Reconsidering Discarded Parameters: Aiding Sparse Subnetwork Fine-tuning

Man Yuan, Bin Wang, Haodong Bian, Xiaochun Yang, Zequan Wang

School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China
2401953@stu.neu.edu.cn, binwang@mail.neu.edu.cn, 2290168@stu.neu.edu.cn,
yangxc@mail.neu.edu.cn, 2472057@stu.neu.edu.cn

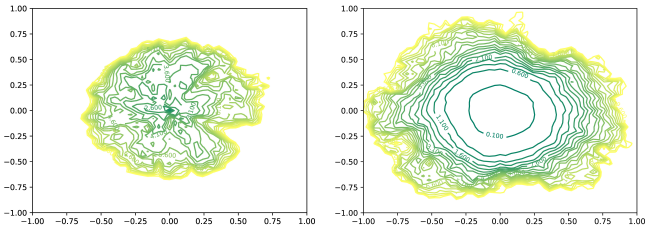
***Abstract*—While Pruning-at-Initialization (PaI) algorithms have evolved rapidly to identify efficient sparse structures, their potential is often bottlenecked by a primitive "hard pruning" training protocol. This paradigm strictly freezes unselected parameters, introducing optimization discontinuities and forcing the model to traverse a rugged loss landscape. To address this, we propose ReFIT, a universal training paradigm that is strictly orthogonal to the mask selection criteria. Rather than proposing a new pruning metric, ReFIT serves as a "plug-and-play" booster that revitalizes existing masks. By replacing rigid truncation with Score-Guided Adaptive Regularization, ReFIT utilizes redundant weights as temporary "optimization scaffolds," dynamically suppressing them to stabilize the optimization trajectory during critical early phases. Theoretically, we prove via Schur Complement analysis that ReFIT strictly reduces the maximum eigenvalue of the effective Hessian, guaranteeing a smoother optimization landscape. Extensive experiments on CIFAR-10/100 and ImageNet across diverse architectures (ResNet, VGG, MobileNet) demonstrate that ReFIT consistently boosts the performance of diverse sparse masks—ranging from classical magnitude pruning to random-weight methods like Edge-Popup. By effectively decoupling topology discovery from optimization dynamics, ReFIT validates that "how we train" is just as critical as "what we keep."**

***Index Terms*—Deep Learning, Fine-Tuning, Sparse Subnetwork, Model Convergence**

I. Introduction

Deep Convolutional Neural Networks (CNNs) have fundamentally revolutionized computer vision, achieving unprecedented success in tasks ranging from image classification [1], [2] to object detection [3], [4]. However, this performance comes at the cost of massive parameter counts and computational complexity, hindering deployment on resource-constrained edge devices. To alleviate this redundancy, Network Pruning has become a standard practice [5], [6], and its relevance has only surged in the era of large foundation models [7], [8].

While traditional methods rely on a computationally expensive "train-prune-finetune" cycle [9]–[11], the field has witnessed a paradigm shift towards **Pruning-at-Initialization (PaI)**. This direction was pioneered by the seminal **Lottery Ticket Hypothesis (LTH)** [12], which demonstrated that sparse subnetworks capable of matching dense performance exist prior to training. Inspired by this finding, a wave of algorithms such as SNIP [13], GraSP [14], SynFlow [15], and recent theoretical expansions [16] have been developed to identify these sparse topologies directly at initialization, promising a streamlined pipeline for efficient training.



(a) Rugged loss landscape (b) Smooth loss landscape

Fig. 1: Loss landscape visualization of ResNet-18 at 90% sparsity using the method from [17]. (a) **Standard Hard Pruning** results in a rugged surface with sharp minima. (b) **ReFIT** induces a significantly smoother and flatter geometry.

However, a critical dichotomy exists in current research: while *Mask Identification* has seen rapid advancements, the subsequent *Training Protocol* remains strikingly primitive. The vast majority of existing methods strictly adhere to a **"Hard Pruning" paradigm**: unselected parameters are rigidly frozen to zero from the very first iteration. We argue that this blind inheritance of standard schedules constitutes a fundamental bottleneck. Theoretical studies on optimization landscapes suggest that the geometry of the loss surface is pivotal for generalization [17], [18], a perspective reinforced by recent topological analysis of loss basins [19], [20]. Unlike dense networks that enjoy redundant pathways for gradient flow [21], sparse subnetworks initialized with binary constraints suffer from severe **gradient blockage** and topological rigidity [22], [23]. As visualized in Figure 1(a), forcing an optimizer to navigate a sparse topology from scratch results in a rugged, ill-conditioned landscape, making the model prone to getting trapped in sharp, suboptimal minima [24]. This implies that the key to unleashing the full potential of sparse networks lies not merely in *what* structure we find, but in *how* we train it.

Guided by this insight, we propose **ReFIT** (Regularized Full-Initialization Fine-Tuning), a universal training paradigm orthogonal to mask selection. ReFIT fundamentally reforms the optimization dynamics by shifting from **"Hard Truncation" to "Soft Landing."** Instead of physically removing weights at initialization, we reformulate pruning as a **Score-**

Guided Adaptive Regularization process. ReFIT dynamically assigns penalty strengths inversely proportional to parameter importance, utilizing the eventually-discarded weights as **"optimization scaffolds."** These scaffolds maintain a smooth loss landscape (Figure 1(b)) during the volatile early training phase [25] and are gradually suppressed only after the model has secured a stable optimization trajectory. Theoretically, we provide a rigorous justification via **Schur Complement analysis**, proving that ReFIT strictly reduces the maximum eigenvalue of the effective Hessian, thereby guaranteeing better condition numbers. Our contributions are summarized as follows:

- **Diagnostic Insight:** We identify the optimization discontinuity caused by the "hard pruning" paradigm as a primary bottleneck, supported by landscape visualization and Hessian analysis referencing flat-minima theories [26], [27].
- **Methodological Innovation:** We propose **ReFIT**, a plug-and-play protocol using **Score-Guided Adaptive Regularization**. By leveraging redundant weights as "optimization scaffolds," it bridges the gap between dense initialization and sparse topology.
- **Validation:** Extensive experiments across diverse architectures (ResNet, VGG, MobileNet) on CIFAR-10/100 and ImageNet demonstrate that ReFIT consistently boosts the performance of state-of-the-art sparse masks.

