TOWARDS COMPACT AND ROBUST DNNS VIA COMPRESSION-AWARE STRUCTURAL STABILITY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sharpness-Aware Minimization (SAM) has recently emerged as an effective technique for improving DNN robustness to input variations. However, its interplay with the compactness requirements of on-device DNN deployments remains less explored. Simply pruning a SAM-trained model can undermine robustness, since flatness in the continuous parameter space does not necessarily translate to robustness under the discrete structural changes induced by pruning. Conversely, applying SAM after pruning may be fundamentally constrained by architectural limitations imposed by an early, robustness-agnostic pruning pattern. Motivated by this mismatch between robustness-aware training and compression, we propose **C**ompression-aware **S**tructural **S**tability **O**ptimization (C-SSO). Rather than applying SAM before or after pruning, C-SSO optimizes pruning masks under stochastic structural perturbations, favoring subnetworks whose predictions remain stable under nearby connectivity changes while preserving robustness to input variations. Extensive experiments on CelebA-HQ, Flowers-102, CIFAR-10-C, and Tiny-ImageNet, spanning ResNet-18, GoogLeNet, MobileNet-V2, and DeiT-Ti, show that C-SSO consistently achieves higher certified robustness than competing baselines, with improvements of up to 42%, while maintaining task accuracy comparable to the corresponding unpruned models.

1 INTRODUCTION

DNN robustness has attracted increasing research attention. A model is considered robust if it maintains accurate and consistent outputs when its inputs are subject to natural variations encountered in real-world settings, a property closely tied to its generalization capability. Recently, many studies have leveraged Sharpness-Aware Minimization (SAM) (Mi et al., 2025; Li et al., 2024; Andriushchenko & Flammarion, 2022; Liu et al., 2022a;b; Foret et al., 2021) to enhance model robustness. The underlying philosophy is that a smoother, flatter loss landscape correlates with stronger generalization (Luo et al., 2024; Wen et al., 2022; Li et al., 2018). Accordingly, SAM explicitly promotes a flat loss landscape during training by optimizing model parameters against a local worst-case perturbation in weight space and minimizing the task loss.

Despite SAM's effectiveness in improving DNN robustness, deploying it on edge devices introduces new challenges. Strict memory limits necessitate compressing a full DNN into a lightweight variant, yet typical edge applications, such as facial user authentication and in-home activity monitoring, still demand strong model robustness in real-world deployments.

A seemingly intuitive approach is to prune a SAM-trained model. However, SAM is formulated over a dense and continuous parameter space, whereas pruning induces discrete structural changes. As a result, a model that lies in a flat region with respect to weight-magnitude perturbations may still be highly sensitive to structural connectivity changes, and its robustness is therefore not guaranteed to be preserved after pruning. Recent work, S^2-SAM (Ji et al., 2024), can be used to fine-tune the pruned model with SAM to enforce robustness. In addition, AdaSAP (Bair et al., 2024) adopts a progressive strategy that alternates between pruning to an intermediate sparsity level and applying SAM, iterating this process until the target pruning ratio is reached. However, in these pipelines, the pruning stage remains largely agnostic to robustness objectives, and the achievable robustness may be

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

fundamentally bounded by pruning-induced structural constraints. We also notice that some earlier efforts, e.g., HYDRA, Flying Bird Sehwag et al. (2020); Chen et al. (2022), reconciled robustness with compactness primarily through adversarial training under pixel-level ℓ_p-norm perturbation assumptions (Goodfellow et al., 2015; Madry et al., 2018). Consequently, compared with SAM, the resulting models are less robust to realistic input variations that arise along the underlying data manifold rather than in pixel space, such as changes in illumination, weather conditions, or facial expressions (Yuan et al., 2023; Mirman et al., 2021). Therefore, a natural research question arises: *Is it feasible to identify an effective pruning pattern that simultaneously optimizes model compactness and robustness to input variations?*

To this end, we propose **C**ompression-aware **S**tructural **S**tability **O**ptimization (C-SSO). Instead of treating pruning as a post-processing step around SAM, C-SSO brings robustness-aware optimization directly into the mask search stage. During training, we shift optimization from isolated per-parameter masks to the model structure level: by injecting stochastic mask perturbations across the parameter space, C-SSO encourages the selected subnetwork topology to remain stable under nearby structural changes. This yields a notion of *mask-space structural stability* tailored to compression while preserving robustness to input variations.

To materialize our idea, the key step is to define an effective objective over the mask variables that drives the learning of a pruning pattern achieving both the target pruning ratio and strong robustness. Specifically, C-SSO features a composite objective that jointly accounts for structural stability, input semantic robustness and deployment consistency, as illustrated in Figure 1. We introduce a stability regularizer $\mathcal{L}_{\text{stab}}$ to minimize prediction variance under mask perturbations, ensuring that nearby subnetworks yield consistent predictions. Additionally, $\mathcal{L}_{\text{consis}}$ penalizes the discrepancy between continuous soft masks and discrete deployed masks to prevent deployment degradation. Finally, we derive a mathematical sufficiency condition that enables the design of a rigorous margin-aware ratio loss, $\mathcal{L}_{\text{ratio}}$ to enforce robustness against input variations generated by diffusion models (Preechakul et al., 2022; Kim et al., 2022). Together, these losses steer training toward mask configurations that are structurally stable under pruning perturbations while remaining semantically robust and deployable after binarization, ultimately yielding a compact yet robust DNN. Our key contributions are summarized as follows:

• We introduce the notion of mask-space structural stability, shifting robust compression from isolated per-mask tuning to the model structure level, operationalized via global mask perturbations and stability regularization.

• We propose C-SSO, a compression-aware training framework that optimizes pruning masks via a composite objective integrating mask-space structural stability, certified semantic robustness, and deployment consistency.

• Extensive experiments across multiple benchmarks, including CelebA-HQ (Lee et al., 2020), Flowers-102 (Nilsback & Zisserman, 2008), CIFAR-10-C (Krizhevsky et al., 2009; Hendrycks & Dietterich, 2019), and Tiny-ImageNet, and architectures including ResNet-18 (He et al., 2016), MobileNet-V2 (Sandler et al., 2018), GoogLeNet (Szegedy et al., 2015), and DeiT-Ti (Touvron et al., 2021), demonstrate that C-SSO identifies pruning patterns that meet the compactness target while achieving substantially higher certified robustness than competing baselines. Notably, on CelebA-HQ, C-SSO improves certified robustness by up to 42% and even surpasses unpruned models by 8%–27% at a 50% compression ratio in Table 7.

2 Related Work

Model Compression. Model compression aims to reduce a model's memory footprint. Pruning is one of the effective approaches among others and has been extensively studied, including *unstructured* weight sparsification (Han et al., 2015; Guo et al., 2016; Han et al., 2016; Frankle & Carbin, 2019; Lee et al., 2019; Wang et al., 2020; Tanaka et al., 2020) and *structured* channel/filter removal (Fang et al., 2023; Chen et al., 2024; 2021; 2023; Li et al., 2017; He et al., 2018; Liu et al., 2021; Yin et al., 2023; You et al., 2019; Yang et al., 2025; Huang et al., 2025).

Model Robustness. Model robustness is crucial for real-world deployment, as models must maintain accurate and consistent outputs even when inputs undergo natural variations in the operating environment. In our work, we adopt the more rigorous notion of certified robustness (Pautov et al., 2022; Li et al., 2023a) as our optimization objective.

Compact and Robust DNNs. On-device DNNs must be compact to satisfy stringent memory/storage constraints, while remaining robust to input variations. Early efforts combine adversarial training with compression (Sehwag et al., 2020; Chen et al., 2022; Zhao & Wressnegger, 2023; Gui et al., 2019; Shumailov et al., 2019; Diffenderfer et al., 2021; Dhillon et al., 2018; Sehwag et al., 2019; Ye et al., 2019; Li et al., 2023b): HYDRA (Sehwag et al., 2020) employs a learnable soft mask to minimize adversarial loss, while Flying Bird (Chen et al., 2022) jointly optimizes weights and connectivity. Crucially, such methods optimize masks on an isolated per-mask basis under pixel attacks, and their reliance on ℓ_p-bounded perturbations limits their effectiveness against natural semantic variations. Meanwhile, SAM has emerged as an effective tool for enhancing general robustness (Foret et al., 2021; Mi et al., 2025; Li et al., 2024; Andriushchenko & Flammarion, 2022; Mi et al., 2022; Liu et al., 2022a;b) by encouraging convergence to flat minima in the parameter loss landscape. Recent works attempt to integrate SAM with pruning: S^2-SAM (Ji et al., 2024) fine-tunes pruned models with SAM to mitigate chaotic loss landscapes in sparse networks and derive theoretical bounds on generalization error, while AdaSAP (Bair et al., 2024) progressively alternates between intermediate pruning and SAM updates, demonstrating robustness against out-of-distribution shifts such as Gaussian noise or fog corruptions (Hendrycks & Dietterich, 2019). However, these pipelines optimize pruning and robustness sequentially: pruning first removes parameters on an isolated per-mask basis, and subsequent robustness fine-tuning is then fundamentally bounded by this fixed topology. In contrast, C-SSO unites both objectives at the model structure level: by injecting stochastic structural perturbations during mask search, it directly identifies subnetworks whose overall topology is resilient to connectivity changes while preserving semantic consistency.

3 DESIGN

3.1 PRELIMINARY

Given a classification task represented by the dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m$, $x_i \in \mathbb{R}^n$ and $y_i \in \{1, \ldots, K\}$. We employ a DNN-based classifier f, parameterized by θ, such that $f : \mathbb{R}^n \to [0,1]^K$ maps each input to a probability distribution over K classes. The final prediction is determined by the class with the highest probability, defined as $\hat{y}(\cdot) \triangleq \arg\max f(\cdot)$.

Following Chernoff-Cramer Certification (CC-Cert) framework (Pautov et al., 2022), for an input x with true class c and transform $T : \mathbb{R}^n \to \mathbb{R}^n$, we obtain a space of transformed inputs $\mathbb{S}_T(x)$, under T. The model f is said to be probabilistically robust at x with confidence $1 - \varepsilon$ if the consistency of the prediction $\hat{y}$ over $\mathbb{S}_T(x)$ satisfies

$$\mathbb{P}_{x_T \sim \mathbb{S}_T}\Big(\hat{y}(x_T) = c\Big) \;\geq\; 1 - \varepsilon. \tag{1}$$

To operationalize this, we quantify the prediction discrepancy between x and its transformed counterpart $x_T \sim \mathbb{S}_T(x)$ as

$$Z(x;T) \;\triangleq\; \|\mathbf{p}(x) - \mathbf{p}(x_T)\|_\infty \tag{2}$$

where $\mathbf{p}(x) = f(x)$ denotes the class-probability vector, and we copy entries from $\mathbf{p}(x)$ and sort them in descending order, $p_1 > p_2 > \cdots > p_K$ to measure the margin between the top-2 predictions of input x as $d(x) \triangleq \frac{p_1 - p_2}{2}$.

Thus, assuming the model correctly classifies input x, a sufficient condition for satisfying probabilistic robustness defined in Equation (1) is $Z(x;T) < d(x)$, and we provide the complete proof in Appendix A.2. To summarize, increasing the certified passing probability is a surrogate for optimizing probabilistic robustness

$$\max\; \mathbb{P}\big(Z(x;T) < d(x)\big) \tag{3}$$

which can be achieved by minimizing the prediction discrepancy $Z(x;T)$ or by maximizing the margin $d(x)$.

Following Mirman et al. (2021); Yuan et al. (2023); Pérez et al. (2023), let $G : z \in \mathcal{Z} \to x \in \mathcal{M}$ be a generator that approximates the natural image manifold $\mathcal{M}$ by mapping a lower-dimensional latent vector $z \in \mathcal{Z}$ to the image space. Formally, given an input $x = G(z)$ and a target semantic attribute associated with a unit direction vector $\mathbf{v} \in \mathbb{R}^d$, the mutated image x_T is generated as:

$$x_T = T(x, \delta, \mathbf{v}) = G(z + \delta \cdot \mathbf{v})$$

where $\delta \in [0, 1]$ represents the magnitude of the semantic shift, e.g., the intensity of smiling. The set of all semantically mutated images is defined as $\mathbb{S}_T(x) = \{G(z + \delta \cdot \mathbf{v}) \mid \delta \in [0, 1]\}$. An example of a transformation space is provided in Figure 5.

