery ticket hypothesis for graph neural networks. In *International Conference on Machine Learning*, pages 1695–1706. PMLR, 2021a.

Tianlong Chen, Zhenyu Zhang, Jun Wu, Randy Huang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Can you win everything with a lottery ticket? *Transactions of Machine Learning Research*, 2022. URL `https://openreview.net/forum?id=JL6MU9XFzW`.

Xiaohan Chen, Yu Cheng, Shuohang Wang, Zhe Gan, Jingjing Liu, and Zhangyang Wang. The elastic lottery ticket hypothesis. *Advances in Neural Information Processing Systems*, 34, 2021b.

Wojciech M Czarnecki, Simon Osindero, Max Jaderberg, Grzegorz Swirszcz, and Razvan Pascanu. Sobolev training for neural networks. *Advances in neural information processing systems*, 30, 2017.

Pau de Jorge, Amartya Sanyal, Harkirat S Behl, Philip HS Torr, Gregory Rogez, and Puneet K Dokania. Progressive skeletonization: Trimming more fat from a network at initialization. *arXiv preprint arXiv:2006.09081*, 2020.

Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *Advances in neural information processing systems*, 27, 2014.

Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in Neural Information Processing Systems*, 30, 2017.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. *arXiv preprint arXiv:1810.02054*, 2018.

Gabriel Erion, Joseph D Janizek, Carly Hudelson, Richard B Utarnachitt, Andrew M McCoy, Michael R Sayre, Nathan J White, and Su-In Lee. A cost-aware framework for the development of ai models for healthcare applications. *Nature Biomedical Engineering*, 6(12):1384–1398, 2022.

Utku Evci, Yani Ioannou, Cem Keskin, and Yann Dauphin. Gradient flow in sparse neural networks and how lottery tickets win. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 6577–6586, 2022.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR, 2020a.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel M Roy, and Michael Carbin. Pruning neural networks at initialization: Why are we missing the mark? *arXiv preprint arXiv:2009.08576*, 2020b.

Jonathan Frankle, David J Schwab, and Ari S Morcos. The early phase of neural network training. *arXiv preprint arXiv:2002.10365*, 2020c.

Tommaso Furlanello, Zachary Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. In *International Conference on Machine Learning*, pages 1607–1616. PMLR, 2018.

Trevor Gale, Erich Elsen, and Sara Hooker. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*, 2019.

Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019.

Sharath Girish, Shishira R Maiya, Kamal Gupta, Hao Chen, Larry S Davis, and Abhinav Shrivastava. The lottery ticket hypothesis for object recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 762–771, 2021.

Yifan Gong, Geng Yuan, Zheng Zhan, Wei Niu, Zhengang Li, Pu Zhao, Yuxuan Cai, Sijia Liu, Bin Ren, Xue Lin, et al. Automatic mapping of the best-suited dnn pruning schemes for real-time mobile acceleration. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 27(5):1–26, 2022.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013.

Shupeng Gui, Haotao Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. *Advances in Neural Information Processing Systems*, 32, 2019.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017.

Arjun Gupta, Eliu Huerta, Zhizhen Zhao, and Issam Moussa. Deep learning for cardiologist-level myocardial infarction detection in electrocardiograms. In *8th European Medical and Biological Engineering Conference: Proceedings of the EMBEC 2020, November 29–December 3, 2020 Portorož, Slovenia*, pages 341–355. Springer, 2021.

Saurabh Gupta, Judy Hoffman, and Jitendra Malik. Cross modal distillation for supervision transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2827–2836, 2016.

Guy Gur-Ari, Daniel A Roberts, and Ethan Dyer. Gradient descent happens in a tiny subspace. *arXiv preprint arXiv:1812.04754*, 2018.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *Advances in neural information processing systems*, 28:1135–1143, 2015.

Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

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

Melissa Heikkilä. We're getting a better idea of ai's true carbon footprint. *Technology Review. https://www. technologyreview. com/2022/11/14/1063192/were-getting-a-betteridea-of-ais-true-carbon-footprint*, 2022.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

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

Sara Hooker, Aaron Courville, Gregory Clark, Yann Dauphin, and Andrea Frome. What do compressed deep neural networks forget? *arXiv preprint arXiv:1911.05248*, 2019.

Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *The Journal of Machine Learning Research*, 18(1):6869–6898, 2017.

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Xiaoqian Jiang, Melanie Osl, Jihoon Kim, and Lucila Ohno-Machado. Calibrating predictive model estimates to support personalized medicine. *Journal of the American Medical Informatics Association*, 19(2):263–274, 2012.

Neha Mukund Kalibhat, Yogesh Balaji, and Soheil Feizi. Winning lottery tickets in deep generative models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8038–8046, 2021.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. *arXiv preprint arXiv:1606.07947*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 1990.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018a.

Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised knowledge distillation using singular value decomposition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–350, 2018b.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Xingyu Liu, Alex Leonardi, Lu Yu, Chris Gilmer-Hill, Matthew Leavitt, and Jonathan Frankle. Knowledge distillation for efficient sequences of training runs. *arXiv preprint arXiv:2303.06480*, 2023.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. *Advances in neural information processing systems*, 27, 2014.

Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. *arXiv preprint arXiv:1712.01312*, 2017.

Xiaolong Ma, Geng Yuan, Xuan Shen, Tianlong Chen, Xuxi Chen, Xiaohan Chen, Ning Liu, Minghai Qin, Sijia Liu, Zhangyang Wang, et al. Sanity checks for lottery tickets: Does your winning ticket really win the jackpot? *Advances in Neural Information Processing Systems*, 34:12749–12760, 2021a.

Yilin Ma, Ruizhu Han, and Weizhong Wang. Portfolio optimization with return prediction using deep learning and machine learning. *Expert Systems with Applications*, 165:113973, 2021b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Eran Malach, Gilad Yehudai, Shai Shalev-Shwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. *arXiv preprint arXiv:2002.00585*, 2020.

Andrey Malinin, Bruno Mlodozeniec, and Mark Gales. Ensemble distribution distillation. *arXiv preprint arXiv:1905.00076*, 2019.

Zelda Mariet and Suvrit Sra. Diversity networks: Neural network compression using determinantal point processes. *arXiv preprint arXiv:1511.05077*, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11264–11272, 2019.

Ari S Morcos, Haonan Yu, Michela Paganini, and Yuandong Tian. One ticket to win them all: generalizing lottery ticket initializations across datasets and optimizers. *arXiv preprint arXiv:1906.02773*, 2019.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29, 2015.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. Towards understanding the role of over-parametrization in generalization of neural networks. *arXiv preprint arXiv:1805.12076*, 2018.

Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic pruning is all you need. *Advances in Neural Information Processing Systems*, 33, 2020.

Michela Paganini. Prune responsibly. *arXiv preprint arXiv:2009.09936*, 2020.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

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

Nikolaos Passalis, Maria Tzelepi, and Anastasios Tefas. Heterogeneous knowledge distillation using information flow modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2339–2348, 2020.

Tony Peng and Michael Sarazen. The staggering cost of training sota ai models. *published online*, 2019.

Gean T Pereira and André CPLF de Carvalho. Bringing robustness against adversarial attacks. *Nature Machine Intelligence*, 1(11):499–500, 2019.

Gregory Plumb, Maruan Al-Shedivat, Ángel Alexander Cabrera, Adam Perer, Eric Xing, and Ameet Talwalkar. Regularizing black-box models for improved interpretability. *Advances in Neural Information Processing Systems*, 33:10526–10536, 2020.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. *arXiv preprint arXiv:2005.00561*, 2020.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11893–11902, 2020.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in Neural Information Processing Systems*, 33:20378–20389, 2020.

Adepu Ravi Sankar, Yash Khasbage, Rahul Vigneswaran, and Vineeth N Balasubramanian. A deeper look at the hessian eigenspectrum of deep neural networks and its applications to regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 9481–9488, 2021.

Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization? In *Advances in Neural Information Processing Systems*, pages 2483–2493, 2018.

Pedro Savarese, Hugo Silva, and Michael Maire. Winning the lottery with continuous sparsification. *arXiv preprint arXiv:1912.04427*, 2019.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Kartik Sreenivasan, Jy-yong Sohn, Liu Yang, Matthew Grinde, Alliot Nagle, Hongyi Wang, Eric Xing, Kangwook Lee, and Dimitris Papailiopoulos. Rare gems: Finding lottery tickets at initialization. *Advances in Neural Information Processing Systems*, 35:14529–14540, 2022.

Suraj Srinivas and François Fleuret. Knowledge transfer with jacobian matching. In *International Conference on Machine Learning*, pages 4723–4731. PMLR, 2018.

Suraj Srinivas, Akshayvarun Subramanya, and R Venkatesh Babu. Training sparse neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 138–145, 2017.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.

Jingtong Su, Yihang Chen, Tianle Cai, Tianhao Wu, Ruiqi Gao, Liwei Wang, and Jason D Lee. Sanity-checking pruning methods: Random tickets can win the jackpot. *Advances in neural information processing systems*, 33:20390–20401, 2020.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27, 2014.

Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1701–1708, 2014.

Hidenori Tanaka, Daniel Kunin, Daniel LK Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. *arXiv preprint arXiv:2006.05467*, 2020.

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017.

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

Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th international conference on software engineering*, pages 303–314, 2018.

Bindya Venkatesh, Jayaraman J Thiagarajan, Kowshik Thopalli, and Prasanna Sattigeri. Calibrate and prune: Improving reliability of lottery tickets through prediction calibration. *arXiv preprint arXiv:2002.03875*, 2020.

Roberto Verdecchia, June Sallou, and Luís Cruz. A systematic review of green ai. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, page e1507, 2023.

Stijn Verdenius, Maarten Stol, and Patrick Forré. Pruning via iterative ranking of sensitivity statistics. *arXiv preprint arXiv:2006.00896*, 2020.

Chaofei Wang, Qisen Yang, Rui Huang, Shiji Song, and Gao Huang. Efficient knowledge distillation from model checkpoints. *Advances in Neural Information Processing Systems*, 35:607–619, 2022.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. *arXiv preprint arXiv:2002.07376*, 2020.

Ji Wang, Weidong Bao, Lichao Sun, Xiaomin Zhu, Bokai Cao, and S Yu Philip. Private model compression via knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1190–1197, 2019.

Yue Wang, Tan Nguyen, Yang Zhao, Zhangyang Wang, Yingyan Lin, and Richard Baraniuk. Energynet: Energy-efficient dynamic inference. 2018.

Junru Wu, Yue Wang, Zhenyu Wu, Zhangyang Wang, Ashok Veeraraghavan, and Yingyan Lin. Deep k-means: Re-training and parameter sharing with harder cluster assignments for compressing deep convolutions. In *International Conference on Machine Learning*, pages 5363–5372. PMLR, 2018.

Liang Xiao, Ruili Wang, Bin Dai, Yuqiang Fang, Daxue Liu, and Tao Wu. Hybrid conditional random field based camera-lidar fusion for road detection. *Information Sciences*, 432:543–558, 2018.

Tien-Ju Yang, Yu-Hsin Chen, and Vivienne Sze. Designing energy-efficient convolutional neural networks using energy-aware pruning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5687–5695, 2017.

Shaokai Ye, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 111–120, 2019.

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4133–4141, 2017.

Haoran You, Chaojian Li, Pengfei Xu, Yonggan Fu, Yue Wang, Xiaohan Chen, Richard G Baraniuk, Zhangyang Wang, and Yingyan Lin. Drawing early-bird tickets: Towards more efficient training of deep networks. *arXiv preprint arXiv:1909.11957*, 2019.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S Morcos. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. *arXiv preprint arXiv:1906.02768*, 2019.

Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Visual relationship detection with internal and external linguistic knowledge distillation. In *Proceedings of the IEEE international conference on computer vision*, pages 1974–1982, 2017.

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

Marcos Zampieri, Preslav Nakov, and Yves Scherrer. Natural language processing for similar languages, varieties, and dialects: A survey. *Natural Language Engineering*, 26(6):595–612, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115, 2021a.

Zhenyu Zhang, Xuxi Chen, Tianlong Chen, and Zhangyang Wang. Efficient lottery ticket finding: Less data is more. In *International Conference on Machine Learning*, pages 12380–12390. PMLR, 2021b.

Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *Advances in Neural Information Processing Systems*, pages 3597–3607, 2019.

Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

Appendix

A Advancing Beyond Inheritance: Embracing Knowledge Distillation for Comprehensive Assessment **31**

B Ablation Studies: Key Observations and Insights for Future Research Directions **32**

C Can You Win Everything with A Winning Ticket "at Initialization"? **35**

C.1 Generalization to Distribution Shifts 35

C.2 Calibration and Reliability 38

C.3 Interpretability 41

C.4 Transferability 42

D Supermasks **42**

E Future Work **44**

F Additional experiments **45**

G Experimental Setup **51**

G.1 Datasets 51

G.2 Network Architectures 51

G.3 Training Settings 51

H Benchmarking Sparsification Techniques: A Comparative Study **52**

A Advancing Beyond Inheritance: Embracing Knowledge Distillation for Comprehensive Assessment

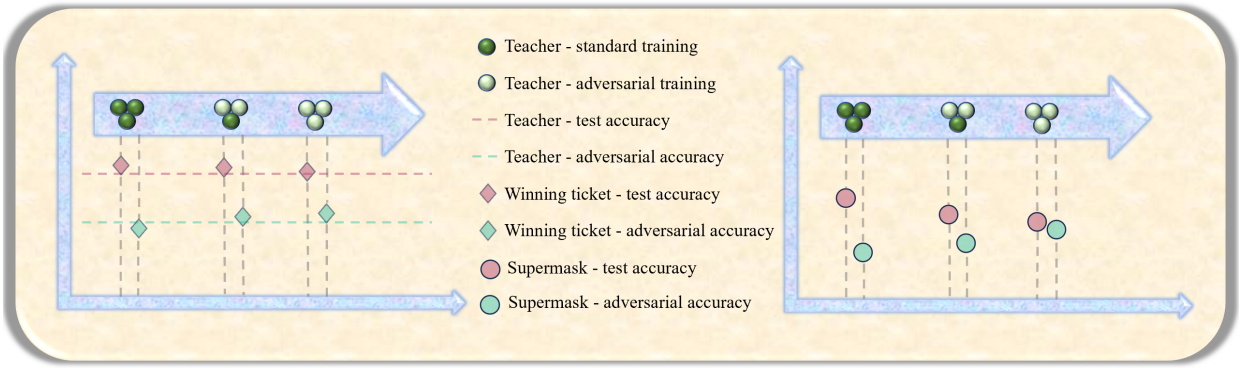

Figure 12: A sketchmap of two counter-intuitive insights: It demonstrates how the KD-LTS model can facilitate the identification of adversarially robust supermasks (on the left) and adversarially robust winning tickets (on the right), utilizing an adversarially robust teacher, all while preserving test performance.

KD-LTS framework capitalizes on a synergy between ground truth label training and the rich, heterogeneous information sourced from an ensemble of teacher networks. As highlighted in the previous section, this innovative approach extends beyond merely enhancing the test accuracy of subnetworks derived from student models. We strategically broaden the already laid out scope and utilize the knowledge distillation platform to further enhance the traits of the subnetworks. This is achieved by utilizing a carefully chosen set of teacher networks with unique attributes, as opposed to standard teacher networks.

Key Insight

KD-LTS uncovers an unexplored and non-intuitive connection: Knowledge distillation, when executed with teacher networks that are robust to adversarial attacks, goes beyond simply identifying robust supermasks, even when distillation is performed with standard data. These supermasks evolve into winning tickets after training, with enhanced adversarial robustness while maintaining test performance, achieved through standard training without any adversarial data. This contrasts with supermasks and winning tickets derived from standard teacher networks, underscoring the distinctive advantage of our approach.

Looking ahead, there is an intention to further explore the potential of KD-LTS framework. The aim is to endow winning tickets with greater interpretability, reinforce confidence in their predictions, bolster their overall reliability and improve transferability beyond adversarial robustness. This will be achieved by using teacher networks that exhibit qualities such as interpretability, confidence, reliability, and those trained on source optimizer, dataset, model or application, to enhance transferability.

Key Findings

- **Transcending merely savoring overall performance: Actively embedding traits in winning tickets.** KD-LTS subnetworks derived using standard teacher models exhibit a 2-3% enhancement over dense networks and subnetworks extracted using early-phase pruning. Figure 13 demonstrates that winning tickets derived from student networks, which incorporate lottery-ticket information from adversarially robust teacher networks, achieve an additional 3% improvement in adversarial robustness. This increment is in addition to the 2-3% improvement already observed with standard teacher models. Furthermore, we have observed a 2-4% increase in the robustness of supermasks.

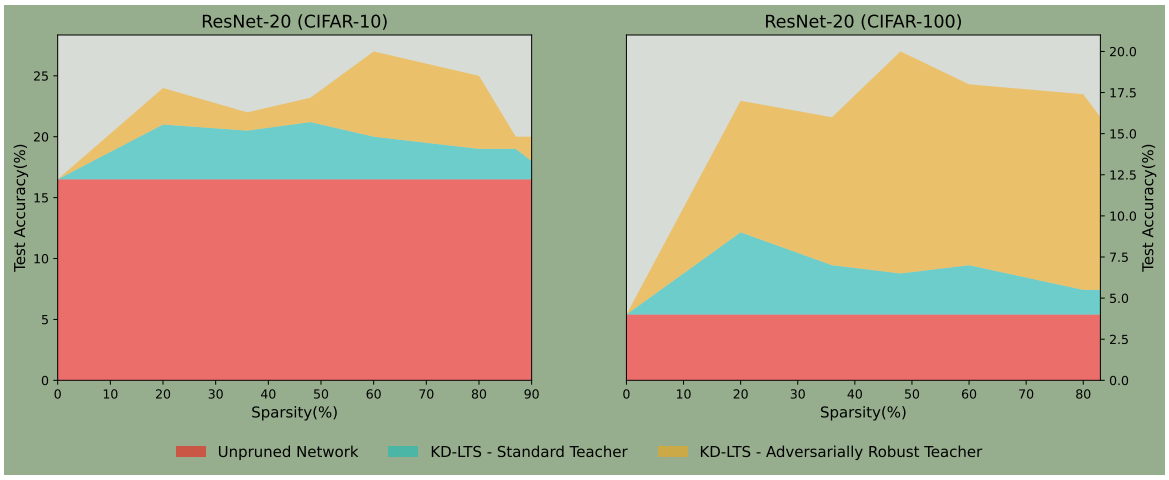

Figure 13: Adversarial accuracy of winning tickets obtained in a ResNet-20 network across datasets CIFAR-10 (left) & CIFAR-100 (right) using standard teacher and adversarially robust teacher.

B Ablation Studies: Key Observations and Insights for Future Research Directions

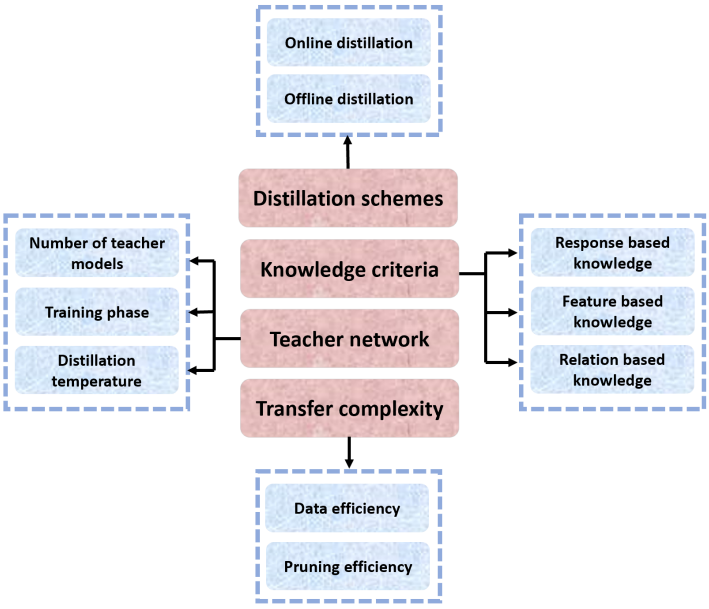

Figure 14: Dissection on the major components of KD-LTS framework.

This section details ablation studies within the KD-LTS framework, highlighting variations across knowledge categories, distillation schemes, teacher model configurations, and the complexities in knowledge transfer. These analyses are conducted using a modest experimental setup, leveraging the LeNet-300-100 architecture trained on the MNIST dataset. Although extending these studies beyond this scale is outside this paper's purview, the ablation studies presented aim to map out multifaceted avenues for future research within the KD-LTS platform. The exploration not only seeks to provide in-depth insights into the framework's effectiveness as a research tool - demonstrating the feasibility of model pruning at initiation - but also to affirm its practical value in identifying effective winning tickets from the outset.

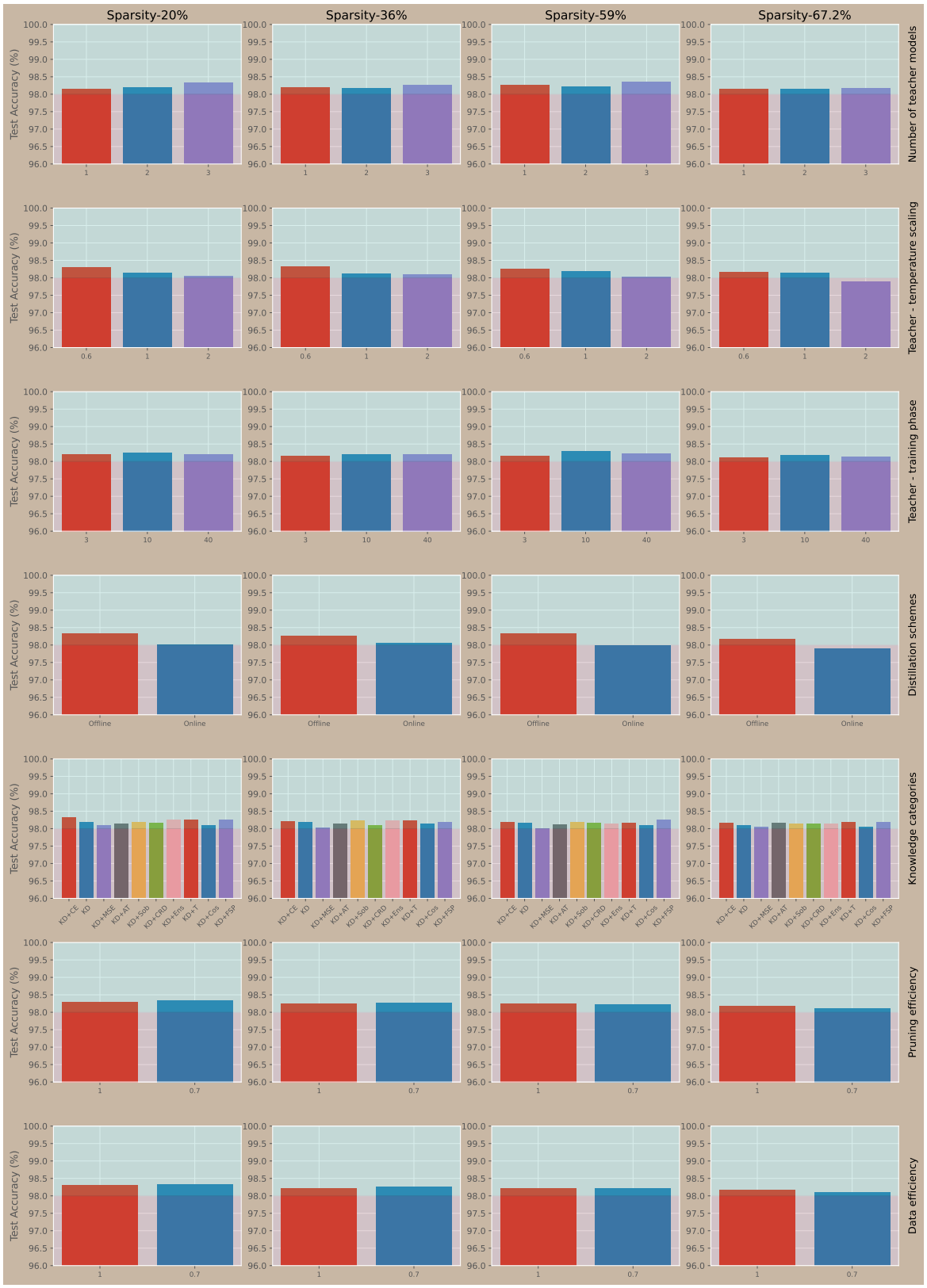

Figure 15: Ablation studies on key componenets of KD-LTS framework in a LeNet-300-100 (MNIST) environment.

- **Number of teacher models.** Figure 15 demonstrate that leveraging representational and structural insights from multiple teacher models helps in identifying high-performing winning tickets, notably in the high sparsity regime. Interestingly, the accuracy improvements achieved through a multi-teacher distillation strategy in the rare sparsity regime, compared to employing a single teacher model, are marginal. This finding suggests that a singular teacher model is capable of encoding the crucial lottery-ticket information necessary for identifying winning tickets. To facilitate the transfer of knowledge from an ensemble of teacher models, we initially employed a straightforward averaging of their responses and features, a technique that incurs a computational cost of $\mathcal{O}(n)$. Conversely, by implementing a strategy that involves randomly selecting a single teacher model at each iteration, we significantly reduced the computational requirements to $\mathcal{O}(1)$, while still tapping into the diversity offered by the ensemble. This computationally efficient approach yielded results comparable to those of the vanilla multi-teacher averaging in terms of winning ticket performance, underscoring the efficacy of simpler distillation methodologies in the context of KD-LTS.

