Vision Paper: Teacher-Guided One-Shot Pruning via Context-Aware Knowledge Distillation

Md. Samiul Alim[1] Sharjil Khan[1] Amrijit Biswas[1]
Fuad Rahman[2] Shafin Rahman[1] Nabeel Mohammed[1]

[1] Apurba-NSU R&D Lab,
Department of Electrical and Computer Engineering,
North South University, Dhaka, Bangladesh

[2] Apurba Technologies, Sunnyvale, CA 94085, USA

{samiul.alim01, sharjil.khan, amrijit.biswas01, shafin.rahman, nabeel.mohammed}@northsouth.edu
fuad@apurbatech.com

***Abstract*—Unstructured pruning remains a powerful strategy for compressing deep neural networks, yet it often demands iterative train–prune–retrain cycles, resulting in significant computational overhead. To address this challenge, we introduce a novel teacher-guided pruning framework that tightly integrates Knowledge Distillation (KD) with importance score estimation. Unlike prior approaches that apply KD as a post-pruning recovery step, our method leverages gradient signals informed by the teacher during importance score calculation to identify and retain parameters most critical for both task performance and knowledge transfer. Our method facilitates a one-shot global pruning strategy that efficiently eliminates redundant weights while preserving essential representations. After pruning, we employ sparsity-aware retraining with and without KD to recover accuracy without reactivating pruned connections. Comprehensive experiments across multiple image classification benchmarks, including CIFAR-10, CIFAR-100, and TinyImageNet, demonstrate that our method consistently achieves high sparsity levels with minimal performance degradation. Notably, our approach consistently outperforms state-of-the-art baselines such as EPG, and EPSD at high sparsity levels, while offering a more computationally efficient alternative to iterative pruning schemes like COLT. The proposed framework offers a computation-efficient, performance-preserving solution well-suited for deployment in resource-constrained environments.**

***Index Terms*—Pruning, Knowledge Distillation (KD), One-Shot Pruning, Lottery Ticket Hypothesis (LTH)**

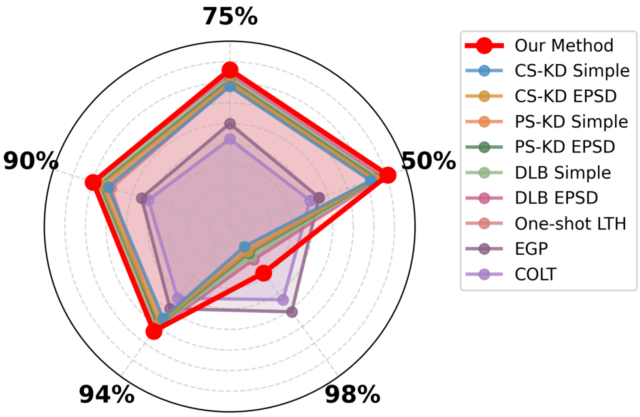


Fig. 1: Overview of accuracy comparison between our method and existing approaches across sparsity levels (50%, 75%, 90%, 94%, 98%). In the radar plot, a larger enclosed area indicates higher accuracy, where **Our Method** consistently achieves superior performance on CIFAE-10 datasets, clearly outperforming competing baselines. Detailed results including CIFAR-10 and other datasets are presented in the Results section.

I Introduction

Deep Neural Networks (DNNs), particularly Convolutional Neural Networks (CNNs), have achieved state-of-the-art performance across numerous computer vision tasks such as image classification [1], [2], object detection, and semantic segmentation [3]. However, these performance gains often come with a high computational and memory cost [2], limiting their deployment on resource-constrained environments such as mobile devices and edge platforms. To address this, the community has explored various model compression techniques aimed at reducing model complexity while retaining performance. Common approaches include parameter quantization [4], efficient architecture design [1], knowledge distillation (KD) [5], and network pruning [6], [7].

Among these, network pruning stands out for its ability to significantly reduce parameter count by removing redundant weights, filters, or neurons. It is broadly categorized into structured pruning [8], [9], which removes entire channels or filters for hardware efficiency, and unstructured pruning [10], [11], which eliminates individual weights and can achieve higher sparsity levels. While structured pruning improves practical deployment, it often sacrifices fine-grained control and performance. Unstructured pruning, in contrast, allows finer sparsity control and higher compression ratios but typically relies on less informative heuristics such as magnitude-based criteria [10]-to determine parameter importance.

Several limitations persist in the current landscape of unstructured pruning. First, importance estimation is often based on heuristics (e.g., weight magnitude) that fail to capture dynamic learning signals. This limitation is evident in approaches like the Lottery Ticket Hypothesis (LTH) [11], which require multiple train-prune-retrain iterations to isolate performant subnetworks, resulting in excessive computational cost. Second, while gradient-based criteria [12], [13] offer a more principled alternative by leveraging sensitivity, they often rely on noisy single-step gradients and are detached from any auxiliary supervision such as that from a teacher network. Third, knowledge distillation (KD) - a powerful framework to transfer knowledge from a large teacher to a smaller student - has been predominantly used as a post-pruning recovery tool [14], [15], rather than being integrated into the pruning process itself. Consequently, pruning decisions are made without the benefit of the teachers informative soft targets. While EPSD [16] addresses this challenge through self-distillation-aware early pruning, our approach extends this by using a pretrained teacher to actively guide the pruning process via task-aligned gradients.

Our goal is to address these limitations by introducing a teacher-guided unstructured pruning framework that leverages both gradient sensitivity and knowledge distillation in a unified pipeline. Unlike iterative methods such as LTH [11] and COLT [17], which require costly train-prune-retrain cycles, our approach enables efficient one-shot global pruning. While prior efforts like EPSD [16] incorporate self-distillation to improve early-stage pruning, they lack explicit teacher supervision during importance estimation. In contrast, we incorporate the teacher's guidance directly into the pruning signal through a distillation-aware loss function specifically, Context-Aware Kullback Leibler Divergence (CA-KLD) [18] augmented with logit normalization [19] for stable optimization. Gradient signals informed by this combined loss are aggregated using an exponential moving average with bias correction and used to rank parameter importance. The resulting scores enable aggressive one-shot pruning, followed by sparsity-aware retraining either with or without KD while strictly preserving the pruned structure.

The main contributions of our work are:

- A novel teacher-guided gradient importance metric that utilizes gradients derived from both task loss and an advanced KD loss (CA-KLD).
- A demonstration of integrating advanced KD not just for post-pruning recovery, but as an active component guiding the identification of critical parameters during the importance score calculation.
- Our One-Shot Global Pruning method minimizes computational cost compared to iterative pruning methods such as Lottery Ticket Hypothesis [11] and COLT [17]. Furthermore, our method outperforms six different methods evaluated in the EPSD [16], including CS-KD Simple, CS-KD EPSD, PS-KD Simple,PS-KD EPSD, DLB Simple, and DLB EPSD, as well as the EGP method [20].
- Extensive empirical evaluation on benchmark datasets (CIFAR-10, CIFAR-100, TinyImageNet) using standard architectures (ResNet18, ResNet34), demonstrating competitive performance compared to other methods.

II Related Work

Deep neural networks, particularly convolutional neural networks (CNNs), have achieved remarkable success across various domains, but their growing computational and memory demands pose challenges for deployment on resource-constrained devices [2]. This has motivated extensive research into model compression techniques, including parameter quantization [4], knowledge distillation (KD) [5], designing compact architectures [1], and network pruning [6], [7]. Our work focuses on pruning enhanced by KD.

Network Pruning: Network pruning reduces model complexity by removing redundant weights, neurons, or filters while minimizing performance loss. **Unstructured pruning** eliminates individual weights, often based on magnitude [10], [11], resulting in sparse networks that require specialized hardware or libraries for efficient inference. The Lottery Ticket Hypothesis (LTH) [11] suggests that dense networks contain sparse "winning ticket" subnetworks that can match the original models performance when trained from early initialization. **Structured pruning**, on the other hand, removes entire filters, channels, or layers [8], [9], producing smaller dense networks compatible with standard hardware, though potentially more detrimental to accuracy, especially in architectures like ResNets [15].

Pruning criteria include magnitude-based methods [10], [11], activation-based metrics such as APoZ [21] or channel entropy [9], [20], and gradient-based approaches [12], [13]. Our method belongs to the gradient-based category but incorporates teacher-student supervision, leveraging first-order gradients weighted by a task-distillation loss to estimate importance efficiently.

Knowledge Distillation for Compression: KD [5] transfers knowledge from a large teacher model to a smaller student using softened output probabilities, capturing richer inter-class relationships than hard labels alone. KD can improve the performance of compact models [22] and is often combined with pruning to help recover accuracy lost during compression [14], [15]. Advanced variants, such as Context Aware KLD (CA-KLD) [18], consider both teacher confidence and uncertainty to produce better-calibrated students.