For model compression, C-SSO is applicable to both structured and unstructured pruning. Let $\mathcal{P}$ denote the set of minimal prunable units, such as an individual weight for unstructured pruning or a channel for structured pruning, with $N = |\mathcal{P}|$ representing the total number of such units. We introduce a learnable soft mask $C \in [0, 1]^N$ as a continuous surrogate. To apply this mask to the weights, we define a broadcasting operator $\mathcal{T} : \mathbb{R}^N \to \mathbb{R}^{|\theta|}$, resulting in the compressed model $f_C(\cdot) = f(\cdot; \mathcal{T}(C) \odot \theta)$. Specifically, for unstructured pruning, $N = |\theta|$ and $\mathcal{T}$ reduces to the identity mapping. In contrast, for structured pruning, $N = C_{\text{out}}$ and $\mathcal{T}$ acts as a broadcasting operator, where C_{out} denotes the number of output channels in CNNs (Luo et al., 2017; Liu et al., 2017).

Predefined pruning ratio (pr) can be written as: $\text{pr} = 1 - \|\theta_{\hat{m}}\|_0/|\theta|$, where $\theta_{\hat{m}} = \mathcal{T}(\hat{m}) \odot \theta$ denotes the active parameters of the deployed model. The binary mask $\hat{m} \in \{0, 1\}^N$ is derived by projecting C onto the target density k, where $k = 1 - \text{pr}$. Specifically, we prune the model in a layer-wise way with a layer-wise threshold τ_k^i such that the top-k proportion of elements in the i-th layer of C are preserved:

$$\hat{m}^i = \mathbf{1}(C^i \geq \tau_k^i), \quad \text{s.t.} \quad \|\theta_{\hat{m}}\|_0 \leq k \cdot |\theta| \tag{4}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

3.2 Our Approach

In this section, we first introduce the initialization method for the soft mask C, followed by a detailed discussion of the composite loss components: (1) $\mathcal{L}_{\text{stab}}$, (2) $\mathcal{L}_{\text{ratio}}$, and (3) $\mathcal{L}_{\text{consis}}$. Since pruning acts by changing which connections remain active, the relevant notion of locality is more naturally defined in mask space than in the dense weight space. Our goal is therefore to identify masks whose predictions remain stable under small structural changes while staying consistent under the target semantic transformation family and after hard-mask deployment.

Percentile-Scaled Initialization. To facilitate an effective mask search, we adopt a percentile-scaled initialization strategy. Instead of naive random initialization, the soft mask values are set proportional to the magnitudes of the pre-trained weights, thereby ensuring that critical connections are prioritized at the beginning of training. Specifically, for the i-th layer, the soft mask C^i is initialized as follows:

$$C^i = \text{clip}\left(\frac{|\theta_i|}{Q(|\theta_i|, \tau)}, 0, 1\right), \tag{5}$$

where $Q(|\theta_i|, \tau)$ is defined as a threshold such that the top $\tau\%$ largest weight magnitudes in the i-th layer exceed it. Consequently, weights within the top $\tau\%$ are assigned a value of 1 after clipping, while the remaining entries are smoothly scaled within the range $[0, 1)$.

After initialization, the soft mask C is treated as a stochastic variable to explore nearby mask configurations and measure local prediction sensitivity under structural perturbations. Specifically, we generate individual mask instances by injecting uniform noise, followed by a clipping operation to maintain the values within the feasible range:

$$C_\xi = \text{clip}(C + \xi, 0, 1), \quad \xi \sim \mathcal{U}(-\mu, \mu), \tag{6}$$

where μ denotes the hyperparameter controlling the magnitude of the injected noise.

Mask-Space Structural Stability and Semantic Consistency. As shown in Equation (3), minimizing the prediction discrepancy Z directly enhances model robustness. For a specific mask instance C, the upper bound of Z can be derived via triangle decomposition as follows:

$$Z_C(x; T) \leq \underbrace{\sqrt{\left\|\mathbf{p}_C(x) - \bar{\mathbf{p}}(x)\right\|_2^2}}_{\text{(A)}} + \underbrace{\left\|\bar{\mathbf{p}}(x) - \bar{\mathbf{p}}(x_T)\right\|_\infty}_{\text{(B)}} + \underbrace{\sqrt{\left\|\bar{\mathbf{p}}(x_T) - \mathbf{p}_C(x_T)\right\|_2^2}}_{\text{(C)}}. \tag{7}$$

where $\bar{\mathbf{p}}(x) = \mathbb{E}_\xi[\mathbf{p}_{C_\xi}(x)]$ denotes the *ensemble mean* over stochastic mask instances. The complete derivation is provided in Appendix A.1. The upper bound formulated in Equation (7) consists of three synergistic components: Terms (A) and (C) quantify the prediction variance induced by stochastic

compression masks, whereas Term (B) characterizes the model's intrinsic sensitivity to semantic transformations.

We decompose the optimization into two synergistic objectives: (1) stability optimization, which explicitly minimizes Terms (A) and (C) via $\mathcal{L}_{\text{stab}}$ to reduce local prediction sensitivity under mask perturbations; and (2) margin-aware semantic optimization, which targets Term (B) via $\mathcal{L}_{\text{ratio}}$ to enforce semantic consistency.

(1) Stability Loss. Considering two independent stochastic mask instances C_m and C_n generated via the stochastic noise injection process (Equation (6)), the stability loss is formulated as follows:

$$\begin{aligned}\mathcal{L}_{\text{stab}} &\triangleq \mathbb{E}_x \, \mathbb{E}_{C_m, C_n} \left[\|\mathbf{p}_{C_m}(x) - \mathbf{p}_{C_n}(x)\|_2^2 \right] \\ &= 2 \, \mathbb{E}_x \, \mathbb{E}_{\xi} \left[\|\mathbf{p}_{C_\xi}(x) - \bar{\mathbf{p}}(x)\|_2^2 \right] \end{aligned} \tag{8}$$

The full proof is provided in Appendix A.3. Minimizing $\mathcal{L}_{\text{stab}}$ effectively reduces the prediction variance induced by stochastic compression masks, i.e., Terms (A) and (C), which helps tighten the theoretical upper bound on the prediction discrepancy Z. In essence, $\mathcal{L}_{\text{stab}}$ is a regularization term that penalizes inconsistent predictions across nearby subnetwork topologies, thereby encouraging mask configurations with lower local prediction sensitivity.

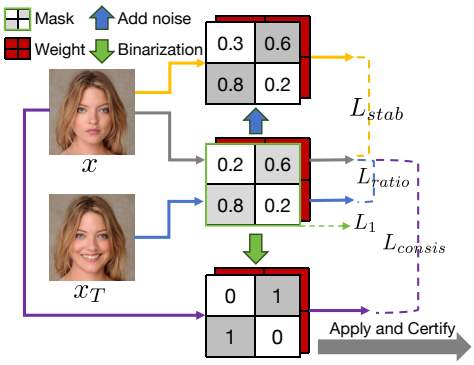


Figure 1: An illustration of the C-SSO framework, where pruning masks are optimized for stability under mask perturbations ($\mathcal{L}_{\text{stab}}$), robustness against semantic input variations ($\mathcal{L}_{\text{ratio}}$), soft-hard mask consistency ($\mathcal{L}_{\text{consis}}$), and L_1 regularization.

(2) Ratio Loss. While $\mathcal{L}_{\text{stab}}$ controls the compression-induced variance terms, it does not by itself constrain the semantic sensitivity captured by Term (B). We therefore introduce ratio loss $\mathcal{L}_{\text{ratio}}$, which penalizes semantic prediction discrepancies $Z(x;T)$ relative to the classification margin $d(x)$. Building on Equation (3), we define the robustness ratio as:

$$r(x;T) \triangleq \frac{Z(x;T)}{d(x) + \epsilon}, \tag{9}$$

where $\epsilon > 0$ is a small constant introduced to prevent division by zero.

To satisfy the certification condition, it is essential to penalize the robustness ratio whenever $r(x;T) > 1$. Accordingly, we introduce a safety threshold $\eta \in (0, 1]$ and employ the $\mathrm{softplus}$ function to implement a smooth penalty when the ratio exceeds this threshold:

$$\mathcal{L}_{\text{ratio}} \triangleq \mathbb{E}_{x, C_m, C_s} \Big[\mathrm{softplus} \left(r(x;T) - \eta \right) \Big]. \tag{10}$$

In practice, to prevent overfitting to a specific subnetwork, the discrepancy $Z(x;T)$ within $r(x;T)$ is computed using two independently sampled stochastic masks C_m and C_s.

From Continuous to Binary. While continuous soft masks facilitate differentiable optimization, practical deployment necessitates discrete binary masks. To mitigate the performance degradation typically induced by binarization, we enforce behavioral alignment between the soft and hard masks during the training process.

(3) Consistency Loss. Specifically, we minimize the divergence between the predictions generated by the soft mask C_m and those from its binarized counterpart $\hat{m}$. This alignment is operationalized through the Kullback–Leibler (KL) divergence loss:

$$\mathcal{L}_{\text{consis}} \triangleq \mathbb{E}_{x, C_m} \Big[\mathrm{KL} \big(\mathbf{p}_{C_m}(x) \| \mathbf{p}_{\hat{m}}(x) \big) \Big]. \tag{11}$$

To optimize this objective, we employ the Straight-Through Estimator (STE) (Bengio et al., 2013) to bypass the non-differentiable binarization step, enabling gradients to directly update the underlying soft mask. This scheme ensures that the robustness properties acquired in the continuous optimization space are effectively preserved within the discrete model used for inference.

Putting it All Together. In addition to the objectives defined above, we incorporate an ℓ_1 regularization term on the soft mask C to induce sparsity, denoted as $\mathcal{L}_1 \triangleq \|C\|_1$. The overall objective function is formulated as:

$$\mathcal{L} = \lambda_{\text{stab}}\,\mathcal{L}_{\text{stab}} + \lambda_{\text{ratio}}\,\mathcal{L}_{\text{ratio}} + \lambda_{\text{consis}}\mathcal{L}_{\text{consis}} + \lambda_1\mathcal{L}_1. \tag{12}$$

The training pipeline of C-SSO, summarized in Algorithm 1 and detailed in Appendix A.4, proceeds in three stages. First, we pre-train the full-precision model using standard cross-entropy loss $\mathcal{L}_{\text{CE}}$ to establish discriminative semantic features. Next, we perform a robust mask search. In this stage, we freeze the backbone parameters and optimize only the soft masks using $\mathcal{L}$. Finally, we binarize the resulting masks and fine-tune the compressed network to recover accuracy.

4 Experiments Setup

Datasets. We evaluate our approach on CelebA-HQ (Lee et al., 2020), Flowers-102 (Nilsback & Zisserman, 2008), CIFAR-10-C (Krizhevsky et al., 2009; Hendrycks & Dietterich, 2019), and Tiny-ImageNet (Le et al., 2015). For CelebA-HQ, we use DIFF-AE (Preechakul et al., 2022) to generate smiling-face variations (Figure 5); for Flowers-102, we use DiffusionCLIP (Kim et al., 2022) to generate illumination-related semantic variations; for CIFAR-10-C and Tiny-ImageNet, we use the fog transformation family. Detailed instantiations of the transformation family T for each dataset are provided in Appendix A.5.

To analyze the effect of the augmentation scale, we define two additive augmentation levels, L1 and L2. For CelebA-HQ and Flowers-102, L1 and L2 add 1,000 and 4,000 transformed samples, respectively. For CIFAR-10-C, L1 and L2 add 10,000 and 20,000 transformed samples. For Tiny-ImageNet, we use 10,000 transformed samples, matching the CIFAR-10-C L1 setting.