- **Training phase of teacher models.** Substantial research (Frankle et al., 2020c; Gur-Ari et al., 2018; Achille et al., 2017) has dissected the complexities of DNNs across various training stages. This study extends the exploration to assess the impact of different training stages on the identification of high-performance winning tickets. By examining the teacher model at distinct phases of training, this investigation aims to pinpoint the optimal stage for transferring critical information to the student network, thereby facilitating the efficient discovery of winning tickets. Remarkably, it was found that teacher models at intermediate training stages—specifically around the 10th epoch—yielded winning tickets with superior performance compared to those at their early 3rd epoch or their later 40th epoch. This observation aligns with findings from traditional knowledge distillation research on model checkpoints, as reported by Wang et al. (2022). Surprisingly, this observation suggests that even teacher networks in the early stages of training possess enough valuable information for the effective extraction of winning tickets, challenging previous assumptions.

- **Pruning efficiency and data efficiency.** LTH and early-epoch pruning techniques have attracted considerable attention for their promising performance. However, the computationally intensive cycles required to identify winning tickets present a significant barrier to their wider practical deployment. While You et al. (2019) and Zhang et al. (2021b). have made strides in addressing this issue - You et al. (2019). by employing Hamming distance to track the evolution of masks across epochs, and Zhang et al. (2021b). by introducing the Pruning -Aware Critical (PrAC) set to optimize data selection - their experiments have predominantly focused on the realm of early-epoch pruning. Building upon this foundation, this study expands into the domain of KD-LTS and pruning at initialization. Findings, depicted in the plots, highlight the efficiency of KD-LTS in significantly reducing the number of training epochs required, achieving comparable results with just 70% of the typical training duration. Furthermore, we demonstrate data efficiency by successfully extracting winning tickets utilizing merely 70% of the full dataset.

- **Distillation schemes.** Ablation studies contrast offline and online distillation approaches, finding that pre-trained teacher models are not essential for identifying high-performing winning tickets. A randomly initialized teacher model and online transfer of knowledge also produce strikingly similar performance.

- **Distillation temperature.** Observations disclose that teacher models trained with varying levels of distillation temperature T for soft-target response-based distillation result in distinct performances of winning tickets, aligning with findings on conventional knowledge distillation by Müller et al. (2019). Specifically, models trained with a T less than 1, which form less discernible clusters at the penultimate layer activations, lead to the identification of higher-quality winning tickets. Conversely, teacher models with T greater than 1, which form defined but broader clusters, correlate with a decrease in the performance of winning tickets. — effectively guiding the student to mimic the teacher's response more closely and thus transferring the "dark knowledge" embedded in the teacher's output distribution.

- **Knowledge categories.** We explore various knowledge domains to effectively identify winning tickets, aiming to enrich the quality of our distillation framework by leveraging, diversity within ensembles through Adaptive Ensemble Knowledge Distillation (AE-KD) (Malinin et al., 2019), first-order Jacobian information (Czarnecki et al., 2017), attention maps, data diversity employing a Contrastive Representation

Distillation (CRD) framework (Tian et al., 2019), (Zagoruyko and Komodakis, 2016), critical period-aware mutual information between features Passalis et al. (2020); Park et al. (2019), layer diversity by exploiting the solution of flow procedure (FSP) (Yim et al., 2017), and, feature-based simple Mean Squared Error (MSE) loss (Ba and Caruana, 2014). A notable observation is that, although exploiting the diversity among heterogeneous features and layers indicates that improved performance winning tickets can be obtained, a simple response-based distillation with a KL-divergence loss on the logits of the student and teacher models, along with cross-entropy, is sufficient for extracting winning tickets. This study not only aids in understanding the effectiveness of KD-LTS framework but also analyses various signal telemetry to identify optimal signals for future research in developing a scoring-based function for pruning at initialization.

Key Findings

• **Commensurate reduction in complexity.** Ablation studies focusing on the number of teacher models, training phases of the teacher model, data efficiency, and pruning efficiency underscore the KD-LTS framework's contributions to computational and storage efficiency, showcasing its practicality as a real-world tool for identifying winning tickets. • **Dynamics of pruning at initialization.** Ablation studies on knowledge distillation categories and schemes not only assist in deconstructing the intricate dynamics of KD-LTS but also highlight critical signals that can systematically facilitate the development of techniques for identifying winning tickets at initialization, thereby eliminating the need to train the dense network altogether.

C Can You Win Everything with A Winning Ticket "at Initialization"?

In this paper, we meticulously evaluate the test accuracy of KD-LTS subnetworks across diverse state-of-the-art DNNs (He et al., 2016; Simonyan and Zisserman, 2014), renowned for their exceptional real-world task performance. However, we recognize the imperative need to assess neural networks beyond mere test accuracy. In today's world, where billions rely on DNN-powered technologies such as ChatGPT chatbots, known for their advanced prompt learning capabilities (Brown et al., 2020; Radford et al., 2019), a comprehensive assessment of neural network performance becomes crucial. A glimpse into key performance dimensions and their practical applications reveals, • **Interpretability:** Paramount in sectors like medical diagnosis (Gupta et al., 2021; Jiang et al., 2012), interpretability ensures model transparency and user trust by making complex models understandable. • **Confidence:** Vital in safety-critical applications such as autonomous driving (Tian et al., 2018; Bojarski et al., 2016), where the reliability of a model's predictions is directly linked to safety outcomes. • **Robustness:** Essential for security, robustness involves the network's resilience to adversarial attacks (Goodfellow et al., 2014) and noisy data, guaranteeing consistent performance under uncertain conditions, such as in cybersecurity applications Pereira and de Carvalho (2019). • **Transferability:** Critical for the versatility of neural networks, enabling models to effectively adapt across various domains with minimal retraining (Long et al., 2015), like language model adaptation across different languages or dialects Zampieri et al. (2020). • **Out-of-Distribution (OoD) Performance:** Crucial for reliability, OoD performance measures a model's ability to handle data significantly different from its training distribution, ensuring competent operation in dynamic settings (Liang et al., 2017), such as in financial modeling Ma et al. (2021b), where models must adapt to unforeseen conditions. Each of these dimensions is vital for ensuring that neural networks meet the ever evolving demands of real-world applications.

Having underscored the necessity for a holistic evaluation framework, we now meticulously analyze the performance of KD-LTS subnetworks extracted right from initialization across a wider array of critical performance dimensions, as outlined in (Chen et al., 2022; Morcos et al., 2019; Chen et al., 2021b).

C.1 Generalization to Distribution Shifts

In the forthcoming subsections, we first define the metrics employed for evaluation, ensuring a clear understanding of our assessment criteria. We then present the principal findings, thoroughly discussing the implications and insights derived from the experiments. The following examination evaluates how KD-LTS

subnetworks fare against altered or shifted data distributions, focusing particularly on their adaptability to scenarios involving natural corruptions, adversarial perturbations, and OoD data.

Definition C.1 (Accuracy)**.** The accuracy of a pruned student network, denoted by $f_L^{\mathrm{s}}(x; m \odot \theta)$, and parameterized by weights θ and a binary mask m, is the ratio of the number of correctly predicted labels to the total number of input samples. Formally, accuracy is defined as:

$$\text{Accuracy} = \frac{1}{I} \sum_{(x,y)\in\mathbb{D}} \delta\left(\arg\max_{c\in\{1,\ldots,C\}} f_{cL}^{\mathrm{s}}(x; m \odot \theta) = \arg\max_{c\in\{1,\ldots,C\}} y_c\right)$$

where $\delta(\cdot)$ is the indicator function, which evaluates to 1 if the predicted label matches the true label and 0 otherwise. Here, $\arg\max_{c\in\{1,\ldots,C\}} f_{cL}^{\mathrm{s}}(x; m \odot \theta)$ signifies the predicted label and $\arg\max_{c\in\{1,\ldots,C\}} y_c$ denotes the ground-truth label for the input sample x.

The following concepts, directly related to evaluating a model's test accuracy, are integral in validating its capacity to generalize across distribution shifts:

- **Generalization Gap:** It is the discrepancy between the accuracy on the training data $\mathbb{D}_{\text{train}} \subset \mathbb{D}$ and the testing data $\mathbb{D}_{\text{test}} \subset \mathbb{D}$, reflecting the model's ability to generalize from the training data to data unseen over the course of training.

- **Natural Corruption Robustness:** The mean corruption error (mCE) assesses the model's robustness to various natural corruptions and is defined as:

$$\text{mCE} = \frac{1}{N_{\mathbb{C}}} \sum_{\kappa=1}^{N_{\mathbb{C}}} \left[\frac{\sum_{\rho=1}^{5} E_{\rho,\kappa}^{m\odot\theta}}{\sum_{\rho=1}^{5} E_{\rho,\kappa}^{\theta}} \right].$$

  Here, $N_{\mathbb{C}}$ represents the total number of corruption types considered,$\kappa$is used as the variable to iterate over the corruption types. The metric leverages the top-1 error rates $E_{\rho}^{m\odot\theta}$ and E_{ρ}^{θ} for the sparse and dense models, respectively, across a range of corruption types such as noise, blur, weather, digital process applied across input samples x. Each corruption type has five corruption severity levels (i.e., $1 \leq \rho \leq 5$). CIFAR-10/100-C from (Hendrycks and Dietterich, 2019) are adopted in our experiments for corrupted data.

  Type of natural corruptions. (Hendrycks and Dietterich, 2019) provides 19 natural corruptions: "gaussian noise", "shot noise", "impulse noise", "defocus blur", "glass blur", "motion blur", "zoom blur", "snow", "frost", "fog", "brightness", "contrast", "elastic transform", "pixelate", "jpeg compression", "speckle noise", "gaussian blur", "spatter", "saturate". More information can be found in the official repository `https://github.com/hendrycks/robustness/tree/master/ImageNet-C/imagenet_c`.

- **Adversarial Robustness:** The model's robustness against adversarial attacks is gauged by its accuracy on adversarially perturbed images. These perturbations for input samples x are crafted using a targeted Projected Gradient Descent (PGD) attack (Madry et al., 2017), formulated as:

$$x_{n+1} = \Pi_{x+\upsilon}\left(x_n + \epsilon_{\text{adv}} \text{sgn}\left(\nabla_{x_n} L_{\text{CE}}\left(x_n; (m \odot \theta)\right)\right)\right)$$

  where ϵ_{adv} is the magnitude of the perturbations, Π is a projection operator, $\upsilon \in \mathcal{R}^d$ represents the set of perturbations illustrating the model's stability under adversarial conditions. The epsilon value (ϵ) and the number of iterations i employed in this iterative targeted PGD attack are detailed alongside the experimental plots.

- **Out-of-Distribution (OoD) Performance:** The model's ability to make reliable predictions on OoD data is assessed using the ROC-AUC metric. ROC-AUC stands for the area under the receiver operating characteristic (ROC) curve. This evaluation is crucial for comprehending the model's predictive confidence when confronted with data that exhibits statistical characteristics divergent from those present in the training set. Such an understanding is indispensable to ensure reliability in real-world applications where data distribution may constantly change. Following (Hendrycks and Gimpel, 2016; Krizhevsky et al., 2009), for CIFAR-10 experiments, CIFAR-100 is regarded as the OoD dataset; for CIFAR-100 experiments, CIFAR-10 is selected as the OoD dataset.

Figure 16: Juxtaposition of KD-LTS subnetworks on metrics such as Natural Corruption Robustness (mCE), Out-of-Distribution (OoD) performance (ROC-AUC), Expected Calibration Error, Negative Log Likelihood, Fidelity, and Stability, against early pruning, initialization pruning, and dense networks in a ResNet-20/CIFAR-10 environment.

The results related to natural corruption robustness and out-of-distribution (OoD) performance are presented in Figure 16, while the findings on adversarial robustness are comprehensively detailed in Figure 17. It is well-established that sparse networks typically demonstrate superior generalization to distribution shifts, a characteristic often linked to their improved Lipschitz interpretability. Therefore, in our experiments designed to showcase the KD-LTS subnetworks' generalization to distribution shifts, we extend our comparisons beyond dense networks to include sparse networks extracted through existing early and initialization pruning methods. This extension offers a comprehensive understanding and thorough evaluation of their competitive performance.

Key Findings

• **Robustness.** KD-LTS subnetworks demonstrate a significant 3-4% improvement in Mean Corruption Error (mCE), a 1-2% increase in ROC-AUC, and a 2-3% enhancement in adversarial robustness compared to subnetworks derived through pruning at initialization. Furthermore, they attain performance comparable to subnetworks extracted by early pruning methods, highlighting KD-LTS's superior efficacy across various sparsity levels.

How does KD-LTS assert generalization to distribution shifts? Figure 17 showcases sample classifications of KD-LTS winning tickets on adversarial examples, demonstrating how misclassification only occurs when the features of the sample appear closer to the adversarial target class compared to random subnetworks and the dense network. Figures 24, 25, and 18 delineate the robust decision-making capabilities of KD-LTS subnetworks against random subnetworks and the dense network in a LeNet-300-100/MNIST experiment, as examined through the lens of the LIME explainer (Ribeiro et al., 2016). This serves to substantiate their inherent robustness. Figures 26 and 19 focus on the activations of the penultimate layer in KD-LTS, dense, and random subnetworks. In KD-LTS subnetworks, our experiments reveal that projections form well-defined, tight clusters indicating clear class boundaries. In contrast, for dense and random subnetworks, these projections appear more scattered and less discernible, highlighting the superior adaptability of KD-LTS subnetworks to distribution shifts.

