SOLAR: Switchable Output Layer for Accuracy and Robustness in Once-for-All Training

Shaharyar Ahmed Khan Tareen Lei Fan Xiaojing Yuan Qin Lin Bin Hu
University of Houston, USA
stareen@cougarnet.uh.edu, lfan8@central.uh.edu, xyuan@uh.edu,
qlin21@central.uh.edu, bhu11@central.uh.edu

Abstract

*Once-for-All (OFA) training enables a single super-net to generate multiple sub-nets tailored to diverse deployment scenarios, supporting flexible trade-offs among accuracy, robustness, and model-size without retraining. However, as the number of supported sub-nets increases, excessive parameter sharing in the backbone limits representational capacity, leading to degraded calibration and reduced overall performance. To address this, we propose **SOLAR** (**S**witchable **O**utput **L**ayer for **A**ccuracy and **R**obustness in Once-for-All Training), a simple yet effective technique that assigns each sub-net a separate classification head. By decoupling the logit learning process across sub-nets, the Switchable Output Layer (SOL) reduces representational interference and improves optimization, without altering the shared backbone. We evaluate SOLAR on five datasets (SVHN, CIFAR-10, STL-10, CIFAR-100, and TinyImageNet) using four super-net backbones (ResNet-34, WideResNet-16-8, WideResNet-40-2, and MobileNetV2) for two OFA training frameworks (OATS and SNNs). Experiments show that SOLAR outperforms the baseline methods: compared to OATS, it improves accuracy of sub-nets up to **1.26%**, **4.71%**, **1.67%**, and **1.76%**, and robustness up to **9.01%**, **7.71%**, **2.72%**, and **1.26%** on SVHN, CIFAR-10, STL-10, and CIFAR-100, respectively. Compared to SNNs, it improves TinyImageNet accuracy by up to **2.93%**, **2.34%**, and **1.35%** using ResNet-34, WideResNet-16-8, and MobileNetV2 backbones (with 8 sub-nets), respectively. The code of **SOLAR** is publicly available at: https://github.com/NAIL-UH/SOLAR and its website can be accessed at https://saktx.github.io/solar.github.io/.*

1. Introduction

Deploying deep neural networks across a wide range of devices—from high-performance servers to resource-constrained edge platforms—requires customized models that balance accuracy, robustness [6, 14, 30, 47], and model-size (or efficiency). Once-for-All (OFA) training [3, 8, 24, 25, 44, 45] addresses this by optimizing a single versatile super-network containing many sub-networks that are tailored to different deployment constraints. The sub-nets can then be selected post-training to meet trade-offs among accuracy, adversarial robustness [7, 28, 48], model-size, or computational cost, without retraining from scratch [3, 24, 38, 44, 45]. While OFA training [38, 44, 45] offers flexibility and efficiency, scaling to a large number of sub-nets introduces a fundamental challenge: *excessive parameter sharing*. When all sub-nets share a single output layer, representational interference occurs, preventing each sub-net from optimizing independently. This coupling of parameters degrades accuracy, calibration, and robustness, particularly for sub-nets with differing capacities.

In this paper, we identify the shared output layer as a bottleneck in OFA frameworks and propose **SOLAR** (*Switchable Output Layer for Accuracy and Robustness in Once-for-All Training*), a simple yet effective approach that introduces separate classification heads for the sub-nets. SOLAR decouples the logit learning process in the common output layer, mitigating logit interference during training and improving the sub-net specific optimization while maintaining the training efficiency.

Key Contributions: Our main contributions are summarized below:

- We identify the shared output layer as a bottleneck in OFA training, that leads to representational interference and performance degradation across the sub-nets. To address this, we propose **SOLAR** (*Switchable Output Layer for Accuracy and Robustness*), a simple and effective method that assigns each sub-net a separate classification head while preserving the shared backbone.
- We incorporate SOLAR into two OFA frameworks: Slimmable Neural Networks (SNNs) [45], which vary network width dynamically during standard training, and Once-for-All Adversarial Training and Slimming (OATS)