Models. We validate our approach using ResNet-18 (He et al., 2016), GoogLeNet (Szegedy et al., 2015), and MobileNet-V2 (Sandler et al., 2018). We further include DeiT-Ti (Touvron et al., 2021) to examine whether the proposed framework extends to transformer architectures. We select these architectures to represent lightweight model families commonly deployed on edge devices, rather than large-scale, over-parameterized networks.

Baselines. To isolate the impact of compression, we first evaluate three unpruned models: Vanilla, trained with standard cross-entropy loss, AdvTrain, which incorporates adversarial loss (Madry et al., 2018), and SAM (Foret et al., 2021), which applies standard weight-space SAM training without pruning. We categorize the pruning baselines into unstructured and structured approaches, with comprehensive descriptions and configurations provided in Appendix A.6. All pruning baselines are retrained from scratch on the same augmented dataset as ours to ensure a fair comparison.

Unstructured Pruning. We select representative methods from three distinct strategies: (1) standard magnitude-based pruning, including Least Magnitude Pruning (LMP) (Han et al., 2015) and its robust variant Van-Adv and Adv-LMP; (2) pixel-level robust pruning, specifically HYDRA (Sehwag et al., 2020) and Flying Bird (FB)+ (Chen et al., 2022); and (3) SAM-style sparse training, including S^2-SAM (Ji et al., 2024) and SAM-LMP.

Structured Pruning. To demonstrate the versatility of the C-SSO framework, we further compare it against representative structured pruning methods, specifically DepGraph (Fang et al., 2023) and HESSO (Chen et al., 2024), to realize actual deployment gains in FLOPs and inference latency.

Metrics. We evaluate performance using two primary metrics: (1) Task Accuracy and (2) Probabilistically Certified Accuracy (PCA), which quantifies the model's robustness against semantic perturbations. All accuracy and PCA values in the paper are reported in percentage units. Following the CC-Cert framework (Pautov et al., 2022), we randomly sample $m = 100$ [1] test instances and set the error bound $\varepsilon = 10^{-3}$ to compute PCA as follows:

$$\text{PCA}(\mathcal{S}, \varepsilon) = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\left[\mathbb{P}_{x_T \sim \mathbb{S}_T(x_i)}\left(\hat{y}(x_T) = y_i\right) \geq 1 - \varepsilon\right].$$

where $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^m$ denotes the evaluation subset with $m = |\mathcal{S}|$, $\mathbf{1}[\cdot]$ is the indicator function, and $\hat{y}(\cdot)$ represents the prediction of the DNN classifier.

[1] The choice $m = 100$ follows common certification practice and is consistent with the VNN-COMP protocol (Brix et al., 2024).

Table 1: Main results on ResNet-18 under different pruning ratios.

Method	Aug.	Pruning Ratio = 30%						Pruning Ratio = 50%						Pruning Ratio = 70%					
		CelebA-HQ		Flowers-102		CIFAR-10-C		CelebA-HQ		Flowers-102		CIFAR-10-C		CelebA-HQ		Flowers-102		CIFAR-10-C	
		Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA
Vanilla	None	83.62	50.0	97.80	58.0	93.04	26.0	83.62	50.0	97.80	58.0	93.04	26.0	83.62	50.0	97.80	58.0	93.04	26.0
	L1	82.96	66.0	97.56	84.0	91.84	80.0	82.96	66.0	97.56	84.0	91.84	80.0	82.96	66.0	97.56	84.0	91.84	80.0
	L2	83.79	71.0	96.70	81.0	93.32	86.0	83.79	71.0	96.70	81.0	93.32	86.0	83.79	71.0	96.70	81.0	93.32	86.0
AdvTrain	None	72.26	2.0	90.33	28.0	73.09	0.0	72.26	2.0	90.33	28.0	73.09	0.0	72.26	2.0	90.33	28.0	73.09	0.0
	L1	72.59	12.0	88.14	32.0	73.37	2.0	72.59	12.0	88.14	32.0	73.37	2.0	72.59	12.0	88.14	32.0	73.37	2.0
	L2	72.02	14.0	86.80	26.0	75.05	1.0	72.02	14.0	86.80	26.0	75.05	1.0	72.02	14.0	86.80	26.0	75.05	1.0
SAM	None	87.90	57.0	98.29	72.0	92.70	53.0	87.90	57.0	98.29	72.0	92.70	53.0	87.90	57.0	98.29	72.0	92.70	53.0
	L1	85.68	62.0	97.80	95.0	92.11	75.0	85.68	62.0	97.80	95.0	92.11	75.0	85.68	62.0	97.80	95.0	92.11	75.0
	L2	85.43	75.0	97.80	92.0	92.35	84.0	85.43	75.0	97.80	92.0	92.35	84.0	85.43	75.0	97.80	92.0	92.35	84.0
LMP	None	85.68	41.0	95.72	69.0	87.22	11.0	83.87	46.0	88.51	71.0	87.53	9.0	62.06	20.0	40.22	11.0	86.65	6.0
	L1	83.29	65.0	96.58	71.0	86.92	9.0	82.47	54.0	89.24	71.0	86.47	51.0	63.70	39.0	70.90	46.0	86.02	48.0
	L2	83.62	55.0	96.45	71.0	87.12	12.0	81.98	66.0	87.90	55.0	86.63	56.0	67.90	46.0	69.93	36.0	84.64	46.0
Van-Adv	None	84.61	11.0	92.30	34.0	82.32	3.0	78.60	15.0	90.46	34.0	82.89	1.0	67.74	1.0	77.75	8.0	82.57	1.0
	L1	82.14	29.0	91.56	49.0	82.94	8.0	75.06	17.0	89.12	42.0	76.04	10.0	59.42	1.0	76.04	10.0	81.86	13.0
	L2	79.09	28.0	88.26	40.0	81.02	10.0	77.12	22.0	88.26	34.0	76.77	12.0	66.83	5.0	76.77	12.0	74.06	3.0
Adv-LMP	None	74.30	5.0	91.20	23.0	66.58	0.0	73.09	5.0	88.51	25.0	75.13	0.0	71.69	4.0	89.36	28.0	72.02	0.0
	L1	71.19	7.0	89.25	25.0	76.87	4.0	71.19	10.0	89.24	27.0	74.17	12.0	67.65	12.0	86.92	25.0	70.90	2.0
	L2	72.59	12.0	89.12	25.0	78.00	7.0	69.22	39.0	85.21	23.0	72.23	3.0	70.95	22.0	85.45	26.0	74.06	3.0
HYDRA	None	52.02	25.0	79.10	56.0	86.41	29.0	53.33	28.0	79.46	51.0	86.02	35.0	51.19	28.0	79.10	56.0	85.47	29.0
	L1	48.56	34.0	76.16	64.0	86.11	57.0	51.44	26.0	77.87	63.0	85.99	53.0	48.97	34.0	76.16	64.0	84.41	50.0
	L2	53.25	44.0	78.61	68.0	86.23	60.0	51.19	48.0	77.87	64.0	86.43	64.0	53.99	40.0	78.61	68.0	84.52	49.0
FB+	None	49.79	9.0	79.09	27.0	63.18	28.0	47.98	6.0	79.82	38.0	65.67	27.0	38.52	1.0	69.32	28.0	74.70	6.0
	L1	49.87	7.0	77.26	32.0	68.40	10.0	49.05	4.0	77.27	32.0	69.31	21.0	39.67	8.0	66.63	34.0	74.44	15.0
	L2	45.26	6.0	72.49	34.0	71.77	34.0	45.84	5.0	74.08	37.0	70.06	30.0	38.35	4.0	61.86	35.0	73.39	12.0
S^2-SAM	None	81.90	41.0	87.16	66.0	80.89	3.0	75.96	45.0	87.65	59.0	80.63	4.0	62.47	33.0	80.92	53.0	78.21	4.0
	L1	81.56	61.0	87.28	78.0	80.21	18.0	78.27	58.0	85.33	72.0	79.88	33.0	65.43	41.0	81.05	65.0	78.26	26.0
	L2	80.41	64.0	88.38	74.0	79.29	29.0	75.06	55.0	87.41	76.0	79.42	21.0	62.55	51.0	79.71	57.0	77.09	28.0
SAM-LMP	None	**87.49**	54.0	93.89	66.0	86.86	12.0	**84.44**	47.0	94.25	71.0	87.87	11.0	64.44	27.0	85.82	56.0	86.34	10.0
	L1	85.10	57.0	90.34	68.0	87.01	10.0	83.29	51.0	94.50	69.0	87.09	16.0	66.26	30.0	85.21	61.0	86.41	9.0
	L2	83.62	60.0	93.40	74.0	87.18	14.0	83.62	58.0	92.91	72.0	87.18	19.0	70.45	44.0	85.82	60.0	86.46	17.0
C-SSO	L1	84.20	**67.0**	**97.19**	83.0	91.76	**81.0**	82.72	**74.0**	**96.45**	88.0	91.15	71.0	76.13	56.0	94.62	79.0	89.42	57.0
	L2	82.88	64.0	96.33	**86.0**	**92.71**	**81.0**	79.09	**74.0**	96.33	**91.0**	**92.70**	**82.0**	**76.46**	**63.0**	**96.70**	**90.0**	**91.77**	**67.0**

Intuitively, a PCA of 80% with $\varepsilon = 10^{-3}$ indicates that for 80% of the test samples, the probability of the prediction flipping under semantic transformations is bounded by 10^{-3}. Detailed descriptions of the certification framework and implementation are provided in Appendix A.7 and Appendix A.8, respectively.

5 Empirical Results

In this section, we conduct extensive experiments to address the following research questions (RQs):

RQ1. Why do prior pruning baselines break under semantic perturbations, and why does C-SSO achieve a better robustness–accuracy trade-off?

RQ2. How does each component of C-SSO's objective contribute to performance?

RQ3. What are the computational overhead and deployment gains of C-SSO?

5.1 Overall Performance Analysis (RQ1)

Table 1 summarizes the results on CelebA-HQ, Flowers-102, and CIFAR-10-C. Across pruning ratios and augmentation scales, C-SSO consistently achieves the highest PCA. The overall pattern is coherent: prior baselines fail since their pruning objectives are misaligned with semantic robustness after structure removal, whereas C-SSO optimizes robustness together with the mask itself.

Magnitude-based methods such as LMP and Adv-LMP rely on weight magnitude as a pruning signal, but that heuristic does not constrain the semantic discrepancy $Z(x;T)$. As a result, their robustness degrades once the semantic shift becomes stronger. For example, on Flowers-102 at pr$= 50\%$, LMP drops from 71% PCA under L1 augmentation to 55% under L2.

Pixel-level robust pruning methods such as Adv-LMP, HYDRA, and FB+ are also misaligned with semantic robustness: they optimize robustness to off-manifold ℓ_p perturbations rather than to semantic transformations. They therefore transfer poorly to CC-Cert, often yielding much lower PCA than C-SSO despite sometimes improving standard adversarial robustness.

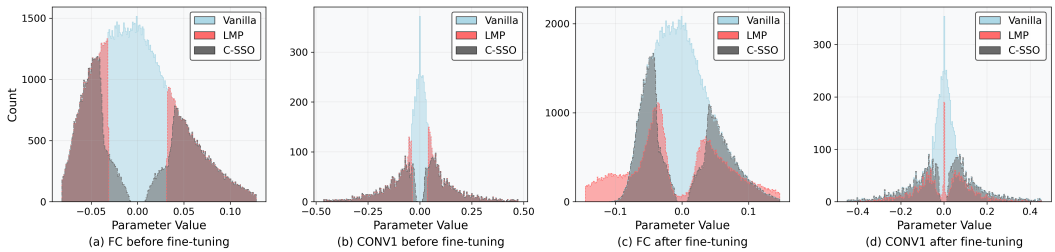


Figure 2: Parameter distributions for the FC layer and `conv1`. Panels (a)–(b) show the distributions immediately after pruning and before fine-tuning; panels (c)–(d) show the corresponding distributions after fine-tuning.