C.2 Calibration and Reliability

Confidence calibration (Guo et al., 2017; Venkatesh et al., 2020) is of paramount importance to ensure reliability in situations where incorrect predictions could lead to serious consequences. To evaluate the calibration of confidence in the predictions made by KD-LTS subnetworks, we adopt the following standard metrics:

Definition C.2 (Confidence)**.** Let $f_L^{\mathrm{s}}(x; m \odot \theta)$ denote the prediction of a pruned student model for a given input x from a dataset $\mathbb{D}$. The confidence of this prediction is defined as the predicted probability that the model assigns to the most likely class. Formally, for a dataset $\mathbb{D}$, the average confidence over the set of predictions is given by:

$$\text{Confidence} = \frac{1}{I} \sum_{x \in \mathbb{D}} \frac{e^{f^{\mathrm{s}}_{c^* L}(x; m \odot \theta)}}{\sum_{c=1}^{C} e^{f^{\mathrm{s}}_{cL}(x; m \odot \theta)}},$$

where c^* represents the class with the highest probability predicted by the model for the input x.

Leveraging both accuracy and confidence, we present explanations of the following metrics associated with confidence calibration.

- **Expected Calibration Error (ECE)**: ECE (Naeini et al., 2015) is defined as the weighted average of the differences between accuracy and confidence across Z bins:

$$\text{ECE} = \sum_{z=1}^{Z} \frac{N_{B_z}}{I} \left| \text{Accuracy}(B_z) - \text{Confidence}(B_z) \right|,$$

  where I represents the number of samples in the dataset $\mathbb{D}$ and N_{B_z} is the number of samples in the z-th bin. This metric serves as a proxy for the mismatch between the model's confidence and its actual performance.

Figure 17: Sample classification of adversarial examples in an MNIST data perturbed using a targetted PGD attack (target label '3') by a dense LeNet-300-100, KD-LTS extracted 45% sparse winning ticket, and 45% sparse randomly pruned subnetwork.

- **Negative Log Likelihood (NLL)**: NLL is a measure of how well the log probability distribution outputted by the model, aligns with the true labels:

$$\text{NLL} = -\sum_{(x,y)\in\mathbb{D}_{\text{val}}} y^{\mathsf{T}} \log\left(\mathcal{P}(z|x)\right),$$

  where $\mathbb{D}_{\text{val}}$ is a validation subset of the dataset $\mathbb{D}$. NLL (LeCun et al., 2015) is minimized when the predicted distribution converges to the true distribution of the labels, indicating optimal calibration.

The results for ECE and NLL in the ResNet-20 experiment on CIFAR-10 are illustrated in Figure 16. Similarly, the outcomes of these metrics for the CIFAR-100 experiment are depicted in Figure 32.

Key Findings

• **Reliability.** KD-LTS subnetworks demonstrate the ability to maintain or enhance confidence levels compared to dense and early-pruned subnetworks, across a range from denser 20% to rarer 60% sparsity levels. This attribute renders KD-LTS subnetworks viable for reliable deployments in real-world settings.

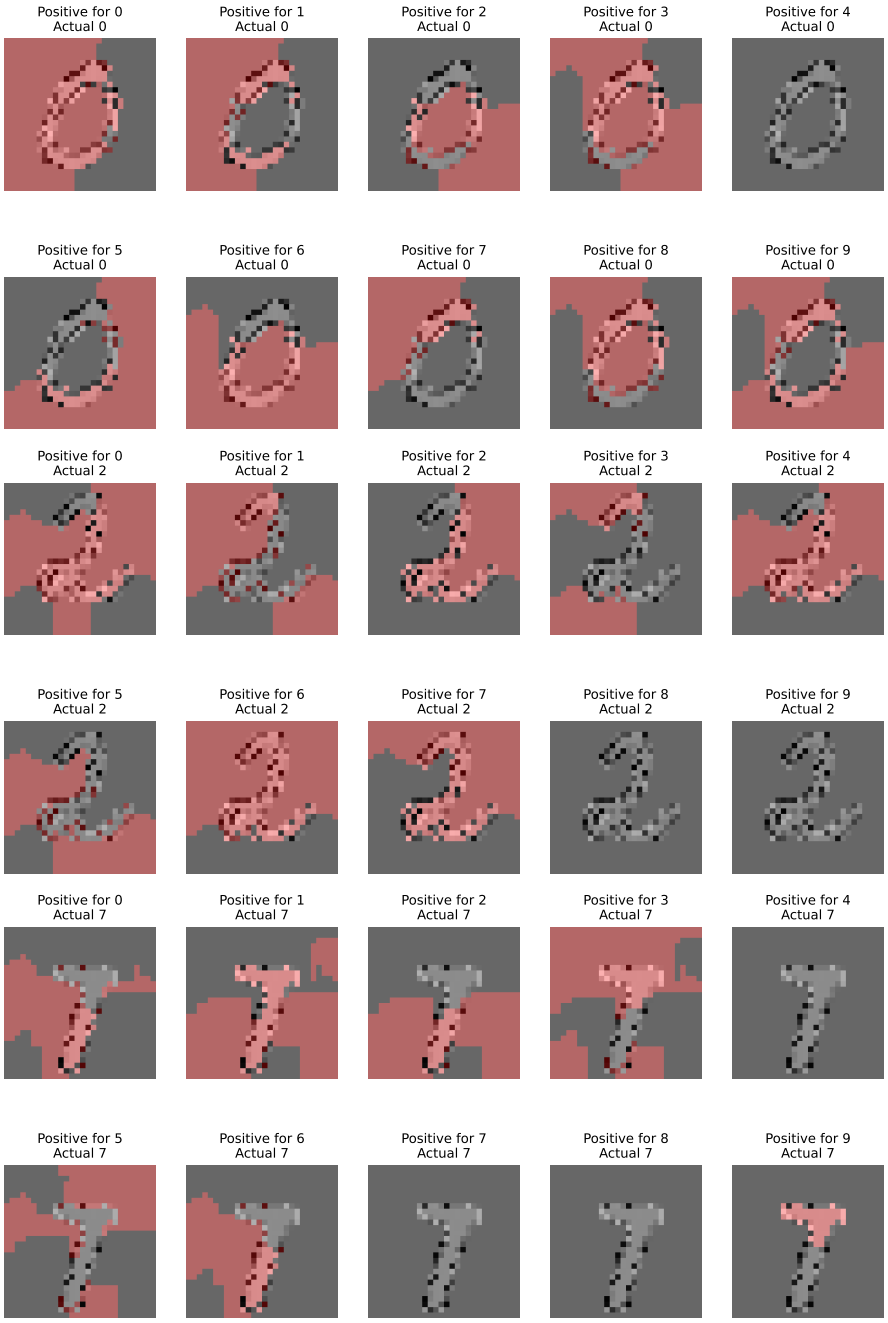

Figure 18: Explaining the classifier: A KD-LTS subnetwork, unpruned network, and a random subnetwork in a LeNet-300-100 in terms of positive regions associated with classification of a ground truth label against positive regions for misclassification.

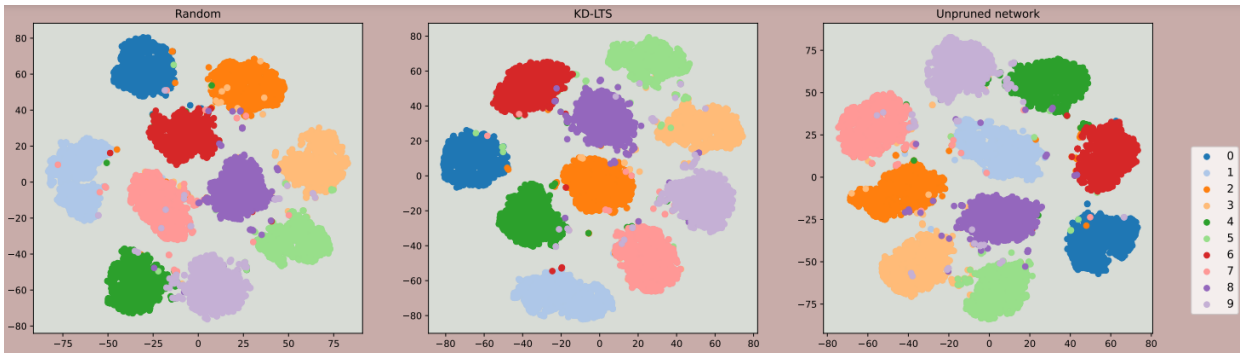

Figure 19: Visualization of penultimate layer's activations across the ten classes in KD-LTS subnetwork, random subnetwork, and unpruned VGG-16 network trained on CIFAR-10.

C.3 Interpretability

We investigate the capability of winning tickets identified by KD-LTS to preserve interpretability. The assessment involves a quantitative analysis of explainability, grounded in functional representation, employing the most widely-recognized metrics: namely, *fidelity* (Plumb et al., 2020) and *stability* (Ghorbani et al., 2019).

- **Fidelity**: Given a validation dataset D_{val}, fidelity quantifies the extent to which the local explanations of a model's predictions align with the actual outcomes of the model. Formally, it is defined as the expected difference in the model's predictions for the original input and a perturbed input within a defined neighborhood, and is given by:

$$\text{Fidelity} = \mathbb{E}_{x \in D_{\text{val}}} \left[\mathbb{E}_{x' \sim \mathcal{N}_x} \left[|g(x') - f(x')|^2 \right] \right],$$

  where $g(x')$ represents the explanation of model's prediction, and $f(x')$ denotes the target model's prediction for the perturbed input x'.

- **Stability**: Stability S focuses on the consistency of local explanations under input perturbations, reflecting the robustness of the explanation model. It is defined as:

$$\text{Stability} = \mathbb{E}_{x \in D_{\text{val}}} \left[\mathbb{E}_{x' \sim \mathcal{N}_x} \left[\|\mathbf{e}(x, f) - \mathbf{e}(x', f)\|_2^2 \right] \right],$$

  where $\mathbf{e}(x, f)$ and $\mathbf{e}(x', f)$ are the embeddings of the explanations for the original and perturbed inputs, respectively.

Following Chen et al. (2022), to compute these metrics, we first perturb each input image x and build its neighborhood set $\mathcal{N}_x$ with a size of $1,000$ samples. Then, we generate a class of interpretable functions $G = \{g_x \in G | x \in D_{\text{val}}\}$, where g_x is a linear function obtained from a regression to the corresponding model's output on $\mathcal{N}_x$. In the above formulation, $f(\cdot)$ denotes the target model we want to interpret, $\mathbf{e}(x, f)$ and $\mathbf{e}(x', f)$ are the learned weights of linear models g_x and $\tilde{g}_x$. Each training sample $\tilde{x} \in \mathcal{N}_x$ is weighted by the Hamming distance of $(\tilde{x}, x)$ and $(\tilde{x}, x')$, respectively. All our experiments replicate the implementation detailed in (Ribeiro et al., 2016). Intuitively, fidelity quantifies how accurately the explainer g_x models the target network f in a neighborhood $\mathcal{N}_x$; Stability measures the degree to which the explanation changes across points in $\mathcal{N}_x$.

The stability and fidelity results from the ResNet-20/CIFAR-10 experiment are presented in Figure 16, while the corresponding results for the CIFAR-100 experiment are shown in Figure 32

Key Findings

• **Stability and precision.** KD-LTS winning tickets showcase a 2-5% improvement in stability and fidelity compared to LTH-Rewind, reaffirming our claim that KD-LTS is not just an alternative but a viable replacement for early pruning techniques, ensuring both software and hardware cost reduction.

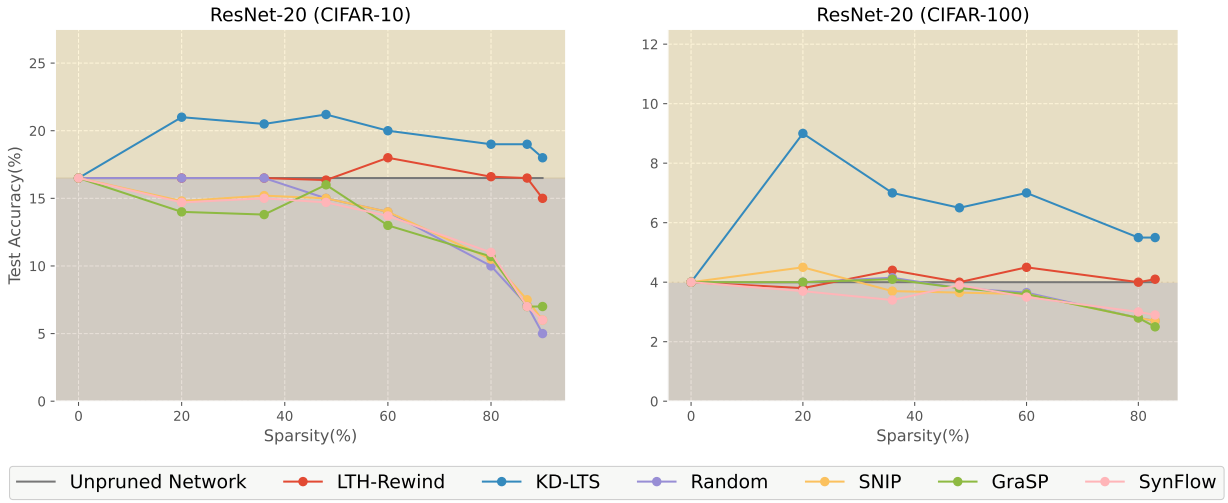

Figure 20: Adversarial accuracy of winning tickets obtained in a ResNet-20 network across datasets CIFAR-10 (left) & CIFAR-100 (right).