II. Related Work

A. Pruning-at-Initialization (PaI)

The Lottery Ticket Hypothesis (LTH) [12] demonstrated that dense networks contain sparse subnetworks capable of matching full accuracy when trained in isolation. To avoid the high cost of iterative magnitude pruning, numerous Pruning-at-Initialization (PaI) methods have been proposed to identify these masks prior to training. SNIP [13] retains connections with high saliency based on gradient sensitivity. GraSP [14] considers the Hessian-gradient product to preserve gradient flow. SynFlow [15] iteratively conserves synaptic flow to prevent layer collapse. More recently, Edge-Popup [28] and Gem-Miner [29] employ latent scores to dynamically optimize the topology. While these methods employ diverse criteria for connection selection, they predominantly share a common limitation: they rely on standard training protocols designed for dense networks, rigidly freezing pruned parameters (Hard Pruning) and neglecting the specific optimization needs of sparse architectures.

B. Optimization Challenges in Sparse Training

Training sparse networks from scratch is notoriously difficult due to their restricted topological capacity. Research indicates that sparse subnetworks suffer from poor gradient propagation [22] and traverse a significantly more rugged loss landscape compared to dense counterparts [17]. Recent works have attempted to mitigate these issues by modifying the architecture or optimization trajectory. RigL [30] and Top-KAST [31] dynamically update the mask topology during training to escape local minima. ToST [32] introduces architectural modifications, such as "ghost" skip connections and soft activations, to improve trainability. However, these methods often incur additional computational overhead. In contrast, ReFIT addresses the optimization bottleneck solely through a refined regularization protocol, maintaining a fixed mask without architectural changes.

C. Regularization Strategies in Pruning

Regularization has been widely used to guide pruning. Approaches like **Polarization Regularization** [33] impose penalties to push weights towards zero for structure learning. **MaskSparsity** [34] applies targeted regularization to accelerate the zeroing of pruned weights. However, a key distinction of our work is the *role* of regularization. Prior methods typically use regularization to *enforce* sparsity. Conversely, **ReFIT** employs regularization as a *smoothing mechanism*. By treating discarded parameters as "optimization scaffolds," ReFIT utilizes the full capacity of the initialization to navigate the early training phase, thereby smoothing the optimization trajectory before final truncation.

III. Preliminaries and Problem Setup

A. Sparse Subnetworks

Consider a deep neural network $f(x;\theta)$ with parameters $\theta \in \mathbb{R}^d$. A sparse subnetwork is typically defined by a binary mask $m \in \{0,1\}^d$. The mask m partitions the parameter space into two disjoint sets:

- **Active Set ($\mathcal{A}$):** Indices where $m_i = 1$. These parameters are preserved and trained.
- **Discarded Set ($\mathcal{D}$):** Indices where $m_i = 0$. In standard protocols, these are pruned.

B. The Hard Pruning Paradigm

Current Pruning-at-Initialization (PaI) methods predominantly employ a "Hard Pruning" protocol. Mathematically, this is a constrained optimization problem:

$$\min_{\theta} \mathcal{L}(\theta) \quad \text{s.t.} \quad \theta_i = 0, \quad \forall i \in \mathcal{D} \tag{1}$$

This forces the optimization trajectory to strictly adhere to a lower-dimensional subspace $\mathbb{R}^{|\mathcal{A}|}$.

C. Hessian and Loss Landscape

The geometry of the loss landscape is locally characterized by the Hessian matrix $H = \nabla^2 \mathcal{L}(\theta)$. For a sparse network, we partition the Hessian into blocks corresponding to the Active ($\mathcal{A}$) and Discarded ($\mathcal{D}$) sets:

$$H = \begin{bmatrix} H_{\mathcal{A}\mathcal{A}} & H_{\mathcal{A}\mathcal{D}} \\ H_{\mathcal{D}\mathcal{A}} & H_{\mathcal{D}\mathcal{D}} \end{bmatrix} \tag{2}$$

where $H_{\mathcal{A}\mathcal{A}}$ is the curvature within the active subspace, and $H_{\mathcal{A}\mathcal{D}}$ captures the interactions. A larger maximum eigenvalue $\lambda_{\max}(H)$ implies a sharper, more unstable landscape.

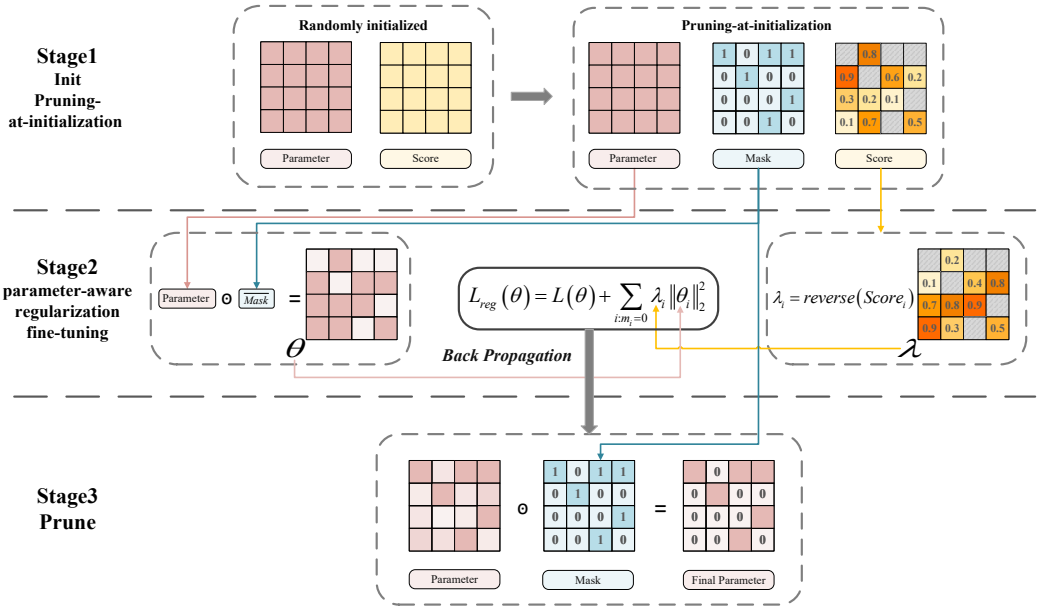


Fig. 2: Overview of the score regularization process in ReFIT, a three-stage pruning process: Stage 1 applies pruning-at-initialization by assigning scores and generating masks for parameters; Stage 2 incorporates Score-Guided regularization during fine-tuning to refine the model; Stage 3 applies the masks to prune the parameters, resulting in the final sparse model.