SAM-based baselines exhibit another failure mode: sequential pipelines inherit weight-space flatness from SAM but leave the pruning pattern oblivious to robustness. On CIFAR-10-C without augmentation, the dense SAM model achieves 53% PCA, but SAM-LMP plunges to 11% and 10% at 50% and 70% sparsity. S^2-SAM partially recovers robustness via post-pruning fine-tuning, yet lags behind C-SSO because the topology is frozen before robustness optimization and mask-induced variance remains unconstrained. These failure modes directly explain why C-SSO works better. Instead of pruning first and repairing robustness later, C-SSO searches for subnetworks that are already stable under nearby mask perturbations and semantically consistent under transformed inputs. On Flowers-102 at pr$= 70\%$ with L2 augmentation, C-SSO reaches 90% PCA, exceeding the nearest competitor HYDRA by 22%.

Figure 2 further supports this interpretation. In Figure 2(a)–(b), LMP aggressively removes parameters near zero, whereas C-SSO preserves a moderate band of small-magnitude weights. In Figure 2(c)–(d), after fine-tuning, the distributions of C-SSO shift only mildly, while LMP undergoes a much larger redistribution. This suggests that C-SSO identifies subnetworks that are already better aligned with the final sparse structure and therefore require less post-pruning re-adjustment. Full-layer versions of this comparison are provided in Appendix A.10.

These empirical gains are not confined to the ResNet-18. In Appendix A.9, we report results on additional CNN backbones, namely GoogLeNet and MobileNet-V2 in Table 7, on a transformer backbone DeiT-Ti against Isomorphic pruning (Fang et al., 2024) in Table 8, under structured pruning against DepGraph and HESSO in Table 9, and on the harder Tiny-ImageNet benchmark in Table 10. We provide detailed analyses on transformation transfer in Table 11, the impact of paired semantic image supervision in Table 12, and stability across random seeds in Table 13.

The PCA gains are not expected to be strictly monotonic in the pruning ratio. Pruning introduces a trade-off between regularization and capacity: moderate sparsity can improve robustness, while more aggressive pruning eventually removes critical parameters and shifts the optimal sparsity in a dataset-dependent way (Bartoldson et al., 2020; Jordao & Pedrini, 2021). This directly explains that PCA may increase from 30% to 50%, and in some cases continue improving at 70% if the optimal sparsity has not yet been reached.

5.2 Ablation Study and Variant Analysis (RQ2)

Component-wise Ablation Study. Table 2 details the contribution of each component within the C-SSO objective. The coefficients are reported as $\{\lambda_{\text{consis}}/\lambda_{\text{ratio}}/\lambda_{\text{stab}}/\lambda_1\}$. The removal of any single loss term causes a distinct PCA degradation, confirming that the components are complementary. In particular, $\mathcal{L}_{\text{consis}}$ and $\mathcal{L}_{\text{ratio}}$ are the most critical terms: reducing or removing either causes the largest drops in both accuracy and PCA. $\mathcal{L}_{\text{stab}}$ and $\mathcal{L}_1$ provide essential regularization, as omitting either degrades PCA without collapsing accuracy. The default setting $\{1/1/5/10^{-4}\}$ achieves the best overall trade-off and is used throughout the main experiments.

Initialization sensitivity. Table 3 studies how the soft-mask initialization affects optimization. Full-scaled initialization collapses, while percentile-scaled initialization consistently outperforms random initialization, with $\tau = 30$ giving the best robustness-accuracy trade-off. This suggests that the search benefits from a moderate prior from weight magnitude, but not from over-constraining the mask variables at initialization.

Hyperparameter sensitivity. Table 4 further examines the mask noise magnitude μ and the safety margin η. A moderate mask noise level ($\mu = 0.5$) achieves the highest PCA, indicating that some

Table 2: Objective coefficient analysis on CelebA-HQ (pr= 70%).

Config $\{\lambda_C/\lambda_R/\lambda_S/\lambda_1\}$	Acc	PCA	Drop
{1 / 1 / 5 / 1e-4}	**76.13**	**56.0**	–
{0 / 1 / 5 / 1e-4}	36.05	9.0	↓ 47.0
{0.3 / 1 / 5 / 1e-4}	65.35	41.0	↓ 15.0
{2 / 1 / 5 / 1e-4}	66.50	49.0	↓ 7.0
{1 / 0 / 5 / 1e-4}	1.23	0.0	↓ 56.0
{1 / 0.3 / 5 / 1e-4}	70.21	50.0	↓ 6.0
{1 / 2 / 5 / 1e-4}	73.83	51.0	↓ 5.0
{1 / 1 / 0 / 1e-4}	66.91	36.0	↓ 20.0
{1 / 1 / 2 / 1e-4}	72.76	52.0	↓ 4.0
{1 / 1 / 10 / 1e-4}	69.96	49.0	↓ 7.0
{1 / 1 / 5 / 0}	67.74	39.0	↓ 17.0
{1 / 1 / 5 / 1e-2}	67.74	44.0	↓ 12.0
{1 / 1 / 5 / 1e-5}	72.18	53.0	↓ 3.0

Table 3: Impact of the initialization percentile τ on model performance. Results are evaluated on ResNet-18 (CelebA-HQ, pr= 50%, τ%).

Metric	Random	Percentile-Scaled (τ)				Full-Scaled (τ)
	(Baseline)	40	30	20	10	0
Acc	78.11	80.82	**82.72**	82.55	83.05	0.49
PCA	59.0	63.0	**74.0**	66.0	71.0	0.0

Table 4: Sensitivity analysis of training hyperparameters on ResNet-18 (CelebA-HQ, pr = 50%). Default settings are $\mu = 0.5$, $\eta = 1.0$.

Metric	Noise Magnitude (μ)			Safety Margin (η)		
	0.1	0.5	0.8	0.95	0.98	1.0
Acc	82.55	82.72	82.30	82.06	82.55	82.72
PCA	70.0	**74.0**	66.0	67.0	68.0	**74.0**

stochasticity is needed to regularize the mask search, whereas too much noise destabilizes it. Similarly, $\eta = 1.0$ gives the best trade-off, consistent with the margin-based interpretation of $\mathcal{L}_{\text{ratio}}$.

Training Variant. Appendix A.9 investigates two training variants: parameter-mask joint search and Hard-mask with STE. Joint search further boosts performance in certain settings, confirming compatibility with joint optimization, while C-SSO consistently outperforms Hard-STE on CelebA-HQ and Flowers-102.

5.3 Overhead and Deployment Gain (RQ3)

Although C-SSO adopts a three-stage pipeline, it does not increase the overall training budget: the 50/100/50 epoch schedule matches the same total of 200 epochs used by single-stage baselines such as FB+. The additional cost is concentrated in Stage 2, where stochastic mask sampling and multiple forward passes are required; nevertheless, Table 5 shows that its per-epoch overhead remains substantially lower than HYDRA and FB+. At inference, C-SSO introduces no additional mechanism.

Table 5: Training overhead on CelebA-HQ (L1, pr=50%, 3 runs).

Method	Time / Epoch (s)
Vanilla (Stage 1&3)	11.01 ± 0.02
HYDRA (Stage 2)	122.26 ± 0.56
FB+ (Single Stage)	61.37 ± 0.83
C-SSO (Stage 2)	22.51 ± 0.09
C-SSO (Stage 3)	11.62 ± 0.27

Table 6 demonstrates the deployment efficiency of structured C-SSO: it nearly halves FLOPs from 1.82G to 0.98G and parameters from 11.3M to 5.8M while reducing inference time from 4.02s to 3.24s, confirming practical deployment gains alongside certified robustness, with corresponding robustness gains reported in Appendix A.9.

Table 6: Deployment efficiency under structured pruning on CelebA-HQ (L1, pr=50%).

Method	FLOPs	Params	Inference Time
C-SSO$_{\text{uns}}$	1.82G	11.3M	4.02s
C-SSO$_{\text{str}}$	0.98G	5.8M	3.24s

6 Conclusion

This work presents C-SSO, a unified training framework for obtaining compact and certified robust DNNs against semantic perturbations. By coupling mask-space structural stability with semantic consistency in a single composite objective, C-SSO provides a principled route to compact models with certified robustness to semantic perturbations. Extensive evaluations across ResNet-18, GoogLeNet, MobileNet-V2, and DeiT-Ti, as well as across CelebA-HQ, Flowers-102, CIFAR-10-C, and Tiny-ImageNet, demonstrate that C-SSO consistently achieves superior certified robustness, with improvements up to 42% over competing baselines, while maintaining task accuracy. Future work will explore C-SSO's interplay with multi-attribute semantic variations using more advanced generative models, and further reduce its computational overhead for optimizing certified robustness.

AI USE STATEMENT

In this work, we used generative AI tools exclusively for grammatical polishing, phrasing refinement, and LaTeX formatting assistance. We have not used generative AI tools for developing models, proposing hypotheses, and the rest of the required disclosure tasks are not applicable to this work. We have thoroughly reviewed and verified all AI-assisted text, and we take full responsibility for the entirety of the contents, claims, and artifacts presented in this work.

ETHICS STATEMENT

This work complies with the ICLR Code of Ethics. Our research focuses on improving the certified robustness and deployment efficiency of deep neural networks against semantic perturbations, thereby enhancing the reliability and safety of vision systems in real-world edge applications. All datasets used in our experiments, including CelebA-HQ, Flowers-102, CIFAR-10, and Tiny-ImageNet, are publicly available and widely adopted within the research community, and we strictly adhere to their intended usage and licenses. Our study does not involve human subject experiments, clinical trials, or the collection of private personal data. While robust visual recognition models could potentially be integrated into automated surveillance systems, our primary goal is to provide defensive certified guarantees that mitigate unpredictable failures under environmental variations.

REPRODUCIBILITY STATEMENT

To ensure complete reproducibility of our findings, we have detailed all theoretical formulations, architectural specifications, and implementation details throughout the paper. Specifically, the certified robustness guarantee and its mathematical derivation are presented in Section 3 and Section 4. The three-stage training pipeline, loss objectives, and optimization procedures are formulated in Section 4. Comprehensive experimental setups, including dataset preprocessing, baseline configurations, training hyperparameter selections, and ablation protocols, are specified in Section 5 and expanded in Appendix A.9. To facilitate future research and reproduction, our complete codebase, including training scripts, and evaluation pipelines, will be made publicly available upon paper acceptance.

REFERENCES

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International conference on machine learning*, pp. 639–668. PMLR, 2022.

Anna Bair, Hongxu Yin, Maying Shen, Pavlo Molchanov, and Jose M. Alvarez. Adaptive sharpness-aware pruning for robust sparse networks. In *The Twelfth International Conference on Learning Representations*, 2024.

Brian Bartoldson, Ari Morcos, Adrian Barbu, and Gordon Erlebacher. The generalization-stability tradeoff in neural network pruning. *Advances in neural information processing systems*, 33: 20852–20864, 2020.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Stéphane Boucheron, Gábor Lugosi, and Olivier Bousquet. Concentration inequalities. In Olivier Bousquet, Ulrike von Luxburg, and Gunnar Rätsch (eds.), *Advanced Lectures on Machine Learning, ML Summer Schools 2003, Canberra, Australia, February 2-14, 2003, Tübingen, Germany, August 4-16, 2003, Revised Lectures*, volume 3176 of *Lecture Notes in Computer Science*, pp. 208–240. Springer, 2003.

Christopher Brix, Stanley Bak, Taylor T Johnson, and Haoze Wu. The fifth international verification of neural networks competition (vnn-comp 2024): Summary and results. *arXiv preprint arXiv:2412.19985*, 2024.

Tianlong Chen, Zhenyu Zhang, Pengjun Wang, Santosh Balachandra, Haoyu Ma, Zehao Wang, and Zhangyang Wang. Sparsity winning twice: Better robust generalization from more efficient training. In *International Conference on Learning Representations*, 2022.