Our Approach in Context: Unlike conventional pruning strategies based on magnitude, activation sparsity, or entropy [6], [9], [21], our framework embeds teacher-guided supervision directly into gradient-based importance estimation. By using smoothed first-order gradients aligned with CA-KLD soft targets [18], [19], we perform global one-shot pruning, reducing computational cost compared to iterative methods such as LTH [11] or COLT [17]. Our dual retraining regimes standard and KD-awarefurther ensure that pruned models retain high accuracy while preserving the benefits of compression.

III Methodology

A. Problem Formulation

Let $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ denote a labeled training dataset, where $x_i \in \mathbb{R}^{h \times w \times c}$ are input images and $y_i \in \{1, \ldots, C\}$ are the corresponding class labels. Where, N is the total number of training samples, C is the number of classes, and h, w, and c denote the height, width, and number of channels of each image, respectively. We aim to prune a dense student network θ_S under teacher supervision θ_T, such that the resulting sparse model maintains strong performance under a high sparsity budget. Unlike magnitude-based heuristics or post-hoc distillation, our approach integrates knowledge distillation (KD) directly into the pruning pipeline through teacher-informed importance score estimation. The teacher θ_T is a high-capacity model pre-trained on the classification task, while the student θ_S, initialized with pre-trained weights, is first fine-tuned via KD with a combined objective of cross-entropy and Context-Aware Kullback-Leibler Divergence (CA-KLD).

After fine-tuning, we compute a parameter-wise importance score for θ_S as

$$I_{\text{raw}} = \Psi(\theta_S, \theta_T; L), \tag{1}$$

where L is the total loss, L_{Total}, combining cross-entropy and CA-KLD, and $\Psi(\theta_S, \theta_T; L)$ is the importance score function. To stabilize against batch-wise noise, exponential moving average (EMA) smoothing with bias correction is applied (details in Subsection III-C). The stabilized score at iteration t is

$$I_{\text{final}} = \Phi_{\text{final}}(S; \gamma, t), \tag{2}$$

where $\gamma \in [0, 1)$ is the decay factor, and $\Phi_{\text{final}}(S; \gamma, t)$ denotes the EMA with bias correction. Using the aggregated scores, we construct a global pruning mask $M \in \{0, 1\}^{|\theta_S|}$ by retaining the top $(1 - r) \times 100\%$ weights, where r is the sparsity target. The pruned model is then obtained as $\theta_S^{\text{pruned}} = \theta_S \odot M$. Finally, retraining is performed (with and without KD) under the fixed mask to recover performance while strictly preserving sparsity. This enables one-shot pruning guided by both task and teacher signals, offering a more efficient and informed alternative to iterative pruning (see Fig. 2).

B. Knowledge Distillation Using CA-KLD Loss with Logit Normalization

We employ a knowledge distillation (KD) framework in which a compact student model is trained to mimic the behavior of a high-capacity teacher model. To enhance the effectiveness of distillation, we adopt the Context-Aware KullbackLeibler Divergence (CA-KLD) loss [23] in combination with logit normalization [19], ensuring the training signal remains both stable and informative. This same loss is later used to guide the pruning process through gradient-based importance estimation.

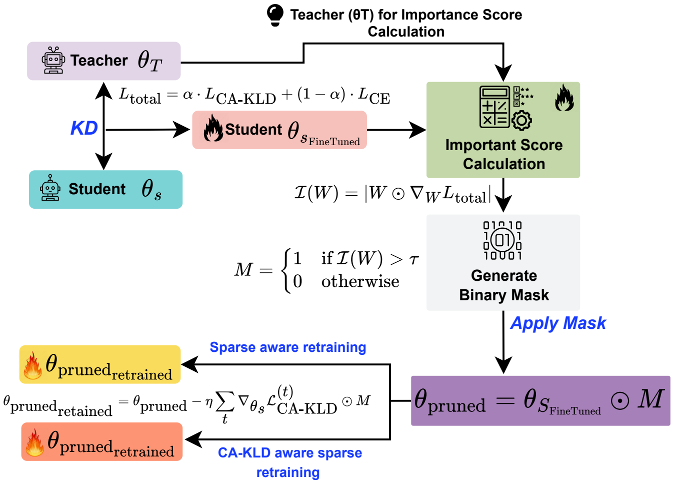

Teacher (θT) for Importance Score Calculation
Teacher θ_T
$L_{\text{total}} = \alpha \cdot L_{\text{CA-KLD}} + (1-\alpha) \cdot L_{\text{CE}}$
KD
Student $\theta_{s_{\text{FineTuned}}}$
Student θ_s
Important Score Calculation
$\mathcal{I}(W) = |W \odot \nabla_W L_{\text{total}}|$
$M = \begin{cases} 1 & \text{if } \mathcal{I}(W) > \tau \\ 0 & \text{otherwise} \end{cases}$
Generate Binary Mask
Apply Mask
Sparse aware retraining
$\theta_{\text{pruned}_{\text{retrained}}}$
$\theta_{\text{pruned}_{\text{retained}}} = \theta_{\text{pruned}} - \eta \sum_t \nabla_{\theta_s} \mathcal{L}_{\text{CA-KLD}}^{(t)} \odot M$
$\theta_{\text{pruned}} = \theta_{S_{\text{FineTuned}}} \odot M$
$\theta_{\text{pruned}_{\text{retrained}}}$
CA-KLD aware sparse retraining

Fig. 2: Overview of the proposed teacher-guided one-shot pruning framework. The student is first trained with a combined KD and task loss, then a gradient-based importance score is used to generate a binary pruning mask. The pruned model is retrained under two regimes: (1) standard fine-tuning and (2) KD-guided retraining, to recover performance while maintaining sparsity.

1) Logit Normalization and Temperature Scaling

Let $\mathbf{z}_T$ and $\mathbf{z}_S$ denote the teacher and student logits, respectively. Before applying softmax, we normalize these logits to reduce distributional variance and emphasize informative features:

$$\text{normalize}(\mathbf{z}) = \frac{\mathbf{z} - \mu_{\mathbf{z}}}{\sigma_{\mathbf{z}} + \epsilon}, \quad \mu_{\mathbf{z}} = \mathbb{E}[\mathbf{z}], \quad \sigma_{\mathbf{z}} = \sqrt{\mathbb{E}[(\mathbf{z} - \mu_{\mathbf{z}})^2]} \tag{3}$$

where ϵ ensures numerical stability. The normalized logits are then scaled by a temperature $T > 1$ to soften the probability distribution:

$$\mathbf{z}_T^{\tau} = \frac{\text{normalize}(\mathbf{z}_T)}{T}, \quad \mathbf{z}_S^{\tau} = \frac{\text{normalize}(\mathbf{z}_S)}{T} \tag{4}$$

2) Context-Aware Distillation Loss

Using the softened logits, we compute probability distributions via softmax:

$$\mathbf{P}_T = \text{softmax}(\mathbf{z}_T^{\tau}), \quad \mathbf{P}_S = \text{softmax}(\mathbf{z}_S^{\tau}) \tag{5}$$

The CA-KLD loss combines forward and reverse Kullback-Leibler divergences to ensure both models are aligned in terms of prediction confidence and uncertainty:

$$\mathcal{L}_{\text{Fwd-KL}} = \sum_{c=1}^{C} P_T(c) \log \frac{P_T(c)}{P_S(c)} \tag{6}$$

$$\mathcal{L}_{\text{Rev-KL}} = \sum_{c=1}^{C} P_S(c) \log \frac{P_S(c)}{P_T(c)} \tag{7}$$

$$\mathcal{L}_{\text{CA-KLD}} = \beta \cdot \mathcal{L}_{\text{Rev-KL}} + (1 - \beta) \cdot \mathcal{L}_{\text{Fwd-KL}} \tag{8}$$

To stabilize the gradient magnitude during training, the loss is scaled by the square of the temperature:

$$\mathcal{L}_{\text{CA-KLD}} \leftarrow \mathcal{L}_{\text{CA-KLD}} \cdot T^2 \tag{9}$$

3) Training Objective

The final training objective incorporates both the distillation signal and the supervised cross-entropy loss:

$$\mathcal{L}_{\text{Total}} = \alpha \cdot \mathcal{L}_{\text{CA-KLD}} + (1-\alpha) \cdot \mathcal{L}_{\text{CE}} \tag{10}$$

where $\alpha \in [0, 1]$ determines the balance between teacher-driven soft supervision and hard label learning. This combined objective serves not only to train the student effectively but also provides informative gradients for pruning via importance score computation.

