Verification and Training of Neural Networks for Robustness Against Neuron Pruning

Anonymous authors
Paper under double-blind review

Abstract

Structured neuron pruning removes entire hidden units to reduce model size and computation, but often leads to unpredictable accuracy degradation. Existing pruning methods typically rely on heuristic importance scores and provide no formal guarantees on the behavior of pruned models. In this work, we propose a certifiable approach for structured neuron pruning in fully connected layers of feedforward neural networks that guarantees robustness against all pruning masks satisfying a given layer-wise sparsity budget. We further develop a computable upper bound on the worst-case change in pairwise class margins induced by neuron pruning. The analysis models pruning as row-zeroing (equivalently, neuron gating via binary masks) in the weight matrices and bounds the resulting deviation via operator-norm-based error propagation. These bounds are then used to develop a margin-aware robust training objective for certifiable pruning robustness. Experiments on MNIST and CIFAR-10 show that the resulting models achieve non-trivial certified accuracy under a range of pruning budgets and that our robust training substantially improves both certified and empirical robustness over standard baselines.

1 Introduction

Deep neural networks (DNNs) achieve state-of-the-art performance across a wide range of applications, but their large model size and computational cost hinder deployment on resource-constrained devices (Chen et al., 2020). This has motivated extensive research on model compression techniques, including quantization, pruning, and knowledge distillation (Rokh et al., 2023; Cheng et al., 2024; Alkhulaifi et al., 2021). Among these, structured pruning is particularly appealing, as it removes entire neurons, channels, or filters to enable real-world acceleration without specialized hardware (Cheng et al., 2024; He & Xiao, 2023). In this work, we focus on neuron-level structured pruning in the fully connected (FC) layers of neural networks.

Although structured pruning effectively reduces computation with minimal accuracy loss, most methods remain heuristic-driven, relying on metrics such as weight magnitude, norms, loss estimates, gradients, or filter similarity (Cheng et al., 2024; Lahav & Katz, 2021). Learning-based approaches, including sparsity regularization (Huang & Wang, 2018; Jiang et al., 2023) and dynamic sparse training (Liu et al., 2020), also depend on such ad hoc signals (e.g., ℓ_1/group-lasso penalties with external thresholds), which require external criteria and hyperparameters. Consequently, pruned models often exhibit unstable performance without guarantees, typically requiring fine-tuning to recover accuracy.

While some works provide guarantees for pruning (Dong et al., 2017; Pitas et al., 2018; Ye et al., 2020; El Halabi et al., 2022), these guarantees are tied to specific pruning algorithms and do not generalize beyond the masks they produce. In practice, however, pruning masks are often not fully controlled by the designer. For example, hardware accelerators often enforce SIMD-aligned or block-aligned pruning patterns that may not coincide with the mask chosen by a given algorithm (Jin et al., 2024), and in federated or on-device learning, clients may apply different pruning strategies due to non-IID data distributions and device-specific constraints (Yi et al., 2024). Moreover, hardware faults can effectively silence neurons during inference, giving rise to functionally pruned subnetworks whose masks are determined by the fault pattern rather than

by a pruning algorithm (Zhang et al., 2018b; Ahmadilivani et al., 2024). These scenarios motivate the need for guarantees that hold uniformly over all admissible pruning configurations under a given sparsity budget.

In this work, we take a different perspective and view structured neuron pruning as a form of combinatorial model perturbation, where each pruning mask defines a distinct sub-network within an exponentially large family. Unlike prior work that focuses on identifying a single performant sub-network, we aim to provide *worst-case, mask-agnostic* guarantees over all subnetworks satisfying a layer-wise sparsity budget. This perspective connects model compression with certified robustness, enabling guarantees under discrete structural perturbations arising from algorithmic choices, system constraints, or hardware faults.

Formally, we consider multiclass classification with feedforward networks. Given a trained network f and a layer-wise pruning budget S, which specifies the maximum number of neurons that may be removed in each fully connected layer, we define $\mathcal{F}_{\text{pruned}}$ as the family of all pruned subnetworks of f that satisfy S. The verification problem is to verify that every $\hat{f} \in \mathcal{F}_{\text{pruned}}$ preserves the correct prediction for a given input. Since this family grows combinatorially with network size, exhaustive evaluation is intractable. Leveraging advances in non-convex global optimization for deep learning (Tjeng et al., 2017; Wong & Kolter, 2018; Chiu & Zhang, 2023), we reformulate pruning robustness as a *mixed-integer linear program* (MILP), solvable by standard solvers such as Gurobi (Cheng & Li, 2022).

To enable training-time robustness, we further derive a computable upper bound on the worst-case deviation in pairwise class margins between the original network and any admissible pruned sub-network. Based on this analysis, we propose a margin-aware, bound-guided training objective that improves robustness to pruning.

We evaluate our methods on different architectures trained on the MNIST and CIFAR10 datasets. We show that our verification techniques based on MILP and worst-case margin deviation yield non-vacuous bounds across a range of sparsity budgets. Moreover, our robust training substantially improves the verified accuracy and empirical performance compared to standard baselines. In summary, our main contributions are summarized as follows:

- We formalize robustness against structured neuron pruning in FC layers as a worst-case guarantee over a family of sub-networks defined by a layer-wise sparsity budget and develop a MILP-based verification method (Section 4.1).
- We derive computable upper bounds on worst-case pairwise margin deviation under pruning for single-layer, multi-layer, and all-layer settings (Section 4.2).
- We propose a theory-driven training objective that leverages our margin-based bounds to improve robustness to pruning under a given layer-wise sparsity budget (Section 4.3). We observe improvements in both certified and empirical robustness over standard baselines.

2 Related Work

There are a number of works that explored pruning techniques with performance guarantees (Lahav & Katz, 2021; Li et al., 2021; Gokulanathan et al., 2020; Aghasi et al., 2017; Dong et al., 2017; Pitas et al., 2018; Ye et al., 2020; El Halabi et al., 2022). For instance, Lahav & Katz (2021) and Gokulanathan et al. (2020) use verification techniques to identify redundant neurons whose removal provably preserves network outputs. Similarly, Aghasi et al. (2017) formulated an optimization problem to prune a network while keeping the pruned model's activations consistent with the original model on the training data. While these works demonstrate that formal guarantees in pruning are possible, their guarantees apply only to the particular masks produced by the corresponding selection strategies, such as layer-wise optimal brain surgeon (Aghasi et al., 2017), difference-of-convex-based pruning (Pitas et al., 2018), or submodular greedy selection (El Halabi et al., 2022). Consequently, no guarantee holds when the network is pruned using other techniques, and in such cases, a new verification routine must be developed and executed. In contrast, we provide guarantees that hold across all possible pruned models satisfying a given layer-wise sparsity budget.

Another line of work aims to prepare neural networks during training so that they are more amenable to post-hoc pruning. One approach focuses on shaping the loss landscape to improve robustness to parameter

perturbations. In particular, flatness-based methods, including sharpness-aware minimization, encourage solutions that are less sensitive to weight changes, thereby improving post-pruning performance (Peste et al., 2022; Bair et al., 2023; Lee et al., 2025; Na et al., 2022). Complementary approaches explicitly promote sparsity or structural adaptability during training. For example, Khan & Stavness (2020) propose sparsity-inducing regularization schemes, while Gomez et al. (2019) introduce targeted dropout to encourage robustness to neuron removal. Related methods, such as slimmable networks and once-for-all models, train shared weights that can be adapted to multiple subnetworks at different widths or sparsity levels (Yu et al., 2018; Cai et al., 2019). While these methods improve empirical performance across different pruning configurations, their evaluation is typically limited to a small subset of subnetworks or specific pruning strategies. As a result, they do not assess worst-case behavior and provide no formal guarantees on model robustness under arbitrary admissible pruning masks.

Our work is also closely related to certified robustness, which aims to provide guarantees on model predictions under bounded perturbations. Existing methods primarily focus on continuous perturbations, including adversarial input perturbations and weight perturbations (Hein & Andriushchenko, 2017; Wong & Kolter, 2018; Weng et al., 2018; Xu et al., 2020; Tsai et al., 2021a; Dang et al., 2025). However, these approaches do not capture the discrete, combinatorial nature of pruning, where perturbations are constrained binary masks over network structure. In contrast, we study structured neuron pruning as a form of combinatorial model perturbation under layer-wise sparsity constraints and enable worst-case guarantees over the resulting family of subnetworks. This setting differs fundamentally from continuous robustness and introduces unique challenges due to its combinatorial nature.

3 Problem Formulation

In this section, we present the notations used throughout the paper and define pruned models under a sparsity budget. We then formalize the two main problems addressed in this work: pruning-robust verification and pruning-robust training.

Notations. Sets and spaces are denoted by capital letters, except for K, the number of hidden layers in a neural network (NN), and C, the number of classes. For any positive integer N, we denote $[N] = \{1, \ldots, N\}$. Matrices are denoted by bold uppercase letters (e.g., $\mathbf{W}$), and vectors by bold lowercase letters (e.g., $\mathbf{x}$). We use $\mathbf{W}[i]$ to denote the i-th row of a matrix $\mathbf{W}$ and $\mathbf{x}[i]$ to denote the i-th entry of a vector $\mathbf{x}$. We use $\|\cdot\|_2$ to denote the Euclidean norm for vectors and the induced spectral norm for matrices, i.e., $\|\mathbf{W}\|_2 = \sup_{\|\mathbf{x}\|_2=1} \|\mathbf{W}\mathbf{x}\|_2$. These are the only norms used throughout the analysis.

Model. We consider a multiclass classification problem with C classes, where pruning is applied to the fully connected (FC) layers of a neural network. To isolate the effect of neuron-level pruning on these layers, we model the network as a fully connected feedforward network with K hidden layers. Let $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_0}$ denote the input, and define the forward pass as

$$\begin{aligned} \mathbf{x}_0 &= \mathbf{x}, \\ \mathbf{z}_k &= \mathbf{W}_k \mathbf{x}_{k-1}, \qquad k = 1, \ldots, K, \\ \mathbf{x}_k &= \sigma(\mathbf{z}_k), \qquad k = 1, \ldots, K, \\ \mathbf{f}(\mathbf{x}) &= \mathbf{W}_{K+1} \mathbf{x}_K, \end{aligned}$$

where $\mathbf{W}_k \in \mathbb{R}^{d_k \times d_{k-1}}$, $k \in [K]$ are the hidden-layer weight matrices, and $\mathbf{W}_{K+1} \in \mathbb{R}^{C \times d_K}$ is the output-layer weight matrix. For simplicity, we omit bias terms. This does not restrict generality, as biases can be incorporated by augmenting the input with a constant dimension.

3.1 Sparsity Budgets and Pruned Models

We define structured neuron pruning via layer-wise sparsity budgets $S = \{s_1, \ldots, s_K\}$, where s_k specifies the maximum number of neurons that can be removed at hidden layer k ($0 \le s_k \le d_k - 1$).