Tianyi Chen, Bo Ji, Tianyu Ding, Biyi Fang, Guanyi Wang, Zhihui Zhu, Luming Liang, Yixin Shi, Sheng Yi, and Xiao Tu. Only train once: A one-shot neural network training and pruning framework. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Tianyi Chen, Luming Liang, Tianyu DING, Zhihui Zhu, and Ilya Zharkov. OTOv2: Automatic, generic, user-friendly. In *The Eleventh International Conference on Learning Representations*, 2023.

Tianyi Chen, Xiaoyi Qu, David Aponte, Colby Banbury, Jongwoo Ko, Tianyu Ding, Yong Ma, Vladimir Lyapunov, Ilya Zharkov, and Luming Liang. Hesso: Towards automatic efficient and user friendly any neural network training and pruning. *arXiv preprint arXiv:2409.09085*, 2024.

Guneet S. Dhillon, Kamyar Azizzadenesheli, Zachary C. Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

James Diffenderfer, Brian R. Bartoldson, Shreya Chaganti, Jize Zhang, and Bhavya Kailkhura. A winning hand: Compressing deep networks can improve out-of-distribution robustness. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 664–676, 2021.

Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16091–16101, 2023.

Gongfan Fang, Xinyin Ma, Michael Bi Mi, and Xinchao Wang. Isomorphic pruning for vision models. In *European Conference on Computer Vision*, pp. 232–250. Springer, 2024.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Shupeng Gui, Haotao Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 1283–1294, 2019.

Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pp. 1379–1387, 2016.

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

Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

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

Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pp. 2234–2240. ijcai.org, 2018.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Weiyu Huang, Yuezhou Hu, Guohao Jian, Jun Zhu, and Jianfei Chen. Pruning large language models with semi-structural adaptive sparse training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24167–24175, 2025.

Jie Ji, Gen Li, Jingjing Fu, Fatemeh Afghah, Linke Guo, Xiaoyong Yuan, and Xiaolong Ma. A single-step, sharpness-aware minimization is all you need to achieve efficient and accurate sparse training. *Advances in Neural Information Processing Systems*, 37:44269–44290, 2024.

Artur Jordao and Hélio Pedrini. On the effect of pruning on adversarial robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1–11, 2021.

Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2426–2435, 2022.

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

Yann Le, Xuan Yang, et al. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 5548–5557. Computer Vision Foundation / IEEE, 2020.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. Snip: single-shot network pruning based on connection sensitivity. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

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

Linyi Li, Tao Xie, and Bo Li. SoK: Certified robustness for deep neural networks. In *44th IEEE Symposium on Security and Privacy, SP 2023, San Francisco, CA, USA, May 21-25, 2023*, pp. 1289–1310. IEEE, 2023a.

Tao Li, Pan Zhou, Zhengbao He, Xinwen Cheng, and Xiaolin Huang. Friendly sharpness-aware minimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5631–5640, 2024.

Zhangheng Li, Tianlong Chen, Linyi Li, Bo Li, and Zhangyang Wang. Can pruning improve certified robustness of neural networks? *Trans. Mach. Learn. Res.*, 2023b.

Liyang Liu, Shilong Zhang, Zhanghui Kuang, Aojun Zhou, Jing-Hao Xue, Xinjiang Wang, Yimin Chen, Wenming Yang, Qingmin Liao, and Wayne Zhang. Group Fisher Pruning for Practical Network Compression. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, 2021.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12360–12370, 2022a.

Yong Liu, Siqi Mai, Minhao Cheng, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Random sharpness-aware minimization. *Advances in neural information processing systems*, 35:24543–24556, 2022b.

Zhuang Liu, Jianguo Li, Zhiqiang Shen, Gao Huang, Shoumeng Yan, and Changshui Zhang. Learning efficient convolutional networks through network slimming. In *Proceedings of the IEEE international conference on computer vision*, pp. 2736–2744, 2017.

Haocheng Luo, Tuan Truong, Tung Pham, Mehrtash Harandi, Dinh Phung, and Trung Le. Explicit eigenvalue regularization improves sharpness-aware minimization. *Advances in Neural Information Processing Systems*, 37:4424–4453, 2024.

Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE international conference on computer vision*, pp. 5058–5066, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. *Advances in Neural Information Processing Systems*, 35:30950–30962, 2022.

Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Tianshuo Xu, Xiaoshuai Sun, Tongliang Liu, Rongrong Ji, and Dacheng Tao. Systematic investigation of sparse perturbed sharpness-aware minimization optimizer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Matthew Mirman, Alexander Hägele, Pavol Bielik, Timon Gehr, and Martin T. Vechev. Robustness certification with generative models. In *PLDI '21: 42nd ACM SIGPLAN International Conference on Programming Language Design and Implementation, Virtual Event, Canada, June 20-25, 2021*, pp. 1141–1154. ACM, 2021.

Dongbin Na, Sangwoo Ji, and Jong Kim. Unrestricted black-box adversarial attack using GAN with limited queries. In *Computer Vision - ECCV 2022 Workshops - Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part I*, volume 13801 of *Lecture Notes in Computer Science*, pp. 467–482. Springer, 2022.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pp. 722–729. IEEE, 2008.

REAC Paley and Antoni Zygmund. On some series of functions (1). In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 26, pp. 337–357. Cambridge University Press, 1930.

Mikhail Pautov, Nurislam Tursynbek, Marina Munkhoeva, Nikita Muravev, Aleksandr Petiushko, and Ivan Oseledets. CC-Cert: A probabilistic approach to certify general robustness of neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7975–7983, 2022.

Juan C. Pérez, Motasem Alfarra, Ali K. Thabet, Pablo Arbeláez, and Bernard Ghanem. Towards characterizing the semantic robustness of face recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023 - Workshops, Vancouver, BC, Canada, June 17-24, 2023*, pp. 315–325. IEEE, 2023.

Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10609–10619. IEEE, 2022.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Towards compact and robust deep neural networks. *arXiv*, abs/1906.06110, 2019.

Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. HYDRA: pruning adversarially robust neural networks. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Ilia Shumailov, Yiren Zhao, Robert Mullins, and Ross Anderson. To compress or not to compress: Understanding the interactions between adversarial attacks and neural network compression. *Proceedings of Machine Learning and Systems*, 1:230–240, 2019.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357. PMLR, 2021.

Chaoqi Wang, Guodong Zhang, and Roger B. Grosse. Picking winning tickets before training by preserving gradient flow. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How sharpness-aware minimization minimizes sharpness? In *The Eleventh International Conference on Learning Representations*, 2022.

Mingzhe Yang, Sihao Lin, Changlin Li, and Xiaojun Chang. Let LLM tell what to prune and how much to prune. In *Forty-second International Conference on Machine Learning*, 2025.

Shaokai Ye, Xue Lin, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, and Yanzhi Wang. Adversarial robustness vs. model compression, or both? In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pp. 111–120. IEEE, 2019.

Lu Yin, Gen Li, Meng Fang, Li Shen, Tianjin Huang, Zhangyang Wang, Vlado Menkovski, Xiaolong Ma, Mykola Pechenizkiy, and Shiwei Liu. Dynamic sparsity is channel-level sparsity learner. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16*, 2023.

Zhonghui You, Kun Yan, Jinmian Ye, Meng Ma, and Ping Wang. Gate decorator: Global filter pruning method for accelerating deep convolutional neural networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2130–2141, 2019.

Yuanyuan Yuan, Shuai Wang, and Zhendong Su. Precise and generalized robustness certification for neural networks. In *32nd USENIX Security Symposium, USENIX Security 2023, Anaheim, CA, USA, August 9-11, 2023*, pp. 4769–4786. USENIX Association, 2023.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Qi Zhao and Christian Wressnegger. Holistic adversarially robust pruning. In *The Eleventh International Conference on Learning Representations*, 2023.

A Appendix A

A.1 Proof for $Z(x;T)$

Proof. To derive a manageable upper bound for the prediction discrepancy $Z(x;T)$ under a specific mask instance C, we utilize the triangle inequality and the relationship between vector norms.

Let $\mathbf{p}_C(x)$ be the probability vector generated by the model with mask C, and let $\bar{\mathbf{p}}(x) = \mathbb{E}_\xi[\mathbf{p}_{C_\xi}(x)]$ denote the ensemble mean over all possible stochastic mask instances. We decompose the total discrepancy by adding and subtracting the ensemble means for both the original input x and the transformed input x_T:

$$\begin{aligned} Z(x;T) &= \left\|\mathbf{p}_C(x) - \mathbf{p}_C(x_T)\right\|_\infty \\ &= \left\|\mathbf{p}_C(x) - \bar{\mathbf{p}}(x) + \bar{\mathbf{p}}(x) - \bar{\mathbf{p}}(x_T) + \bar{\mathbf{p}}(x_T) - \mathbf{p}_C(x_T)\right\|_\infty \\ &\le \left\|\mathbf{p}_C(x) - \bar{\mathbf{p}}(x)\right\|_\infty + \left\|\bar{\mathbf{p}}(x) - \bar{\mathbf{p}}(x_T)\right\|_\infty + \left\|\bar{\mathbf{p}}(x_T) - \mathbf{p}_C(x_T)\right\|_\infty \\ &\le \underbrace{\sqrt{\left\|\mathbf{p}_C(x) - \bar{\mathbf{p}}(x)\right\|_2^2}}_{\text{(A)}} + \underbrace{\left\|\bar{\mathbf{p}}(x) - \bar{\mathbf{p}}(x_T)\right\|_\infty}_{\text{(B)}} + \underbrace{\sqrt{\left\|\bar{\mathbf{p}}(x_T) - \mathbf{p}_C(x_T)\right\|_2^2}}_{\text{(C)}} \end{aligned}$$

□

A.2 Proof for Label Invariance

Let y be the true label of input x. We assume the classifier correctly predicts the input x, i.e., $c = \arg\max \mathbf{p}(x) = y$. We denote $c = \arg\max\ \mathbf{p}$ and $\tilde{c} = \arg\max\ \mathbf{p}_T$. Given the assumption $c = y$, if $\|\mathbf{p}(x) - \mathbf{p}(x_T)\|_\infty < d = \frac{p_1 - p_2}{2}$ holds, then $\tilde{c} = c = y$.

Proof. Assume, for contradiction, that $\tilde{c} \neq c$. In this case one has

$$p_{T\tilde{c}} > p_{Tc}, \qquad p_c > p_{\tilde{c}}.$$

Moreover, the condition $\|\mathbf{p} - \mathbf{p}_T\|_\infty < d$ implies that

$$|p_k - p_{Tk}| < d \quad \text{for all } k \in \{1, \ldots, K\}.$$

In particular, for indices c and $\tilde{c}$ it follows that

$$p_{Tc} > p_c - d, \qquad p_{T\tilde{c}} < p_{\tilde{c}} + d.$$

Consequently,

$$p_{Tc} - p_{T\tilde{c}} \ > \ (p_c - d) - (p_{\tilde{c}} + d) = p_c - p_{\tilde{c}} - 2d.$$

On the other hand, the assumption $p_{T\tilde{c}} > p_{Tc}$ requires $p_{Tc} - p_{T\tilde{c}} < 0$. This contradicts the fact that the right-hand side above is non-negative whenever $p_c - p_{\tilde{c}} \ge 2d$. Hence, the assumption $\tilde{c} \neq c$ is invalid, and we conclude that $\tilde{c} = c$. □