IV. Methodology

To overcome the optimization discontinuity and gradient blockage inherent in the traditional "Hard Pruning" paradigm, we introduce **ReFIT** (Regularized Full-Initialization Fine-Tuning). Unlike prior approaches that enforce rigid topological constraints from initialization, ReFIT reformulates sparse training as a continuous **"Soft Suppression"** process. By leveraging a **Score-Guided Adaptive Regularization** mechanism, we utilize the eventually-discarded weights as "optimization scaffolds"—preserving essential gradient flow during critical early training phases before gradually suppressing them.

In this section, we first formalize this transition from hard constraints to continuous regularization (Section IV-A). We then provide a rigorous theoretical justification via **Schur Complement analysis**, proving that ReFIT strictly reduces the maximum eigenvalue of the effective Hessian, thereby guaranteeing a smoother optimization landscape (Section IV-B). Finally, we detail the three-stage protocol that renders ReFIT a universal, plug-and-play framework for diverse pruning criteria (Section IV-C).

A. From Hard Constraints to Soft Suppression

The core insight of ReFIT is to relax the rigid constraint in Eq. (1) into a continuous regularization term. Instead of strictly freezing $\theta_{\mathcal{D}}$ to zero, we allow them to vary but impose a penalty to suppress their magnitude. The ReFIT objective is:

$$\mathcal{L}_{\text{ReFIT}}(\theta) = \mathcal{L}(\theta) + \sum_{i \in \mathcal{D}} \frac{\lambda_i}{2} \theta_i^2 \tag{3}$$

Here, $\lambda_i > 0$ is a scalar hyperparameter controlling the suppression strength.

- **Active ($\mathcal{A}$):** Update normally based on task gradients.
- **Discarded ($\mathcal{D}$):** Receive gradients but are continuously pulled towards zero by $\lambda_i \theta_i$.

This transforms the optimization from traversing a rigid "cliff" edge to moving through a smoothed "valley," using discarded parameters as "optimization scaffolds."

B. Theoretical Analysis: Spectral Smoothing via Schur Complement

To rigorously justify ReFIT, we analyze the local curvature. We prove that ReFIT strictly reduces the maximum eigenvalue of the effective Hessian observed by the active parameters.

1) Effective Hessian of Hard Pruning: Recalling the Hessian partition defined in Eq. (2), the standard Hard Pruning paradigm enforces $\delta_{\mathcal{D}} = 0$. Consequently, the curvature observed by the active parameters is restricted to the principal submatrix:

$$H_{\text{Hard}}^{\text{eff}} \triangleq H_{\mathcal{A}\mathcal{A}} \tag{4}$$

2) Effective Hessian of ReFIT: In ReFIT, the discarded parameters act as a regularized buffer. The local quadratic approximation of the ReFIT objective involves the full Hessian from Eq. (2) augmented by the regularization term λI:

$$\mathcal{L}_{\text{ReFIT}}(\delta) \approx \frac{1}{2} \begin{bmatrix} \delta_{\mathcal{A}}^\top & \delta_{\mathcal{D}}^\top \end{bmatrix} \begin{bmatrix} H_{\mathcal{A}\mathcal{A}} & H_{\mathcal{A}\mathcal{D}} \\ H_{\mathcal{D}\mathcal{A}} & H_{\mathcal{D}\mathcal{D}} + \lambda I \end{bmatrix} \begin{bmatrix} \delta_{\mathcal{A}} \\ \delta_{\mathcal{D}} \end{bmatrix} \tag{5}$$

By minimizing out the discarded parameters $\delta_{\mathcal{D}}$ (solving $\nabla_{\delta_{\mathcal{D}}} \mathcal{L} = 0$), we derive the **Effective Hessian** via the Schur Complement:

$$H_{\text{ReFIT}}^{\text{eff}} \triangleq H_{\mathcal{A}\mathcal{A}} - H_{\mathcal{A}\mathcal{D}} (H_{\mathcal{D}\mathcal{D}} + \lambda I)^{-1} H_{\mathcal{D}\mathcal{A}} \tag{6}$$

3) Spectral Smoothing Theorem: **Theorem 1 (Spectral Reduction).** *For any $\lambda > 0$ such that $(H_{\mathcal{D}\mathcal{D}} + \lambda I)$ is positive definite, the maximum eigenvalue of the ReFIT effective Hessian is strictly bounded by that of Hard Pruning:*

$$\lambda_{\max}(H^{\text{eff}}_{\text{ReFIT}}) \leq \lambda_{\max}(H^{\text{eff}}_{\text{Hard}}) \tag{7}$$

Proof. Let $K(\lambda) = H_{\mathcal{A}\mathcal{D}}(H_{\mathcal{D}\mathcal{D}} + \lambda I)^{-1} H_{\mathcal{D}\mathcal{A}}$. Since $(H_{\mathcal{D}\mathcal{D}} + \lambda I)$ is Positive Definite (PD), its inverse is PD. Thus, $K(\lambda)$ is Positive Semi-Definite (PSD). According to Weyl's Inequality, subtracting a PSD matrix reduces the eigenvalues:

$$\lambda_{\max}(H^{\text{eff}}_{\text{ReFIT}}) = \lambda_{\max}(H_{\mathcal{A}\mathcal{A}} - K(\lambda)) \leq \lambda_{\max}(H_{\mathcal{A}\mathcal{A}}) \tag{8}$$

□

Implication. The term $-K(\lambda)$ acts as a "Spectral Dampener," theoretically guaranteeing a smoother landscape and better condition number than hard pruning.

C. The ReFIT Protocol

ReFIT operates in three distinct stages to ensure a smooth transition from dense initialization to sparse topology, as illustrated in Figure 2. The detailed procedure is summarized in Algorithm 1.

1) Stage 1: Mask Identification: We initialize the network with dense weights θ_0. The computation of importance scores S is determined by the specific Pruning-at-Initialization (PaI) criterion adopted. For instance, magnitude-based methods (e.g., LTH [12]) define importance as absolute weight values ($S_i = |\theta_i|$), whereas gradient-based methods (e.g., SNIP [13]) calculate connection sensitivity ($S_i = |\theta_i \odot g_i|$). Based on these method-specific scores, a binary mask $m \in \{0,1\}^d$ is generated by selecting the top-κ parameters, identifying the Active Set $\mathcal{A}$ and Discarded Set $\mathcal{D}$.