For each layer $k \in [K]$, let $M_k = \left\{\mathbf{m}_k \in \{0,1\}^{d_k} : \|\mathbf{1} - \mathbf{m}_k\|_0 \leq s_k\right\}$ denote the set of admissible pruning masks, where $\mathbf{m}_k[i] = 0$ indicates that neuron i is pruned. Given $\mathbf{m}_k \in M_k$, we define the pruning operator $\mathcal{P}_{\mathbf{m}_k}(\mathbf{x}_k) = \mathbf{m}_k \circ \mathbf{x}_k$, where $\circ$ denotes element-wise multiplication.

A pruned network $\hat{f}$ associated with a budget S and masks $\{\mathbf{m}_k\}_{k=1}^K$ is defined recursively as

$$\hat{\mathbf{x}}_0 = \mathbf{x}, \tag{1}$$
$$\hat{\mathbf{z}}_k = \mathbf{W}_k \hat{\mathbf{x}}_{k-1}, \tag{2}$$
$$\hat{\mathbf{x}}_k = \sigma(\hat{\mathbf{z}}_k), \tag{3}$$
$$\hat{\mathbf{x}}_k^p = \mathcal{P}_{\mathbf{m}_k}(\hat{\mathbf{x}}_k), \qquad k = 1, \ldots, K, \tag{4}$$

subject to $\mathbf{m}_k \in M_k$. The final output is given by $\hat{f}(\mathbf{x}) = \mathbf{W}_{K+1}\hat{\mathbf{x}}_K^p$.

Remark 1 *The operator $\mathcal{P}_{\mathbf{m}_k}$ models neuron removal by zeroing out the corresponding activations. This is equivalent to removing the associated rows and columns in the weight matrices during forward propagation if activation functions satisfy $\sigma(0) = 0$, ensuring that pruned neurons do not contribute to subsequent layers.*

3.2 Verification and Training for Pruning Robustness

We consider two problems in this work. First, we aim to provide robustness guarantees for all pruned models satisfying a given sparsity budget. Second, we tackle the problem of training a NN such that it is robust against such pruning.

Problem 1 (Pruning-robust verification) *Let f be a neural network with K hidden fully connected layers and a layer-wise sparsity budget $S = \{s_1, \ldots, s_K\}$, where $0 \leq s_k \leq d_k - 1$. Let $M_{\text{pruned}} = \{\mathbf{m} = \{\mathbf{m}_1, \ldots, \mathbf{m}_K\} : \|\mathbf{1} - \mathbf{m}_k\|_0 \leq s_k,\ \forall k \in [K]\}$ denote the set of admissible pruning masks. For a mask $\mathbf{m} \in M_{\text{pruned}}$, let $\hat{f}_{\mathbf{m}}$ denote the corresponding pruned network, and let $\hat{f}_{\mathbf{m}}^{(i)}(\mathbf{x})$ denote the logit of class $i \in [C]$. Given an input $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_0}$ with true label t, the goal is to solve*

$$\min_{\mathbf{m} \in M_{\text{pruned}}} \gamma_{\mathbf{m}}(\mathbf{x}, t) = \min_{\mathbf{m} \in M_{\text{pruned}}} \min_{i \in [C],\, i \neq t} \left(\hat{f}_{\mathbf{m}}^{(t)}(\mathbf{x}) - \hat{f}_{\mathbf{m}}^{(i)}(\mathbf{x})\right), \tag{5}$$

which corresponds to the worst-case pairwise margin over all admissible pruning masks.

In the case of classification, if the optimal value of equation 5 is strictly positive, then the prediction remains unchanged and correct under all admissible pruning masks. In this case, the model is said to be *pruning-robust* at input $\mathbf{x}$.

Reasoning over the set of admissible pruning masks involves a combinatorial discrete space, rendering Problem 1 NP-hard in general. Instead of solving it exactly, in this work, we develop tractable formulations that compute certified lower bounds on the worst-case margin.

To train a neural network that is robust against pruning under a given sparsity budget, we seek to incorporate pruning-induced robustness into the training objective. Motivated by prior work on margin-based robustness and weight perturbations (Hein & Andriushchenko, 2017; Tsai et al., 2021a;b), we treat the worst-case pruning-induced margin deviation as a training signal for pruning robustness.

Problem 2 (Pruning-robust training) *Let f be a neural network with parameters $\mathbf{W}$, a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{D}|}$, and a layer-wise sparsity budget $S = \{s_1, \ldots, s_K\}$. For an input $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_0}$ with true label t, and for any admissible pruning mask $\mathbf{m} \in M_{\text{pruned}}$, let $\hat{f}_{\mathbf{m}}$ denote the corresponding pruned network. Define the worst-case pairwise class-margin deviation as*

$$\Delta(S, \mathbf{x}, t) = \max_{\mathbf{m} \in M_{\text{pruned}}} \max_{i \in [C],\, i \neq t} \left| \left(\hat{f}_{\mathbf{m}}^{(t)}(\mathbf{x}) - \hat{f}_{\mathbf{m}}^{(i)}(\mathbf{x})\right) - \left(f^{(t)}(\mathbf{x}) - f^{(i)}(\mathbf{x})\right) \right|. \tag{6}$$

The objective is to train the network parameters so that this deviation remains small relative to the nominal margin $\gamma(\mathbf{x}, t) = f^{(t)}(\mathbf{x}) - \max_{i \neq t} f^{(i)}(\mathbf{x})$, which ensures that the prediction is preserved under all admissible pruning masks.

The maximization in equation 6 over the discrete mask space M_{pruned} is highly non-linear and non-differentiable. Instead, we derive tractable upper bounds on its optimal value, denoted by $\delta(S, \mathbf{x}, t)$. These bounds serve as surrogates of the worst-case deviation and enable the design of margin-aware, bound-guided training objectives for pruning-robust neural networks.

4 Methodology

In this section, we first show how to certify pruning robustness under all possible pruning masks via global optimization (Section 4.1). We then derive analytical upper bounds on the worst-case deviation of pairwise class margin induced by pruning at a given sparsity budget in three cases: single-layer pruning, all-layer pruning, and multi-layer pruning (Section 4.2). Finally, we leverage these bounds to design a surrogate training objective for pruning-robust learning (Section 4.3).

4.1 Mixed-Integer Linear Programming for Pruning-Robust Verification

Directly solving Problem 1 requires enumerating all admissible pruning masks, which is infeasible due to the combinatorial size of M_{pruned}. To address this, we formulate the verification problem as a mixed-integer linear program (MILP) that jointly encodes network activations, pruning masks, and sparsity constraints. This formulation builds on standard techniques in neural network verification (Tjeng et al., 2017; Zhang et al., 2018a; McCormick, 1976), and can be solved using off-the-shelf solvers such as Gurobi (Cheng & Li, 2022), which provide global optimality guarantees via branch-and-bound.

(1) Activation constraints. For each layer $k \in [K]$ and neuron $i \in [d_k]$, let $\underline{\hat{z}}_k[i]$ and $\overline{\hat{z}}_k[i]$ denote valid finite bounds on the pre-activation $\hat{z}_k[i]$. For ReLU activations, we introduce binary variables $a_{k,i} \in \{0, 1\}$ and exactly encode all feasible ReLU activation patterns (Tjeng et al., 2017):

$$\hat{x}_k[i] \geq \hat{z}_k[i], \quad \hat{x}_k[i] \geq 0, \quad \hat{x}_k[i] \leq \overline{\hat{z}}_k[i]\, a_{k,i}, \quad \hat{x}_k[i] \leq \hat{z}_k[i] - \underline{\hat{z}}_k[i](1 - a_{k,i}). \tag{7}$$

For general activation functions, we instead use affine upper and lower bounds over the interval $[\underline{\hat{z}}_k[i], \overline{\hat{z}}_k[i]]$ (Zhang et al., 2018a).

(2) Pruning constraints. For each layer $k \in [K]$ and neuron $i \in [d_k]$, neuron pruning is modeled as a binary gating operation $\hat{x}^p_k[i] = m_k[i]\hat{x}_k[i]$, where $m_k[i] \in \{0, 1\}$ indicates whether neuron i is retained. Given valid bounds $L_k[i] \leq \hat{x}_k[i] \leq U_k[i]$, the bilinear terms are linearized exactly using McCormick envelopes (McCormick, 1976):

$$\begin{aligned} &\hat{x}^p_k[i] \leq U_k[i]m_k[i], \\ &\hat{x}^p_k[i] \geq L_k[i]m_k[i], \\ &\hat{x}^p_k[i] \leq \hat{x}_k[i] - L_k[i](1 - m_k[i]), \\ &\hat{x}^p_k[i] \geq \hat{x}_k[i] - U_k[i](1 - m_k[i]). \end{aligned} \tag{8}$$

The sparsity budget is enforced via: $\sum_{i=1}^{d_k} m_k[i] \geq d_k - s_k$.

(3) Bound propagation. We obtain valid finite bounds on all pre-activations and activations using interval bound propagation (IBP) from a bounded input domain $\mathcal{X} \subseteq \mathbb{R}^{d_0}$ (Gowal et al., 2018). Given bounds $L_{k-1}[r] \leq \hat{x}^p_{k-1}[r] \leq U_{k-1}[r]$, the pre-activation bounds are computed via interval arithmetic:

$$\underline{\hat{z}}_k[i] = \sum_r \big(W_k[i,r]^+ L_{k-1}[r] + W_k[i,r]^- U_{k-1}[r]\big), \quad \overline{\hat{z}}_k[i] = \sum_r \big(W_k[i,r]^+ U_{k-1}[r] + W_k[i,r]^- L_{k-1}[r]\big). \tag{9}$$

The activation bounds $[L_k[i], U_k[i]]$ are then obtained by applying σ over the interval $[\underline{\hat{z}}_k[i], \overline{\hat{z}}_k[i]]$.

(4) Objective. For a fixed input $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_0}$ with true label t, the pruning-robust verification problem is formulated as the following mixed-integer linear program:

$$\min_{\mathbf{m} \in M_{\text{pruned}}} \min_{i \in [C],\, i \neq t} \hat{f}^{(t)}_{\mathbf{m}}(\mathbf{x}) - \hat{f}^{(i)}_{\mathbf{m}}(\mathbf{x}), \tag{10}$$

subject to the activation constraints in equation 7, the pruning constraints in equation 8 and bound constraints described in equation 9.

Remark 2 *For ReLU networks, the binary variables in equation 7 exactly encode all feasible ReLU phase assignments, and the bilinear mask-activation products in equation 8 are exactly represented via the McCormick formulation since the mask variables are binary. Therefore, provided that valid finite bounds on all pre-activations and activations are available - obtained via bound propagation from a bounded input domain - and all integer variables are preserved, solving the MILP to global optimality yields an exact certificate for the input x, jointly over all admissible pruning masks and ReLU activation patterns. Note that in this setting, interval bounds obtained from IBP are used solely to provide valid bounds for the mixed-integer encoding; tighter bounds improve computational efficiency but are not required for exactness. For non-ReLU activations, where the activation is replaced by affine upper and lower bounds, the resulting formulation becomes a relaxation, and the optimal value provides a sound lower bound on the true verification objective.*

The scalability of the MILP formulation is primarily limited by the number of binary variables, which grows with network size. Each neuron introduces binary variables for both activation phases and pruning masks, leading to exponential worst-case complexity. In multi-class settings, the verification is typically decomposed into $C-1$ sub-problems (one per competing class), further multiplying the computational cost. These factors make the approach challenging for larger networks and complex datasets.