C. Teacher-Guided Gradient Importance Calculation

This method identifies important weights for pruning by computing parameter importance scores based on the gradient flow induced by a joint training signal that combines both supervised and distillation objectives. As detailed in Section III-B, the total loss $\mathcal{L}_{\text{Total}}$ incorporates both the cross-entropy loss and the CA-KLD distillation loss, allowing the teacher model to guide the student not only during learning but also in identifying the parameters most critical for task performance. The pruning decision is thus driven by the interaction between parameter magnitudes and their gradient alignment with respect to this joint loss, and the resulting raw importance scores are stabilized over time using an Exponential Moving Average (EMA) with bias correction. The overall process of computing gradient-based importance is depicted in Fig. 3 and Algorithm 1.

To identify weights that are critical for both task performance and alignment with the teacher model, we compute teacher-guided gradient-based importance scores. As previously defined in Section III-B, the total loss $\mathcal{L}_{\text{Total}}$ integrates both supervised and distillation signals, enabling the teacher to influence not only the students learning process but also the assessment of parameter relevance. This formulation allows the pruning process to retain weights that are essential for task success and effective teacher-student alignment.

The corresponding gradient with respect to the model parameters W is then:

$$\nabla_W \mathcal{L}\text{Total} = \alpha \cdot \nabla_W \mathcal{L}\text{CA-KLD} + (1-\alpha) \cdot \nabla_W \mathcal{L}_{\text{CE}}, \tag{11}$$

where $\alpha \in [0, 1]$ balances the contributions of the distillation and supervised objectives. This gradient captures both the teachers guidance and the ground-truth supervision, allowing us to assess how essential each parameter is to the combined learning signal. This formulation corresponds to the function $\Psi(\theta_S, \theta_T; \mathcal{L})$ introduced in the problem formulation. To quantify the relevance of each weight, we compute the element-wise product between the weight and its gradient magnitude, yielding a raw importance score as introduced in the problem formulation:

$$I_{\text{raw}}(W) = |W \odot \nabla_W \mathcal{L}_{\text{Total}}| \,. \tag{12}$$

Because raw gradient values are inherently noisy across batches, we smooth them using an exponential moving average (EMA):

$$I_t(W) = \gamma \cdot I_{t-1}(W) + (1-\gamma) \cdot I_{\text{raw}}(W), \tag{13}$$

where $\gamma = 0.9$ controls the rate of decay and determines how much past importance values influence the current estimate. To account for the initialization bias introduced by EMA in early steps, we apply a bias correction factor that normalizes the smoothed score:

$$I_{\text{final}}(W) = \frac{I_t(W)}{1-\gamma^t}, \tag{14}$$

where t is the number of batches seen. This bias-corrected EMA corresponds to $\Phi_{\text{final}}(S; \gamma, t)$ as defined in the problem formulation. The resulting score $I_{\text{final}}(W)$ serves as a robust, teacher-guided measure of parameter importance and forms the basis for global pruning in our pipeline.

D. Global Pruning via Gradient Magnitude Thresholding

Perform a one-shot, globally-aware pruning of the model using precomputed teacher-guided importance scores I(W) (Section III-C), enabling high sparsity while preserving performance-critical parameters. Unlike iterative pruning, which gradually removes weights and retrains over multiple cycles, our method applies aggressive, one-shot sparsification by pruning a portion (e.g., 95%) of the model's weights in a single forward pass. The pruning decision is guided by gradient-based importance scores that encode both task supervision and teacher knowledge. While this one-shot approach offers efficiency and simplicity, it comes at the cost of an initial performance drop due to the severity of the pruning. To address this, the pruned model undergoes post-pruning retraining, allowing it to recover and adapt to its new sparse structure.

Step-by-Step Algorithm

1) Compute Global Threshold for Pruning

Let $v = \bigcup_l \text{vec}(I_l) \in \mathbb{R}^D$ denote the concatenated importance scores across all prunable weights in the network, where $\text{vec}(\cdot)$ flattens the layer-wise importance tensor I_l, and D is the total number of prunable parameters. To prune a target percentage $p\%$ of the student models weights, compute the global pruning threshold τ by identifying the $k = \lfloor (1-p) \cdot D \rfloor$-th smallest value in v: $\tau = \text{TopK}\big(v,\, k = \lfloor (1-p) \cdot D \rfloor\big)_{\text{min}}$. This ensures that exactly $p\%$ of the weights with the lowest importance scores are pruned globally.

2) Generate Binary Pruning Mask

The binary pruning mask $M(W)$ is generated by thresholding the importance scores $I(W)$, which quantify the contribution of each weight to both task performance and knowledge distillation fidelity. Given the computed global threshold τ, weights are retained if their importance exceeds τ, and pruned otherwise. Formally, for a weight W in layer l, the binary mask is defined as

$$M_l(W) = \begin{cases} 1 & \text{if } I_l(W) > \tau \quad \text{(retained)} \\ 0 & \text{otherwise} \quad \text{(pruned)} \end{cases}$$

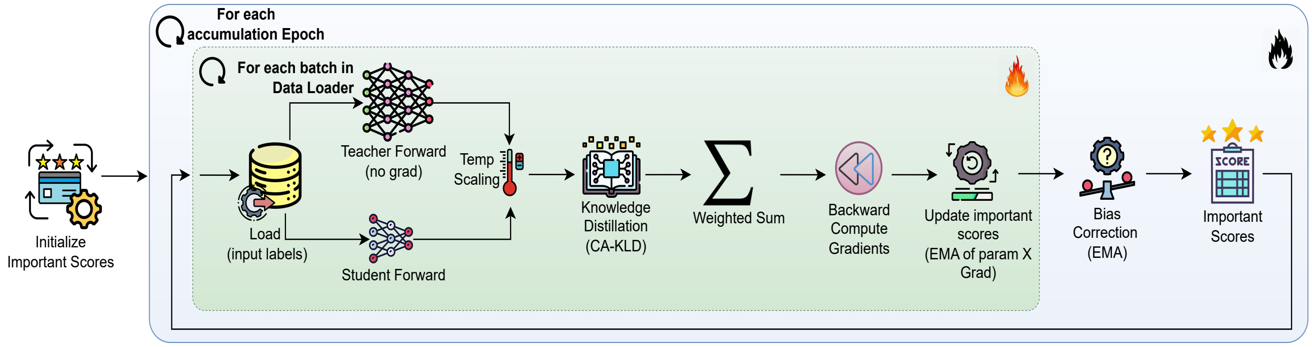

For each accumulation Epoch
For each batch in Data Loader
Initialize Important Scores
Load (input labels)
Teacher Forward (no grad)
Student Forward
Temp Scaling
Knowledge Distillation (CA-KLD)
Weighted Sum
Backward Compute Gradients
Update important scores (EMA of param X Grad)
Bias Correction (EMA)
SCORE
Important Scores

Fig. 3: Teacher Guided Important Score Computation

This enforces a global sparsity pattern while preserving the most critical weights, as identified by the teacher-guided gradient importance mechanism (see Section III-C).

3) Apply Mask to Prune the Model

The final pruned student model θ_S^{pruned} is obtained by element-wise masking of the original weights: $\theta_S^{\text{pruned}} = \theta_S \odot M$, where $\odot$ denotes element-wise multiplication.All weights with importance scores below the global threshold are masked out (set to zero), resulting in a sparse model.

Algorithm 1 Teacher Guided Gradient-Based Importance Score Calculation for Conv Layers

Requires: Teacher θ_T, Student θ_S, Dataset $\mathcal{D}$, Temperature τ, Distillation weight α, Mixing factor β_{prob}, Epochs E, Logits $\hat{z}$

Output: Layer-wise importance scores I

```
Initialize $I_l \leftarrow 0$ for all conv layers $l$ in $\theta_S$, momentum $\mu \leftarrow 0.9$, batch counter $b \leftarrow 0$
Set $\theta_T$ to eval mode, $\theta_S$ to train mode.
for epoch $= 1$ to $E$ do
    for $(x, y) \in \mathcal{D}$ do
        Move $(x, y)$ to Device, zero gradients in $\theta_S$
        with torch.no_grad(): $\hat{z}_T \leftarrow \theta_T(x)$
        $\hat{z}_S \leftarrow \theta_S(x)$
        Scale logits: $\hat{z}_T^{\tau} \leftarrow \hat{z}_T/\tau$, $\hat{z}_S^{\tau} \leftarrow \hat{z}_S/\tau$
        $L_{\text{KD}} \leftarrow \text{CA-KLD}(\hat{z}_S^{\tau}, \hat{z}_T^{\tau}, \beta_{\text{prob}}) \cdot \tau^2$
        $L_{\text{CE}} \leftarrow \text{CrossEntropy}(\hat{z}_S, y)$
        $L \leftarrow \alpha \cdot L_{\text{KD}} + (1 - \alpha) \cdot L_{\text{CE}}$
        Backprop: $L$.backward(), $b \leftarrow b + 1$
        for each conv layer $l$ in $\theta_S$ do
            if $\nabla_l \neq$ None then
                Compute layer contribution: $\Delta_l \leftarrow |\theta_l \cdot \nabla_l|$
                (EMA): $I_l \leftarrow \mu \cdot I_l + (1 - \mu) \cdot \Delta_l$
            end if
        end for
    end for
end for
for each layer $l$ do Normalization
    $I_l \leftarrow I_l/(1 - \mu^b)$
end for
return $I$
```

E. Sparse-Aware Retraining

After pruning, the student model is retrained to restore accuracy while strictly maintaining its sparsity pattern. To achieve this, we apply two key mechanisms: masked gradient updates and momentum correction.