2) Stage 2: Soft Suppression via Generic Score Adaptation: Given the raw importance scores S and the identified sets $\mathcal{A}$ and $\mathcal{D}$ from Stage 1, we proceed to the core soft suppression phase.

Generic Score Normalization. Different pruning criteria produce scores at vastly different scales (e.g., gradient norms vs. weight magnitudes). To ensure ReFIT is agnostic to the specific metric used, we first normalize the raw scores S_{raw} into a unified range $[0,1]$ via min-max normalization:

$$S_{\text{norm},i} = \frac{S_{\text{raw},i} - \min(S_{\text{raw}})}{\max(S_{\text{raw}}) - \min(S_{\text{raw}})} \tag{9}$$

Adaptive Penalty Assignment. We then dynamically assign a specific regularization strength λ_i to each parameter in the discarded set $\mathcal{D}$. The strength is designed to be inversely proportional to the parameter's relative importance:

$$\lambda_i = \beta \cdot (1 - S_{\text{norm},i}), \quad \forall i \in \mathcal{D} \tag{10}$$

where β is a global hyperparameter controlling the maximum suppression force. This mechanism ensures that "irrelevant" weights (low score) face strong suppression, while "near-critical" weights (high score) are only lightly penalized, allowing them to serve as "optimization scaffolds" for longer.

Algorithm 1 The ReFIT Training Protocol

Require: Network $f(x;\theta)$, Dataset $\mathcal{D}_{ata}$, Target Sparsity κ, Base Reg Strength β, Criterion $C(\cdot)$.

Stage 1: Mask & Score Calculation
Compute scores $S \leftarrow C(\theta_0)$
$S_{norm} \leftarrow \text{Normalize}(S)$ to $[0,1]$
Mask $m \leftarrow \text{TopK}(S, \kappa)$
Define Sets: $\mathcal{A} = \{i | m_i = 1\}$, $\mathcal{D} = \{i | m_i = 0\}$
Stage 2: Score-Guided Soft Suppression
Compute Adaptive Penalty: $\lambda_i \leftarrow \beta \cdot (1 - S_{norm,i})$ for $i \in \mathcal{D}$
for epoch $t = 1, \ldots, E$ **do**
  Sample batch (x, y)
  $\mathcal{L}_{task} = \ell(f(x;\theta), y)$
  $\mathcal{L}_{reg} = \sum_{i \in \mathcal{D}} \frac{\lambda_i}{2} \theta_i^2$
  Update $\theta \leftarrow \theta - \eta \nabla_\theta (\mathcal{L}_{task} + \mathcal{L}_{reg})$
end for
Stage 3: Final Truncation
Hard prune: $\theta_i \leftarrow 0, \forall i \in \mathcal{D}$
return Sparse Subnetwork θ^*

Training Objective. Consequently, the optimization in Stage 2 minimizes the following composite loss function:

$$\mathcal{L}_{\text{Stage2}}(\theta) = \mathcal{L}_{\text{task}}(f(x;\theta), y) + \underbrace{\sum_{i \in \mathcal{D}} \frac{\lambda_i}{2} \theta_i^2}_{\text{Soft Suppression Term}} \tag{11}$$

where $\mathcal{L}_{\text{task}}$ denotes the standard task-specific loss (e.g., Cross-Entropy for classification). By minimizing Eq. (11), the optimizer smoothly drives the discarded parameters towards zero according to their individual λ_i, avoiding the optimization shock caused by immediate hard truncation.

3) Stage 3: Hard Truncation: Upon completion of the training, we enforce the exact sparsity constraint by setting $\theta_i = 0, \forall i \in \mathcal{D}$.

V. Experiments

A. Experimental Setup

Datasets and Architectures. To evaluate the universality of ReFIT, we conduct extensive experiments on diverse benchmarks: **CIFAR-10/100, Tiny-ImageNet** for standard evaluation and **ImageNet (ILSVRC-2012)** for large-scale verification. We employ a wide range of architectures, including **ResNet-18/50**, **VGG-16**, and the compact **MobileNet-V1**. For extensive validation across different datasets and architectures, we selected the most popular **LTH** [12] as the primary representative, demonstrating how ReFIT generalizes robustly across diverse architectural backbones.

Baselines and Pruning Methods. We integrate ReFIT with various state-of-the-art sparse mask identification algorithms, covering magnitude-based (**LTH** [12]), gradient-based (**SNIP** [13], **GraSP** [14]), and dynamic topology methods (**Edge-Popup** [28], **RareGem** [29]). We compare our method against the "Standard" training protocol, which employs the

TABLE I: Classification accuracies of various pruning algorithms for varying sparsities on CIFAR-10, CIFAR-100, and Tiny-ImageNet. We compare the baseline pruning methods with our proposed ReFIT method. ReFIT demonstrates consistent improvements across all datasets, especially on the larger-scale Tiny-ImageNet.