4.2 Upper bound on the worst-case margin deviation for robust training objective

Assumptions for bound analysis. For the derivation of margin-based deviation bounds, we assume that the activation function σ is 1-Lipschitz and satisfies $\sigma(0) = 0$. Note that these conditions are not required for the MILP-based verification framework. Formally, let $\mathbf{m}_k \in \{0,1\}^{d_k}$ and define the masked weight matrix: $\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k) := \mathrm{Diag}(\mathbf{m}_k)\mathbf{W}_k$. Under these assumption $\sigma(0) = 0$, pruning can be modeled as row masking, the pruned network $\hat{f}$ associated with masks $\{\mathbf{m}_k\}_{k=1}^K$ is defined recursively as

$$\begin{aligned}\hat{\mathbf{x}}_1^p &= \sigma\big(\mathcal{P}_{\mathbf{m}_1}(\mathbf{W}_1)\mathbf{x}_0\big),\\ \hat{\mathbf{x}}_k^p &= \sigma\big(\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k)\hat{\mathbf{x}}_{k-1}^p\big), \quad k = 2,\dots,K,\\ \hat{f}(\mathbf{x}) &= \mathbf{W}_{K+1}\hat{\mathbf{x}}_K^p.\end{aligned}$$

Proposition 1 (Worst-case margin deviation) *Let f be a neural network with K hidden fully connected layers and sparsity budget $S = \{s_1,\dots,s_K\}$, which defines the set of admissible pruning masks M_{pruned}. For any $\mathbf{m} \in M_{\text{pruned}}$, let $\hat{f}_{\mathbf{m}}$ denote the corresponding pruned network. Define $\|\Delta\mathbf{x}_k\|_2 = \|\hat{\mathbf{x}}_k^p - \mathbf{x}_k\|_2$ for $k \in [K]$ as the layer-wise output deviation norm, which accumulates pruning-induced perturbations from all preceding layers. For any input $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^{d_0}$ with true class t, $\Delta(S,\mathbf{x},t)$ denotes the worst-case margin deviation (WMD) such that*

$$\Delta(S,\mathbf{x},t) \le \max_{i\neq t} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2 \cdot \max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta\mathbf{x}_K\|_2 .$$

Proof 1

$$\begin{aligned}\Delta(S,\mathbf{x},t) &= \max_{\mathbf{m}\in M_{\text{pruned}}} \max_{i\neq t, i\in[C]} \Big|(\hat{f}_{\mathbf{m}}^{(t)}(\mathbf{x}) - \hat{f}_{\mathbf{m}}^{(i)}(\mathbf{x})) - (f^{(t)}(\mathbf{x}) - f^{(i)}(\mathbf{x}))\Big|\\ &\overset{\text{(i)}}{=} \max_{\mathbf{m}\in M_{\text{pruned}}} \max_{i\neq t, i\in[C]} \big|(\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i])^\top(\hat{\mathbf{x}}_K^p - \mathbf{x}_K)\big|\\ &\overset{\text{(ii)}}{\le} \max_{\mathbf{m}\in M_{\text{pruned}}} \max_{i\neq t, i\in[C]} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2 \, \|\Delta\mathbf{x}_K\|_2\\ &\overset{\text{(iii)}}{=} \max_{i\neq t, i\in[C]} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2 \cdot \max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta\mathbf{x}_K\|_2.\end{aligned}$$

where (i) follows from expanding the logits and using the fact that the output layer is not pruned. (ii) follows from the Cauchy-Schwarz inequality. (iii) holds since $\|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2$ is independent of $\mathbf{m}$.

Details of the proof are provide in Appendix A.1. The remaining challenge is to bound the layerwise output deviation $\|\Delta\mathbf{x}_k\|_2$ for $k \in [K]$. We consider three pruning scenarios—single-layer, selected multi-layer, and all-layer pruning—and derive tractable bounds inspired by Lipschitz-style perturbation analyses, but specialized to the discrete combinatorial space of structured neuron removal under layerwise sparsity budgets.

Proposition 2 (Single-layer pruning bound) *Let f be a neural network with K hidden fully connected layers and sparsity budget $S = \{s_1, \ldots, s_K\}$ such that only layer $\ell \in [K]$ is pruned, i.e., $s_\ell > 0$ and $s_k = 0$ for all $k \neq \ell$. Let M_{pruned} denote the corresponding set of admissible pruning masks. For any input $\mathbf{x} \in \mathcal{X}$ with true label t, define*

$$\begin{aligned}\delta_\ell &= \|tops_{s_\ell}(\mathbf{x}_\ell)\|_2, \\ \delta_k &= \|\mathbf{W}_k\|_2\, \delta_{k-1}, \qquad k = \ell+1, \ldots, K.\end{aligned}$$

where $tops_{s_\ell}(\mathbf{x}_\ell)$ denotes the vector obtained by keeping the s_ℓ entries of $\mathbf{x}_\ell$ with largest absolute values and setting the rest to zero. The worst-case margin deviation (WMD) is bounded by

$$\Delta(S, \mathbf{x}, t) \leq \max_{i \neq t, i \in [C]} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2\, \delta_K.$$

Proof 2 *Since only layer ℓ is pruned, the hidden representations before layer ℓ are unchanged. Hence,*

$$\max_{\mathbf{m} \in M_{\text{pruned}}} \|\Delta \mathbf{x}_\ell\|_2 = \max_{\mathbf{m}_\ell \in M_\ell} \|\mathbf{m}_\ell \circ \mathbf{x}_\ell - \mathbf{x}_\ell\|_2 = \max_{\mathbf{m}_\ell \in M_\ell} \|\bar{\mathbf{m}}_\ell \circ \mathbf{x}_\ell\|_2 = \|tops_{s_\ell}(\mathbf{x}_\ell)\|_2 = \delta_\ell.$$

For $k > \ell$, no further pruning is applied, so

$$\begin{aligned}\max_{\mathbf{m} \in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 &= \max_{\mathbf{m} \in M_{pruned}} \|\sigma(\mathbf{W}_k \hat{\mathbf{x}}_{k-1}^p) - \sigma(\mathbf{W}_k \mathbf{x}_{k-1})\|_2 \\ &\overset{\text{(i)}}{\leq} \max_{\mathbf{m} \in M_{pruned}} \|\mathbf{W}_k(\hat{\mathbf{x}}_{k-1}^p - \mathbf{x}_{k-1})\|_2 \\ &\overset{\text{(ii)}}{\leq} \|\mathbf{W}_k\|_2 \max_{\mathbf{m} \in M_{pruned}} \|\Delta \mathbf{x}_{k-1}\|_2.\end{aligned}$$

Thus $\max_{\mathbf{m} \in M_{pruned}} \|\Delta \mathbf{x}_k\|_2 \leq \delta_k, k = \ell+1, \ldots, K$. The result then follows from Proposition 1. Here (i) follows from the $1-$Lipschitz property of σ. (ii) follows from the definition of the spectral norm. Details of the proof are provided in Appendix A.2

Proposition 3 (All-layer pruning bound) *Let f be a neural network with K hidden fully connected layers and sparsity budget $S = \{s_1, \ldots, s_K\}$, and let M_{pruned} denote the corresponding set of admissible pruning masks. Assume that pruning is applied to all hidden layers, i.e., $s_k > 0$ for $k \in [K]$. For any input $\mathbf{x} \in \mathcal{X}$ with true label t, define*

$$\begin{aligned}\delta_1 &= \|tops_{s_1}(\mathbf{x}_1)\|_2, \\ \delta_k &= \|\mathbf{W}_k\|_2\, \delta_{k-1} + \|tops_{s_k}(\mathbf{x}_k)\|_2, \qquad k = 2, \ldots, K,\end{aligned}$$

where $tops_{s_k}(\mathbf{x}_k)$ denotes the vector obtained by keeping the s_k entries of $\mathbf{x}_k$ with largest absolute values and setting the rest to zero. Then the worst-case margin deviation (WMD) satisfies

$$\Delta(S, \mathbf{x}, t) \leq \max_{i \neq t, i \in [C]} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2\, \delta_K.$$

Proof 3 *The first-layer deviation satisfies*

$$\max_{\mathbf{m} \in M_{\text{pruned}}} \|\Delta \mathbf{x}_1\|_2 = \|tops_{s_1}(\mathbf{x}_1)\|_2 = \delta_1.$$

For $k \geq 2$, by the triangle inequality, the masking equivalence under $\sigma(0) = 0$, the 1-Lipschitz property of σ, and the spectral norm bound,

$$\begin{aligned}\max_{\mathbf{m} \in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 &= \max_{\mathbf{m} \in M_{\text{pruned}}} \|\mathbf{m}_k \circ \sigma(\mathbf{W}_k \hat{\mathbf{x}}_{k-1}^p) - \sigma(\mathbf{W}_k \mathbf{x}_{k-1})\|_2 \\ &\leq \max_{\mathbf{m} \in M_{\text{pruned}}} \|\mathbf{m}_k \circ \sigma(\mathbf{W}_k \hat{\mathbf{x}}_{k-1}^p) - \mathbf{m}_k \circ \sigma(\mathbf{W}_k \mathbf{x}_{k-1})\|_2 + \max_{\mathbf{m}_k \in M_k} \|\mathbf{m}_k \circ \mathbf{x}_k - \mathbf{x}_k\|_2 \\ &\leq \max_{\mathbf{m} \in M_{\text{pruned}}} \|\sigma(\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k) \hat{\mathbf{x}}_{k-1}^p) - \sigma(\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k) \mathbf{x}_{k-1})\|_2 + \|tops_{s_k}(\mathbf{x}_k)\|_2 \\ &\leq \max_{\mathbf{m} \in M_{\text{pruned}}} \|\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k)(\hat{\mathbf{x}}_{k-1}^p - \mathbf{x}_{k-1})\|_2 + \|tops_{s_k}(\mathbf{x}_k)\|_2 \\ &\leq \|\mathbf{W}_k\|_2 \max_{\mathbf{m} \in M_{\text{pruned}}} \|\Delta \mathbf{x}_{k-1}\|_2 + \|tops_{s_k}(\mathbf{x}_k)\|_2.\end{aligned}$$

Applying Proposition 1 completes the proof. Details of the proof are provided in Appendix A.3

The two terms in the upper bound above highlight that pruning affects the network through two mechanisms: a local perturbation introduced at the pruned layer, and its subsequent amplification through downstream layers. We now generalize to the case where pruning is applied to a subset of layers. In this case, if a layer is unpruned, the deviation follows the single-layer propagation rule; otherwise, it incorporates an additional local deviation term as in the all-layer pruning case.