First, during backpropagation, gradients corresponding to pruned weights are masked out to prevent updates. For each layer l, the masked gradient is computed as:

$$\nabla W_{\text{masked}}^l = \nabla W^l \odot M^l \tag{15}$$

where M^l is the binary pruning mask and $\odot$ denotes element-wise multiplication. This ensures that no learning signal reaches the pruned weights.

Second, to avoid reactivation of pruned weights through residual momentum, we apply momentum correction. For SGD optimizers with momentum μ, the velocity update is constrained by the same binary mask:

$$v^l(t+1) = \mu \cdot v^l(t) \odot M^l + \nabla W_{\text{masked}}^l \tag{16}$$

This guarantees that momentum is only accumulated for unpruned parameters, preserving sparsity throughout retraining.

1) KD-Aware Sparse Retraining

We further enhance retraining through knowledge distillation, using the CA-KLD loss with logit normalization and temperature scaling to align the sparse student with the dense teacher. This process guides the pruned student to mimic teacher outputs while adhering to the fixed pruning mask. The student parameters are updated as:

$$\theta_S^{\text{final}} = \theta_S^{\text{pruned}} - \eta \sum_t \nabla_{\theta_S} \mathcal{L}_{\text{CA-KLD}}(t) \odot M \tag{17}$$

where θ_S^{pruned} are the weights of the sparse student, $\mathcal{L}_{\text{KD}}$ is the CA-KLD distillation loss, η is the learning rate, and M is the global pruning mask.

By enforcing sparsity constraints during optimization and coupling it with teacher-guided supervision, this retraining strategy enables the student to recover or even surpass pre-pruning accuracy without violating its compressed architecture.

TABLE I: Teacher and Student Model's Accuracies Across Different Datasets

Dataset	ResNet50 (Teacher) (%)	ResNet18 (Student) (%)	ResNet34 (Student) (%)
CIFAR-10	95.40	95.92	96.50
CIFAR-100	80.89	81.12	82.77
TinyImageNet	78.46	62.19	

IV Experiments and Results

All experiments were conducted on a single NVIDIA RTX 3090 GPU (24 GB VRAM) with 16 vCPUs and 125 GB RAM. This setup provided sufficient computational resources to support both dense and sparse model training, ensuring reproducible results across pruning and distillation scenarios.

A. Datasets

We evaluate our approach on three widely used image classification benchmarks datasets:

CIFAR-10 [24]: CIFAR-10 contains 60,000 color images (32×32) across 10 balanced classes, split into 50,000 training and 10,000 test samples. Its low resolution and limited categories make it suitable for lightweight model evaluation.

CIFAR-100 [24]: CIFAR-100 has the same size and resolution as CIFAR-10 but with 100 classes (600 images each). This finer granularity and fewer samples per class pose greater challenges in learning discriminative features under data scarcity.

Tiny ImageNet [25]: Tiny ImageNet includes 100,000 images (64×64) over 200 classes, with 500 training samples each. Its higher resolution and class diversity approximate real-world recognition tasks while remaining computationally feasible, making it a common benchmark for efficient architectures.

B. Training Settings

1) Gradient Importance Calculation Setting

Gradient importance scores are computed with fixed hyperparameters for stability and reproducibility. The balancing factor $\alpha = 0.7$ emphasizes the CA-KLD loss over cross-entropy, aligning student predictions with the teacher while maintaining task performance. The bidirectional KL in CA-KLD uses $\beta = 0.5$, giving equal weight to preserving teacher knowledge (forward KL) and reducing student overconfidence (reverse KL). Importance scores are accumulated over 3 epochs and smoothed using exponential moving averages (EMA, $\gamma = 0.9$) to reduce batch-level noise. For non-KD retraining, a temperature $T = 5.0$ stabilizes gradients, while KD retraining uses two configurations: $(T = 3.0, \alpha = 0.7)$ and $(T = 5.0, \alpha = 0.7)$.

2) Post-Pruning Training Setting

To recover accuracy after aggressive pruning, pruned models are retrained with: (i) standard fine-tuning without KD using early stopping (patience=5 epochs), and (ii) KD-based fine-tuning using the total loss L_{Total} combining task-specific loss and CA-KLD. Two KD configurations mirror the gradient importance settings. This approach ensures balanced supervision from ground-truth labels and teacher outputs, enhancing performance across all sparsity levels while maintaining stability and generalization.

TABLE II: Performance of Sparse Student (ResNet18) when Teacher (ResNet50) across CIFAR-10, CIFAR-100, and Tiny-ImageNet (Our Method)

Dataset	Sparsity (%)	Acc (%)	# Epoch (Retrain)
CIFAR-10	98.41	90.79	39
	93.92	94.28	30
	90.00	94.97	10
	80.00	95.44	18
	75.00	95.62	9
	50.46	96.08	8
CIFAR-100	98.01	67.06	24
	93.53	76.35	25
	90.55	77.29	26
	80.60	79.54	18
	74.69	80.14	19
	50.75	80.99	10
TinyImageNet	97.56	50.64	19
	93.11	53.79	19
	89.22	52.36	8
	79.31	56.54	20
	74.35	57.42	13
	50.02	59.29	22

V Results

A. Teacher and Student Model Performance Before Pruning

Table I shows performance of the teacher (ResNet50) and student models (ResNet18, ResNet34) prior to pruning. All models were initialized with ImageNet-pretrained weights and fine-tuned on the target datasets, with students trained under KD supervision. On CIFAR-10, ResNet50 achieved 95.40%, while ResNet18 and ResNet34 slightly outperformed with 95.92% and 96.50%, respectively. For CIFAR-100, ResNet50 reached 80.89%, compared to 81.12% for ResNet18 and 82.71% for ResNet34. On TinyImageNet, ResNet50 achieved 78.46%, while ResNet18 obtained 62.19%.

B. Main Results

We evaluate the proposed KD-guided pruning framework, which employs the CA-KLD loss function with $T = 3$ and $\alpha = 0.7$, on CIFAR-10, CIFAR-100, and TinyImageNet. Using ResNet18 as the student distilled from a ResNet50 teacher, the method demonstrates strong robustness across varying sparsity levels, from moderate compression to extreme pruning, as summarized in Table II. On CIFAR-10, the framework maintains high accuracy even under severe pruning, achieving 90.79% at 98.41% sparsity. Accuracy improves as sparsity decreases, reaching 94.97% at 90.00% and peaking at 96.08% at 50.46%. This highlights KDs role in mitigating the performance degradation typically observed with aggressive pruning. For CIFAR-100, the student shows resilience on this more challenging dataset, recording 66.32% at 98.01% sparsity. Accuracy increases steadily with reduced sparsity, attaining 81.01% at 50.75%. These results indicate that the framework effectively preserves generalization in classification tasks. On TinyImageNet, the model achieves 50.64% accuracy at 97.56% sparsity, improving to 59.29% at 50.02%. This demonstrates adaptability to larger-scale tasks while maintaining discriminative power. Overall, the results confirm that KD-guided pruning enables deep networks to achieve high compression without sacrificing accuracy. Additional evaluations, including ResNet34 experiments and non-KD baselines, are presented in the Appendix table VII,VIII.

VI State-of-the-art Comparison

A. Comparison With six Different Methods

We evaluate our pruning-and-distillation framework against six different method baselines: CS-KD Simple [26] [16], CS-KD EPSD [16], PS-KD Simple [27] [16], PS-KD EPSD [16], DLB Simple [28] [16], and DLB EPSD [16]. All experiments use ResNet-18 on CIFAR-100, TinyImageNet, and CIFAR-10, under five target sparsities. Results are displayed in Figures 4, 5, and 6.