A.3 Variance Identity

Proof. Let C_m and C_n be two independent stochastic mask instances derived from C via the noise injection process defined in Equation (6), i.e., $C_\xi = \text{clip}(C + \xi, 0, 1)$ with $\xi \sim \mathcal{U}(-\mu, \mu)$. For a fixed

input x, let $\mathbf{p}_{C_\xi} = f(x; \mathcal{T}(C_\xi) \odot \theta)$, and $\bar{\mathbf{p}} = \mathbb{E}_\xi[\mathbf{p}_{C_\xi}]$ denote the ensemble mean. Then:

$$\begin{aligned}
\mathbb{E}_{C_m,C_n}\left[\|\mathbf{p}_{C_m} - \mathbf{p}_{C_n}\|_2^2\right] &= \mathbb{E}_{C_m,C_n}\left[\langle \mathbf{p}_{C_m} - \mathbf{p}_{C_n}, \mathbf{p}_{C_m} - \mathbf{p}_{C_n}\rangle\right] \\
&= \mathbb{E}_{C_m}\|\mathbf{p}_{C_m}\|_2^2 + \mathbb{E}_{C_n}\|\mathbf{p}_{C_n}\|_2^2 - 2\,\mathbb{E}_{C_m,C_n}[\langle \mathbf{p}_{C_m}, \mathbf{p}_{C_n}\rangle] \\
&\quad \text{(Since } C_m \text{ and } C_n \text{ are independent, } \mathbb{E}[XY] = \mathbb{E}[X]\mathbb{E}[Y]\text{)} \\
&= 2\,\mathbb{E}_\xi\|\mathbf{p}_{C_\xi}\|_2^2 - 2\,\langle \mathbb{E}_{C_m}[\mathbf{p}_{C_m}],\ \mathbb{E}_{C_n}[\mathbf{p}_{C_n}]\rangle \\
&= 2\,\mathbb{E}_\xi\|\mathbf{p}_{C_\xi}\|_2^2 - 2\,\langle \bar{\mathbf{p}}, \bar{\mathbf{p}}\rangle \\
&= 2\,\mathbb{E}_\xi\|\mathbf{p}_{C_\xi}\|_2^2 - 2\,\|\bar{\mathbf{p}}\|_2^2 \\
&= 2\left(\mathbb{E}_\xi\|\mathbf{p}_{C_\xi}\|_2^2 - 2\|\bar{\mathbf{p}}\|_2^2 + \|\bar{\mathbf{p}}\|_2^2\right) \\
&= 2\left(\mathbb{E}_\xi\|\mathbf{p}_{C_\xi}\|_2^2 - 2\langle \mathbb{E}_\xi[\mathbf{p}_{C_\xi}], \bar{\mathbf{p}}\rangle + \|\bar{\mathbf{p}}\|_2^2\right) \\
&= 2\,\mathbb{E}_\xi\left[\|\mathbf{p}_{C_\xi}\|_2^2 - 2\langle \mathbf{p}_{C_\xi}, \bar{\mathbf{p}}\rangle + \|\bar{\mathbf{p}}\|_2^2\right] \\
&= 2\,\mathbb{E}_\xi\left[\|\mathbf{p}_{C_\xi} - \bar{\mathbf{p}}\|_2^2\right].
\end{aligned}$$

□

A.4 Training Algorithm

Algorithm 1 summarizes the full three-stage training pipeline used throughout the paper. Stage 1 learns a dense semantic backbone, Stage 2 freezes the backbone and searches for a robust mask under stochastic structural perturbations, and Stage 3 binarizes the selected mask and fine-tunes the resulting compressed network.

A.5 Detailed Description of Dataset and Augmentations

This section details the source datasets and the corresponding transformation families used to construct the augmented training data.

Dataset Introduction. (1) CelebA-HQ (Na et al., 2022): A facial identity dataset comprising 307 identities (4,263 training, 1,215 testing images). We employ DIFF-AE (Preechakul et al., 2022) to introduce semantic variations, specifically transforming faces to have “smiling" expressions. (2) Oxford Flowers-102 (Nilsback & Zisserman, 2008): A dataset of 102 flower categories (6,552 training, 818 testing images). We utilize DiffusionCLIP (Kim et al., 2022) to generate mutated images based on the prompt “A flower in focus, with the sun rising behind it, casting a warm golden glow." (3) CIFAR-10 (Krizhevsky et al., 2009): To evaluate the generalization of C-SSO beyond generative models, we adopt the “fog" corruption (Severity Level 2) from the CIFAR-10-C benchmark (Hendrycks & Dietterich, 2019). (4) Tiny-ImageNet (Le et al., 2015): We use Tiny-ImageNet with the same fog transformation family as CIFAR-10-C to evaluate the method on a more challenging dataset.

Augmentation Settings. For the diffusion-based datasets, we set the perturbation magnitude $\|\delta\| = 1$ to produce the largest semantic shift. To analyze the impact of data scale, we define two additive augmentation levels, meaning that transformed samples are added to the original training set. For CelebA-HQ and Flowers-102, L1 and L2 add 1,000 and 4,000 mutated samples, respectively. For CIFAR-10-C, L1 and L2 add 10,000 and 20,000 fog-corrupted samples. For Tiny-ImageNet, we use 10,000 fog-corrupted samples, matching the CIFAR-10-C L1 setting.

To evaluate the scalability of C-SSO against corruption-style perturbations, we incorporate the “fog" transformation from the CIFAR-10-C benchmark (Hendrycks & Dietterich, 2019). Unlike the diffusion-based semantic mutations used for CelebA-HQ and Flowers-102, the transformation space for CIFAR-10-C and Tiny-ImageNet is constructed via pixel-level linear interpolation. Specifically, we define the transformation function T as follows:

$$T(x;\delta) = \text{Clip}\Big(\text{LinearInterp}\big(x, \text{Fog}(x, L_2), \delta\big), 0, 1\Big), \quad \delta \in [0,1] \tag{13}$$

where $Fog(\cdot)$ denotes the foggy augmentation algorithm at severity level 2. The parameter δ controls the interpolation intensity, transitioning from the original image ($\delta = 0$) to the fully corrupted version ($\delta = 1$). During the certification process, we sample $\delta_i \sim \mathcal{U}(0,1)$ to obtain the set of transformed inputs for estimating the upper bound of the prediction discrepancy Z.

Algorithm 1 Training Procedure of C-SSO

Require: DNN θ, dataset $\mathcal{S}$, transformation $T(\cdot, \delta)$, pruning ratio pr, broadcasting operator $\mathcal{T}$, safety threshold η, small constant ϵ, coefficients $\lambda_{\{\text{stab,consis,ratio,1}\}}$, learning rate α.

Construct augmented dataset $\mathcal{S}' = \mathcal{S} \cup \{(T(x, \delta), y) \mid (x, y) \in \mathcal{S}\}$, paired dataset $\mathcal{B}_p = \{(x, T(x, \delta))\}$ derived from $\mathcal{S}'$

Stage 1: Pre-training

Train full-precision model on augmented dataset: $\theta_{\text{pre}} \leftarrow \arg\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{S}'}[\mathcal{L}_{\text{CE}}(\theta, x, y)]$

Stage 2: Robust Mask Search

Initialize soft mask C for each layer via Equation (5)

while not converged **do**

  Sample a batch: $(x, x_T) \leftarrow \mathcal{B}_p$

  // 1. Stability Loss

  Generate stochastic masks C_m, C_n, C_s by injecting noise into C via Equation (6)

  Probabilistic outputs: $\mathbf{p}_{C_m} \leftarrow f(x; \mathcal{T}(C_m) \odot \theta_{\text{pre}})$, $\mathbf{p}_{C_n} \leftarrow f(x; \mathcal{T}(C_n) \odot \theta_{\text{pre}})$

  $\mathcal{L}_{\text{stab}} \leftarrow \|\mathbf{p}_{C_m} - \mathbf{p}_{C_n}\|_2^2$

  // 2. Soft-Hard Consistency

  Compute binarization threshold τ_k^i to satisfy pr

  Binarize masks: $\hat{m}^i \leftarrow \mathbf{1}(C^i \geq \tau_k^i)$

  STE: $\hat{m} \leftarrow (\hat{m} - C).detach() + C$

  Hard output: $\mathbf{p}_{\hat{m}} \leftarrow f(x; \mathcal{T}(\hat{m}) \odot \theta_{\text{pre}})$

  $\mathcal{L}_{\text{consis}} \leftarrow \text{KL}(\mathbf{p}_{C_m} \| \mathbf{p}_{\hat{m}})$

  // 3. Ratio Loss

  Semantic output: $\mathbf{p}_{C_s} \leftarrow f(x_T; \mathcal{T}(C_s) \odot \theta_{\text{pre}})$

  Calculate discrepancy $Z \leftarrow \|\mathbf{p}_{C_m}(x) - \mathbf{p}_{C_s}(x_T)\|_\infty$

  Calculate margin $d \leftarrow \frac{p_1 - p_2}{2}$, where p_1, p_2 are top 2 entries of $\mathbf{p}_{C_m}$

  $\mathcal{L}_{\text{ratio}} \leftarrow \text{Softplus}(\frac{Z}{d+\epsilon} - \eta)$

  // 4. Update Mask

  $\mathcal{L} \leftarrow \lambda_{\text{stab}}\mathcal{L}_{\text{stab}} + \lambda_{\text{consis}}\mathcal{L}_{\text{consis}} + \lambda_{\text{ratio}}\mathcal{L}_{\text{ratio}} + \lambda_1\mathcal{L}_1$

  Update mask $C \leftarrow C - \alpha\nabla_C\mathcal{L}$

end while

Stage 3: Post-training

Compute binarization threshold τ_k^i to satisfy pr

Binarize masks: $\hat{m}^i \leftarrow \mathbf{1}(C^i \geq \tau_k^i)$ to meet pr

Fine-tune model parameters θ from θ_{pre}:
$\theta_f \leftarrow \arg\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{S}'}[\mathcal{L}_{\text{CE}}(\mathcal{T}(\hat{m}) \odot \theta, x, y)]$

Ensure: Final compressed model $\mathcal{T}(\hat{m}) \odot \theta_f$

A.6 Detailed Description of Baselines

This section provides the technical definitions and strengthens the contrast between these methods and our proposed C-SSO framework.

Robustness and Unstructured Pruning Baselines. We compare C-SSO against standard and robust training strategies, as well as methods that integrate pruning with adversarial defense:

- **Vanilla**: The standard baseline where the model is trained using only cross-entropy loss without any pruning or robust training.
- **AdvTrain**: A standard defense method that incorporates adversarial training (Madry et al., 2018) to optimize robustness against pixel-level L_p-norm perturbations, ignoring semantic-level variations.
- **SAM** (Foret et al., 2021): A dense weight-space SAM baseline that perturbs model parameters during training to improve robustness without performing pruning.
- **Least Magnitude Pruning (LMP)** (Han et al., 2015): A classic magnitude-based pruning approach that removes weights with the smallest absolute values, without explicitly accounting for structural stability or semantic robustness.

- **Van-Adv**: A sequential baseline that first obtains a pruned model via LMP and then applies adversarial fine-tuning, testing whether pixel-level robustness can recover after pruning.
- **Adv-LMP**: A sequential combination that applies LMP to an adversarially trained model, representing a naive attempt to combine compactness and robustness.
- **HYDRA** (Sehwag et al., 2020): A representative robust pruning framework that optimizes the pruning mask using the TRADES (Zhang et al., 2019) adversarial loss, yet remains limited to pixel-level robustness objectives.
- **FB+** (Chen et al., 2022): A dynamic sparsity method that initializes a random sparse subnetwork and jointly optimizes weights and topology, focusing on efficient training rather than certified stability.
- **S^2-SAM** (Ji et al., 2024): A SAM-style sparse fine-tuning approach designed for pruned models. While it performs weight-space SAM updates after one-shot pruning to improve generalization and sparse robustness, it lacks explicit constraints for semantic robustness or mask-induced prediction variance.
- **SAM-LMP**: A sequential baseline that applies LMP to a dense SAM-trained model, testing whether weight-space SAM robustness survives post-hoc pruning.

For the adversarial-training-based baselines used in our experiments, we employ an L_2-norm PGD attack, setting the perturbation budget to $\epsilon = 0.5$ for CelebA-HQ and CIFAR-10-C, and $\epsilon = 2.0$ for Flowers-102. We do not include adversarial baselines on Tiny-ImageNet.

Structured Pruning Baselines. To evaluate the versatility of C-SSO in structured pruning settings, we compare with advanced structured pruning methods:

- **DepGraph** (Fang et al., 2023): A method that constructs a dependency graph to group coupled parameters, ensuring structural consistency but without robustness guarantees.
- **HESSO** (Chen et al., 2024): A most recent structured pruning technique from the "Only-Train-Once" (OTO) (Chen et al., 2021; 2023) family, which focuses purely on training efficiency and structural redundancy removal.