Proposition 4 (Selected multi-layer pruning bound) *Let f be a neural network with K hidden fully connected layers and sparsity budget $S = \{s_1, \ldots, s_K\}$. Let $I_p \subseteq [K]$ denote the set of pruned layers, i.e., $s_k > 0$ for $k \in I_p$ and $s_k = 0$ otherwise, and let $\ell = \min I_p$ be the first pruned layer. Let M_{pruned} denote the corresponding set of admissible masks. For any input $\mathbf{x} \in \mathcal{X}$ with true label t, define*

$$\delta_\ell = \|tops_{s_\ell}(\mathbf{x}_\ell)\|_2,$$
$$\delta_k = \begin{cases} \|\mathbf{W}_k\|_2\, \delta_{k-1}, & k \notin I_p, \\ \|\mathbf{W}_k\|_2\, \delta_{k-1} + \|tops_{s_k}(\mathbf{x}_k)\|_2, & k \in I_p, \end{cases} \qquad k = \ell+1, \ldots, K,$$

where $tops_{s_k}(\mathbf{x}_k)$ denotes the vector obtained by keeping the s_k entries of $\mathbf{x}_k$ with largest absolute values and setting the rest to zero. The worst-case margin deviation (WMD) satisfies

$$\Delta(S, \mathbf{x}, t) \leq \max_{i \neq t, i \in [C]} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2\, \delta_K.$$

Remark 3 *The proposed bounds can be computed efficiently via a single forward-style recursion once the required activations and spectral norms are available. The recursion itself scales linearly with depth. The spectral norm of each weight matrix can be estimated using r steps of power iteration, the total cost is $O\Big(\sum_{k=1}^{K} r\, d_k d_{k-1}\Big)$. The additional cost of computing the local pruning terms $tops_{s_k}(\mathbf{x}_k)$ is at most $O(d_k \log d_k)$ per layer using sorting. This makes the bound substantially more efficient than MILP-based verification. The gain in scalability, however, comes at the expense of tightness because the use of spectral norms can be conservative, especially in deep networks where repeated multiplication may amplify over-approximation errors.*

4.3 Pruning-robust training loss

Building on the margin-bound analysis in Section 4.2, we design a training objective that directly promotes robustness against pruning by enforcing the condition that the worst-case margin deviation remains below the nominal class margin. Recall that a sufficient condition for preserving the prediction under all admissible pruning masks is

$$\delta(S, \mathbf{x}, t) \leq \gamma(\mathbf{x}, t),$$

where $\delta(S, \mathbf{x}, t)$ is the worst-case margin deviation bound and $\gamma(\mathbf{x}, t)$ is the margin between the true class logit and the largest competing logit.

Motivated by this condition, we introduce the following margin-aware loss:

$$\mathcal{L}_{\text{pr}}\big(f(\mathbf{x}), y, \mathcal{F}_{\text{pruned}}\big) \;=\; \text{CE}\big(f(\mathbf{x}), y\big) \;+\; \lambda_1 \max\big(\delta(S, \mathbf{x}, t) \;-\; \gamma(\mathbf{x}, t), 0\big) \;+\; \lambda_2 \frac{\delta(S, \mathbf{x}, t)}{\max(\gamma(\mathbf{x}, t), \epsilon)}. \quad (11)$$

Here, $\text{CE}(f(\mathbf{x}), y)$ is the standard cross-entropy loss, $\delta(S, \mathbf{x}, t)$ denotes the worst-case margin deviation bound, and $\gamma(\mathbf{x}, t)$ is the nominal margin. The hyperparameters $\lambda_1, \lambda_2 \geq 0$ control the trade-off between accuracy and pruning robustness.

The hinge term $\max(\delta - \gamma, 0)$ serves as the primary robustness term, as it directly penalizes violations of the sufficient condition $\delta \leq \gamma$ and therefore encourages margin preservation under pruning. The ratio term δ/γ_ϵ, with $\gamma_\epsilon = \max(\gamma, \epsilon)$, acts as an auxiliary scale-aware regularizer that normalizes the deviation relative to the nominal margin and remains invariant to global logit scaling. Minimizing equation 11 promotes certified pruning robustness by reducing the deviation bound and increasing the margin.

Remark 4 *The bound $\delta(S, \mathbf{x}, t)$ is composed of standard operations such as matrix norms, maximum, and top-k selection, all of which are piecewise differentiable. This allows the certified deviation bound to be integrated directly into gradient-based optimization without requiring relaxation or surrogate approximations.*

5 Experiments

We evaluate the suitability of the proposed MILP verification framework (Section 4.1) and the pruning-robust training objective (Section 4.3) for Problems 1 and 2. First, we compare the verified accuracy under different pruning budgets for standard-trained models (trained with cross-entropy) and pruning-robust models (Section 5.1). We then analyze the verification tightness and scalability of MILP and margin-based bounds as the pruning budget increases (Section 5.2). Additionally, to assess the effect of pruning-robust training in adversarial settings, we evaluate models against three attacks: (i) projected gradient descent (PGD) adapted to pruning (Madry et al., 2017), (ii) ℓ_2-based adversarial pruning, where neurons with the largest weight norms are removed, and (iii) random pruning (Section 5.3). Finally, we study the stability of model performance across different pruning strategies (Section 5.4).

All models are trained on MNIST (LeCun et al., 2010) and CIFAR-10 (Krizhevsky et al., 2009) using PyTorch. Experiments are conducted on a Dell Precision 7680 with an Intel i9-13950HX (32 Core) CPU, 64GB RAM, and NVIDIA GeForce RTX 4090 Laptop GPU with 16GB VRAM.[1]

5.1 Verification of pruning-robustness

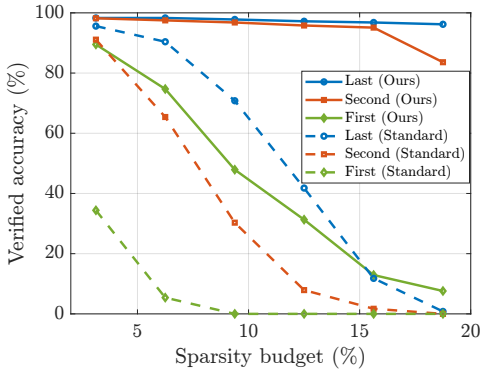


Figure 1: Verified accuracy on MNIST test set under single-layer pruning for robust-trained and standard models. “First”, “Second”, and “Last” denote pruning applied to the corresponding hidden layer.

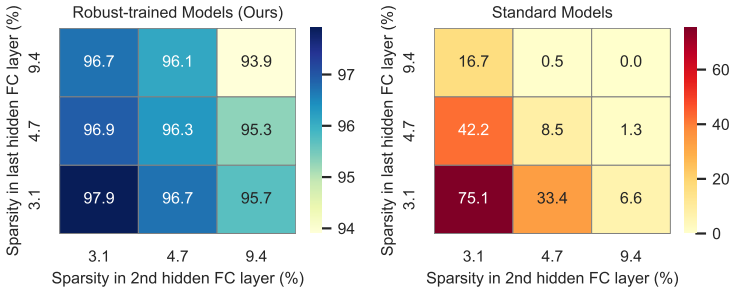


Figure 2: Verified accuracy on MNIST test set under multi-layer pruning settings for our trained-robust vs. standard models across different sparsity budgets in the last and second hidden FC layers.

We compute the verified accuracy of both robust-trained and standard-trained models on a fully connected network with four layers (128-64-32-10) using ReLU activations without bias terms. Verified accuracy is

[1]The source code will be made publicly available upon acceptance.

defined as the proportion of test samples for which the worst-case margin over all admissible pruning masks remains strictly positive. We report results obtained via MILP on 1000 MNIST test samples in Figure 1 (single-layer pruning), Figure 2 (multi-layer pruning), and Figure 3 (all-layer pruning). The MILP problems are solved to optimality, yielding exact certificates. Across all settings, robust training significantly improves certified robustness. For standard models, verified accuracy degrades rapidly as the sparsity budget increases, often collapsing at relatively small pruning levels. In contrast, robust-trained models maintain high verified accuracy over a substantially wider range of sparsity, particularly when pruning is applied to deeper layers. We also observe that pruning the first hidden layer leads to a significantly larger degradation in verified accuracy compared to deeper layers. This is expected, as early layers directly encode input features, and pruning at this stage removes fundamental representations that cannot be recovered by subsequent layers.

Additional experiments are provided in Appendix A.5. In particular, more results for multi-layer pruning are shown in Figures 6 and 7. The trade-off between verified accuracy and clean accuracy, along with verification runtimes, is reported in Table 3. Computational overhead of robust training compared to standard training is discussed in Table 4. A grid search over λ_1 and λ_2 is presented in Figure 5, and additional results on CIFAR-10 are given in Figure 8.

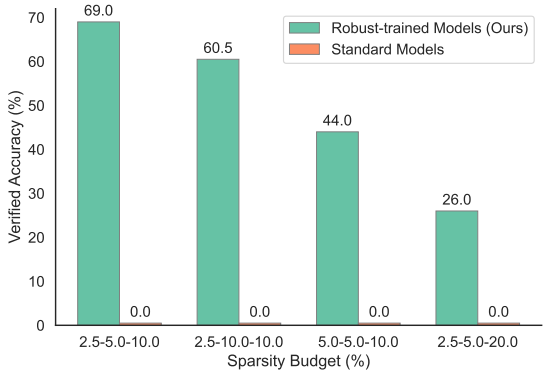


Figure 3: Verified accuracy on MNIST test set under all-layer pruning settings for trained-robust vs. standard models. Sparsity budgets denote pruning ratios to the first, second, last hidden FC layers, respectively.

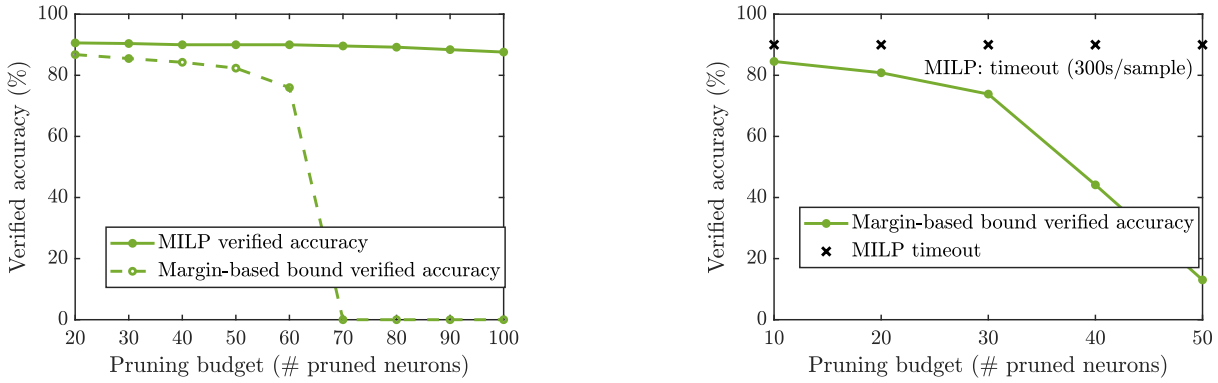
5.2 Verification Tightness and Scalability

We compare the verified accuracy of VGG11 on CIFAR-10 using MILP and margin-based bounds (Figure 4a and 4b). For pruning of the last fully connected layer (Figure 4a), margin-based bounds achieve comparable tightness to MILP under mild sparsity, but become increasingly loose as the pruning budget grows. Notably, in the tight-bound regime, margin-based bounds are significantly more efficient, verifying all 10k test images takes approximately 7 seconds on our machine, whereas MILP requires almost 5 hours (1.77s/image).