C.4 Transferability

We expand on the conjecture proposed in (Morcos et al., 2019; Chen et al., 2021b) regarding the transferability of winning tickets across various datasets, optimizers, and architectures. While LTH-Rewind has been the dominant method for identifying transferable winning ticket, our research re-evaluates and broadens this concept to extract transferable subnetowrks right at the initialization stage. Our empirical findings clearly demonstrate that winning tickets identified through the KD-LTS framework are indeed transferable across different datasets, optimizers, and architectures. Notably, they even exhibit performance that is comparable, and in most instances, superior to those identified by LTH-Rewind experiments.

Key Findings

- **Versatility.** KD-LTS subnetworks show an improvement of 0.6%, 0.7%, and 0.42% in performance when transferred across datasets (CIFAR-10 ← CIFAR-100 and CIFAR-100 ← CIFAR-10), optimizers (Adam ← SGD), and architectures (ResNet-14 ← ResNet-20), respectively, compared to LTH-Rewind transferability.
- **Universality.** The results underscore the existence of shared and transferable patterns in DNNs that make sparse networks trainable, not just from early epochs but right from the initialization stage. Furthermore, they demonstrate the KD-LTS framework's capability in identifying these patterns.

In summary, KD-LTS subnetworks, extracted right from the onset of training, not only enjoy superior test performance but also provide sparse networks that are accurate, transferable, interpretable, reliable, secure, and confident.

D Supermasks

Definition D.1 (Supermask)**.** A supermask is a subnetwork, derived from a randomly initialized neural network, represented as $f_L(x; m \odot \theta^{(0)})$, capable of achieving a test accuracy surpassing the chance accuracy without the need for training the underlying parameters. Here, the term chance accuracy refers to the test accuracy achieved by the original, untrained dense network, denoted as $f_L(x; \theta^{(0)})$.

Recent theoretical studies (Malach et al., 2020; Orseau et al., 2020; Ramanujan et al., 2020) have substantially strengthened the proposal of LTH. These works suggest that a sufficiently overparameterized neural network contains a subnetwork that, with only logarithmic increase in size compared to the target network, can achieve comparable accuracy to the large network - even without any training. In response to these groundbreaking

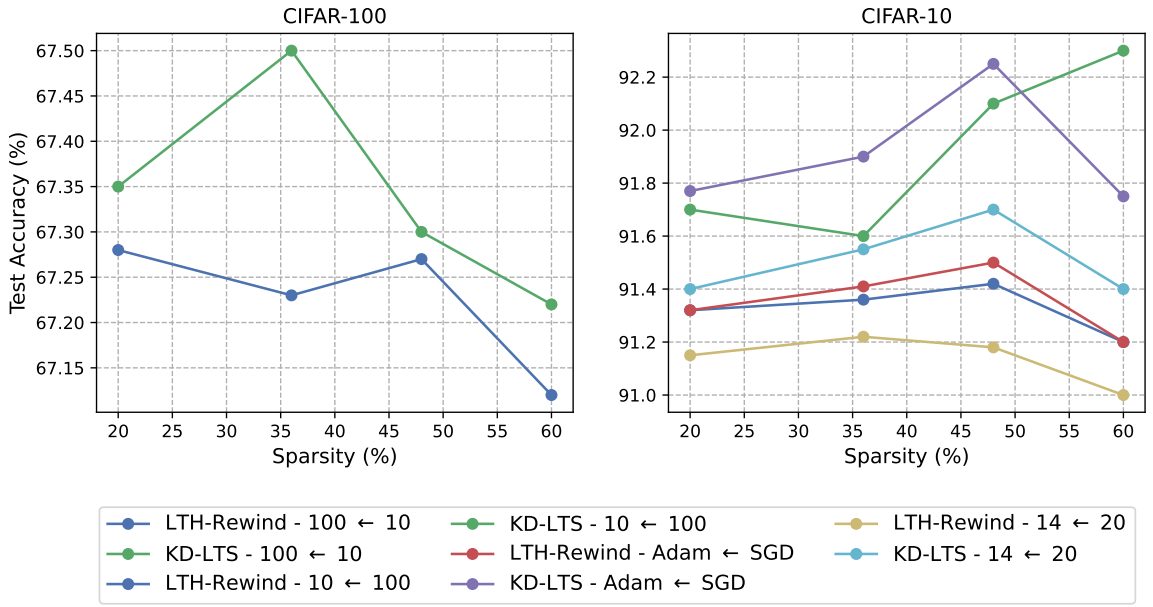

Figure 21: Test accuracy of winning tickets transferred across datasets (CIFAR-10 & CIFAR-100), architectures (ResNet-20 & ResNet-14) and Optimizers (Adam & SGD).

insights on LTH, a considerable amount of experimental research has been conducted. Efforts include CS (Savarese et al., 2019), Stochastic Sparsification (SS) (Zhou et al., 2019), Edge Popup (EP), Smart Ratio (Su et al., 2020), and GEM-MINER (Sreenivasan et al., 2022), all aimed at developing algorithms for identifying *supermasks*. This burgeoning area of research underscores the latent potential of subnetworks at initialization.

These theoretical results indicate that pruning alone might suffice as a training method. This challenges the conventional belief that binary mask training, perceived as having lesser representational power compared to continuous parameter training, cannot yield comparable outcomes. Supermasks confer a notable advantage in terms of storage efficiency, necessitating only the binary mask and a single random seed for the reconstruction of the entire network's weights. KD-LTS further strengthens this characteristic, obtaining high-quality supermasks that can be harnessed for their own line-up of applications in real-world scenarios.

Exploration into a novel avenue of knowledge distillation. KD-LTS paves the way for various avenues of exploration, extending beyond pruning to include a pioneering premise in knowledge distillation. Unlike traditional knowledge distillation methods that transfer knowledge from larger teacher models to smaller student networks via weight training, KD-LTS involves mask-training large student networks using heterogeneous information from teacher networks. The pruned larger networks, known as supermasks, exhibit exceptional standalone performance without any supplementary training. These supermasks are not only receptive to further weight-training from scratch but also exceed the accuracy levels and parameter efficiency of dense, small student models trained through conventional knowledge distillation methods. For example, consider the case of ResNet-50, with its approximately 24 million parameters. Utilizing KD-LTS for pruning at initialization, we obtain a subnetwork with 90% sparsity that exceeds the performance of a fully-trained ResNet-18, which has around 12 million parameters and is trained using standard knowledge distillation techniques. This achievement is particularly remarkable, considering that the KD-LTS-identified subnetwork benefits from an 80% reduction in parameter count. Notably, our knowledge distillation framework highlights that these winning tickets not only achieve a notable 3.4% improvement in test accuracy but also demonstrate that supermasks, even without any additional training, experience only a marginal 3% reduction in accuracy compared to their traditional knowledge distillation counterparts.

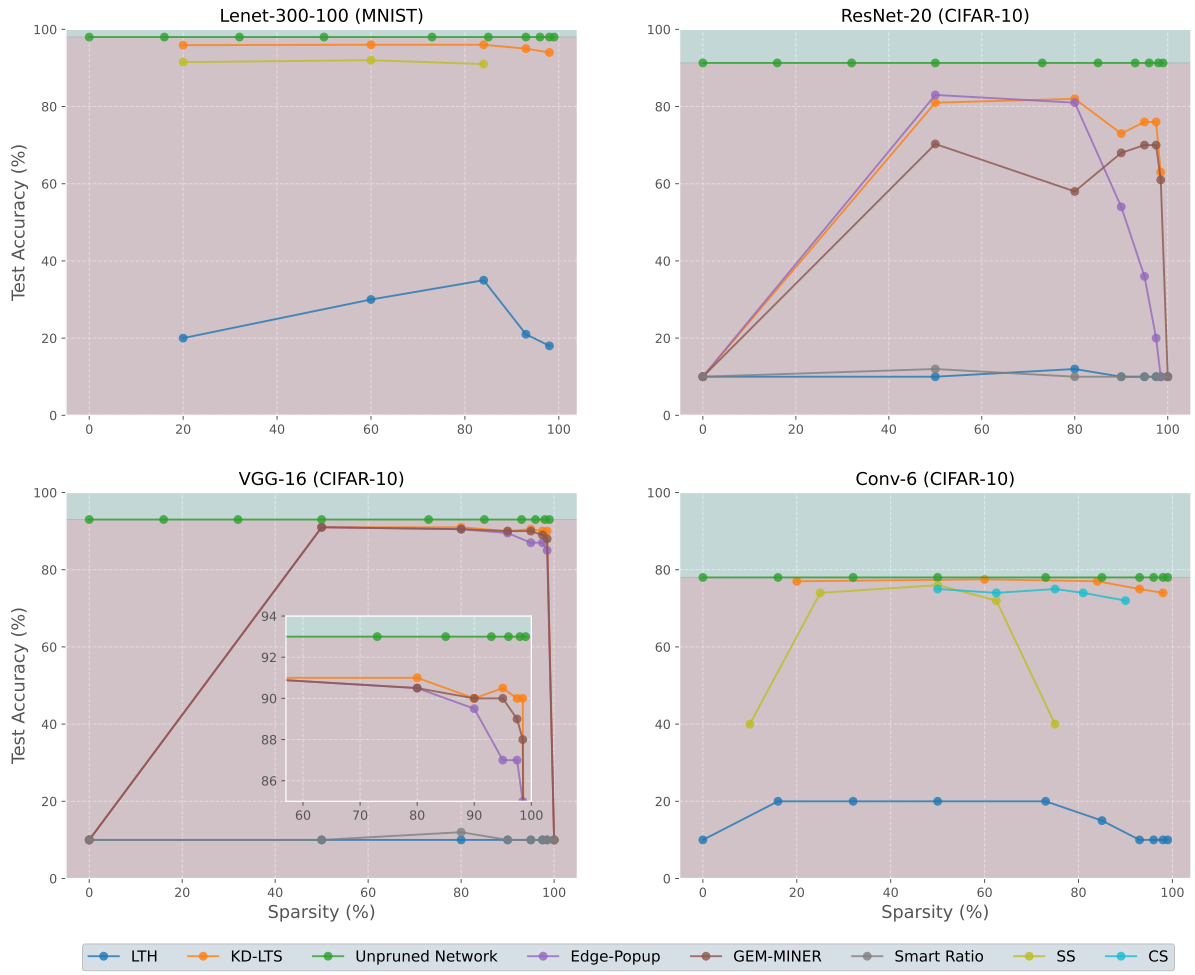

Figure 22: Evaluating the supermask performance of KD-LTS in comparison to CS, SS, GEM-MINER, Edge-Popup, Smart Ratio, and LTH.

Key Findings

• **Test efficiency.** Among the comparative techniques, IMP and SR produce supermasks with accuracy levels barely surpassing random guessing, SS struggles to learn masks with sparsity exceeding 50%, and EP identifies subnetworks with high pre-training accuracy but faces limitations in the high sparsity regime, particularly beyond 95%. In stark contrast, KD-LTS method stands out, with its highest-performing supermasks achieving 94%, 66.4%, 90.1%, and 74.4% test accuracy at 98% sparsity for LeNet-300-100, ResNet-20, VGG-16, and a 6-layer CNN, respectively. These results significantly surpass the leading results presented in the current literature using other techniques, which are 60%, 61.5%, 88.2%, and 68.2%, establishing a marked improvement over existing methods. • **Unexplored arena.** KD-LTS not only pioneers new avenues in the field of neural network pruning at initialization but also offers a novel perspective on knowledge distillation, significantly contributing to the realm of knowledge distillation-assisted model compression.

E Future Work

Acknowledging room for improvement within KD-LTS, the key areas for our future investigation include: • Extend KD-LTS beyond computer vision to diverse domains such as natural language processing. • Investigate the transferability of winning tickets across applications for broader impact, surpassing the

current focus on optimizers, networks, and datasets. • Integrate KD-LTS to identify subnetworks specifically optimized for modern libraries and hardware, achieving tangible computational reductions. • Examine the structure, patterns, and characteristics of KD-LTS identified subnetworks to investigate their efficacy and potential as a powerful initialization technique. (**Intriguing Observation:** We observed that KD-LTS, by transferring information on the SVD of parameters (KD+SVD+CE), identified winning tickets achieving an adversarial accuracy of 79% after training. This contrasts with subnetworks identified using cross-entropy and label information alone, which achieved 63% adversarial accuracy. Importantly, both sets of subnetworks underwent only standard training with standard data. This demonstrates the surprising capability of a pruning-at-initialization technique like KD-LTS to embed specialized traits, such as adversarial accuracy, using only standard data - traits previously thought to be achievable solely through computationally intensive training methods, such as adversarial training with adversarial data). • Develop effective techniques for efficient sparsity control within the Phase-3 of KD-LTS.