Sparse Mask	**CIFAR-10**			**CIFAR-100**			**Tiny-ImageNet**		
	90%	95%	98%	90%	95%	98%	90%	95%	98%
ResNet-18 [No Pruning]									
Baseline Accuracy	94.80	-	-	74.64	-	-	65.18	-	-
Edge-Popup [28]	90.11 ± 0.28	78.98 ± 0.74	60.29 ± 1.17	69.44 ± 0.97	45.37 ± 1.28	39.82 ± 2.19	60.45 ± 0.14	42.80 ± 0.76	30.33 ± 1.40
Edge-Popup + ReFIT	$\mathbf{92.01 \pm 0.33}$	$\mathbf{87.60 \pm 0.78}$	$\mathbf{70.54 \pm 2.06}$	$\mathbf{71.30 \pm 0.91}$	$\mathbf{53.49 \pm 0.75}$	$\mathbf{47.22 \pm 0.81}$	$\mathbf{61.27 \pm 0.32}$	$\mathbf{50.43 \pm 0.18}$	$\mathbf{39.98 \pm 1.34}$
LTH [12]	93.26 ± 0.14	92.17 ± 0.25	91.88 ± 0.45	71.27 ± 0.14	70.31 ± 0.12	68.87 ± 0.27	62.15 ± 0.27	60.21 ± 0.52	58.76 ± 1.12
LTH + ReFIT	$\mathbf{94.53 \pm 0.21}$	$\mathbf{93.44 \pm 0.11}$	$\mathbf{92.26 \pm 0.17}$	$\mathbf{72.36 \pm 0.36}$	$\mathbf{71.60 \pm 0.17}$	$\mathbf{69.45 \pm 0.24}$	$\mathbf{63.27 \pm 0.18}$	$\mathbf{61.25 \pm 0.43}$	$\mathbf{59.40 \pm 0.16}$
SNIP [13]	92.12 ± 0.65	91.19 ± 0.37	87.97 ± 0.26	69.43 ± 0.42	65.43 ± 0.25	54.31 ± 0.98	60.12 ± 0.45	56.12 ± 0.34	45.25 ± 0.16
SNIP + ReFIT	$\mathbf{92.88 \pm 0.15}$	$\mathbf{92.11 \pm 0.23}$	$\mathbf{88.22 \pm 0.14}$	$\mathbf{70.01 \pm 0.21}$	$\mathbf{66.28 \pm 0.52}$	$\mathbf{55.27 \pm 0.96}$	$\mathbf{61.82 \pm 0.25}$	$\mathbf{57.32 \pm 0.13}$	$\mathbf{46.16 \pm 0.55}$
GraSP [14]	92.28 ± 0.67	91.74 ± 0.15	89.70 ± 0.42	69.15 ± 0.34	66.46 ± 0.21	58.43 ± 0.43	60.80 ± 0.42	59.34 ± 0.38	43.43 ± 0.16
GraSP + ReFIT	$\mathbf{92.98 \pm 0.07}$	$\mathbf{92.07 \pm 0.24}$	$\mathbf{90.25 \pm 0.15}$	$\mathbf{70.04 \pm 0.21}$	$\mathbf{67.21 \pm 0.44}$	$\mathbf{59.63 \pm 0.76}$	$\mathbf{61.75 \pm 0.26}$	$\mathbf{60.58 \pm 0.15}$	$\mathbf{44.69 \pm 0.45}$
RareGem [29]	91.54 ± 0.18	89.98 ± 0.12	87.22 ± 0.65	70.21 ± 0.19	67.25 ± 0.24	58.92 ± 0.22	61.25 ± 0.28	58.44 ± 0.35	46.40 ± 0.50
RareGem + ReFIT	$\mathbf{92.11 \pm 0.23}$	$\mathbf{90.65 \pm 0.70}$	$\mathbf{88.34 \pm 1.06}$	$\mathbf{71.35 \pm 0.65}$	$\mathbf{68.19 \pm 0.65}$	$\mathbf{60.82 \pm 0.11}$	$\mathbf{62.88 \pm 0.20}$	$\mathbf{59.95 \pm 0.25}$	$\mathbf{47.80 \pm 0.35}$
ResNet-50 [No Pruning]									
Baseline Accuracy	95.62	-	-	78.45	-	-	69.80	-	-
Edge-Popup [28]	93.15 ± 0.22	88.42 ± 0.65	74.18 ± 1.05	74.12 ± 0.55	62.35 ± 0.92	48.60 ± 1.85	64.15 ± 0.35	53.20 ± 0.88	38.55 ± 1.52
Edge-Popup + ReFIT	$\mathbf{94.88 \pm 0.18}$	$\mathbf{92.35 \pm 0.45}$	$\mathbf{83.60 \pm 0.95}$	$\mathbf{76.58 \pm 0.42}$	$\mathbf{69.10 \pm 0.65}$	$\mathbf{59.45 \pm 1.25}$	$\mathbf{66.22 \pm 0.28}$	$\mathbf{59.85 \pm 0.62}$	$\mathbf{49.10 \pm 1.15}$
LTH [12]	95.10 ± 0.12	94.85 ± 0.18	93.60 ± 0.35	77.20 ± 0.20	76.15 ± 0.32	73.40 ± 0.45	68.55 ± 0.22	67.10 ± 0.35	64.25 ± 0.65
LTH + ReFIT	$\mathbf{95.58 \pm 0.10}$	$\mathbf{95.25 \pm 0.15}$	$\mathbf{94.10 \pm 0.22}$	$\mathbf{78.05 \pm 0.18}$	$\mathbf{77.25 \pm 0.25}$	$\mathbf{74.80 \pm 0.38}$	$\mathbf{69.45 \pm 0.18}$	$\mathbf{68.30 \pm 0.28}$	$\mathbf{65.85 \pm 0.45}$
SNIP [13]	94.20 ± 0.35	93.10 ± 0.42	89.55 ± 0.55	75.40 ± 0.38	72.20 ± 0.55	62.15 ± 0.95	66.80 ± 0.38	63.45 ± 0.55	52.30 ± 0.85
SNIP + ReFIT	$\mathbf{94.95 \pm 0.15}$	$\mathbf{94.05 \pm 0.28}$	$\mathbf{90.80 \pm 0.35}$	$\mathbf{76.35 \pm 0.25}$	$\mathbf{73.40 \pm 0.40}$	$\mathbf{63.80 \pm 0.75}$	$\mathbf{67.90 \pm 0.25}$	$\mathbf{64.50 \pm 0.42}$	$\mathbf{53.85 \pm 0.65}$
GraSP [14]	94.35 ± 0.40	93.45 ± 0.30	90.10 ± 0.48	75.25 ± 0.45	73.15 ± 0.35	64.40 ± 0.82	67.10 ± 0.40	64.20 ± 0.50	50.15 ± 0.70
GraSP + ReFIT	$\mathbf{95.05 \pm 0.12}$	$\mathbf{94.20 \pm 0.22}$	$\mathbf{91.15 \pm 0.32}$	$\mathbf{76.10 \pm 0.28}$	$\mathbf{74.85 \pm 0.30}$	$\mathbf{67.20 \pm 0.65}$	$\mathbf{68.25 \pm 0.25}$	$\mathbf{66.10 \pm 0.35}$	$\mathbf{52.80 \pm 0.55}$
RareGem [29]	93.85 ± 0.20	92.50 ± 0.25	90.45 ± 0.55	76.10 ± 0.25	73.80 ± 0.40	66.50 ± 0.50	67.50 ± 0.30	65.20 ± 0.45	54.60 ± 0.75
RareGem + ReFIT	$\mathbf{94.60 \pm 0.15}$	$\mathbf{93.80 \pm 0.20}$	$\mathbf{91.90 \pm 0.40}$	$\mathbf{77.05 \pm 0.20}$	$\mathbf{75.45 \pm 0.30}$	$\mathbf{69.20 \pm 0.45}$	$\mathbf{68.80 \pm 0.22}$	$\mathbf{66.95 \pm 0.35}$	$\mathbf{56.20 \pm 0.55}$

TABLE II: **Scalability on ImageNet (ResNet-50).** Top-1 Accuracy (%) comparison. ReFIT significantly boosts performance, particularly for Edge-Popup at high sparsity.