While MILP provides exact verification when solved to optimality, its practical scalability is limited. As shown in Figure 4b, the margin-based bound remains computationally efficient and yields non-trivial certificates, whereas MILP fails to terminate within a time limit of 300 seconds per sample. This is particularly pronounced in wider networks and when pruning is applied to early layers, where loose variable bounds and interactions with downstream activations increase the search complexity, rendering the problem intractable even for small pruning budgets. This highlights a trade-off between tightness and scalability and suggests a hybrid strategy in which analytical bounds are used to tighten intermediate variable bounds and pre-screen easy instances prior to selective exact MILP verification.

5.3 Robustness under Adversarial Pruning.

We observe that certified robustness is primarily achieved in the mild pruning regime (Section 5.1 and 5.2). As sparsity increases, the combinatorial nature of pruning induces increasingly adverse worst-case configurations, leading to a rapid collapse of certifiable margins even under exact MILP verification. To



(a) Last hidden FC layer.

(b) First hidden FC layer.

Figure 4: Verified accuracy on CIFAR-10 (VGG11) comparing MILP and margin-based bounds. Left: pruning the last fully connected layer. Right: pruning the first fully connected layer—MILP times out beyond 10 pruned neurons (300s/sample), while the bound remains tractable and provides non-trivial certificates.

evaluate robustness beyond this regime, we consider adversarial pruning attacks, including PGD-based, ℓ_2-based, and random pruning. We adapt PGD (Madry et al., 2017) by optimizing continuous neuron-importance scores and projecting them onto top-k binary masks (with $k = d_k - s_k$), using a straight-through estimator for gradient propagation. The attack is run for 400 steps. For the ℓ_2-based attack, we prune neurons with the largest row-wise weight norms, while for random pruning, we sample masks uniformly under the budget and report mean $\pm$ standard deviation over 1000 samples. We conduct this analysis on the SmallConv model on CIFAR-10, where pruning is applied to two hidden FC layers with 200 and 100 neurons. As these layers are relatively less redundant, pruning leads to more pronounced degradation.

able 1 shows that our method consistently outperforms both standard training and SAM (Foret et al., 2020) across all pruning budgets and attack types. Here, SAM (Foret et al., 2020) is chosen as a flatness-based robustness baseline. The gains are especially significant under strong adversarial pruning (PGD), where our model maintains higher accuracy even at high sparsity levels. Improvements are also observed under random pruning, indicating that the learned robustness generalizes beyond worst-case attacks.

5.4 Analysis under Different Pruning Methods

We further evaluate robustness under common structured pruning heuristics, including L1-norm-based pruning (Li et al., 2017), L2-norm pruning (Han et al., 2015), Taylor (Molchanov et al., 2017b), and ActMean (Molchanov et al., 2017a). This experiment is conducted on the SmallConv model on CIFAR-10. As shown in Table 2, our method achieves the highest accuracy in most settings, with significantly smaller average accuracy drops compared to standard and SAM, especially under aggressive pruning. These results indicate that our training objective not only improves worst-case robustness under adversarial pruning, but also enhances stability under practical pruning heuristics.

We also evaluated Targeted Dropout (TD) (Gomez et al., 2019) as a pruning-aware baseline in Appendix A.7. TD performs strongly under several benign pruning heuristics and exhibits relatively small clean-accuracy trade-offs. However, it is less reliable under adversarial pruning settings, where our method yields substantially stronger robustness. These results highlight a distinction between empirical resilience to heuristic pruning and robustness to worst-case pruning masks.

6 Conclusion and Discussion

We proposed a verification framework based on MILP and margin-based bounds to certify robustness against all admissible pruning patterns under layerwise sparsity constraints, together with a training objective for

Table 1: Empirical robustness under L2-adversarial pruning, PGD attack, and Random pruning (mean ± std over 1000 samples). Pruned (fc1, fc2) denotes the number of neurons removed from the corresponding hidden FC layers.

Pruned (fc1, fc2)	**Model**	**Unpruned**	**L2-adv**	**PGD**	**Random** ($\mu \pm \sigma$)
0-30	Standard	76.7	71.35	46.24	70.93 ± 1.42
	SAM	74.33	70.51	54.32	70.06 ± 0.92
	Ours	75.43	**73.57**	**61.12**	**74.84 ± 0.88**
0-40	Standard	76.7	67.36	29.58	67.98 ± 2.1
	SAM	74.33	67.21	42.63	68.52 ± 1.36
	Ours	75.33	**73.8**	**61.21**	**74.56 ± 1.2**
60-0	Standard	76.7	60.07	33.71	66.74 ± 1.5
	SAM	74.33	56.03	37.72	65.95 ± 1.27
	Ours	73.18	**69.10**	**53.42**	**71.12 ± 0.81**
80-0	Standard	76.7	52.8	26.18	62.28 ± 1.93
	SAM	74.33	47.96	22.7	62.15 ± 1.69
	Ours	73.48	**65.29**	**45.1**	**69.9 ± 1.2**
40-20	Standard	76.7	63.45	26.74	67.15 ± 1.71
	SAM	74.33	61.23	37.98	66.96 ± 1.27
	Ours	72.68	**66.28**	**55.38**	**70.77 ± 0.86**
60-30	Standard	76.7	54.85	12.21	61.16 ± 2.44
	SAM	74.33	51.61	16.53	62.09 ± 1.84
	Ours	72.88	**60.81**	**54.96**	**69.57 ± 1.23**

Table 2: Empirical robustness under representative structured pruning heuristics: L2-norm, L1, Taylor, and ActMean pruning, along with the average accuracy drop from the corresponding unpruned model. Pruned (fc1, fc2) denotes the number of neurons removed from the corresponding hidden FC layers.

Pruned (fc1, fc2)	**Model**	**L2**	**L1**	**Taylor**	**ActMean**	**Avg. drop**
0-40	Standard	68.7	69.01	74.46	74.60	4.71
	SAM	66.95	66.90	72.08	70.29	5.28
	Ours	**75.30**	**75.25**	**75.36**	**75.39**	**0.04**
80-0	Standard	66.20	62.78	69.62	69.41	9.70
	SAM	67.54	66.70	70.36	69.92	5.70
	Ours	**73.20**	**73.20**	**73.50**	**73.43**	**0.16**
60-30	Standard	65.53	66.36	72.73	72.62	7.39
	SAM	65.54	65.15	**72.89**	72.23	5.38
	Ours	**71.59**	**71.60**	72.75	72.57	**0.75**
80-40	Standard	58.18	56.91	66.08	67.8	14.46
	SAM	59.73	57.97	66.72	64.84	12.02
	Ours	**71.54**	**71.37**	**72.42**	**72.23**	**0.60**

pruning-robust models. While formal verification becomes challenging for larger networks, our results demonstrate non-trivial certified robustness in the mild pruning regime and strong empirical robustness under more aggressive pruning. Our framework focuses on neuron-level structured pruning in fully connected layers via row-masking operators. Extending the analysis to other structured pruning schemes, such as channel or filter pruning, is an interesting direction for future work. Finally, we note that our guarantees are deterministic and per-input; thus, the reported verified accuracy does not imply distribution-level generalization guarantees.

References

Kingma DP Ba J Adam et al. A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 1412 (6), 2014.

Alireza Aghasi, Afshin Abdi, Nam Nguyen, and Justin Romberg. Net-trim: Convex pruning of deep neural networks with performance guarantee. *Advances in neural information processing systems*, 30, 2017.

Mohammad Hasan Ahmadilivani, Mahdi Taheri, Jaan Raik, Masoud Daneshtalab, and Maksim Jenihhin. A systematic literature review on hardware reliability assessment methods for deep neural networks. *ACM Computing Surveys*, 56(6):1–39, 2024.

Abdolmaged Alkhulaifi, Fahad Alsahli, and Irfan Ahmad. Knowledge distillation in deep learning and its applications. *PeerJ Computer Science*, 7:e474, 2021.

Anna Bair, Hongxu Yin, Maying Shen, Pavlo Molchanov, and Jose Alvarez. Adaptive sharpness-aware pruning for robust sparse networks. *arXiv preprint arXiv:2306.14306*, 2023.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

Yanjiao Chen, Baolin Zheng, Zihan Zhang, Qian Wang, Chao Shen, and Qian Zhang. Deep learning on mobile and embedded devices: State-of-the-art, challenges, and future directions. *ACM Computing Surveys (CSUR)*, 53(4):1–37, 2020.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Xin Cheng and Xiang Li. Discretization and global optimization for mixed integer bilinear programming. *Journal of Global Optimization*, 84(4):843–867, 2022.

Hong-Ming Chiu and Richard Y Zhang. Tight certification of adversarially trained neural networks via nonconvex low-rank semidefinite relaxations. In *International Conference on Machine Learning*, pp. 5631–5660. PMLR, 2023.

Hue Dang, Matthew Robert Wicker, Goetz Botterweck, and Andrea Patane. Certifiably quantisation-robust training and inference of neural networks. In *The 28th International Conference on Artificial Intelligence and Statistics*, 2025.

Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Advances in neural information processing systems*, 30, 2017.

Marwa El Halabi, Suraj Srinivas, and Simon Lacoste-Julien. Data-efficient structured pruning via submodular optimization. *Advances in Neural Information Processing Systems*, 35:36613–36626, 2022.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Sumathi Gokulanathan, Alexander Feldsher, Adi Malca, Clark Barrett, and Guy Katz. Simplifying neural networks using formal verification. In *NASA Formal Methods: 12th International Symposium, NFM 2020, Moffett Field, CA, USA, May 11–15, 2020, Proceedings 12*, pp. 85–93. Springer, 2020.

Aidan N Gomez, Ivan Zhang, Siddhartha Rao Kamalakara, Divyam Madaan, Kevin Swersky, Yarin Gal, and Geoffrey E Hinton. Learning sparse networks using targeted dropout. *arXiv preprint arXiv:1905.13678*, 2019.

Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In *NIPS*, pp. 1135–1143, 2015. URL http://papers.nips.cc/paper/5784-learning-both-weights-and-connections-for-efficient-neural-network.

Yang He and Lingao Xiao. Structured pruning for deep convolutional neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 46(5):2900–2919, 2023.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. *Advances in neural information processing systems*, 30, 2017.

Zehao Huang and Naiyan Wang. Data-driven sparse structure selection for deep neural networks. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 304–320, 2018.

Nan-Fei Jiang, Xu Zhao, Chao-Yang Zhao, Yong-Qi An, Ming Tang, and Jin-Qiao Wang. Pruning-aware sparse regularization for network pruning. *Machine Intelligence Research*, 20(1):109–120, 2023.

Yuyang Jin, Runxin Zhong, Saiqin Long, and Jidong Zhai. Efficient inference for pruned cnn models on mobile devices with holistic sparsity alignment. *IEEE Transactions on Parallel and Distributed Systems*, 2024.

Najeeb Khan and Ian Stavness. Pruning convolutional filters using batch bridgeout. *IEEE Access*, 8:212003–212012, 2020.

