DPaI: Differentiable Pruning at Initialization with Node-Path Balance Principle

Lichuan Xiang[1 *], Quan Nguyen-Tri[2,3 *], Lan-Cuong Nguyen[2,3], Hoang Pham[1], Khoat Than[2], Long Tran-Thanh[1], Hongkai Wen[1]

[1]University of Warwick [2]Hanoi University of Science and Technology [3]FPT Software AI Center *Equal Contribution

Introduction

Lottery Ticket Hypothesis (LTH) suggests the existence of sparse networks at initialization that can be trained to full accuracy.

Task: Pruning at Initialization (PaI) identifies LTH before training. → Significantly reduce memory and computational costs.

Motivation:

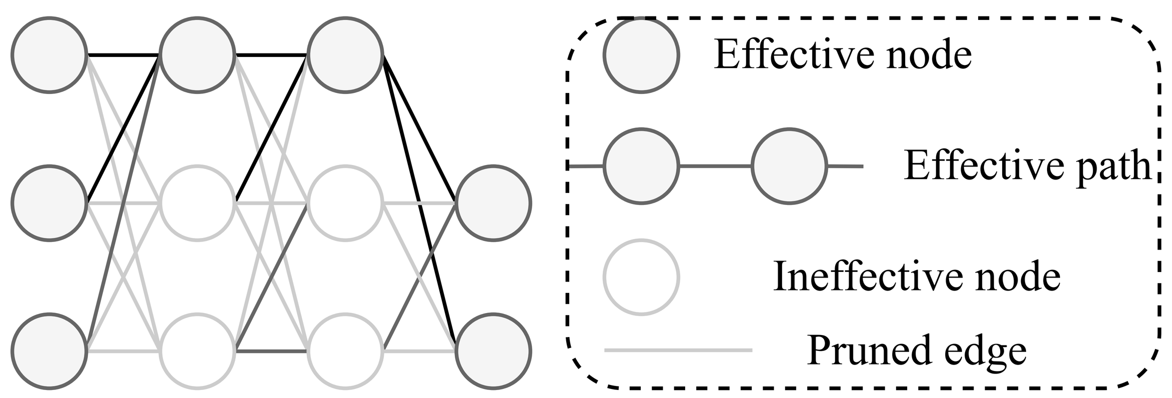
- *Node-Path Balancing (NPB)* principle optimizing subnetwork's topologies.
- NPB implementations require solving large-scale discrete optimization problems.

Contribution: We introduce Differentiable Pruning at Initialization (DPaI):

- Converts discrete NPB optimization into a differentiable formulation.
- Dynamically optimizes pruning masks to enhance network topology.
- Utilizes efficient gradient-based methods for fast, superior pruning.

Node-Path Balancing

Effective Path: connects an input node to an output node without any interruptions.
Effective Node/Channel: at least one effective path goes through it.



Architecture $f(x, \mathbf{W})$, parameter $\mathbf{W} \in \mathbb{R}^N$. NPB objective is to identify a binary mask $\mathbf{M}$ that:

$$\text{Maximize} \quad \mathcal{R}_{NPB} := \alpha \log \mathcal{R}_N + (1-\alpha)\log \mathcal{R}_P$$

$$\text{s.t.} \quad \|\mathbf{M}\|_1 \le N(1-\rho), \quad \rho: \text{desired sparsity}$$

Method Overview

Introduce differentiable score parameters for each weight: $m_{i,j}^{(l)} = \text{Top}_{k^{(l)}}(|s_{i,j}^{(l)}|)$

The number of incoming paths to a node:

$$P(v_j^{(l)}) = \sum_{i=1}^{h^{(l-1)}} m_{i,j}^{(l)} P(v_i^{(l-1)}), \quad \mathcal{R}_P = \sum_{j=1}^{h^{(L)}} P(v_j^{(L)})$$

The number of outgoing paths to a node:

$$\frac{\delta \mathcal{R}_P}{\delta P(v_j^{(l)})} = \sum_{n,p,q,\dots,k} m_{p,n}^{(L)} m_{q,p}^{(L-1)} \dots m_{j,k}^{(l+1)}$$

A node is effective when $N(v_j^{(l)}) > 0$:

$$N(v_j^{(l)}) = P(v_j^{(l)}) \frac{\delta \mathcal{R}_P}{\delta P(v_j^{(l)})}, \quad \mathcal{R}_N = \sum_{l,j} \tanh N(v_j^{(l)})$$