Algorithm	**Fine-tuning Method**	**70%**	**75%**	**80%**	**85%**
Edge-Popup	Standard	61.56	58.76	55.45	33.42
	ReFIT (Ours)	**63.59**	**60.80**	**58.69**	**40.78**
SNIP	Standard	68.24	67.30	65.20	64.40
	ReFIT (Ours)	**69.70**	**68.35**	**66.58**	**65.90**

widely used hard-pruning schedule (freezing pruned weights to zero).

Finally, we empirically validate the theoretical claims made in Section IV.

B. Universal Performance Gains on CIFAR and Tiny-ImageNet

We first investigate whether ReFIT can consistently enhance the trainability of diverse sparse masks. Table I reports the performance on CIFAR-10/100 and Tiny-ImageNet using ResNet-18/50.

Consistent Improvement across Masks. ReFIT consistently outperforms the standard training protocol across all evaluated masks. This advantage is most striking in **Edge-Popup**, an algorithm originally designed for fixed random weights and notoriously incompatible with standard fine-tuning (which disrupts its mask-weight coupling). ReFIT effectively breaks this barrier by using edge scores to guide safe weight updates, achieving a massive **+7.4%** accuracy boost (39.82% $\rightarrow$ 47.22%) at 98% sparsity. Similarly, for **RareGem** and **GraSP**, which identify high-quality but hard-to-optimize masks, ReFIT provides substantial improvements (e.g., boosting RareGem by **1.90%** on CIFAR-100), demonstrating its ability to revive connectivity patterns that are otherwise difficult to optimize under hard constraints.

Robustness at High Sparsity. The performance gap between ReFIT and the baseline widens as sparsity increases. At 98% sparsity on CIFAR-100, ReFIT outperforms the baseline by significant margins across all methods (e.g., **+0.96%** for SNIP). This confirms our hypothesis that the "optimization scaffold" effect is most critical when the parameter space is severely restricted, providing necessary stability where standard optimization falters.

C. Scalability to ImageNet

To strictly validate the scalability of ReFIT on industrial-scale datasets, we conduct experiments on **ImageNet** using ResNet-50. As presented in Table II, the benefits of our approach translate seamlessly to this challenging domain.

Most notably, the substantial gains observed on **Edge-Popup** are amplified at this scale. At 85% sparsity, ReFIT boosts top-1 accuracy by a remarkable **+7.36%** (33.42% $\rightarrow$ 40.78%), confirming that our method effectively scales to large-parameter spaces even when starting from random subnetworks. Similarly, for **SNIP**, ReFIT maintains a robust advantage, consistently improving performance by $\sim$1.5% across all sparsity levels (e.g., 64.40% $\rightarrow$ 65.90% at 85% sparsity). These results verify that ReFIT provides a universal boost independent of dataset scale.

D. Cross-Architecture Generalization

To demonstrate the universality of ReFIT beyond ResNet architectures, we extend our evaluation to **MobileNet-V1** (a compact, depth-wise separable convolutional network) and

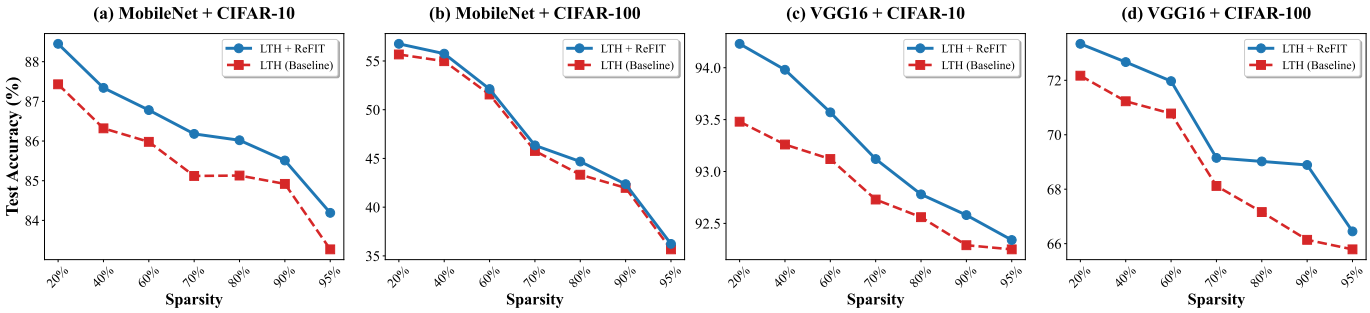


Fig. 3: **Cross-Architecture Generalization on CIFAR-10/100.** Comparison between LTH (Baseline) and ReFIT (Ours) on **MobileNet-V1** [(a), (b)] and **VGG-16** [(c), (d)]. ReFIT consistently outperforms the baseline, exhibiting superior stability particularly at high sparsity levels.

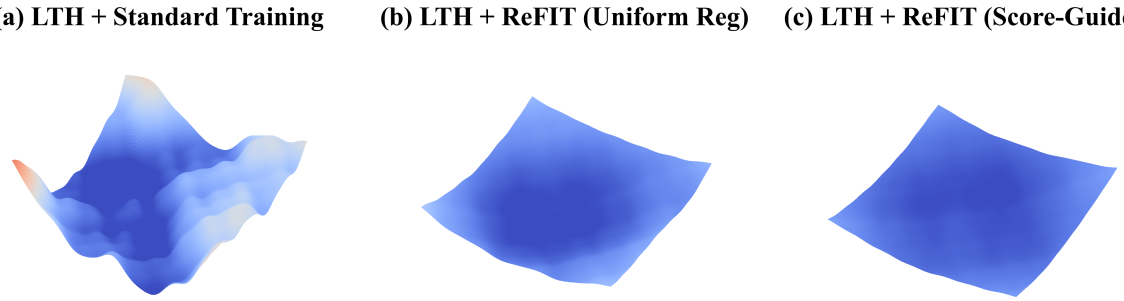


Fig. 4: **3D Visualization of the Loss Landscape. (a) Standard Training:** Converges to a sharp and chaotic minimum. **(b) Uniform Regularization:** Results in a smooth landscape **(c) Score-Guided Regularization:** Induces the significantly flattest and widest basin, demonstrating the optimal optimization geometry.