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

Ori Lahav and Guy Katz. Pruning and slicing neural networks using formal verification. In *2021 Formal Methods in Computer Aided Design (FMCAD)*, pp. 183–192. IEEE, 2021.

Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. att labs, 2010.

Dongyeop Lee, Kwanhee Lee, Jinseok Chung, and Namhoon Lee. Safe: Finding sparse and flat minima to improve pruning. *arXiv preprint arXiv:2506.06866*, 2025.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *International Conference on Learning Representations (ICLR)*, 2017.

Jiaqi Li, Ross Drummond, and Stephen R Duncan. Robust error bounds for quantised and pruned neural networks. In *Learning for Dynamics and Control*, pp. 361–372. PMLR, 2021.

Junjie Liu, Zhe Xu, Runbin Shi, Ray CC Cheung, and Hayden KH So. Dynamic sparse training: Find efficient sparse network from scratch with trainable masked layers. *arXiv preprint arXiv:2005.06870*, 2020.

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

Garth P McCormick. Computability of global solutions to factorable nonconvex programs: Part i—convex underestimating problems. *Mathematical programming*, 10(1):147–175, 1976.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *ICLR (Poster)*, 2017a. URL https://openreview.net/forum?id=SJGCiw5gl.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. In *International Conference on Learning Representations (ICLR)*, 2017b.

Clara Na, Sanket Vaibhav Mehta, and Emma Strubell. Train flat, then compress: Sharpness-aware minimization learns more compressible models. *arXiv preprint arXiv:2205.12694*, 2022.

Alexandra Peste, Adrian Vladu, Eldar Kurtic, Christoph H Lampert, and Dan Alistarh. Cram: A compression-aware minimizer. *arXiv preprint arXiv:2207.14200*, 2022.

Konstantinos Pitas, Mike Davies, and Pierre Vandergheynst. Cheap dnn pruning with performance guarantees, 2018.

Babak Rokh, Ali Azarpeyvand, and Alireza Khanteymoori. A comprehensive survey on model quantization for deep neural networks in image classification. *ACM Transactions on Intelligent Systems and Technology*, 14(6):1–50, 2023.

Vincent Tjeng, Kai Xiao, and Russ Tedrake. Evaluating robustness of neural networks with mixed integer programming. *arXiv preprint arXiv:1711.07356*, 2017.

Yu-Lin Tsai, Chia-Yi Hsu, Chia-Mu Yu, and Pin-Yu Chen. Formalizing generalization and adversarial robustness of neural networks to weight perturbations. *Advances in Neural Information Processing Systems*, 34:19692–19704, 2021a.

Yu-Lin Tsai, Chia-Yi Hsu, Chia-Mu Yu, and Pin-Yu Chen. Non-singular adversarial robustness of neural networks. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3840–3844. IEEE, 2021b.

Lily Weng, Huan Zhang, Hongge Chen, Zhao Song, Cho-Jui Hsieh, Luca Daniel, Duane Boning, and Inderjit Dhillon. Towards fast computation of certified robustness for relu networks. In *International Conference on Machine Learning*, pp. 5276–5285. PMLR, 2018.

Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pp. 5286–5295. PMLR, 2018.

Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33:1129–1141, 2020.

Mao Ye, Lemeng Wu, and Qiang Liu. Greedy optimization provably wins the lottery: Logarithmic number of winning tickets is enough. *Advances in Neural Information Processing Systems*, 33:16409–16420, 2020.

Liping Yi, Xiaorong Shi, Nan Wang, Jinsong Zhang, Gang Wang, and Xiaoguang Liu. Fedpe: Adaptive model pruning-expanding for federated learning on mobile devices. *IEEE Transactions on Mobile Computing*, 23 (11):10475–10493, 2024.

Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018a.

Jeff Jun Zhang, Tianyu Gu, Kanad Basu, and Siddharth Garg. Analyzing and mitigating the impact of permanent faults on a systolic array based neural network accelerator. In *2018 IEEE 36th VLSI Test Symposium (VTS)*, pp. 1–6. IEEE, 2018b.

A Appendix

In this appendix section, we provide details of Proposition 1, 2, 3, 4. Additionally, we present details of the experiments and additional experiments mentioned in the main paper.

A.1 Proof of Preposition 1

From the definition of the worst-case pairwise class margin in the equation 6, we have:

$$\begin{aligned}
\Delta(S,\mathbf{x},t) &= \max_{\mathbf{m}\in M_{\text{pruned}}} \max_{i\neq t, i\in[C]} \left|(\hat{f}^{(t)}_{\mathbf{m}}(\mathbf{x}) - \hat{f}^{(i)}_{\mathbf{m}}(\mathbf{x})) - (f^{(t)}(\mathbf{x}) - f^{(i)}(\mathbf{x}))\right| \\
&\overset{\text{(i)}}{=} \max_{\mathbf{m}\in\mathcal{M}} \max_{i\neq t, i\in[C]} \left|\left(\mathbf{W}_{K+1}[t]^\top \cdot \hat{\mathbf{x}}^p_K - \mathbf{W}_{K+1}[i]^\top \cdot \hat{\mathbf{x}}^p_K\right) - \left(\mathbf{W}_{K+1}[t]^\top \cdot \mathbf{x}_K - \mathbf{W}_{K+1}[i]^\top \cdot \mathbf{x}_K\right)\right| \\
&= \max_{\mathbf{m}\in M_{\text{pruned}}} \max_{i\neq t, i\in[C]} \left|(\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i])^\top(\hat{\mathbf{x}}^p_K - \mathbf{x}_K)\right| \\
&\overset{\text{(ii)}}{\leq} \max_{\mathbf{m}\in M_{\text{pruned}}} \max_{i\neq t, i\in[C]} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2 \, \|\Delta\mathbf{x}_K\|_2 \\
&\overset{\text{(iii)}}{=} \max_{i\neq t, i\in[C]} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2 \cdot \max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta\mathbf{x}_K\|_2.
\end{aligned} \tag{12}$$

In the proof above, equality (i) is because there is no pruning in the output layer. In (ii), we apply the Cauchy-Schwarz inequality: $|\mathbf{a}^\top\mathbf{b}| \leq \|\mathbf{a}\|_2 \, \|\mathbf{b}\|_2$, where $\mathbf{a} = \mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]$ and $\mathbf{b} = \hat{\mathbf{x}}^p_K - \mathbf{x}_K$. and (iii) is beacause $\|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2$ does not depend on the mask $\mathbf{m}$, it can be taken outside the maximization.

Since $\max_{i\neq t,\, i\in[C]} \|\mathbf{W}_{K+1}[t] - \mathbf{W}_{K+1}[i]\|_2$ can be computed as the maximum pairwise Euclidean distance between row t and all other rows of $\mathbf{W}_{K+1}$, the remaining challenge in deriving an upper bound on the WMD is to bound the worst-case layer-wise output deviation norm $\|\Delta\mathbf{x}_K\|_2$. The corresponding results are presented in Propositions 2, 3, and 4, with detailed proofs provided below.

A.2 Proof of Single-layer Pruning Bound (For Proposition 2)

Let ℓ be the only pruned hidden layer with sparsity budget s_ℓ ($s_k = 0$ for all $k \neq \ell$). Since only layer ℓ is pruned, the representations before layer ℓ remain unchanged, i.e., $\hat{\mathbf{x}}^p_{\ell-1} = \mathbf{x}_{\ell-1}$. Hence,

$$\begin{aligned}
\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta\mathbf{x}_\ell\|_2 &= \max_{\mathbf{m}_\ell\in M_\ell} \|\mathbf{m}_\ell \circ \mathbf{x}_\ell - \mathbf{x}_\ell\|_2 \\
&= \max_{\mathbf{m}_\ell\in M_\ell} \|(\mathbf{m}_\ell - \mathbf{1}) \circ \mathbf{x}_\ell\|_2 \\
&= \max_{\mathbf{m}_\ell\in M_\ell} \|\bar{\mathbf{m}}_\ell \circ \mathbf{x}_\ell\|_2 \\
&= \|tops_{s_\ell}(\mathbf{x}_\ell)\|_2 = \delta_\ell,
\end{aligned}$$

where $\bar{\mathbf{m}}_\ell = \mathbf{1} - \mathbf{m}_\ell$, and the maximum is attained by selecting the s_ℓ entries of largest magnitude.

For $k > \ell$, no further pruning is applied, so

$$\begin{aligned}
\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta\mathbf{x}_k\|_2 &= \max_{\mathbf{m}\in M_{\text{pruned}}} \|\sigma(\mathbf{W}_k\hat{\mathbf{x}}^p_{k-1}) - \sigma(\mathbf{W}_k\mathbf{x}_{k-1})\|_2 \\
&\overset{\text{(i)}}{\leq} \max_{\mathbf{m}\in M_{\text{pruned}}} \|\mathbf{W}_k(\hat{\mathbf{x}}^p_{k-1} - \mathbf{x}_{k-1})\|_2 \\
&\overset{\text{(ii)}}{\leq} \|\mathbf{W}_k\|_2 \max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta\mathbf{x}_{k-1}\|_2,
\end{aligned} \tag{13}$$

where (i) uses the 1-Lipschitz property of σ, and (ii) follows from the definition of the spectral norm.

By recursion, $\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta\mathbf{x}_k\|_2 \leq \delta_k$ for $k = \ell+1, \ldots, K$. The result then follows from Proposition 1.

A.3 Proof for All-layer Pruning Bound - Proposition 3

For the first hidden layer,

$$\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_1\|_2 = \max_{\mathbf{m}_1\in M_1} \|\mathbf{m}_1 \circ \mathbf{x}_1 - \mathbf{x}_1\|_2 = \|tops_{s_1}(\mathbf{x}_1)\|_2 = \delta_1.$$

For each layer $k \geq 2$,

$$\begin{aligned}
\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 &= \max_{\mathbf{m}\in M_{\text{pruned}}} \|\mathbf{m}_k \circ \sigma(\mathbf{W}_k \hat{\mathbf{x}}_{k-1}^p) - \sigma(\mathbf{W}_k \mathbf{x}_{k-1})\|_2 \\
&\overset{\text{(i)}}{\leq} \max_{\mathbf{m}\in M_{\text{pruned}}} \|\mathbf{m}_k \circ \sigma(\mathbf{W}_k \hat{\mathbf{x}}_{k-1}^p) - \mathbf{m}_k \circ \sigma(\mathbf{W}_k \mathbf{x}_{k-1})\|_2 + \max_{\mathbf{m}_k\in M_k} \|\mathbf{m}_k \circ \mathbf{x}_k - \mathbf{x}_k\|_2 \\
&\overset{\text{(ii)}}{\leq} \max_{\mathbf{m}\in M_{\text{pruned}}} \|\sigma(\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k) \hat{\mathbf{x}}_{k-1}^p) - \sigma(\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k) \mathbf{x}_{k-1})\|_2 + \|tops_{s_k}(\mathbf{x}_k)\|_2 \\
&\overset{\text{(iii)}}{\leq} \max_{\mathbf{m}\in M_{\text{pruned}}} \|\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k)(\hat{\mathbf{x}}_{k-1}^p - \mathbf{x}_{k-1})\|_2 + \|tops_{s_k}(\mathbf{x}_k)\|_2 \\
&\overset{\text{(iv)}}{\leq} \|\mathbf{W}_k\|_2 \max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_{k-1}\|_2 + \|tops_{s_k}(\mathbf{x}_k)\|_2.
\end{aligned} \tag{14}$$

Here, (i) follows from the triangle inequality after adding and subtracting $\mathbf{m}_k \circ \sigma(\mathbf{W}_k \mathbf{x}_{k-1})$. Step (ii) uses the equivalence between activation masking and row masking under $\sigma(0) = 0$, together with the definition of $tops_{s_k}(\mathbf{x}_k)$. Step (iii) follows from the 1-Lipschitz property of σ. Step (iv) follows from the definition of the spectral norm and the inequality $\|\mathcal{P}_{\mathbf{m}_k}(\mathbf{W}_k)\|_2 \leq \|\mathbf{W}_k\|_2$. Thus, by recursion,

$$\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 \leq \delta_k, \qquad k = 1, \ldots, K.$$

Applying Proposition 1 yields the desired result.