The derivative with respect to $\mathcal{R}_P$ and $\mathcal{R}_N$:

$$\frac{\delta \mathcal{R}_P}{\delta s_{i,j}^{(l)}} \propto \frac{\delta \mathcal{R}_P}{\delta P(v_j^{(l)})} P(v_i^{(l-1)}), \quad \frac{\delta \mathcal{R}_N}{\delta s_{i,j}^{(l)}} \propto \mathbb{1}_{N(v_j^{(l)})=0} \frac{\delta \mathcal{R}_P}{\delta s_{i,j}^{(l)}}$$

Path Objective: promote the score of edges that connect numerous effective paths.

Node Objective: promote the score of edges in an ineffective node.

Algorithm 1 Differentiable PaI (DPaI)

1: **Input:** network $f(x, \mathbf{W})$, final sparsity ρ, iteration steps T, hyperparameter α, β, η
2: Initialize the score parameters: $s_{i,j}^{(l)} \sim \mathcal{N}(0,1)$
3: Layer-wise sparsity: $k^{(l)} \leftarrow \text{ERK}(\rho)$
4: **for** $t \in 1, \dots, T$ **do**
5: Binarize the mask: $m_{i,j}^{(l)} \leftarrow \text{Top}_{k^{(l)}}(|s_{i,j}^{(l)}|)$
6: Number of effective paths: $\mathcal{R}_P \leftarrow f(\mathbb{1}, \mathbf{M})$
7: Calculate the derivatives: $\frac{\delta \mathcal{R}_P}{\delta s_{i,j}^{(l)}}, \frac{\delta \mathcal{R}_N}{\delta s_{i,j}^{(l)}}, \frac{\delta \mathcal{R}_C}{\delta s_{i,j}^{(l)}}$
8: Update the score parameters: $s_{i,j}^{(l)} \leftarrow s_{i,j}^{(l)} + \eta\left((1-\alpha)\frac{\delta \mathcal{R}_P}{\delta s_{i,j}^{(l)}} + \alpha\left((1-\beta)\frac{\delta \mathcal{R}_N}{\delta s_{i,j}^{(l)}} + \beta\frac{\delta \mathcal{R}_C}{\delta s_{i,j}^{(l)}}\right)\right)$
9: **end for**
10: **Output:** pruned network $f(x, \mathbf{M} \odot \mathbf{W})$

Convergence Analysis

Assuming, edge $m_{i,j}^{(l)}$ replaces $m_{p,q}^{(l)}$, and the rest of the sub-network remains fixed.

Optimising $\mathcal{R}_P$: $\left|\frac{\delta \log \mathcal{R}_P}{\delta s_{i,j}^{(l)}}\right| > \left|\frac{\delta \log \mathcal{R}_P}{\delta s_{p,q}^{(l)}}\right|$

Optimising $\mathcal{R}_N$: $N(v_j^{(l)}) = 0 \rightarrow N(v_j^{(l)}) > 0$

If $N(v_q^{(l)}) = 0$: $\left|\frac{\delta \log \mathcal{R}_P}{\delta s_{i,j}^{(l)}}\right| > \left|\frac{\delta \log \mathcal{R}_P}{\delta s_{p,q}^{(l)}}\right|$

If $N(v_q^{(l)}) > 0$: $\left|\frac{\delta \log \mathcal{R}_P}{\delta s_{i,j}^{(l)}}\right| > 0$

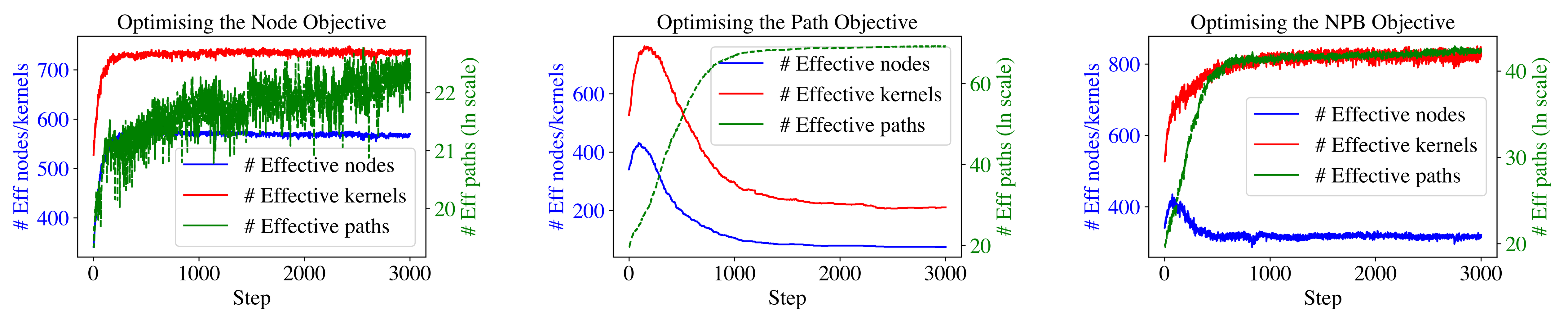
Optimising $\mathcal{R}_{NPB}$:

If $N(v_j^{(l)}) = 0, N(v_q^{(l)}) > 0$: $\left|\frac{\delta \log \mathcal{R}_P}{\delta s_{i,j}^{(l)}}\right| > \epsilon \left|\frac{\delta \log \mathcal{R}_P}{\delta s_{p,q}^{(l)}}\right|$

If $N(v_j^{(l)}) > 0, N(v_q^{(l)}) = 0$: $\left|\frac{\delta \log \mathcal{R}_P}{\delta s_{i,j}^{(l)}}\right| > \frac{1}{\epsilon} \left|\frac{\delta \log \mathcal{R}_P}{\delta s_{p,q}^{(l)}}\right|$

Otherwise: $\left|\frac{\delta \log \mathcal{R}_P}{\delta s_{i,j}^{(l)}}\right| > \left|\frac{\delta \log \mathcal{R}_P}{\delta s_{p,q}^{(l)}}\right|$

Figure 1: The convergence of different objective:



Results

Figure 2: DPaI consistently outperforms prior PaI methods across datasets and sparsity levels.

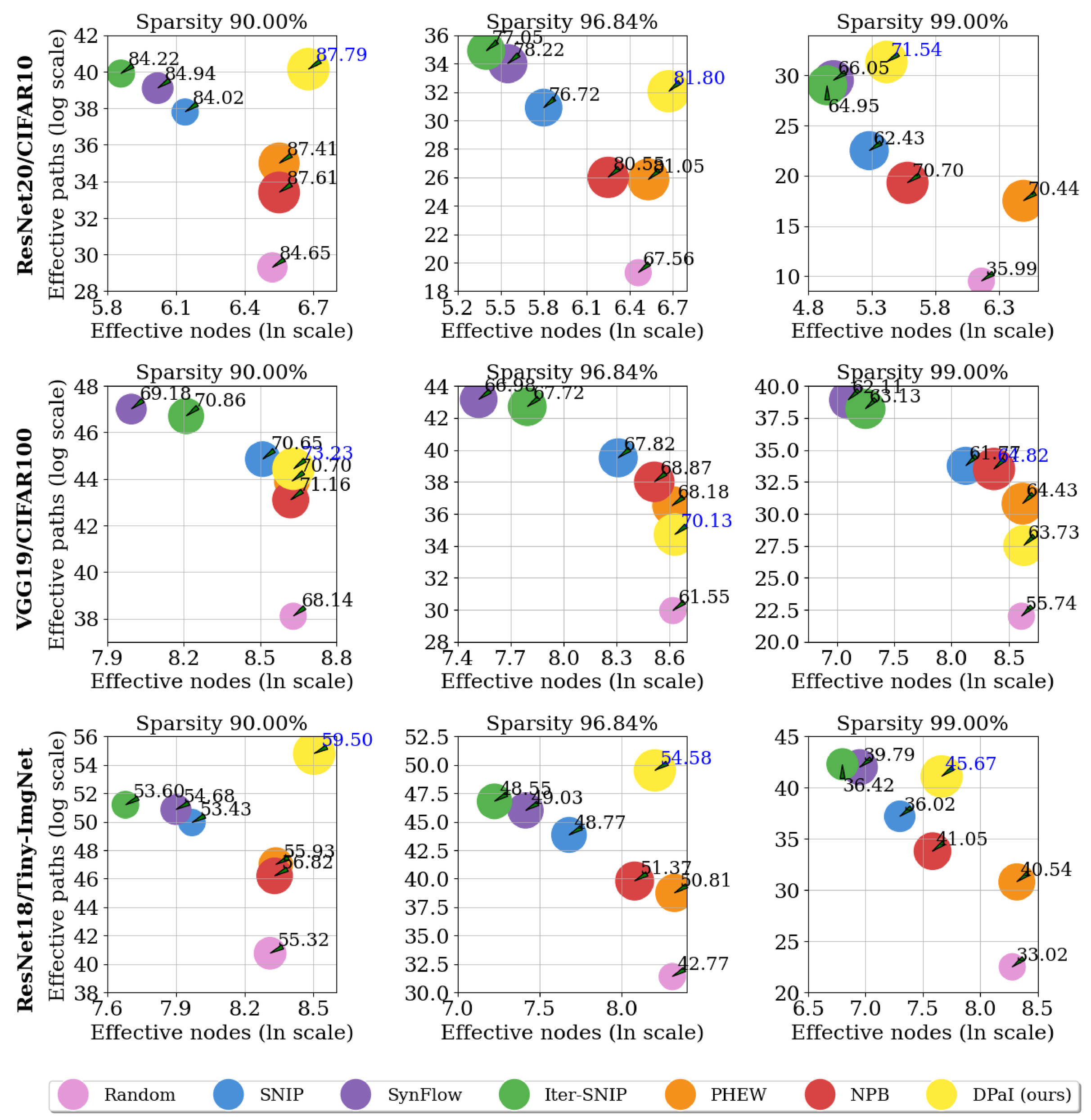


Figure 3: Easy to select effective hyperparameter from a variety of node-path balanced topologies.

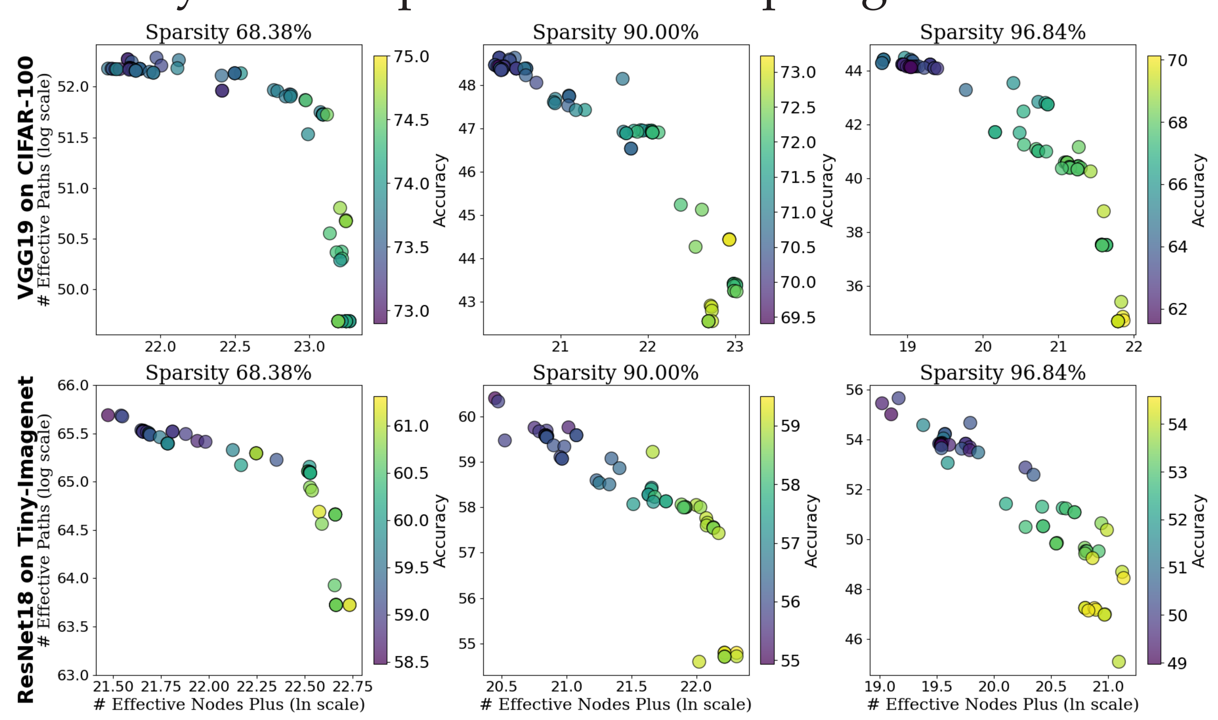


Figure 4: DPaI significantly outperforms PHEW and NPB in pruning speed under large-scale settings.