Transformer Baseline.

- **Isomorphic Pruning** (Fang et al., 2024): A structured pruning method that groups parameters into isomorphic sub-structures based on computational topology and performs importance ranking within each group to ensure comparability across heterogeneous components. In DeiT-Ti experiments, it is applied to the FFN layers as a representative transformer pruning baseline to evaluate whether C-SSO generalizes beyond CNNs.

A.7 Certification Framework: CC-Cert

According to Equation (2) and Equation (3), the label invariance condition is $Z < d$. Hence, our goal is to bound $P(Z \geq d)$ from the above. Using Markov's inequality:

$$\mathbb{P}(Z \geq t) \leq \frac{\mathbb{E}(Z)}{t}, \tag{14}$$

where Z is a non-negative random variable and $t \in \mathbb{R}^+$. This provides an initial bound, but CC-Cert refines this using the Chernoff-Cramer inequality (Boucheron et al., 2003):

$$\mathbb{P}(Z \geq d) = \mathbb{P}(e^{Zt} \geq e^{dt}) \leq \frac{\mathbb{E}(e^{Zt})}{e^{dt}}. \tag{15}$$

This gives a tighter upper bound for $\mathbb{P}(Z \geq d)$, which serves as an upper bound for ε in Equation (1). The optimal value of t is chosen to minimize this bound. However, the true expectation of e^{Zt} is difficult to compute directly.

To address this, CC-Cert (Pautov et al., 2022) estimates $\mathbb{E}(e^{Zt})$ by sampling n transformed inputs $\{x_{T_i}\}_{i=1}^n$, calculating $Z_i = \|\mathbf{p} - \mathbf{p}_{T_i}\|_\infty$ for each, and using:

$$Y = \frac{1}{n \cdot e^{dt}} \sum_{i=1}^{n} e^{Z_i t}. \tag{16}$$

Y is an estimate of the upper bound on ε. While Y could overestimate or underestimate the true value, underestimation is more problematic as it could lead to false certification. To mitigate this risk, we sample Y independently l times, resulting in $\{Y_1, \ldots, Y_l\}$. The probability of underestimation is then bounded using the following inequality (derived from Pautov et al. (2022); Paley & Zygmund (1930)):

$$\mathbb{P}\left(\frac{\max\{Y_1, \ldots, Y_l\}}{\alpha} < \frac{\mathbb{E}(e^{Xt})}{e^{dt}}\right) < \left(\frac{1}{1 + \frac{n(1-\alpha)^2}{C_v^2}}\right)^l, \tag{17}$$

where Z is a random variable in $[0, 1]$, α is a hyperparameter, and $C_v = \frac{\sqrt{\mathrm{Var}(e^{Zt})}}{\mathbb{E}(e^{Zt})} \sim 1$ is the coefficient of variation of e^{Zt}.

To obtain a reliable estimate of the upper bound on ε, we repeat the following process l times. For each repetition, we sample n transformed images $\{x_{T_i}\}_{i=1}^n$ from the semantic transformation set $\mathbb{S}_T$ for a given input x, compute $Z_i = \|\mathbf{p} - \mathbf{p}_{T_i}\|_\infty$ for each, and calculate the corresponding estimate Y_i using Equation (16). The maximum of these l independent estimates, $\max\{Y_1, \ldots, Y_l\}/\alpha$, is then used as a conservative upper bound for ε. This ensures that, for sufficiently large n and l, the probability of underestimating the true ε, and therefore falsely certifying a non-robust model, becomes arbitrarily small. This provides a high-confidence probabilistic robustness guarantee.

For the settings, we set $\alpha = 0.9$, $n = 100$, and $l = 10$ in Equation (17), and sweep the temperature vector t logarithmically over $[10^{-4}, 10^4]$ with 500 intervals.

A.8 Implementation Details

The training pipeline proceeds in three stages. First, the initial pretraining stage is conducted for 50 epochs with a learning rate of 0.01. Next, the compression-aware robustness optimization runs for 100 epochs with a learning rate of 1e-4, utilizing the hyperparameter settings $\lambda_{\text{stab}} = 5$, $\lambda_{\text{ratio}} = 1$, $\lambda_{\text{consis}} = 1$, and λ_1 =1e-4. Finally, the compressed model is fine-tuned for 50 epochs with a learning rate of 0.001. All experiments are implemented in PyTorch and executed on a single NVIDIA RTX A6000 GPU.

A.9 Additional Evaluation Results

This subsection collects additional generalization, ablation, variant results referenced in Section 5.

Generalization across CNN backbones. Table 7 extends the main comparison from ResNet-18 to GoogLeNet and MobileNet-V2 on CelebA-HQ at pr= 50%. Despite the structural differences among ResNet-18, GoogLeNet, and MobileNet-V2, C-SSO consistently delivers the highest certified robustness. The most striking improvement occurs in GoogLeNet, where C-SSO achieves 49.0% PCA, outperforming the LMP baseline (7.0%) by 42%. Notably, C-SSO even surpasses vanilla unpruned models by 8% to 27%. This indicates that C-SSO effectively filters out non-robust redundant parameters that would otherwise degrade semantic stability.

Table 7: Robustness generalization across CNN architectures on CelebA-HQ (pr= 50%, Aug.=L1).

Model	Acc			PCA		
	Vanilla	LMP	**C-SSO**	Vanilla	LMP	**C-SSO**
MobileNet-V2	**71.85**	33.91	<u>67.98</u>	<u>33.0</u>	6.0	**44.0**
GoogLeNet	<u>67.16</u>	44.12	**70.86**	<u>22.0</u>	7.0	**49.0**
ResNet-18	**82.96**	82.47	<u>82.72</u>	<u>66.0</u>	54.0	**74.0**

Generalization to transformer. Table 8 evaluates DeiT-Ti with FFN pruning under the Isomorphic pruning protocol (Fang et al., 2024). Although the backbone and pruning granularity differ substantially from the CNN setting, C-SSO still yields consistently higher PCA than the transformer pruning baseline across pruning ratios on both CelebA-HQ and Flowers-102. Note that we only report results

Table 8: Results on DeiT-Ti under different pruning ratios.

Method	Aug.	Pruning Ratio = 30%				Pruning Ratio = 50%				Pruning Ratio = 70%			
		CelebA-HQ		Flowers-102		CelebA-HQ		Flowers-102		CelebA-HQ		Flowers-102	
		Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA	Acc	PCA
Vanilla	None	75.06	42.0	98.29	80.0	75.06	42.0	98.29	80.0	75.06	42.0	98.29	80.0
	L1	73.58	55.0	97.68	92.0	73.58	55.0	97.68	92.0	73.58	55.0	97.68	92.0
	L2	69.88	57.0	97.68	94.0	69.88	57.0	97.68	94.0	69.88	57.0	97.68	94.0
Isomorphic	None	74.56	26.0	**98.90**	60.0	68.56	17.0	97.68	49.0	51.93	13.0	95.36	48.0
	L1	74.65	37.0	**98.90**	75.0	69.47	30.0	97.56	69.0	53.21	20.0	95.84	68.0
	L2	**75.30**	50.0	**98.90**	85.0	**70.70**	43.0	**98.66**	76.0	60.08	28.0	**97.07**	75.0
C-SSO	L1	74.98	59.0	96.94	**90.0**	66.67	40.0	95.72	90.0	**73.00**	**58.0**	94.87	**86.0**
	L2	69.79	**60.0**	97.07	**90.0**	67.49	**51.0**	96.33	**93.0**	68.89	53.0	95.11	85.0

on CelebA-HQ and Flowers-102, which are compatible with the standard 224×224 setting of Deit-Ti [2].

Generalization to structured pruning. Table 9 shows that the advantage of C-SSO is not limited to unstructured mask search. Under channel-level pruning, C-SSO still achieves the best or tied-best PCA on nearly all dataset and augmentation settings, outperforming both DepGraph and HESSO. This indicates that the method transfers across pruning granularities as long as the optimization is posed in mask space. Gains on DeiT-Ti and structured pruning are more moderate than on unstructured CNNs, as pruning entire channels or blocks imposes rigid structural constraints that narrow the search space. Nonetheless, C-SSO consistently improves PCA across all structured settings and in all 12 DeiT-Ti configurations; for instance, C-SSO surpasses Isomorphic pruning by 38% PCA on CelebA-HQ at a 70% pruning ratio under L1 augmentation.

Table 9: Extension to structured pruning (pr $= 50\%$). Comparison of accuracy and PCA across different augmentation levels.

Method	Aug.	CelebA-HQ		Flowers-102		CIFAR-10-C	
		Acc	PCA	Acc	PCA	Acc	PCA
DepGraph	None	80.99	27.0	94.74	56.0	92.10	20.0
	L1	79.51	50.0	94.87	67.0	90.98	73.0
	L2	80.08	60.0	95.35	77.0	**92.48**	70.0
HESSO	None	56.54	30.0	81.78	57.0	85.26	17.0
	L1	58.18	43.0	80.92	73.0	84.54	61.0
	L2	55.39	36.0	80.80	68.0	84.77	67.0
C-SSO	L1	**85.72**	64.0	**95.97**	80.0	90.21	**76.0**
	L2	83.21	**66.0**	95.60	**83.0**	90.04	73.0

Additional dataset extension. Table 10 provides an additional comparison on Tiny-ImageNet. The same failure mode remains visible on this harder benchmark: sequentially pruning a SAM model leads to large PCA degradation, whereas C-SSO preserves substantially higher certified robustness and much stronger accuracy across all sparsity levels.

Generalization to unseen semantic transformations. We further examine whether the robustness learned during mask optimization transfers beyond the training transformation. We train all models using only the smiling transformation and evaluate them on two unseen attributes, arched eyebrows and oval face. Each attribute contains 50 identities, and all models use a pruning ratio of 50%. As shown in Table 11, C-SSO achieves 54% PCA, exceeding S^2-SAM and SAM-LMP by 20% and

[2] https://huggingface.co/facebook/deit-tiny-patch16-224

Table 10: Results on Tiny-ImageNet under different pruning ratios.

Method	pr = 30%		pr = 50%		pr = 70%	
	Acc	**PCA**	**Acc**	**PCA**	**Acc**	**PCA**
SAM (unpruned)	53.65	30.0	53.65	30.0	53.65	30.0
SAM-LMP	41.58	13.0	37.96	9.0	28.96	7.0
C-SSO	**53.57**	**26.0**	**53.21**	**25.0**	**51.27**	**22.0**

9%, respectively. This result provides evidence that the learned structural stability partially transfers beyond the transformation used for training, while remaining lower than the PCA obtained on the original smiling transformation.

Table 11: Generalization to unseen semantic transformations on CelebA-HQ at a pruning ratio of 50%. All methods are trained using only the smiling transformation.

Method	Training transformation	Unseen test transformations	PCA
S^2-SAM	Smiling	Arched eyebrows and oval face	34%
SAM-LMP	Smiling	Arched eyebrows and oval face	45%
C-SSO	Smiling	Arched eyebrows and oval face	**54%**

Effect of paired semantic supervision. During training, C-SSO employs paired original and transformed images (x, x') together with the semantic ratio loss $\mathcal{L}_{\text{ratio}}$ to bound prediction discrepancies. To investigate whether the certified robustness improvement merely stems from having access to this paired semantic supervision rather than from our mask-space structural optimization, we construct an explicit control baseline: we train LMP with the identical paired training samples and the same ratio loss $\mathcal{L}_{\text{ratio}}$ used by C-SSO. The experiment is conducted on CelebA 1K at a pruning ratio of 50%. As shown in Table 12, providing paired ratio supervision to LMP improves its PCA only marginally from 54% to 56%, while C-SSO achieves 74%. This confirms that paired semantic supervision alone cannot account for the robustness gains, and that C-SSO's performance fundamentally originates from its mask-space structural stability optimization.