F Additional experiments

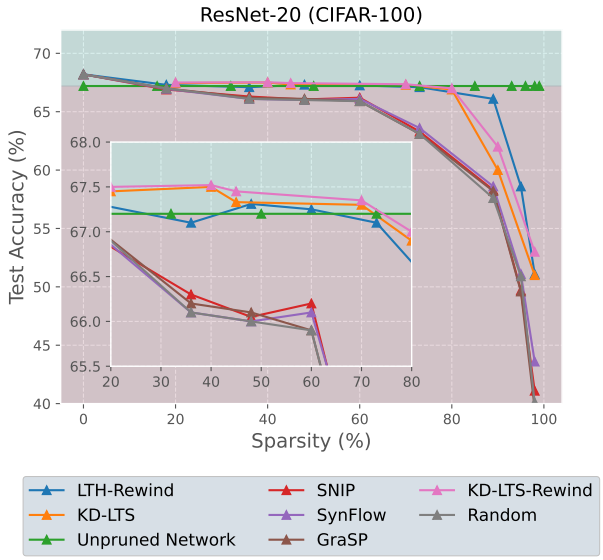

Figure 23: KD-LTS' winning ticket performance, extracted at initialization and early epoch.

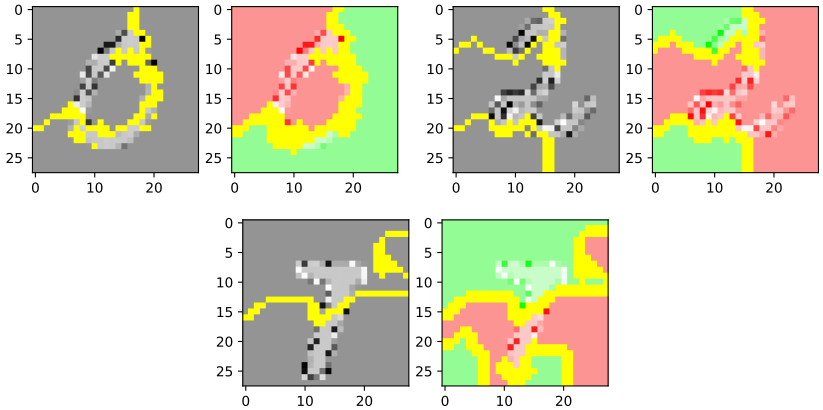

Figure 24: Highlighting the positive/negative regions for classifications of MNIST data by trained KD-LTS subnetwork, unpruned network and a random subnetwork in a LeNet-300-100.

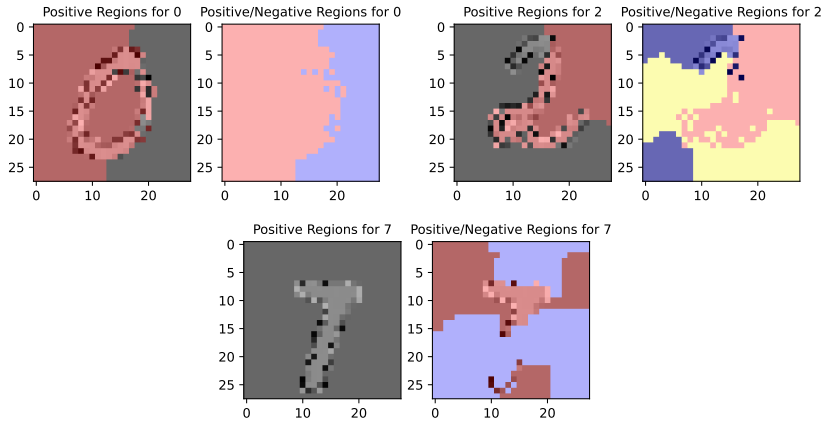

Figure 25: Explaining the positive/negative regions for classifications of MNIST data by trained KD-LTS subnetwork, unpruned network and a random subnetwork in a LeNet-300-100.

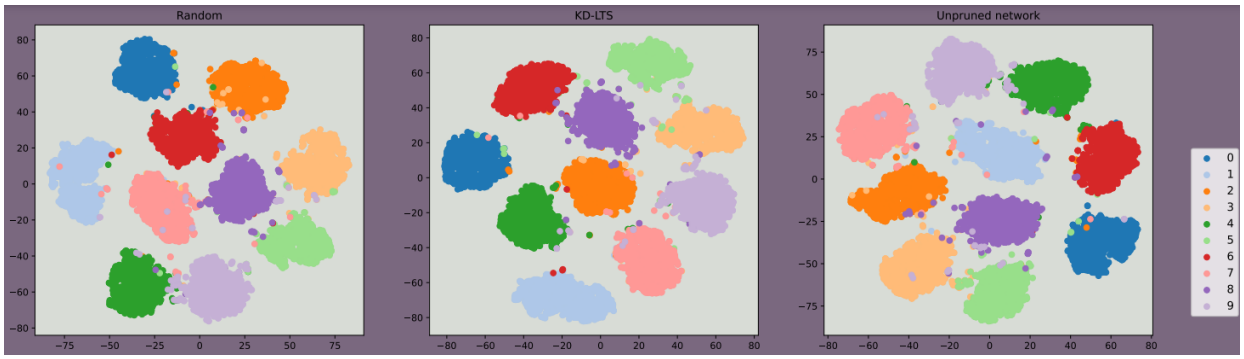

Figure 26: Visualization of penultimate representations in KD-LTS subnetwork, Random subnetwork, and Unpruned VGG-16 network trained on CIFAR-10.

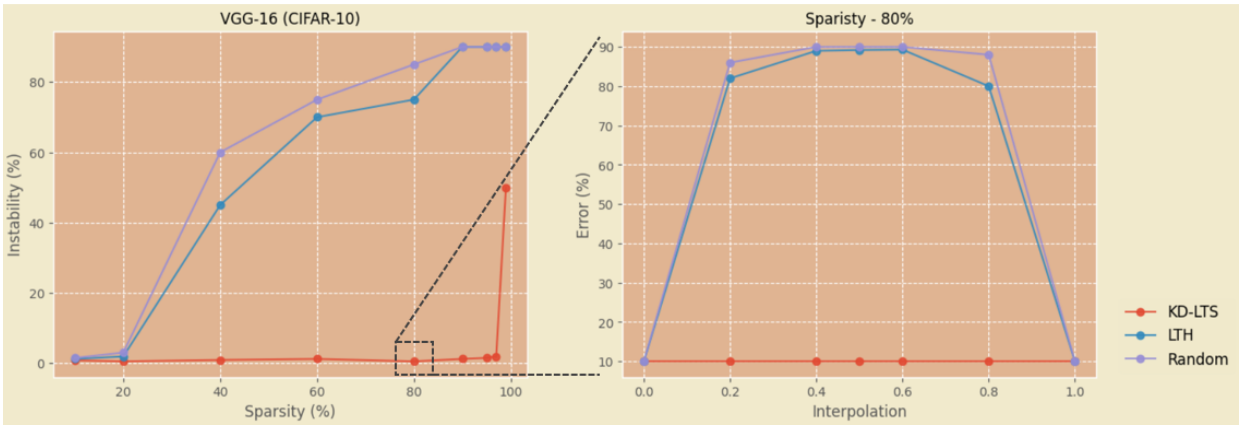

Figure 27: Evaluating stability indicators in LTH, random pruning, and KD-LTS at diverse sparsity levels in VGG-16/CIFAR-10 experiment and examining linear mode connectivity at non-trivial sparsity levels.

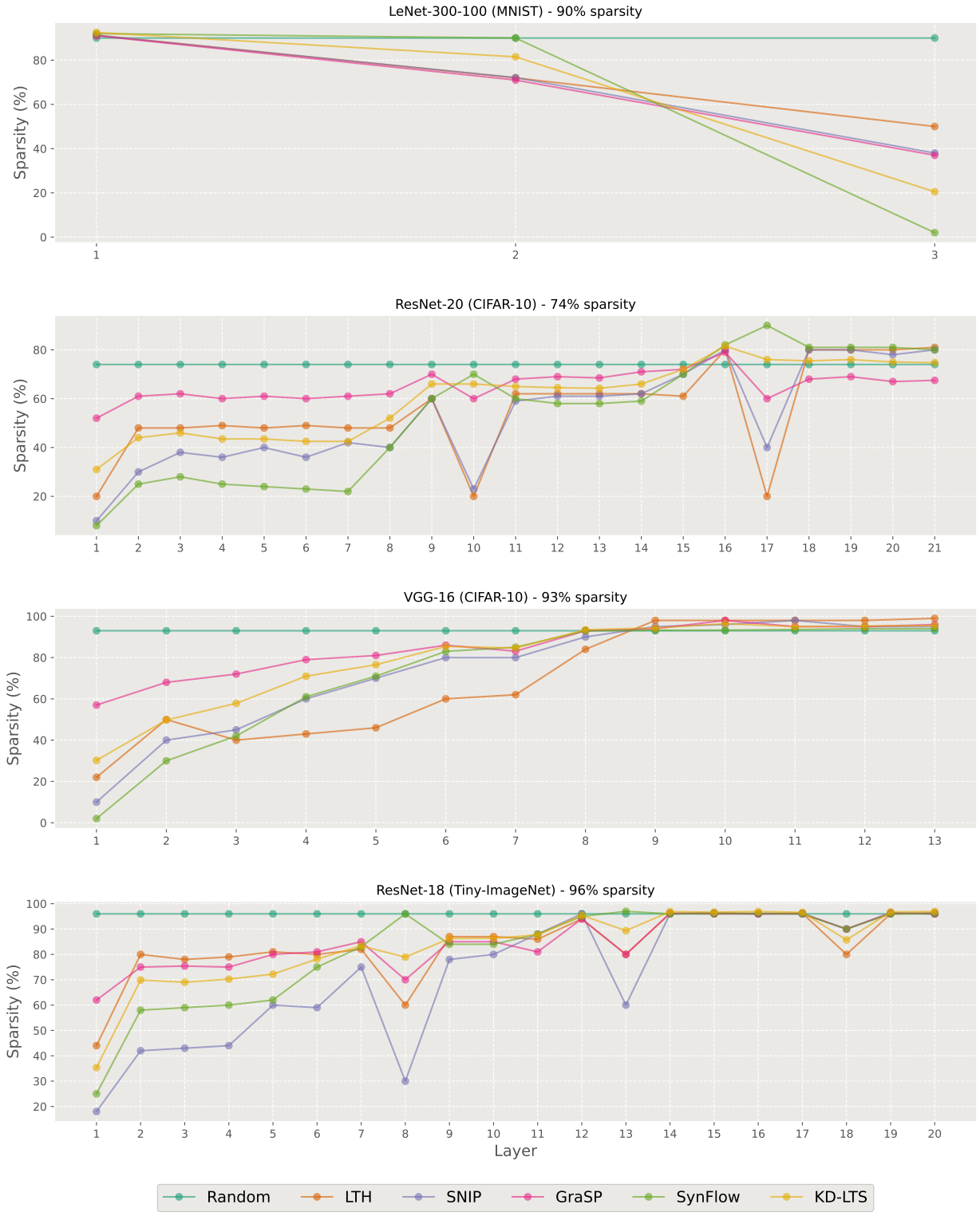

Figure 28: Per-layer sparsities achieved by each pruning at initialization method for a specified sparsity level across various experiments.