1) Comparison On CIFAR-100

As shown in Figure 5, our method outperforms all six competitive baselines across most sparsity levels. At 36% sparsity, we achieve 81.74% accuracy, exceeding the best baseline (DLB EPSD: 81.32%) by +0.42 percentage points (pp). At 59% sparsity, our accuracy remains strong at 81.32%, outperforming all other methods including DLB EPSD (79.54%, +1.78 pp). At 79% sparsity, our method achieves 79.90%, maintaining a slight lead over DLB EPSD (79.32%) and PS-KD EPSD (78.72%). Even under high sparsity (90%), our method reaches 78.49% accuracy, again outperforming DLB EPSD (77.87%) and demonstrating resilience under compression. At the extreme 95% sparsity level, our model retains 74.68% accuracyonly slightly trailing PS-KD EPSD (75.54%) by 0.86 pp, while significantly outperforming DLB EPSD (46.86%) and all CS-KD variants.

2) Comparison On TinyImageNet

Figure 6 shows the top-1 accuracy of ResNet18 across varying sparsity levels on Tiny ImageNet. At 36% sparsity, our method achieves 61.35%, outperforming the strongest baseline (DLB EPSD: 58.21%) by +3.14 percentage points (pp). At 59% sparsity, we retain 59.96% accuracy again ahead of DLB EPSD (57.98%) by +1.98 pp. As the pruning becomes more aggressive, our method maintains its advantage: at 79% sparsity, we achieve 57.90%, exceeding the closest baseline (PS-KD EPSD: 55.19%) by +2.71 pp. Under extreme sparsity, our framework demonstrates notable resilience. At 90% sparsity, we record 53.48% accuracy, outperforming DLB EPSD (53.33%) and all other baselines. At 95% sparsity, our method reaches 53.42%, a significant +3.25 pp improvement over DLB EPSD (50.17%) and nearly +10 pp over the weakest-performing KD variants.

3) Comparison On CIFAR-10

As shown in Figure 4, our method achieves top-1 accuracy across all sparsity levels when benchmarked against state-of-the-art pruning and distillation baselines, including CS-KD, PS-KD, and DLB (both simple and EPSD variants). At 36% and 59% sparsity, our method yields 96.05% and 95.84% accuracy, respectively, outperforming all baselines. Even under higher sparsity constraints, we retain leading performance: 95.68% at 79%, 94.83% at 90%, and 95.88% at 95% sparsity. Notably, our method surpasses the best baseline (DLB-EPSD) at all levels, including the most extreme 95% sparsity setting where it improves by nearly +0.89%. Overall, across three benchmarks our approach achieves the best or highly competitive accuracy at low-to-moderate sparsities, and remains on par with state-of-the-art methods even under extreme pruning. This demonstrates the effectiveness of our gradient-based importance scoring and joint self-distillation strategy in preserving model performance under high compression.

TABLE III: Comparison of ResNet18 performance with COLT against Hossain et al. [17] across different datasets in terms of accuracy and latency.

Dataset	Ours (T=3/5, α=0.7)			Hossain et al. [17]		
	Sparsity. (%)	Acc. (%)	Latency. (min)	Sparsity. (%)	Acc. (%)	Latency. (min)
CIFAR-10	97.7	91.87	**27.82**	97.7	**92.40**	276
TinyImageNet	97.4	51.14	**42.43**	97.4	**53.90**	1756
CIFAR-100	97.4	**68.66**	**19.76**	97.4	68.40	355

B. Comparison with COLT

To assess the effectiveness of our teacher-guided pruning framework at extreme sparsity levels, we benchmark it against Cyclic Overlapping Lottery Tickets (COLT) by Hossain et al. [17], a state-of-the-art iterative pruning method that identifies overlapping sparse subnetworks through cyclic training and pruning phases. Table III details the top-1 accuracy and computational latency for ResNet-18 on CIFAR-10, TinyImageNet, and CIFAR-100 datasets, comparing our best results against COLT-2 at matched sparsity ratios of 97.7% for CIFAR-10 and 97.4% for TinyImageNet and CIFAR-100. The empirical results reveal a compelling trade-off favoring our approach. While COLT-2 attains marginally higher accuracy on CIFAR-10 (92.40% vs. our 91.87%, a 0.53% gap) and TinyImageNet (53.90% vs. 51.14%, a 2.76% gap), our method surpasses it on CIFAR-100 (68.66% vs. 68.40%, a 0.26% advantage). These accuracy differences are relatively minor, particularly considering the datasets' complexities, but the disparities in computational efficiency are stark. COLT-2's iterative cyclic process demands extensive training across multiple overlapping subnetworks, resulting in latencies of 276 minutes for CIFAR-10, 1756 minutes for TinyImageNet, and 335 minutes for CIFAR-100 factors of approximately 10×, 41×, and 18× higher than our respective latencies of 27.82, 42.43, and 19.76 minutes. For fair judgment, all latency measurements were conducted on an RTX 3090 GPU, consistent with the hardware used in COLT. This efficiency stems from our one-shot global pruning strategy, which leverages teacher-guided gradients (as detailed in Section III-C) to perform a single, informed pruning step followed by concise retraining. In contrast, COLT's multi-phase iterations incur prohibitive overhead, limiting its scalability for rapid prototyping or deployment in time-sensitive scenarios. By achieving comparable or superior accuracy with drastically reduced computational demands, our framework demonstrates clear superiority, offering a more practical and resource-efficient solution for

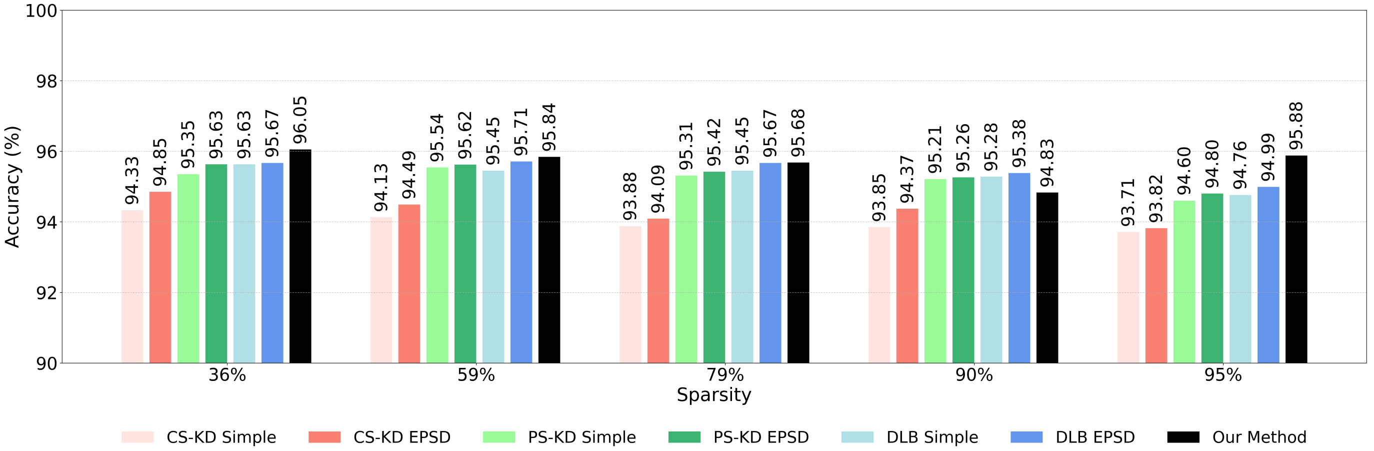


Fig. 4: Top-1 accuracy of sparse ResNet-18 on CIFAR-10 at varying sparsity levels (36% to 95%), comparing our method against six baselines: CS-KD Simple [16] [26], CS-KD EPSD [16], PS-KD Simple [16] [27], PS-KD EPSD [16], DLB Simple [16] [28], and DLB EPSD [16]. Our method consistently outperforms all baselines at high and moderate sparsity levels.