A.4 Proof for Multi-layer Pruning Bound - Proposition 4

Let $I_p \subseteq [K]$ denote the set of pruned hidden layers, and let $\ell = \min I_p$ be the first pruned layer. Since no pruning is applied before layer ℓ, the hidden representations up to layer $\ell-1$ are unchanged, i.e., $\hat{\mathbf{x}}_{\ell-1}^p = \mathbf{x}_{\ell-1}$. Hence,

$$\begin{aligned}
\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_\ell\|_2 &= \max_{\mathbf{m}_\ell\in M_\ell} \|\mathbf{m}_\ell \circ \mathbf{x}_\ell - \mathbf{x}_\ell\|_2 \\
&= \max_{\mathbf{m}_\ell\in M_\ell} \|\bar{\mathbf{m}}_\ell \circ \mathbf{x}_\ell\|_2 \\
&= \|tops_{s_\ell}(\mathbf{x}_\ell)\|_2 = \delta_\ell,
\end{aligned}$$

where $\bar{\mathbf{m}}_\ell = \mathbf{1} - \mathbf{m}_\ell$.

For any layer $k > \ell$, two cases arise.

Case 1: $k \notin I_p$**.** No pruning is applied at layer k, we apply equation 13 from the single-layer pruning bound, that is,

$$\begin{aligned}
\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 &= \max_{\mathbf{m}\in M_{\text{pruned}}} \|\sigma(\mathbf{W}_k \hat{\mathbf{x}}_{k-1}^p) - \sigma(\mathbf{W}_k \mathbf{x}_{k-1})\|_2 \\
&\leq \|\mathbf{W}_k\|_2 \max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_{k-1}\|_2.
\end{aligned}$$

Thus, $\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 \leq \|\mathbf{W}_k\|_2 \delta_{k-1} = \delta_k$.

Case 2: $k \in I_p$**.** Pruning is applied at layer k, we apply equation 14 from the all-layer pruning bound

$$\begin{aligned}\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 = &\max_{\mathbf{m}\in M_{\text{pruned}}} \|\mathbf{m}_k \circ \sigma(\mathbf{W}_k \hat{\mathbf{x}}^p_{k-1}) - \sigma(\mathbf{W}_k \mathbf{x}_{k-1})\|_2 \\ \leq &\|\mathbf{W}_k\|_2 \max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_{k-1}\|_2 + \|tops_{s_k}(\mathbf{x}_k)\|_2.\end{aligned}$$

Therefore,

$$\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 \leq \|\mathbf{W}_k\|_2 \delta_{k-1} + \|tops_{s_k}(\mathbf{x}_k)\|_2 = \delta_k.$$

Combining the two cases, we obtain recursively that

$$\max_{\mathbf{m}\in M_{\text{pruned}}} \|\Delta \mathbf{x}_k\|_2 \leq \delta_k, \qquad k = \ell, \ldots, K.$$

The result then follows from Proposition 1.

A.5 Additional Experiments on MNIST dataset

For experiments on MNIST (LeCun et al., 2010) on the architecture of three hidden layers 128-64-32-10 reported in the manuscript, image pixel values were rescaled from the range $[0, 255]$ to $[0, 1]$ without applying any additional preprocessing. For standard training, the neural network was trained using Adam algorithm (Adam et al., 2014) with a weight decay of 1×10^{-3}. The model was trained for 120 epochs with a batch size of 128 on an NVIDIA GeForce RTX 4090 Laptop GPU.

For robust training, we also used the Adam algorithm with the same hyperparameters as defined in the standard training process. We leverage a learning curriculum by scheduling the values of λ_1 and λ_2. Specifically, after 5 warm-up epochs, we linearly increase the scaling factor λ_1, λ_2 from 0 to the target values of λ_1 and λ_2 over next 10 epochs. The schedule is applied at every batch update, resulting in a total of $10\times$ (number of batches per epoch) linear steps. The target values of λ_1 and λ_2 are selected from a grid search (Figure 5). Morever, we also propose a more principle way to calibrate λ_1 and λ_2 in Section A.6.

We evaluate the verified accuracy on a subset of the MNIST test set using our MILP verification method. For each test sample, the optimization problem is solved using Gurobi 12.0.3 with a 300-second time limit. If Gurobi fails to find a solution within the allotted time or the optimization is unsuccessful due to an out-of-memory error, the sample is classified as non-robust.

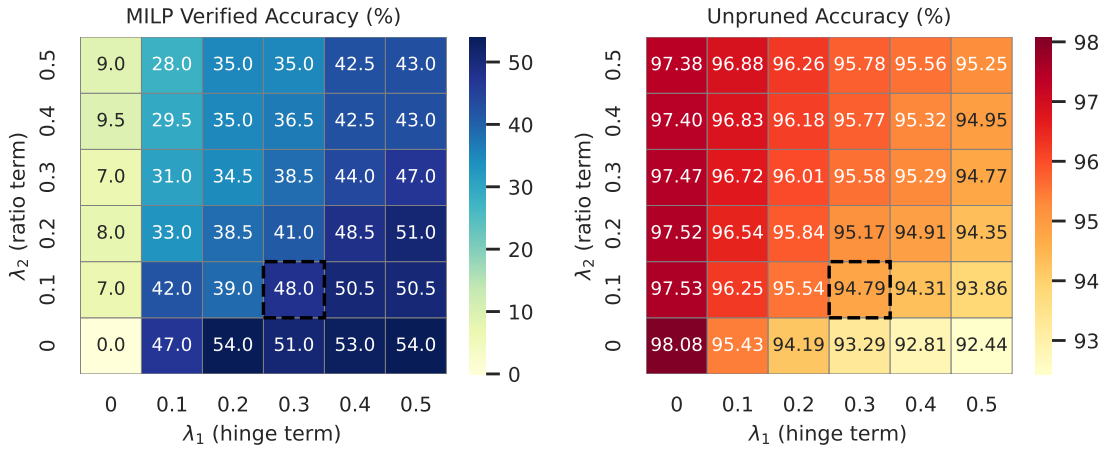


Figure 5: Grid search over λ_1 (hinge term) and λ_2 (ratio term) under 10% neuron pruning at the first hidden FC layer. The left heatmap shows MILP verified accuracy (%), and the right heatmap shows clean accuracy (%). The selected configuration $(\lambda_1, \lambda_2) = (0.3, 0.1)$ achieves a favorable balance between verified and unpruned performance.

Table 3: Representative clean–verified accuracy trade-offs under different pruning settings for the 128–64–32–10 network on MNIST. Sparsity (s_1, s_2, s_3) denotes the percentage of neurons pruned in the first, second, and last hidden layers, respectively. The corresponding number of pruned neurons is obtained by rounding to the nearest integer.

Sparsity ratio (%)	**Clean acc. (%)**		**Verified acc. (%)**		**Runtime (s/sample)**	
	Robust	**Standard**	**Robust**	**Standard**	**Robust**	**Standard**
$(0, 0, 10)$	98.19	98.08	**96.2**	70.8	**0.14**	0.16
$(0, 0, 20)$	98.13	98.08	**96.2**	0.8	0.56	**0.16**
$(0, 10, 0)$	97.43	98.08	**96.8**	30.3	0.55	**0.24**
$(0, 20, 0)$	97.64	98.08	**83.6**	0.0	0.7	**0.46**
$(10, 0, 0)$	94.79	98.08	**47.9**	0.0	**4.85**	17.89
$(13, 0, 0)$	94.99	98.08	**31.3**	0.0	**4.22**	24.41
$(0, 10, 10)$	97.38	98.08	**93.9**	0.0	0.99	**0.50**
$(6, 0, 10)$	95.22	98.08	**56.0**	0.0	**18.74**	30.29
$(6, 10, 0)$	95.14	98.08	**43.90**	0.0	38.66	**18.18**
$(5, 5, 10)$	94.89	98.08	**44.0**	0.0	70.44	**22.34**

Grid Search over λ_1 and λ_2 To select λ_1 and λ_2 for the MNIST experiments reported in the main paper, we perform a grid search over the range $[0, 0.5]$ under a representative single-layer pruning setting (10% pruning at the first hidden layer). The results are shown in Figure 5, where verified accuracy is evaluated on 200 MNIST test samples using MILP.

The row $\lambda_2 = 0$ isolates the effect of the hinge term alone, while the column $\lambda_1 = 0$ isolates the ratio term, effectively serving as an ablation study over the two loss components. This comparison reveals that verified robustness is primarily driven by the hinge term λ_1: increasing λ_1 leads to substantial improvements in verified accuracy, whereas the ratio term λ_2 has a comparatively smaller but non-negligible effect, mainly refining performance for a fixed λ_1. This suggests that enforcing an absolute margin constraint (via the hinge term) is more critical for pruning robustness than relative normalization.

We also observe a clear trade-off between verified and clean accuracy: larger values of λ_1 improve robustness but reduce clean accuracy. The ratio term λ_2 provides additional flexibility in navigating this trade-off without substantially degrading verified accuracy. Based on these observations, we select $(\lambda_1, \lambda_2) = (0.3, 0.1)$ as a configuration that achieves a favorable balance between verified robustness and clean accuracy. All MNIST results reported in the main paper use this configuration; setting-specific tuning could further improve results for individual pruning budgets.

Robustness-Accuracy trade-off and Verification Runtime Analysis Table 3 highlights a clear trade-off between clean (unpruned) and verified accuracy across different pruning settings. Across all settings, the robust-trained model consistently achieves substantially higher verified accuracy than the standard model, particularly under aggressive and multi-layer pruning, while incurring only a modest reduction in clean accuracy (within 4%).

We also report the average verification runtime per MNIST sample. In many cases where the robust model attains high verified accuracy, its verification runtime is higher than that of the standard model, whose verified accuracy often collapses to zero. This suggests that finding counterexamples is easier for standard models, allowing the verifier to terminate earlier. In contrast, when the robust model achieves moderate verified accuracy, its runtime is often lower than that of the standard model. One possible explanation is that robust training leads to tighter intermediate bounds, which reduces the effective search space and improves verification efficiency.