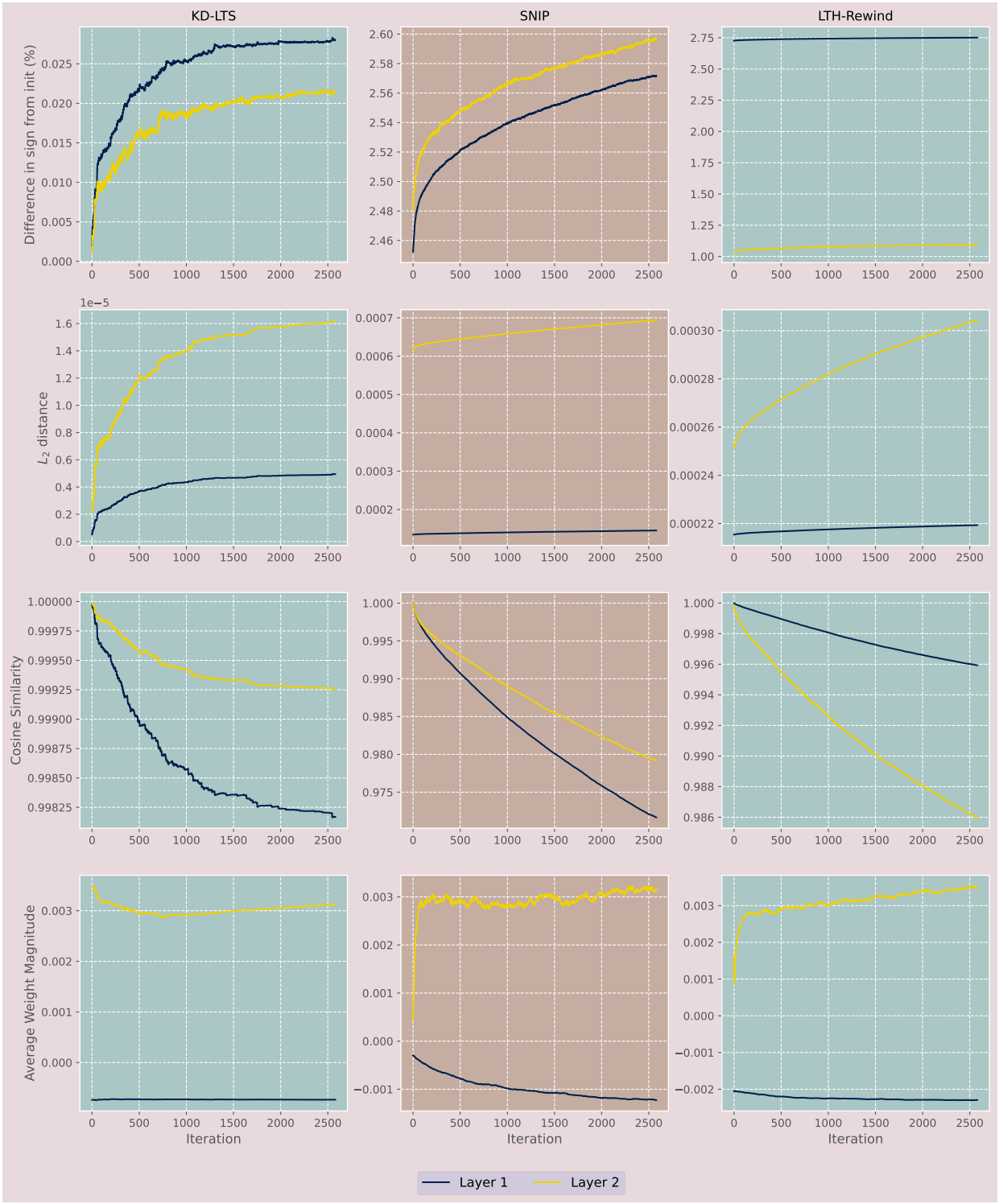

Figure 29: Training dynamics of weight parameters analyzed through the lens of pivotal metrices in a LeNet-300-100 network across the first 2580 iterations.

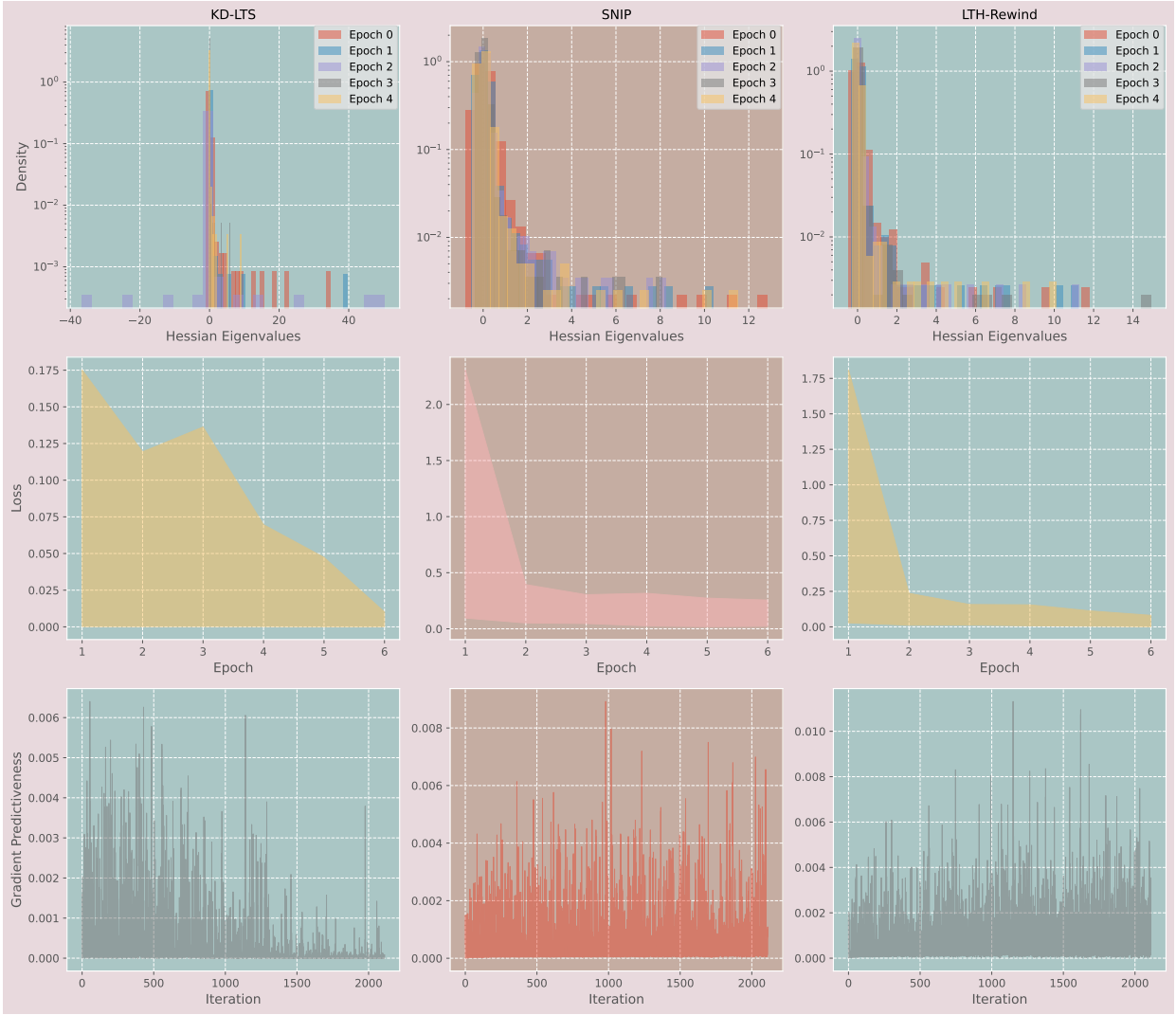

Figure 30: Analyzing the optimization landscape in LeNet-300-100 subnetwork trained on MNIST over the first six epochs.

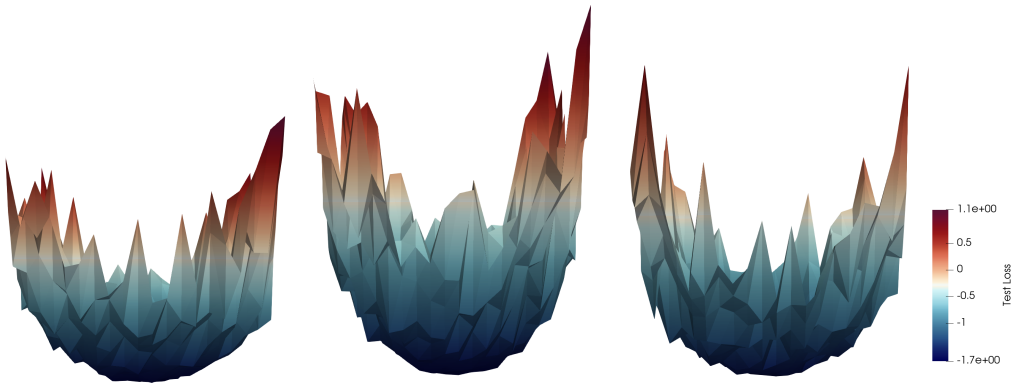

Figure 31: Visualizing the Loss landscape: Comparing LeNet-300-100 subnetworks at the optimum extracted via KD-LTS, Random, and LTH-Rewind, trained on the MNIST dataset

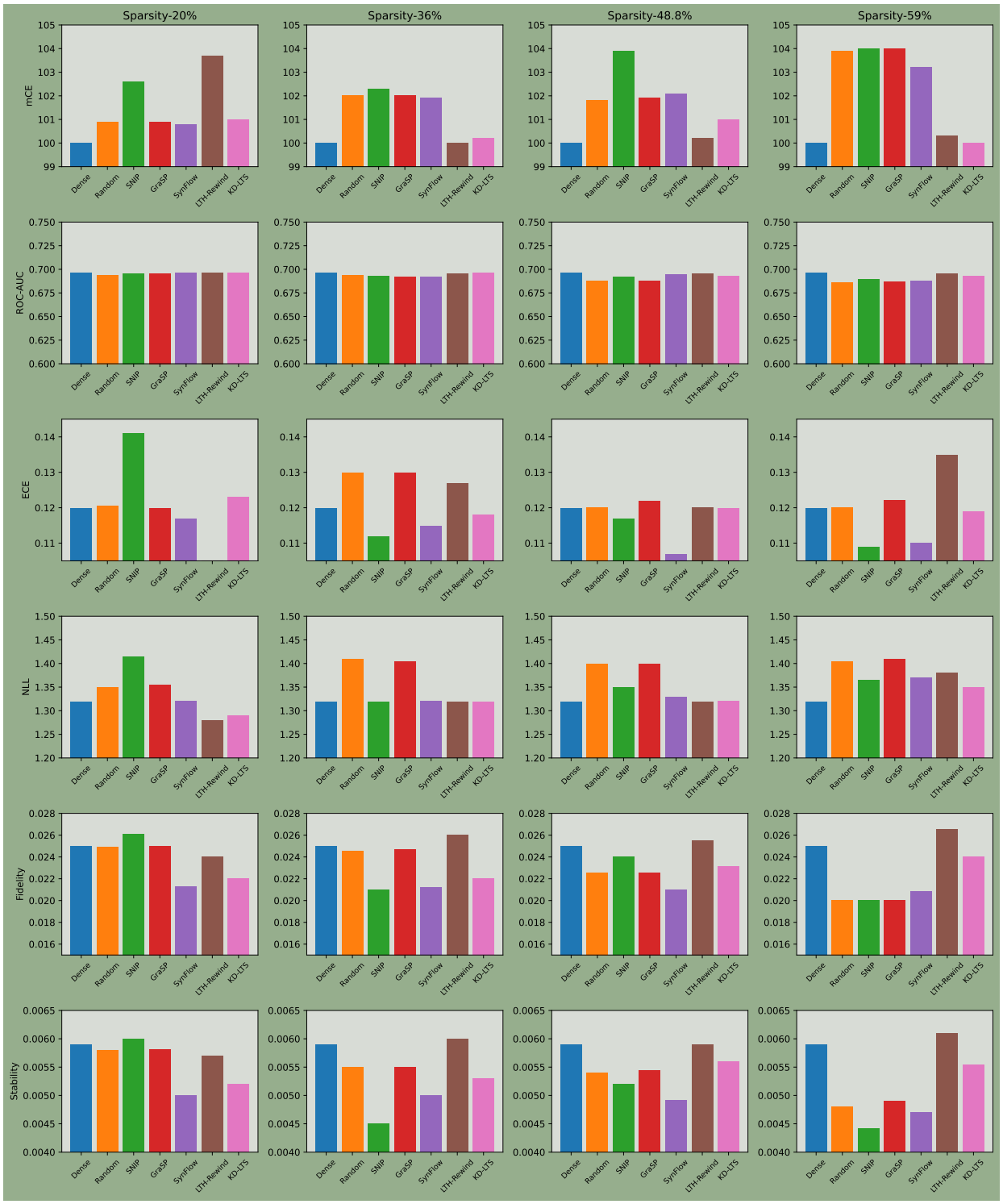

Figure 32: Comparative analysis of KD-LTS subnetworks on Key Metrics: Natural Corruption Robustness (mCE), Out-of-Distribution performance (ROC-AUC), Expected Calibration Error, Negative Log Likelihood, Fidelity, and Stability against early pruning, initialization Pruning, and dense networks in a ResNet-20/CIFAR-100 environment.

G Experimental Setup

This section provides a detailed description of the experimental settings and configurations utilized to evaluate the effectiveness of proposed KD-LTS technique in detecting winning tickets at initialization.

G.1 Datasets

Experiments are conducted on four well-established datasets, each possessing unique characteristics, to comprehensively evaluate winning tickets. For the MNIST database, 55,000 examples were utilized for training and 5,000 were reserved for validation, with images normalized on a per-channel basis. In the case of CIFAR-10 and CIFAR-100 datasets, per-channel normalization, random horizontal flipping, and random pixel shifts of up to four pixels in any direction were applied. The training sets were partitioned into 45,000 training examples and 5,000 validation examples. The model demonstrating the highest validation accuracy was chosen for further evaluation on the test set. For the Tiny-ImageNet dataset, augmentation included per-channel normalization, selection of a patch with a random aspect ratio between 0.8 and 1.25 and a random scale between 0.1 and 1, cropping to 224x224, and applying random horizontal flipping.