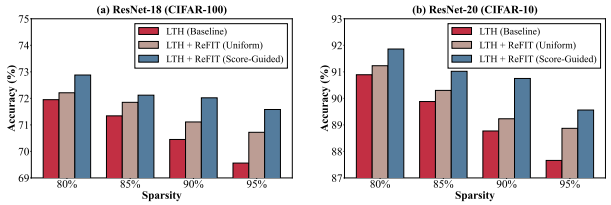
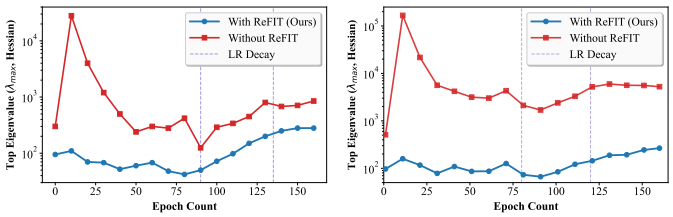


Fig. 5: **Ablation Study on Regularization Strategy.** Performance comparison on CIFAR-100 (ResNet-18) across three settings: *Standard Training* (Red), *Uniform Regularization* (Brown), and *Score-Guided Regularization* (Blue).



(a) LTH Mask (b) SynFlow Mask

Fig. 6: **Evolution of the Top Hessian Eigenvalue ($\lambda_{\max}$).** Comparison between Baseline (red) and ReFIT (blue) using **(a) LTH** and **(b) SynFlow** algorithms. ReFIT consistently suppresses early curvature spikes and maintains a low spectral norm across different pruning methods, ensuring optimization stability.

VGG-16 (a deep plain network without residual connections). Figure 3 illustrates the performance trends comparing the standard LTH baseline (red dashed line) against LTH + ReFIT (blue solid line) across a wide sparsity spectrum.

- **MobileNet-V1 (Figure 3 a-b):** As a lightweight architecture with limited redundancy, MobileNet is inherently sensitive to parameter pruning. On the challenging CIFAR-100 dataset, the baseline suffers a severe performance drop at extreme sparsity (95%), falling to 35.67%. ReFIT effectively mitigates this degradation, recovering the accuracy to **36.23%**. Similarly, on CIFAR-10, ReFIT maintains a consistent advantage throughout the pruning spectrum, elevating the final accuracy from 83.27% to **84.19%** at the highest sparsity level. These results verify that ReFIT provides stable optimization benefits even for compact models with minimal parameter margins.
- **VGG-16 (Figure 3 c-d):** Unlike ResNet, VGG-16 lacks residual connections, making gradient flow more susceptible to topological disruptions. Despite this structural challenge, ReFIT consistently outperforms the baseline across all sparsity levels. For instance, on CIFAR-100, ReFIT achieves an accuracy of **73.34%** at 20% sparsity

(vs. 72.17% baseline) and maintains its lead even at 95% sparsity (**66.45%** vs. 65.79%). This consistent superiority confirms that our score-guided regularization effectively constructs robust gradient scaffolds for plain, chain-like networks, regardless of the architectural topology.

E. Ablation Study: The Efficacy of Score-Guided Regularization

To strictly disentangle the benefits of our proposed mechanism, we conduct an ablation study on CIFAR-100 (ResNet-18) comparing three paradigms: (1) **Standard Training** (Baseline, red bars), (2) ReFIT with **Uniform Regularization** (constant λ, brown bars), and (3) ReFIT with **Score-Guided Regularization** (Ours, blue bars).

As illustrated in Figure 5, a clear performance hierarchy emerges. While **Uniform Regularization** (brown) consistently outperforms the hard-pruning baseline by smoothing the optimization landscape, it remains suboptimal compared to our **Score-Guided** strategy (blue), particularly at extreme sparsities (e.g., yielding a $\sim$0.8% gain at 95% sparsity).

This performance gap highlights a critical limitation of uniform penalties: they suppress all discarded weights equally, inadvertently dampening "near-critical" connections that could facilitate gradient flow. In contrast, our **Score-Guided Regularization** acts as an intelligent filter. By assigning penalty strengths inversely proportional to importance scores, it dynamically preserves high-score weights as *strong optimization scaffolds* while rapidly suppressing irrelevant ones. This mechanism effectively constructs a **score-aware optimization trajectory**, allowing the network to leverage the most valuable latent pathways for a smoother and more precise convergence to the target sparse topology.

F. Mechanism Analysis: Loss Landscape and Hessian

To strictly verify that ReFIT smooths the optimization terrain, we conduct a dual analysis using Hessian spectral norms and 3D landscape visualization.

Hessian Eigenvalue Evolution. Figure 6 tracks the trajectory of the top Hessian eigenvalue ($\lambda_{\max}$), which serves as a proxy for the sharpness of the loss landscape. A clear dichotomy is observed between the two methods. The **Baseline (red curve)** exhibits a massive explosion in curvature during the early training phase, indicating that the optimizer is forced to traverse a highly unstable, rugged region. This "shock" often pushes the weights into sharp, suboptimal minima. In contrast, **ReFIT (blue curve)** effectively suppresses this varying curvature, maintaining a stable and low spectral norm throughout the entire optimization process. This empirical evidence confirms that our soft suppression strategy successfully acts as a "spectral dampener", guiding the network along a smooth trajectory and preventing the optimization instabilities inherent to hard pruning.

Loss Landscape Visualization. We further corroborate these numerical findings with 3D loss landscape visualizations in Figure 4. A striking geometric distinction emerges between the hard pruning baseline and regularization-based approaches:

- **Standard Hard Pruning:** The landscape is characterized by sharp, chaotic non-convexities and narrow valleys. This "rugged cliff" geometry indicates a highly unstable optimization surface, which is notoriously difficult to traverse and is theoretically linked to poor generalization [26].
- **Regularization Strategies (Uniform & Score-Guided):** In contrast, both **Uniform Regularization** and our **Score-Guided ReFIT** exhibit similarly smooth geometries with significantly wider basins. The visual similarity between these two methods suggests that the transition from "hard constraints" to "soft suppression" is the primary driver for landscape smoothing. By avoiding immediate parameter truncation, both strategies successfully transform the landscape into a stable "smooth valley," distinctively outperforming the traditional sparse training paradigm.