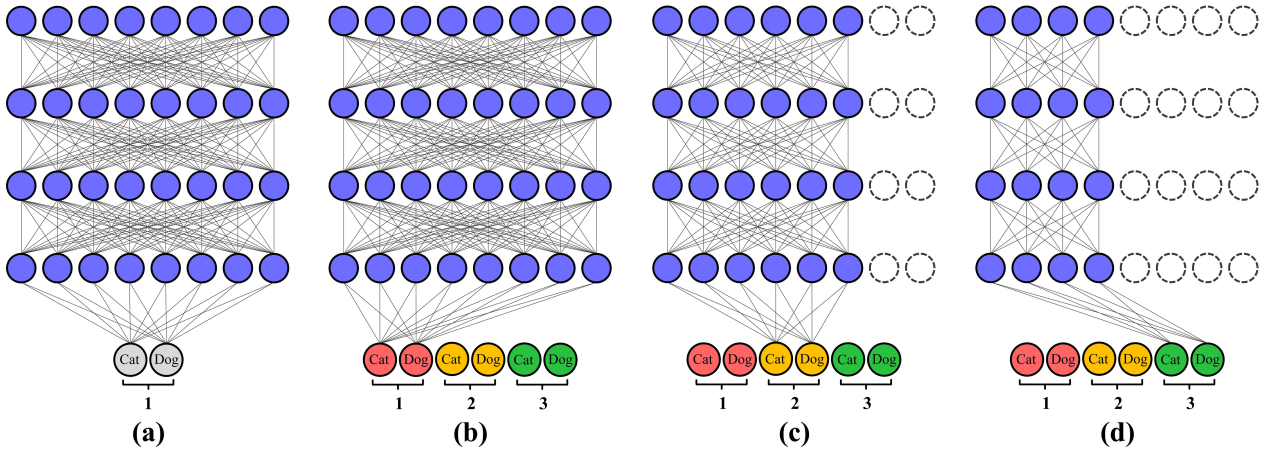

Figure 1. Illustration of a vanilla SNN [45] vs. SNN with Switchable Output Layer (SNN-SOL) in a super-net backbone with three sub-nets of different widths: (a) vanilla SNN with a shared output layer; (b–d) SNN-SOL with 100%, 75%, and 50% widths, respectively. Width refers to the number of channels per layer. SOL assigns a separate classification head to each sub-net, enabling decoupled logit learning, with the number of heads equal to the number of sub-nets in the backbone.

[38], which combines adversarial training with the dynamic width shrinking and uses conditional loss function.

- We perform extensive experiments across five benchmark datasets (SVHN [29], CIFAR-10 [22], STL-10 [9]), CIFAR-100 [22], and TinyImageNet [26], using four different super-net backbones (WideResNet-16-8 [46], ResNet-34 [16], WideResNet-40-2 [46], MobileNetV2 [34]), demonstrating that SOLAR generalizes well and improves both standard accuracy and adversarial robustness across the sub-nets and frameworks.
- Our smallest sub-net from OATS-SOL, trained on the SVHN dataset using WideResNet-16-8 backbone, achieves the best accuracy of **94.01%** and robustness of **53.08%**, surpassing the standard OATS [38] baseline by **0.57%** and **1.57%**, respectively, while maintaining the compact model size of **387 KB**.

2. Related Work

Once-for-All (OFA) Training: OFA framework [3] trains a single over-parameterized super-net from which many sub-nets can be derived by sampling architectures with different depths, widths, kernel sizes, or input resolutions. These sub-nets inherit weights from the super-net, enabling efficient deployment without retraining from scratch. A progressive shrinking strategy [3, 8, 31] is used to jointly optimize all sub-nets. Although OFA enables massive scalability and supports over 10^{19} sub-nets, training a super-net that performs well across all sub-nets is hard, because smaller sub-nets suffer from degraded performance due to conflicting gradients and **"shared parameters"**. When many sub-nets share parameters, gradients from different sub-nets cause interference, making it harder to optimize the shared layers for all sub-net configurations [38, 45].

Slimmable Neural Networks (SNNs): SNNs [45] follow the OFA principle by training a single super-net operating only at four widths (0.25×, 0.5×, 0.75×, 1.0×). SNNs provide a twofold trade-off between accuracy and model size (or efficiency). They address key OFA challenges—particularly performance degradation caused by conflicting feature statistics when all sub-nets share a single Batch Normalization (BN) layer [20]—by introducing Switchable Batch Normalization (SBN) [45], which assigns a separate BN layer to each sub-net. This design reduces training instability and gradient interference, improving performance across widths [12]. However, SNNs still rely on a shared output layer for all sub-nets, which becomes a bottleneck as the sub-net diversity grows. This limits the capacity to fully adapt output representations to varying sub-net complexities. Our proposed Switchable Output Layer (SOL) solves this problem by providing sub-net-specific classification heads, effectively overcoming the common output layer bottleneck and further enhancing sub-net performance without sacrificing training cost.