G.2 Network Architectures

Four distinct neural network models were deployed, carefully chosen to underscore the versatility and robustness of the approach. This selection includes architectures suitable for both small-scale and large-scale settings, aligning with classifications from the work of Frankle et al. (2020a) and further informed by our research considerations. The first model, LeNet-300-100, is characterized by its 266K parameters. Progressing to a larger scale, VGG-16, featuring 14.7M parameters, is specifically utilized in the CIFAR-10/CIFAR-100 experiments. ResNet-20, introduced for the CIFAR-10/100 datasets with 20 layers and 274K parameters in (He et al., 2016), adds diversity to the selection. Lastly, ResNet-50, tailored for the ImageNet dataset with 50 layers and 25.5M parameters, further emphasizes the range of architectures in the experimental framework. The implementations of ResNet-20 and ResNet-50 were sourced from the OpenLTH repository, while the LeNet-300-100 and VGG-16 networks were leveraged from the TorchVision implementations.

G.3 Training Settings

Table 4 offers a comprehensive overview of the training settings used in experiments to assess both KD-LTS and other comparative techniques for extracting subnetworks, both at initialization and from early epochs.

Table 4: Training Settings

Network	Dataset	Epochs	Batchsize	Iterations per epoch	Rewind Iteration
LeNet-300-100	MNIST	40	256	469	1407
ResNet-20	CIFAR-10/100	180	128	391	1173
VGG-16	CIFAR-10/100	180	128	391	1173
ResNet-50	ImageNet	90	1024	1251	3753

Tables 5 and 6 detail the optimizer settings for the post-extraction training of subnetworks and mask training, respectively.

Table 5: Optimizer settings for training weight and bias parameters

Network	Batchsize	Optimizer	Momentum	LR	LR Drop	Weight decay	Initialization
LeNet-300-100	MNIST	SGD	-	0.01	-	-	Orthogonal
ResNet-20	CIFAR-10/100	SGD	0.9	0.1	10x at 90, 135	2e-4	Orthogonal
VGG-16	CIFAR-10/100	SGD	0.9	0.1	10x at 80, 120	2e-4	Orthogonal
ResNet-50	ImageNet	SGD	0.9	0.4	10x at 30, 60, 80	1e-4	Orthogonal

Table 6: Optimizer settings for training mask parameters

Network	Dataset	Optimizer	Momentum	LR	LR Drop	Weight decay	Initialization
LeNet-300-100	MNIST	SGD	-	0.1	-	-	Constant
ResNet-20	CIFAR-10/100	SGD	0.9	1	10x at 90, 135	-	Constant
VGG-16	CIFAR-10/100	SGD	0.9	1	10x at 80, 120	-	Constant
ResNet-50	ImageNet	SGD	0.9	1	10x at 30, 60, 80	-	Constant

Next, we present the hyperparameters for our knowledge distillation framework in Table 7. These are the general settings used across all our experiments unless otherwise stated. The hyperparameters for response-based, feature-based, and relational-based knowledge distillation techniques mentioned in the ablation studies are specified in the respective sections.

Table 7: Hyperparameters for the knowledge distillation framework

Network	Dataset	α	ω	$\beta^{(0)}$	$\beta^{(E)}$	k	ν_k	T
LeNet-300-100	MNIST	0.9	0.2	1	60	2	0.5	10
ResNet-20	CIFAR-10/100	0.9	0.6	0.1	99	3	0.34	2
VGG-16	CIFAR-10/100	0.9	0.2	0.1	300	3	0.34	30
ResNet-50	ImageNet	0.7	0.6	0.1	99	3	0.34	2

H Benchmarking Sparsification Techniques: A Comparative Study

We conduct a comprehensive evaluation of multiple baseline methods and various state-of-the-art sparsification techniques to demonstrate the advantages of our approach in extracting supermasks, conducting pruning at initialization, and performing early training pruning. Additionally, we carry out a series of sanity checks, sensitivity analyses, and stability assessments across the strategies to distinguish the quality of our winning tickets. In this section, we briefly describe the key features of these juxtaposed techniques and the critical parameters in their implementation.

Lottery Ticket Hypothesis: Frankle and Carbin (2018) adopted and customize the popular IMP to an iterative process to obtain subnetworks at initialization. We refer to the standard LTH procedure that involves resetting the weights to their random initial values, as 'LTH' in our plots. As an alternative, when the weights are rewound to an earlier epoch, we denote it as 'LTH-Rewind'. Throughout all LTH-Rewind experiments presented in this study, we rewind the weights specifically to the 3rd epoch (unless otherwise specified). For our experiments, the LTH implementation from the OpenLTH repository is employed. Also, to ensure the functionality of the pruned subnetwork, its weights are proportionally scaled to account for the fraction of weights pruned, as described in Algorithm 2.

Algorithm 2: LTH

1: Randomly initialize a neural network $f_L^s(x; (m \odot \theta^{(0)}))$ where $m = \mathbf{1}$: m is a vector of all-ones with the same dimension as $\theta^{(0)}$.
2: Train the network for E epochs with an optimal learning rate to minimize L_{CE} and obtain $f_L^s(x; (m \odot \theta^{(E)}))$.
3: Prune 20% of the parameters with the least absolute magnitude $|\theta^{(E)}|$, creating an updated mask m'. Here, $|\cdot|$ represents element-wise application of absolute value.
4: Reset the weights of the remaining portion of the network to their values in $\theta^{(0)}$ for 'LTH' experiments and to their values in $\theta^{(3)}$ for 'LTH-Rewind' experiments.
5: Rescale the values of weights based on the sparsity of the network: $\theta^{(0)} \leftarrow \theta^{(0)}/(1 - \frac{\|m'\|_0}{P})$ for 'LTH' experiments and $\theta^{(3)} \leftarrow \theta^{(3)}/(1 - \frac{\|m'\|_0}{P})$ for 'LTH-Rewind' experiments.
6: Let $m = m'$ and repeat steps 2 through 4 until a sufficiently pruned network has been obtained.

SNIP (Single-shot Network Pruning): SNIP, introduced by Lee et al. (2018a), is another pruning at-initialization technique designed to calculate importance scores based on gradients, preserve weights with the most significant impact on the loss function, and thereby selectively eliminate redundant connections in a data-dependent manner. In the standard implementation of SNIP, all of those operations are executed in a

single iteration. To facilitate a fair comparison, we consider a recently introduced iterative variant of SNIP proposed by de Jorge et al. (2020). In our experiments, we utilize the implementation of this iterative SNIP available at `https://github.com/hoonyyhoon/Synflow_SNIP_GraSP`. Also, with 100 iterations performed, weights are gradually pruned in accordance with an exponential schedule as proposed in (de Jorge et al., 2020), ensuring that the final desired level of sparsity is achieved at the 100th iteration.

GraSP (Gradient Signal Preservation): GraSP, introduced by Wang et al. (2020), calculate the Hessian-gradient product for each layer to selectively eliminate weights that diminish gradient flow and prunes weights whose removal results in the least decrease in the gradient norm. We utilize the code available at `https://github.com/hoonyyhoon/Synflow_SNIP_GraSP` for the implementation of GraSP, and adopt the same strategy as implemented by (Wang et al., 2020) for creating the mini-batch used to compute GraSP scores in CIFAR-10 and CIFAR-100 experiments. Specifically, we randomly sample ten examples from each class, aligning with their approach for both CIFAR-10 and CIFAR-100 experiments.

SynFlow: SynFlow, proposed by Tanaka et al. (2020), is a data-agnostic pruning at initialization algorithm designed to preserve the total flow of synaptic strengths through the network and achieve Maximum Critical Compression. Our implementation of SynFlow in our experiments follows the strategy outlined in the repository `https://github.com/hoonyyhoon/Synflow_SNIP_GraSP`. To streamline the process, we conduct 100 iterations, mirroring the approach employed by (Tanaka et al., 2020)

Algorithm 3: Iterative Synaptic Flow Pruning (SynFlow)

Randomly initialize a neural network $f_L^{\mathrm{s}}(x;(m \odot \theta^{(0)}))$ where $m = \mathbf{1}$.
for n in N **do**
  Replace all parameters $\theta^{(0)}$ with their absolute values $|\theta^{(0)}|$.
  Sample a mini-batch of training data $\mathbb{D}_b = \{(x_i, y_i)\}_{i=1}^{b} \subset \mathbb{D}$.
  Forward propagate an input of all ones $\mathbf{1}$ through the network $f_L^{\mathrm{s}}(\mathbf{1};(m \odot |\theta^{(0)}|))$.
  Prune $(1 - S^{n/N})$ fraction of the parameters according to the criterion $\left|\nabla_{\theta^{(0)}} f_L^{\mathrm{s}}(x; m \odot |\theta^{(0)}|) \odot \theta^{(0)}\right|$, creating an updated mask m'.
  Let $m = m'$.
end
Rescale the values of weights based on the sparsity of the network: $\theta^{(0)} \leftarrow \theta^{(0)}/(1-S)$.
Train the network $f_L^{\mathrm{s}}(x;(m \odot \theta^{(0)}))$ for E epochs with the optimal learning rate to obtain $f_L^{\mathrm{s}}(x;(m \odot \theta^{(E)}))$.

ProsPr: Alizadeh et al. (2022) present ProsPr, a method that leverages meta-gradients, and incorporates an estimation of the higher-order effects of pruning on the loss function and the optimization trajectory to pinpoint trainable sparse models during initialization. For our comparative analysis, we utilize their implementation at `https://github.com/mil-ad/prospr`.

GEM-MINER: GEM-MINER, introduced by Sreenivasan et al. (2022), represents a recent advancement in identifying subnetworks at initialization, revealing winning tickets within initialized subnetworks. It is based on one-shot pruning method, edge-popup (EP) (Ramanujan et al., 2020) for identifying supermasks. The code implementation of GEM-MINER from `https://github.com/ksreenivasan/pruning_is_enough` is used for the experiments.

Algorithm 4: GEM-MINER

Initialize $f_L^{\mathrm{s}}(x; (m \odot \theta^{(0)}))$, q.
for epoch e in E **do**
  for iteration n in N **do**
    /*** STE ***/
    Train $f_L^{\mathrm{s}}(x; (r(m) \odot \theta^{(0)}) \odot q)$.
    /*** STE ***/
    $m \leftarrow \mathrm{proj}_{[0,1]^P}\, m$.
  end
  if $\mathrm{mod}(e, V) = 0$ **then**
    $I_1 \leftarrow \{u \mid q_u = 1\}$.
    $m_{\mathrm{sorted}} \leftarrow$ Sort $\{m_u, u \in I_1\}$.
    $m_{\mathrm{bottom}} \leftarrow$ Bottom- 20% of m_{sorted}.
    $q \leftarrow q \odot 1_{m_u \notin m_{\mathrm{bottom}}}$.
  end
end
Rescale the values of weights based on the sparsity of the network: $\theta^{(0)} \leftarrow \theta^{(0)}/(1-S)$.
Train the network $f_L^{\mathrm{s}}(x; (q \odot \theta^{(0)}))$ for a total of E epochs using an optimal learning rate, resulting in $f_L^{\mathrm{s}}(x; (q \odot \theta^{(E)}))$.

Continuous Sparsification (CS): Continuous Sparsification (CS) (Savarese et al., 2019) is designed to identify sparse subnetworks during the early training phases by approximating l_0 regularization through a series of continuous sigmoid functions. We utilize their publicly available code from `https://github.com/lolemacs/continuous-sparsification` to replicate CS results. In our experiments replicating CS, we set the rewinding epoch to the 3rd epoch, employ 3 rounds of training to identify winning tickets, and use hyperparameter values $\lambda = 10^{-8}$, $\beta^{(0)} = 1$, $\beta^{(E)} = 300$. We initialize mask parameters to different constants for varying the sparsity levels across experiments.

Algorithm 5: Continuous Sparsification

for round r in R **do**
  Initialize $\beta \leftarrow 1$
  Train $\min_{s,\theta} L_{\mathrm{CE}}(x; \sigma(\beta s) \odot \theta) + \lambda \|\sigma(\beta s)\|_1$ for E epochs, while increasing β by factor μ each epoch.
  If round $r = R$, output $f_L^{\mathrm{s}}(x; H(s^{(E)}) \odot \theta^{(3)})$.
  Otherwise, set $s \leftarrow \min\{\beta^{(E)} s^{(E)}, s^{(0)}\}$, $\beta \leftarrow 1$. Here, $\min\{\cdot,\cdot\}$ represent element-wise application of the minimum function to compare corresponding elements of the two vectors.
end
Rescale the values of weights based on the sparsity of the network: $\theta^{(3)} \leftarrow \theta^{(3)}/(1 - \frac{\|H(s^{(E)})\|_0}{P})$.
Train the network $f_L^{\mathrm{s}}(x; (H(s^{(E)})) \odot \theta^{(3)}))$ for E epochs with optimal learning rate to obtain $f_L^{\mathrm{s}}(x; (H(s^{(E)}) \odot \theta^{(E)}))$.