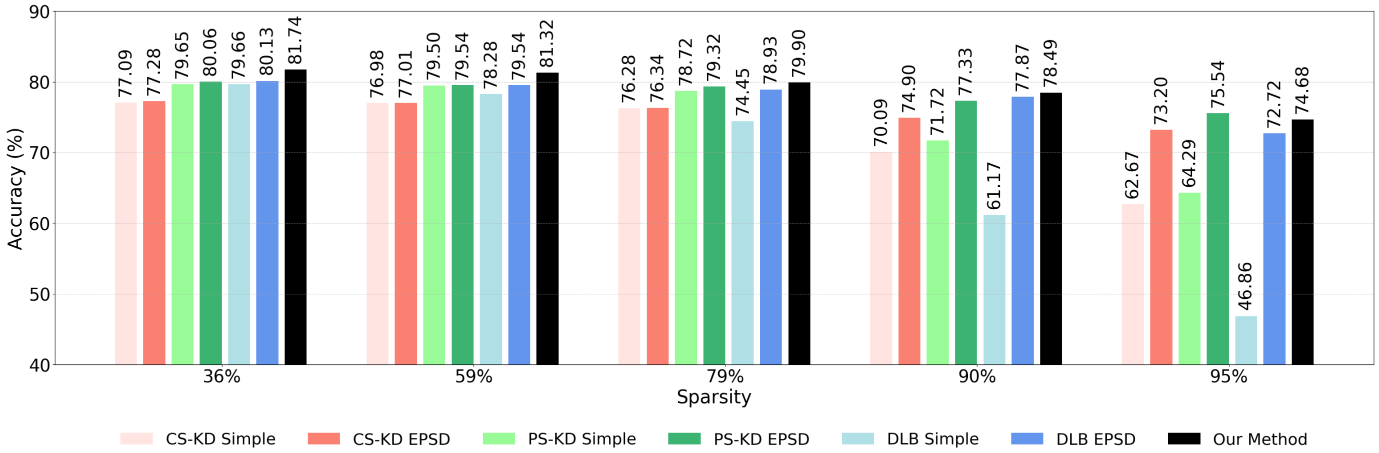


Fig. 5: Top-1 accuracy comparison of ResNet-18 on CIFAR-100 across five sparsity levels (36% to 95%). Results are benchmarked against six baselines: CS-KD Simple [16] [26], CS-KD EPSD [16], PS-KD Simple [16] [27], PS-KD EPSD [16], DLB Simple [16] [28] , and DLB EPSD [16]. Our method achieves consistently higher accuracy, especially at moderate sparsity levels.

high-sparsity model compression in real-world, constrained environments.

C. Comparison with One-Shot Lottery Ticket Hypothesis

To further evaluate the efficacy of our teacher-guided pruning framework, we conduct a direct comparison against a one-shot variant of the Lottery Ticket Hypothesis (LTH) [11]. In the one-shot LTH setting, the dense network is not fully pretrained; instead, starting from the random initialization θ_0, we first train for a short warm-up of E_w epochs, then apply a single round of pruning to reach the target sparsity. The surviving weights are reset to their initial values θ_0, and the resulting sparse subnetwork is trained to convergence. This baseline preserves the computational appeal of non-iterative pruning while aligning with the original LTH retraining protocol. Table IV reports the top-1 accuracy of our approach against the one-shot LTH baseline across different sparsity levels on CIFAR-10, CIFAR-100, and TinyImageNet using ResNet-18. Overall, our method provides consistent improvements over one-shot LTH, with the gap becoming more noticeable at higher sparsities. For example, at 98.41% sparsity on CIFAR-10, our framework achieves 90.79% accuracy compared to 89.47% for LTH, a gain of 1.32%. Similar advantages are observed on TinyImageNet, where at 50% sparsity we obtain 60.93% accuracy versus 56.91% for LTH, reflecting the benefit of teacher-guided importance estimation in more challenging datasets. Even at moderate sparsity levels (e.g., 50%), our method matches or slightly improves upon LTH, showing that the integration of KD signals into the pruning criterion provides stable performance gains without the need for iter-

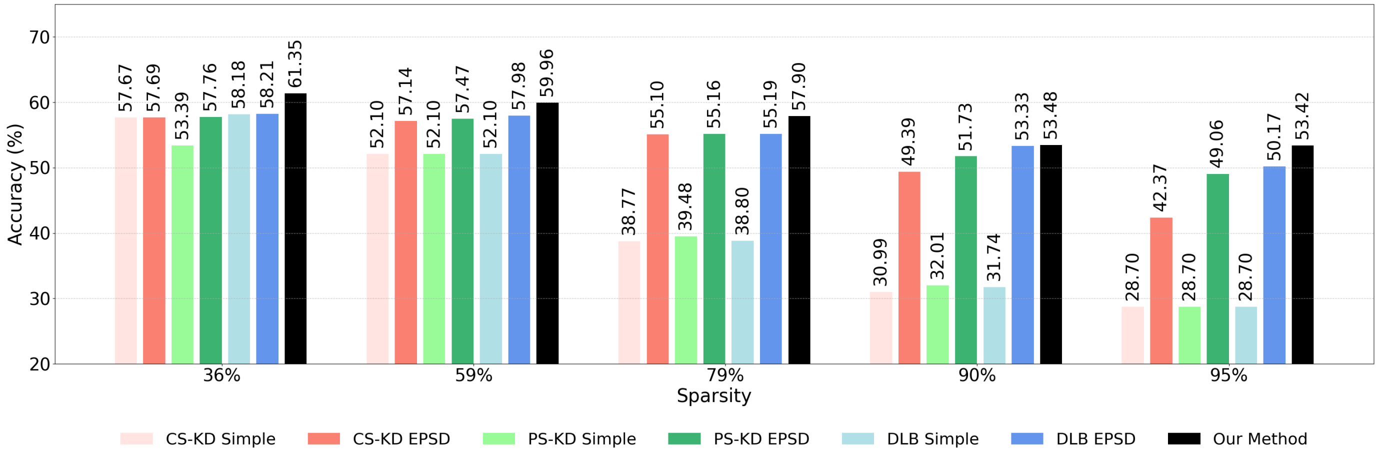


Fig. 6: Accuracy comparison of ResNet-18 on TinyImageNet across five sparsity levels. Our method consistently surpasses six distillation-based baselines: CS-KD Simple [16] [26], CS-KD EPSD [16], PS-KD Simple [16] [27], PS-KD EPSD [16], DLB Simple [16] [28], and DLB EPSD [16]. Even under high sparsity, our method maintains competitive accuracy, demonstrating effective knowledge preservation.

TABLE IV: Comparison of Our Method and One-Shot LTH across different sparsity levels

Dataset	Sparsity (%)	Our Method (Acc)	One-Shot LTH (Acc)
CIFAR-10			
	98.41	**90.79**	89.47
	93.92	**94.40**	94.05
	75.00	**95.62**	94.85
	50.00	**96.08**	95.28
TinyImageNet			
	98.41	**50.64**	47.24
	93.92	53.79	**55.43**
	74.35	**58.58**	56.99
	50.02	**60.93**	56.91
CIFAR-100			
	98.01	**67.06**	65.20
	80.60	**79.54**	79.11
	74.69	**80.14**	79.63
	50.63	**81.01**	80.27

ative refinement. These results demonstrate that the use of teacher-informed gradients during the calculation of important scores leads to more reliable sparse models compared to other pruning methods. Additional, comparison with entropy-guided pruning (EGP) approach by Liao et al. [20] given in the Appendix table VI and Fig 7

VII Ablation Study

To gain deeper insights into the effectiveness of knowledge distillation in the pruning setting, we compare student performance both without KD and with KD at two temperatures ($T = 3$ and $T = 5$). Table V summarizes results for ResNet-18 students across CIFAR-10, CIFAR-100, and TinyImageNet under different sparsity levels. As discussed earlier, pruning alone (w/o KD) causes a significant loss in accuracy, particularly for high sparsity ratios and more complex datasets. For example, CIFAR-10 accuracy falls to 86.9% at 98% sparsity, while CIFAR-100 and TinyImageNet degrade to 62.5% and 47.7%, respectively. Introducing KD clearly alleviates this issue. Across all datasets, KD consistently boosts performance, even at extreme sparsity. On CIFAR-10, KD recovers almost +4% at 98% sparsity, raising accuracy from 86.9% to 90.8%. Similar improvements are observed on CIFAR-100 (from 62.5% to 67.1%) and TinyImageNet (from 47.7% to 50.6%). At moderate sparsity (e.g., 75%–90%), KD nearly closes the gap with dense models, demonstrating its robustness. Comparing temperatures, we find that both $T = 3$ and $T = 5$ provide substantial benefits, with $T = 3$ performing slightly better on CIFAR-10 and CIFAR-100, while $T = 5$ yields marginal gains on TinyImageNet. This suggests that the optimal temperature may depend on dataset complexity, but in all cases, KD stabilizes training and preserves accuracy under high compression.

Overall, these results validate that while pruning alone is insufficient, our context-aware KD strategy enables highly sparse student models to remain competitive with their dense counterparts.