Training Runtime Analysis Table 4 shows that the computational overhead of robust training depends on the model architecture. Robust training introduces additional computations, including spectral norm

Table 4: Per-epoch training run time (in seconds) averaged over 20 epochs for the standard and robust models across different datasets and architectures.

	Standard model	Robust model
MLP (MNIST)	1.012 sec.	2.645 sec.
SmallConv (CIFAR-10)	5.656 sec.	5.717 sec.

estimation with complexity $\mathcal{O}(kd^2)$ for fully connected layers, where k is the number of power iterations and d is the layer width. However, in convolutional neural networks (CNNs), the overall training cost is dominated by convolutional operations, which typically scale as $\mathcal{O}(HWC_{\text{in}}C_{\text{out}}K^2)$. As a result, the additional overhead from the robustness terms becomes negligible in CNN training. In contrast, for smaller architectures such as MLPs, where the baseline computational cost is low, the relative overhead becomes more noticeable.

More experiments on multi-layer pruning More experiments on multi-layer pruning are shown in Figure 6 and 7, showing that robust training significantly improves certified robustness. For standard models, verified accuracy degrades rapidly as the sparsity budget increases, often collapsing at relatively small pruning budgets.

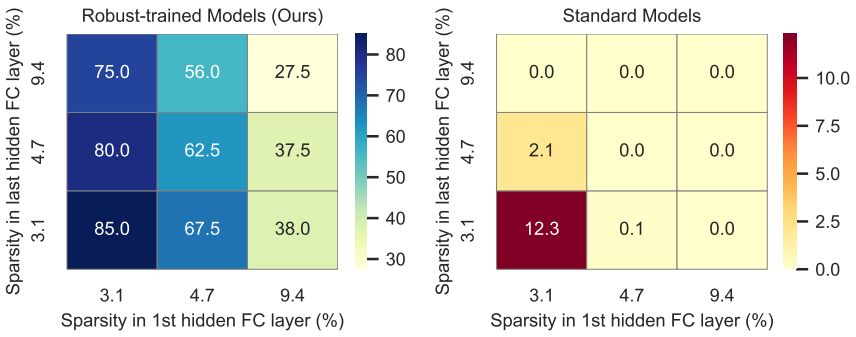


Figure 6: Verified accuracy on MNIST test set under multi-layer pruning settings for our trained-robust vs. standard models across different sparsity budgets in the first and last hidden FC layers.

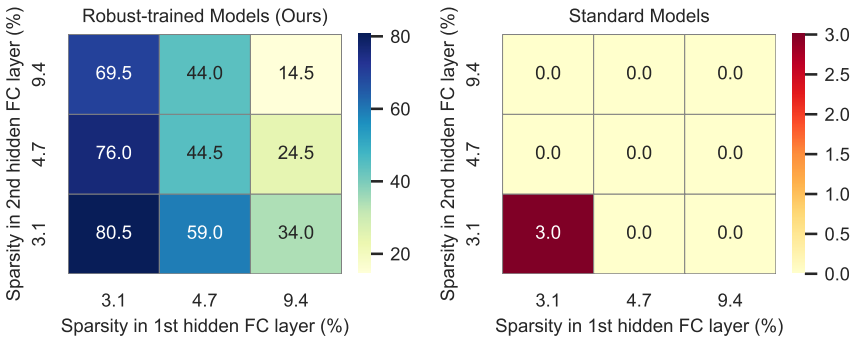


Figure 7: Verified accuracy on MNIST test set under multi-layer pruning settings for our trained-robust vs. standard models across different sparsity budgets in the first and second hidden FC layers.

A.6 Calibration of λ_1 and λ_2 in SmallConv-CIFAR-10

For CIFAR-10 (Krizhevsky et al., 2009), we use a SmallConv architecture consisting of three convolutional layers (with 8, 16, and 32 filters) followed by two hidden fully connected (FC) layers of sizes 200 and 100. Input images are normalized using mean $[0.4914, 0.4822, 0.4465]$ and standard deviation $[0.2470, 0.2435, 0.2616]$. Structured pruning is applied only to the FC layers.

To stabilize optimization, we adopt a curriculum schedule for λ_1 and λ_2: after 20 warm-up epochs, both coefficients are linearly increased from 0 to their target values over the next 20 epochs, with updates applied at each batch.

We adopt a consistent tuning strategy across datasets, where grid search is used to select λ_1 and λ_2. For CIFAR-10, we further leverage gradient scale analysis to guide the search space, providing a more principled and scalable hyperparameter calibration. Specifically, we estimate the relative magnitudes and alignment of the gradients of each loss component. For pruning on the last FC layer (0–10 setting), we observe that $g_1/g_{\text{CE}} \approx 0.15$ and $g_2/g_{\text{CE}} \approx 0.0025$, with $\cos(\text{CE}, L_1) \approx -0.06$ and $\cos(\text{CE}, L_2) \approx 0.29$. This suggests that the hinge term L_1 can be increased moderately without significantly degrading clean accuracy.

We therefore search λ_1 and λ_2 around these ratios. Figure 8 shows the grid search results. We observe trends consistent with MNIST: (i) verified robustness is primarily driven by λ_1, (ii) increasing λ_1 improves robustness at a mild cost to clean accuracy, and (iii) varying λ_2 provides additional flexibility in the verified–clean trade-off. Based on these results, we select $(\lambda_1, \lambda_2) = (10^{-2}, 0.3)$ for pruning on the last fully connected layer. A similar strategy is applied for the first fully connected layer and all-layer pruning experiments, where we select $(\lambda_1, \lambda_2) = (10^{-2}, 0.05)$.

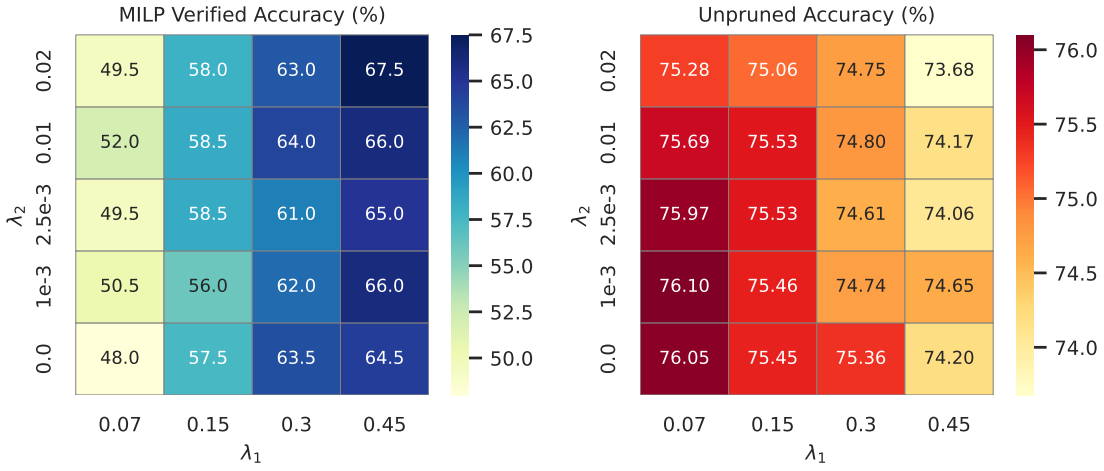


Figure 8: Grid search over λ_1 (hinge term) and λ_2 (ratio term) under 10% neuron pruning at the last hidden FC layer. The left heatmap shows CIFAR-10 verified accuracy (%), and the right heatmap shows clean accuracy (%)

A trade-off between clean (unpruned) and verified accuracy across different pruning settings on Smallconv-CIFAR-10 is shown in Table 5.

A.7 Targeted-dropout Model Performance under Different Pruning Methods

We evaluate Targeted Dropout (TD) (Gomez et al., 2019) as a pruning-aware baseline that encourages resilience to likely removable units during training. Table 6 shows a clear separation between heuristic (benign) pruning and worst-case pruning robustness. While TD performs well under non-adversarial pruning criteria, it is less reliable under adversarial pruning, where our method achieves substantially stronger robustness. Notably, this robustness gap widens as the pruning budget increases, highlighting the vulnerability of TD under more aggressive pruning.

Table 5: Verified accuracy under different settings for SmallConv on 1000 CIFAR-10 test images. Sparsity (s_1, s_2) denotes the number of neurons pruned in the first and last hidden FC layers, respectively.

Sparsity	**Clean acc. (%)**		**Verified acc. (%)**	
	Robust	**Standard**	**Robust**	**Standard**
(0, 5)	74.99	76.70	**69.0**	19.9
(0, 10)	74.80	76.70	**63.5**	0.7
(0, 15)	75.84	76.70	**18.0**	0.0
(5, 0)	70.66	76.70	**65.6**	2.5
(10, 0)	69.79	76.70	**45.0**	0.1
(15, 0)	68.25	76.70	**33.4**	0.0
(5, 10)	70.32	76.70	**32.5**	0.0

Table 6: Performance under L1-norm, L2-norm, Taylor, Actmean, PGD attack, and random pruning (mean ± std over 1000 samples).

Pruned (fc1, fc2)	**Model**	**Unpruned**	**L1**	**L2**	**Taylor**	**Actmean**	**PGD**	**Random** $(\mu \pm \sigma)$
0-30	Standard	76.7	72.17	72.0	75.88	75.64	46.24	70.93 ± 1.42
	TD	75.79	75.12	75.10	75.65	75.52	40.79	67.09 ± 3.09
	Ours	75.43	75.48	75.48	75.42	75.42	**61.12**	**74.84 ± 0.88**
0-40	Standard	76.7	69.01	68.7	74.46	74.60	29.58	67.98 ± 2.1
	TD	76.87	76.15	76.23	76.63	76.18	38.36	61.25 ± 4.96
	Ours	75.33	75.25	75.30	75.36	75.39	**61.21**	**74.56 ± 1.2**
60-0	Standard	76.7	69.91	70.51	73.57	73.38	33.71	66.74 ± 1.5
	TD	76.81	76.65	76.65	76.73	76.56	31.73	66.52 ± 1.74
	Ours	73.18	72.98	72.94	73.18	73.18	**53.42**	**71.12 ± 0.81**
80-0	Standard	76.7	62.78	66.2	69.62	69.41	26.18	62.28 ± 1.93
	TD	76.36	75.90	75.99	76.0	75.69	31.62	61.79 ± 2.25
	Ours	73.48	73.20	73.20	73.50	73.43	**45.1**	**69.9 ± 1.2**
60-30	Standard	76.7	66.36	66.53	72.73	72.62	12.21	61.16 ± 2.44
	TD	76.45	75.85	75.71	76.04	75.76	13.97	60.36 ± 2.78
	Ours	72.88	71.37	71.54	72.42	72.23	**54.96**	**69.57 ± 1.23**

TD also exhibits higher variance under random pruning, suggesting limited generalization beyond the specific pruning heuristic used during training. We attribute this gap to the training objectives: TD promotes robustness to likely removable units, whereas our method directly optimizes a bound on the worst-case pairwise margin degradation over all pruning masks within a given sparsity budget, leading to stronger and more generalizable robustness.