Table 12: Effect of paired semantic supervision on CelebA 1K at a pruning ratio of 50%.

Method	Paired ratio loss	Acc	PCA
LMP	No	82.47%	54%
LMP with L_{ratio}	Yes	82.55%	56%
C-SSO	Yes	82.72%	**74%**

Variation across training seeds. We repeat the CelebA-HQ evaluation three times at a pruning ratio of 50%. Higher PCA indicates that more inputs satisfy the semantic robustness certificate. As reported in Table 13, C-SSO obtains a mean PCA of 74.00% with a standard deviation of 1.00%, and retains the highest PCA in every run. The variation across runs is small relative to the gap between C-SSO and the compared baselines.

Table 13: PCA across three training runs on CelebA-HQ at a pruning ratio of 50%.

Method	Run 1	Run 2	Run 3	Mean $\pm$ Std
S^2-SAM	57%	55%	53%	55.00% $\pm$ 2.00%
SAM-LMP	59%	58%	58%	58.33% $\pm$ 0.58%
C-SSO	**75%**	**74%**	**73%**	**74.00% $\pm$ 1.00%**

Stage-2 variant analysis. In addition to the default decoupled pipeline, we consider a joint-search variant where both mask parameters and network weights are optimized during Stage 2. As shown in Table 14, this variant can further improve the final performance under pr$= 30\%$ in some settings, indicating that our framework is compatible with joint optimization. We note that our main results adopt mask-only optimization followed by finetuning, as this decoupled design is more stable, computationally efficient, and easier to analyze. We therefore view joint optimization as an orthogonal extension rather than the focus of this work.

We include Hard-STE as an additional baseline (pr$= 50\%$). While STE enables gradient propagation through binary masks, our approach differs in that the optimization is defined over a neighborhood of masks rather than a single binarized solution. Specifically, the soft-mask parameterization allows stochastic mask perturbations, which are required by L_{stab} to explicitly minimize prediction variance across multiple mask instances, encouraging structural stability in mask space. In contrast, Hard-STE optimizes a single projected binary mask via surrogate gradients without enforcing such stability.

Empirically, C-SSO consistently outperforms Hard-STE on CelebA-HQ and Flowers-102, while Hard-STE is competitive or slightly better on CIFAR-10-C. This suggests that soft-mask relaxation is more effective under manifold-based semantic perturbations, where local mask-space stability aligns with smooth semantic variations, whereas its advantage is less pronounced for corruption-style shifts.

Table 14: Alternative Stage-2 designs under different pruning ratios.

Method	**Aug.**	**CelebA-HQ**		**Flowers-102**		**CIFAR-10-C**	
		Acc	PCA	Acc	PCA	Acc	PCA
pr = 30%							
C-SSO$_{\text{Joint}}$	L1	83.21	71.0	96.58	85.0	91.37	81.0
	L2	82.72	68.0	96.45	84.0	92.20	80.0
C-SSO	L1	**84.20**	67.0	**97.19**	83.0	91.76	81.0
	L2	82.88	64.0	96.33	86.0	92.71	81.0
pr = 50%							
Hard-STE	L1	82.06	60.0	96.94	84.0	91.67	81.0
	L2	81.32	64.0	95.97	85.0	**92.96**	**85.0**
C-SSO	L1	82.72	**74.0**	96.45	88.0	91.15	71.0
	L2	79.09	**74.0**	96.33	**91.0**	92.70	82.0

A.10 Parameters Distribution of All Layers

The following figures complement Figure 2 by showing the parameter distributions across all layers. The same trend remains visible throughout the network: compared with LMP, C-SSO preserves a wider band of small-magnitude weights after pruning and exhibits noticeably smaller distribution drift after fine-tuning.

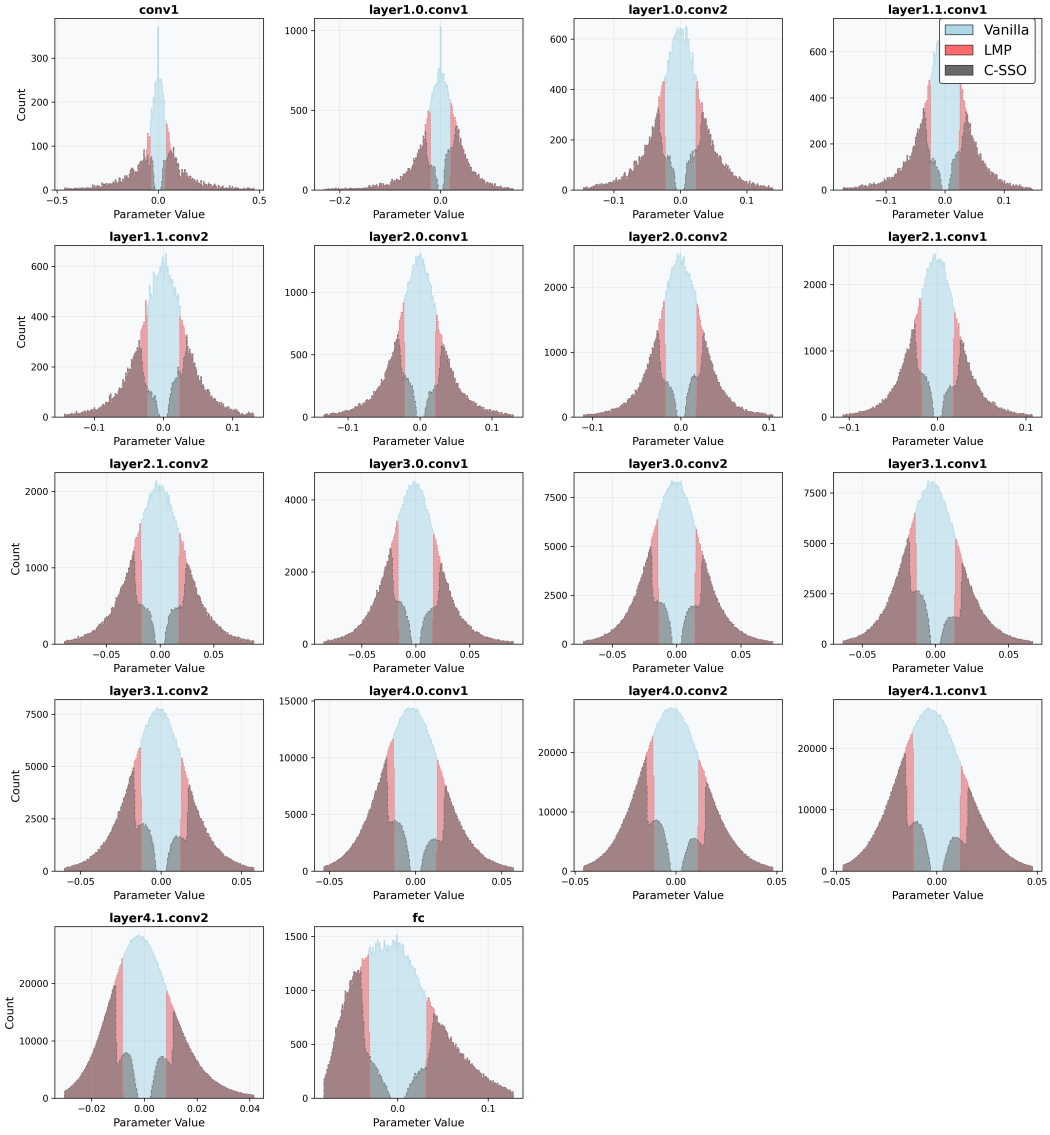


Figure 3: Parameter distributions across all layers immediately after pruning and before fine-tuning. Compared with LMP, C-SSO retains a broader set of small-magnitude weights across the network.

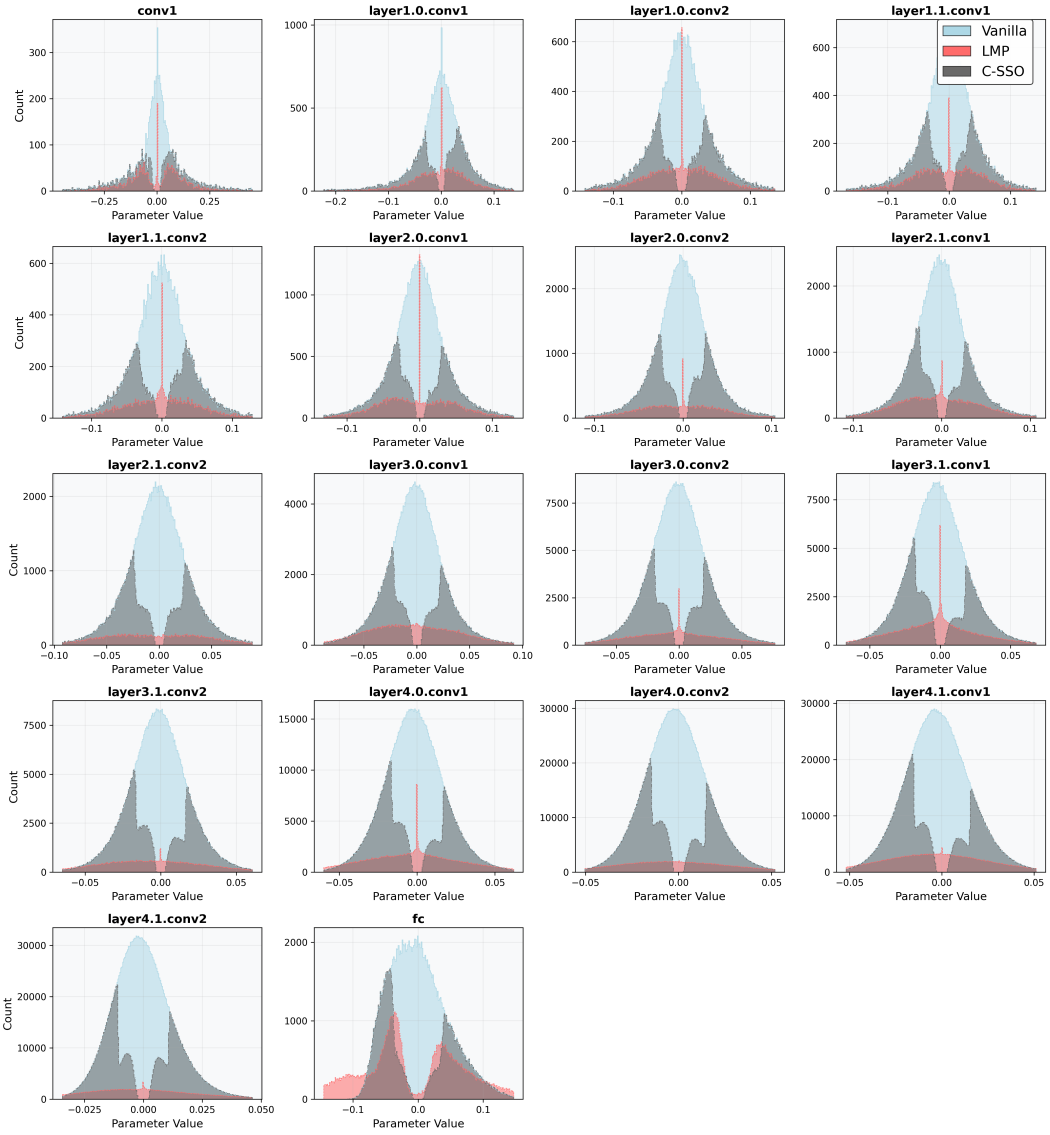


Figure 4: Parameter distributions across all layers after fine-tuning. The distributions produced by C-SSO remain comparatively stable, whereas LMP undergoes much larger post-pruning redistribution.

A.11 Illustration of Transformation Space

This figure visualizes the transformation space $\mathbb{S}_T(x)$ used in certification. The examples illustrate that certification is performed over a continuous family of semantically related samples generated from the same transformation family rather than over a single transformed image.



Figure 5: Illustration of the transformation space $\mathbb{S}_T(x)$, where the leftmost image is the original sample and the remaining images correspond to different transformation magnitudes within the same family.