VIII Discussion

Our one-shot global pruning pipeline, leveraging a teacher-guided CA-KLD signal and smoothed gradient-based importance scores, maintains high accuracy even at extreme sparsity (>98%). For instance, on CIFAR-10 at 98.41% sparsity, we achieve 90.79% accuracy, surpassing EGP, which only outperforms us above 98% sparsity due to its entropy criterion preserving critical neurons. We outperform six KD-based baselines (CS-KD Simple & EPSD, PS-KD Simple & EPSD, DLB Simple & EPSD) by introducing KD in important score calculation, unlike their post-pruning KD application. Compared to iterative methods like COLT, our approach delivers comparable accuracy with significantly lower latency (e.g., 27.82 vs. 276 minutes on CIFAR-10 at 97.7% sparsity).

TABLE V: Ablation study: Accuracy (%) of ResNet-18 students without KD and with KD ($T = 3, 5$) across sparsity levels.

Dataset	Sparsity (%)	Acc. w/o KD (%)	Acc. w/ KD (T=3) (%)	Acc. w/ KD (T=5) (%)
CIFAR-10	98.41	86.99	**90.79**	90.66
	93.92	92.97	94.28	**94.40**
	90.00	94.12	**94.97**	94.83
	80.00	95.21	**95.44**	95.29
	75.00	95.46	**95.62**	95.60
	50.46	96.06	**96.08**	96.03
TinyImgNet	97.56	47.71	**50.64**	50.07
	93.11	51.74	**53.79**	52.11
	89.22	52.53	**52.36**	52.19
	79.31	53.41	56.54	**57.95**
	74.35	54.39	57.42	**58.58**
	50.02	58.53	59.29	**60.93**
CIFAR-100	98.01	62.49	66.32	**67.06**
	93.53	72.00	75.34	**76.35**
	90.55	73.94	**77.40**	77.29
	80.60	77.61	79.44	**79.54**
	74.69	78.95	79.78	**80.14**
	50.75	80.69	**81.01**	80.99

Against one-shot LTH, our superior importance scoring yields better results. Slight sensitivity to KD temperature (T=3 often outperforms T=5) and minor accuracy dips at extreme sparsity suggest exploring hybrid entropy-gradient criteria or adaptive-T schedules. Our framework effectively balances compression, accuracy, and efficiency, making it ideal for resource-constrained edge deployments.

IX Conclusion

We presented a teacher-guided one-shot pruning framework that integrates knowledge distillation directly into pruning through a gradient-based importance metric. By combining cross-entropy and context-aware KL divergence with logit normalization, we derive stable importance scores that enable aggressive and effective global pruning. Across CIFAR-10, CIFAR-100, and TinyImageNet, our method preserves accuracy even under extreme sparsity and surpasses state-of-the-art pruning and distillation baselines. Compared to iterative pruning, it achieves competitive results at a fraction of the cost, offering a practical and scalable solution for resource-constrained environments. This work establishes the value of KD into pruning itself and opens pathways for extending the framework to broader architectures and tasks.

References

[1] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," https://arxiv.org/abs/1602.07360, 2016, arXiv preprint arXiv:1602.07360.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 40, no. 4, pp. 834–848, 2018, arXiv:1606.00915v2 (2017).

[4] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," in *Low-Power Computer Vision*, D. Cavagnino, D. Demarchi, M. Martina, and G. Masera, Eds. Chapman and Hall/CRC, 2022, pp. 291–326.

[5] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," https://arxiv.org/abs/1503.02531, 2015, arXiv preprint arXiv:1503.02531, NIPS Deep Learning Workshop.

[6] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," in *International Conference on Learning Representations (ICLR)*, 2016, arXiv:1510.00149 (2015).

[7] D. Blalock, J. J. G. Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *Proceedings of Machine Learning and Systems (PMLSys)*, vol. 2, pp. 129–146, 2020.

[8] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," in *International Conference on Learning Representations (ICLR) Workshop Track*, 2017, arXiv:1608.08710 (2016).

[9] J.-H. Luo and J. Wu, "An entropy-based pruning method for CNN compression," https://arxiv.org/abs/1706.05791, 2017, arXiv preprint arXiv:1706.05791.

[10] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," in *Advances in Neural Information Processing Systems (NIPS)*, vol. 28, 2015, pp. 1135–1143.

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

[12] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Advances in Neural Information Processing Systems (NIPS)*, D. S. Touretzky, Ed., vol. 2, 1990, pp. 598–605.

[13] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in *International Conference on Learning Representations (ICLR)*, 2017.

[14] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *International Conference on Learning Representations (ICLR)*, 2018.

[15] N. Aghli and E. Ribeiro, "Combining weight pruning and knowledge distillation for CNN compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021, pp. 2569–2576.

[16] D. Chen, N. Liu, Y. Zhu, Z. Che, R. Ma, F. Zhang, X. Mou, Y. Chang, and J. Tang, "Epsd: Early pruning with self-distillation for efficient model compression," 2024. [Online]. Available: https://arxiv.org/abs/2402.00084

[17] M. I. Hossain, M. Rakib, M. M. L. Elahi, N. Mohammed, and S. Rahman, "Colt: Cyclic overlapping lottery tickets for faster pruning of convolutional neural networks," 2022.

[18] Z. Liu, X. Zhang, J. Ye, and H. Liu, "Context-aware knowledge distillation for deep model compression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 8, 2023, pp. 9599–9607.

[19] S. Sun, W. Ren, J. Li, R. Wang, and X. Cao, "Logit standardization in knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 731–15 740.

[20] Z. Liao, V. Quétu, V.-T. Nguyen, and E. Tartaglione, "Can unstructured pruning reduce the depth in deep neural networks?" in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, 2023, pp. 1402–1406.

[21] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," https://arxiv.org/abs/1607.03250, 2016, arXiv preprint arXiv:1607.03250.

[22] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," in *International Conference on Learning Representations (ICLR)*, 2015, arXiv:1412.6550 (2014).

[23] D. Du, Y. Zhang, S. Cao, J. Guo, T. Cao, X. Chu, and N. Xu, "Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation," 2024. [Online]. Available: https://arxiv.org/abs/2402.10631

[24] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[25] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, p. 3, 2015.

[26] S. Yun, J. Park, K. Lee, and J. Shin, "Regularizing class-wise predictions via self-knowledge distillation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 876–13 885.

[27] J. Kim, S. Park, and N. Kwak, "Paraphrasing complex network: Network compression via factor transfer," in *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[28] C. Shen, X. Wang, Y. Song, Y. Li, and J. Yin, "Dynamic label-based distillation," in *Advances in Neural Information Processing Systems*, vol. 35, 2022.

Appendix

X Additional Results

A. In detail Results on Sparse ResNet18

Results on CIFAR-10: As shown in Table II, ResNet18 with KD consistently maintains high accuracy across sparsity levels. At the extreme compression of 98.41%, the model achieves 90.79%, while at 90.00% sparsity accuracy reaches 94.97%. Even at the lowest sparsity (50.46%), KD preserves a strong accuracy of 96.08%. These results illustrate that the proposed KD-guided pruning framework allows the student network to retain remarkable performance despite aggressive parameter reduction, with $T = 3$ playing a key role in stabilizing performance under extreme sparsity.

Results on CIFAR-100: On the more challenging CIFAR-100 dataset, KD enables the student model to maintain competitive accuracy across a wide range of sparsity levels. At 98.01% sparsity, the model still achieves 66.32%, and performance steadily improves as sparsity decreases: 75.34% at 92.53%, 77.40% at 90.55%, and 79.48% at 80.60%. The best accuracy of 81.01% is observed at 50.75% sparsity. This trend demonstrates the resilience of the proposed framework, where even under extreme compression the student model retains strong generalization, and accuracy progressively recovers as more parameters are preserved.

Results on TinyImageNet: The effectiveness of the framework is further validated on TinyImageNet, which poses a more complex recognition task. At extreme sparsity (97.56%), KD achieves 50.64% accuracy, showing the models ability to withstand heavy pruning. As sparsity is reduced, performance improves notably, reaching 58.79% at 93.11%. At moderate sparsity (89.22%), accuracy stabilizes at 52.36%, while at lower sparsity levels KD provides consistent gains, with 56.54% at 79.31% and 58.49% at 74.35%. The best accuracy of 59.29% is obtained at 50.02% sparsity. These results confirm that the KD-guided pruning framework generalizes effectively to large-scale datasets, ensuring stable performance across both extreme and moderate compression regimes.

Overall, the results across CIFAR-10, CIFAR-100, and TinyImageNet demonstrate the robustness and adaptability of the proposed KD-guided pruning framework. By leveraging the CA-KLD objective with $T = 3$, the method consistently recovers or even improves student model accuracy under severe sparsity, establishing its effectiveness as a general solution for compressing deep networks while preserving discriminative power.