VI. Conclusion

In this work, we revisited a fundamental yet overlooked bottleneck in sparse training: the optimization difficulties introduced by the "hard pruning" paradigm. We argued that the abrupt removal of parameters creates a rugged loss landscape, preventing even high-quality sparse masks from reaching their optimal performance. To resolve this, we introduced **ReFIT**, a generic and method-agnostic training framework that transforms sparse training from a rigid truncation process into a smooth, regularization-guided trajectory.

Our contributions are threefold. First, we provided a rigorous theoretical foundation showing that **Score-Guided Adaptive Regularization** acts as a spectral dampener, strictly reducing the effective Hessian eigenvalue and smoothing the optimization terrain. Second, we demonstrated the mechanism of "optimization scaffolds," where eventually-discarded weights are leveraged to maintain gradient flow during the critical early training phase. **Most importantly, we established the universality of ReFIT.** Our extensive experiments reveal that ReFIT is not limited to specific architectures or pruning criteria. It consistently revitalizes a wide spectrum of sparse structures, from gradient-based masks (SNIP, GraSP) to the notably difficult-to-train random topologies (Edge-Popup).

By decoupling the *optimization dynamics* from the *mask identification*, ReFIT offers a powerful, plug-and-play enhancement for the community. It suggests that future research should look beyond merely finding "winning tickets" and focus equally on the trajectory used to train them. We hope ReFIT serves as a robust baseline for this new paradigm of **Soft Sparse Training**.

References

[1] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "Convnext v2: Co-designing and scaling convnets with masked autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[2] M. Dehghani, J. Djolonga, B. Mustafa, P. Padlewski, J. Heek, J. Gilmer *et al.*, "Scaling vision transformers to 22 billion parameters," in *International Conference on Machine Learning (ICML)*, 2023.

[3] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[4] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," *arXiv preprint arXiv:2303.05499*, 2023.

[5] G. Fang, X. Ma, M. Song, M. B. Mi, and X. Wang, "Depgraph: Towards any structural pruning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[6] E. Frantar and D. Alistarh, "Sparsegpt: Massive language models can be accurately pruned in one-shot," in *International Conference on Machine Learning (ICML)*, 2023.

[7] M. Sun, Z. Liu, A. Bair, and J. Z. Kolter, "A simple and effective pruning approach for large language models," in *International Conference on Learning Representations (ICLR)*, 2024.

[8] X. Ma, G. Fang, and X. Wang, "Llm-pruner: On the structural pruning of large language models," *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[9] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 28, 2015.

[10] P. Molchanov, A. Mallya, J. Kautz *et al.*, "Importance estimation for neural network pruning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 11 264–11 272.

[11] Z. Liu, M. Sun, T. Zhou, G. Huang, and T. Darrell, "Rethinking the value of network pruning," in *International Conference on Learning Representations (ICLR)*, 2019.

[12] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *International Conference on Learning Representations (ICLR)*, 2019.

[13] N. Lee, T. Ajanthan, and P. H. Torr, "SNIP: Single-shot network pruning based on connection sensitivity," in *International Conference on Learning Representations (ICLR)*, 2019.

[14] C. Wang, G. Zhang, and R. Grosse, "Picking winning tickets before training by preserving gradient flow," in *International Conference on Learning Representations (ICLR)*, 2020.

[15] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 6377–6389.

[16] S. Liu, Z. Wang, and D. C. Mocanu, "Bridging the gap between pruning and sparsity: A unified perspective," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[17] H. Li, Z. Xu, G. Taylor, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[18] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," in *International Conference on Learning Representations (ICLR)*, 2019.

[19] C. Geniesse, J. Chen, T. Xie *et al.*, "Visualizing loss functions as topological landscape profiles," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[20] R. Ding, T. Li, and X. Huang, "Better loss landscape visualization for deep neural networks with trajectory information," in *Proceedings of the Asian Conference on Machine Learning (ACML)*, 2024.

[21] F. Draxler, K. Veschgini, M. Salmhofer, and F. Hamprecht, "Essentially no barriers in neural network energy landscape," in *International Conference on Machine Learning (ICML)*, 2018.

[22] U. Evci, F. Pedregosa, A. Gomez, and E. Elsen, "The difficulty of training sparse neural networks," in *International Conference on Learning Representations (ICLR)*, 2019.

[23] H. Mostafa and X. Wang, "Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization," in *International Conference on Machine Learning (ICML)*, 2019.

[24] Z. Yao, A. Gholami, K. Keutzer, and M. W. Mahoney, "PyHessian: Neural networks through the lens of the hessian," in *Proceedings of the IEEE International Conference on Big Data (Big Data)*, 2020, pp. 581–590.

[25] A. Achille, M. Rovere, and S. Soatto, "Critical learning periods in deep networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[26] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," in *International Conference on Learning Representations (ICLR)*, 2017.

[27] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997.

[28] V. Ramanujan, M. Wortsman, A. Kembhavi, A. Farhadi, and M. Rastegari, "What's hidden in a randomly weighted neural network?" in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 893–11 902.

[29] K. Sreenivasan, J.-y. Sohn, L. Yang, M. Grail, and D. Papailiopoulos, "Rare gems: Finding lottery tickets at initialization," in *International Conference on Learning Representations (ICLR)*, 2022.

[30] U. Evci, T. Gale, J. Menick, P. S. Castro, and E. Elsen, "Rigging the lottery: Making all tickets winners," in *International Conference on Machine Learning (ICML)*, 2020, pp. 2943–2952.

[31] S. Jayakumar, R. Pascanu, J. Rae, S. Osindero, and E. Elsen, "Top-KAST: Top-K always sparse training," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 20 744–20 754.

[32] A. K. Jaiswal, H. Ma, T. Chen, Y. Ding, and Z. Wang, "Training your sparse neural network better with any mask," in *International Conference on Machine Learning (ICML)*, 2022, pp. 9833–9844.

[33] T. Zhuang, Z. Zhang, Y. Huang, X. Zeng, K. Shuang, and X. Li, "Neuron-level structured pruning using polarization regularizer," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 9865–9877.

[34] N. Jiang, X. Zhao, C. Zhao, Y. An, M. Tang, and J. Wang, "Pruning-aware sparse regularization for network pruning," *International Journal of Automation and Computing*, vol. 20, no. 1, pp. 109–120, 2023.