B. Results On Sparse Resnet34

Results On CIFAR-10: Table VIII presents the results for ResNet34 across sparsity levels from 50.47% to 98.44%. At the highest sparsity of 98.44%, KD with $T = 3$ achieves 92.35%, a 6.13 percentage-point improvement over the non-KD accuracy of 86.22%. At 90.02% sparsity, KD improves performance from 94.93% (non-KD) to 95.71% ($T = 3$). At 50.47% sparsity, the model retains high accuracy with 96.52% even without KD, while KD retraining sustains competitive performance. Interestingly, at 80.02% sparsity, both $T = 3$ and $T = 5$ achieve 96.12%, slightly outperforming the non-KD accuracy of 96.16%. This suggests smoother logits (higher T) may provide subtle gains in denser models. Overall, ResNet34 consistently outperforms ResNet18 under extreme pruning, achieving 92.35% at 98.44% sparsity (KD, $T = 3$) versus 90.79% for ResNet18, highlighting the advantage of deeper student networks in retaining performance under high sparsity conditions.

TABLE VI: Accuracy Comparison for Sparse ResNet18 with EPG

		Our Method		Liao et al. [20]
Dataset	Sparsity (%)	Acc. T=3 (%)	Acc. T=5 (%)	Acc. (%)
CIFAR-10				
	50.00	**96.08**	96.03	92.56
	75.00	**95.62**	95.60	93.00
	93.92 / 93.8	94.28	**94.40**	92.93
	98.41 / 98.4	90.79	90.66	**93.12**
TinyImageNet				
	50.02 / 50.00	59.29	**60.93**	41.70
	74.35 / 75.00	57.42	**58.58**	41.24
	93.92 / 93.8	**53.79**	52.11	41.86
	98.41 / 98.4	**50.64**	50.07	37.44

Results On CIFAR-100: Table VII compares the performance of the deeper ResNet34 student model under the same teacher and conditions. At 98.22% sparsity, KD yields accuracies of 72.38% ($T = 3$) and 72.79% ($T = 5$), improving significantly over the 66.14% baseline. At 90.75% sparsity, performance rises to 81.09% ($T = 3$) and 81.44% ($T = 5$), versus 77.89% without KD. With sparsity at 80.77%, KD achieves 82.21% ($T = 3$) and 81.91% ($T = 5$), outperforming the baseline of 81.09%. At 76.56% sparsity, accuracies of 82.21% ($T = 3$) and 82.25% ($T = 5$) are recorded, compared to 81.70% without KD. Interestingly, at the lowest sparsity level (50.86%), KD slightly underperforms the baseline of 83.18%, with 83.12% ($T = 3$) and 82.83% ($T = 5$). These results highlight ResNet34's stronger baseline and consistent performance with KD, especially under higher sparsity, while maintaining competitive accuracy as sparsity decreases.

XI Additional Comparison Result

A. Comparison with EGP.

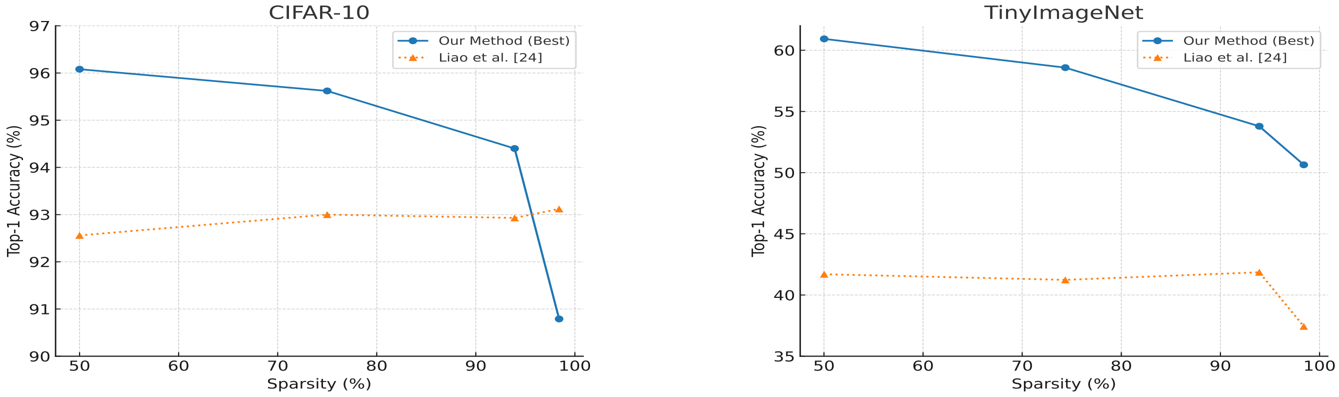
We also compare our method with the entropy-guided pruning(EGP) approach by Liao et al. [20] across various sparsity levels for ResNet18 on CIFAR-10 and TinyImageNet, as shown in Table VI. Our method uses KD configurations with $T = 3$ and $T = 5$, both with $\alpha = 0.7$, and reports the best accuracy between the two. and line plot visualization in Figure 7 Our teacher-guided pruning approach generally outperforms Liao et al. [20] across most sparsity levels (Table VI). On CIFAR-10, at 50.00% sparsity, our method achieves 96.08% accuracy ($T = 3$), surpassing Liao et al.'s 92.56% by 3.52 percentage points. This trend continues at 75.00% sparsity (95.62% vs. 93.00%) and 93.92% sparsity (94.40% vs. 92.93% at 93.8% sparsity for Liao et al. [20]). However, at the highest sparsity of 98.41%, Liao et al.'s method achieves 93.12%

TABLE VII: Performance of the student model (ResNet34) distilled from the teacher model (ResNet50) on CIFAR-100 using our method. Results are reported across different sparsity levels in terms of accuracy without knowledge distillation (w/o KD) and with KD at temperatures $T = 3$ and $T = 5$ with $\alpha = 0.7$.

Sparsity (%)	Accuracy w/o KD (%)	Acc (KD, T=3, α=0.7) (%)	Acc (KD, T=5, α=0.7) (%)	# Epoch w/o KD (Retrain)	# Epoch KD T=3 (Retrain)	# Epoch KD T=5 (Retrain)
98.22	66.14	72.38	**72.79**	22	38	39
93.74	75.35	79.54	**79.78**	10	34	22
90.75	77.89	81.09	**81.44**	8	11	30
80.77	81.09	82.21	**81.91**	13	28	14
75.85	81.70	**82.41**	82.25	9	11	10
50.86	83.18	83.12	**82.83**	14	11	7

TABLE VIII: Performance of the sparse student model (ResNet34) distilled from the teacher model (ResNet50) on CIFAR-10 using our method. Results are reported at various sparsity levels in terms of accuracy without knowledge distillation (w/o KD) and with KD at temperatures $T = 3$ and $T = 5$ with $\alpha = 0.7$.

Sparsity (%)	Accuracy w/o KD (%)	Acc (KD, T=3, α=0.7) (%)	Acc (KD, T=5, α=0.7) (%)	# Epoch w/o KD (Retrain)	# Epoch KD T=3 (Retrain)	# Epoch KD T=5 (Retrain)
98.44	86.22	**92.35**	91.57	16	44	37
93.94	94.07	95.16	**95.38**	34	20	22
90.02	94.93	**95.71**	95.67	20	11	18
80.02	96.03	96.08	**96.12**	15	9	28
75.01	**96.16**	96.13	96.14	11	13	11
50.47	**96.52**	96.51	96.49	13	9	17



(a) Cifar10

(b) Tiny ImageNet

Fig. 7: Top-1 accuracy vs. sparsity for ResNet18 on CIFAR-10 and TinyImageNet. Results are shown for our method (solid blue line) and the method by Liao et al. [20] (dotted orange line) across varying sparsity levels. Accuracy is reported at different sparsity points, with detailed values provided in Table VI

.

accuracy (at 98.4% sparsity), outperforming our 90.79% by 2.33 percentage points, possibly due to their entropy-guided strategy better preserving critical neurons at extreme sparsity. On TinyImageNet, our method consistently excels: at 50.02% sparsity, we achieve 60.93% accuracy ($T = 5$) compared to 41.70% for Liao et al. (at 50.00% sparsity), a 19.23 percentage-point improvement. This gap persists at 74.55% sparsity (76.58% vs. 41.24% at 75.00% sparsity), 93.92% sparsity (53.79% vs. 41.86% at 93.8% sparsity), and 98.41% sparsity (50.64% vs. 37.44% at 98.4% sparsity). The superior performance on TinyImageNet highlights the effectiveness of our CA-KLD-based KD (Section III-B) in transferring fine-grained knowledge, enabling better generalization on more complex datasets with higher class diversity.