Once-for-All Adversarial Training and Slimming (OATS): OATS [38] extends the Once-for-All Adversarial Training (OAT) framework [38] by integrating model compactness across widths like the SNNs [45]. It trains a single super-net supporting three widths (0.5×, 0.75×, 1.0×) via channel-wise slimming, enabling deployment across devices with varying resources. During training, OATS conditions on both adversarial loss weight λ and width fraction, allowing the super-net to enable a balance between accuracy, robustness, and efficiency without requiring retraining from scratch. To handle distribution mismatches between clean and adversarial samples [12], OATS introduces Switchable Dual Batch Normalization (SDBN) [38]

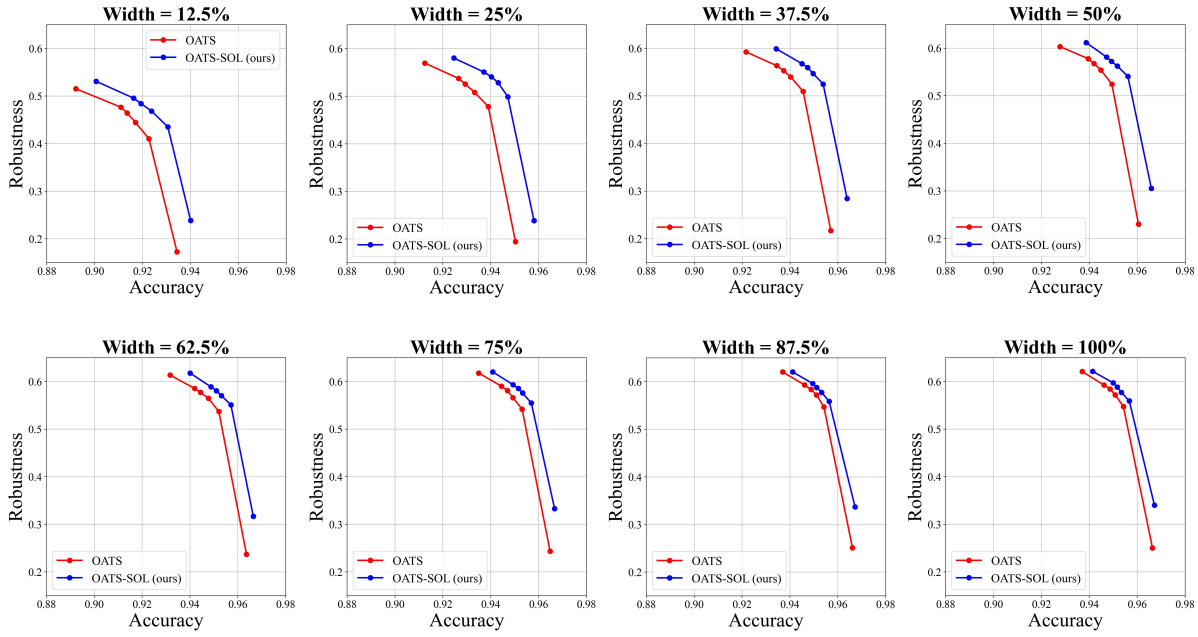

Figure 2. Comparison of OATS [38] and OATS-SOL on SVHN dataset using WideResNet-16-8 backbone packed with 8 sub-nets. OATS-SOL provides superior performance than OATS for all the sub-nets in terms of accuracy and PGD-7 robustness.

with separate BN layers for each data type, and sub-net width, ensuring stable, high-performance training. Conditional learning techniques such as FiLM layers [33, 38] or scaled noise injection [24, 25] enable adaptive behavior based on input conditions. OATS [38] employs FiLM layers for this purpose. Our proposed Switchable Output Layer (SOL) improves the performance of both SNNs and OATS frameworks across diverse datasets and architectures, offering superior accuracy, robustness, and efficiency trade-offs.

3. Preliminaries

Consider a multi-class classification setting with N training samples and C classes. For each sample i, let $y_i \in \{1, \ldots, C\}$ denote the ground-truth label, and let $\mathbf{p}_i = (p_{i1}, \ldots, p_{iC})$ denote the predicted class probabilities, computed via the softmax function from the model logits z_{ic}:

$$p_{ic} = \frac{\exp(z_{ic})}{\sum_{j=1}^{C} \exp(z_{ij})} \tag{1}$$

Given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, C\}$, a neural network $f : \mathbb{R}^d \to \mathbb{R}^C$ with parameters θ maps inputs to logits. The model is typically trained using empirical risk minimization (ERM) with the cross-entropy loss:

$$\mathcal{L}_{\text{CE}} = \mathcal{L}(f(\mathbf{x}_i; \theta), y_i) = -\log p_{iy_i} \tag{2}$$

Adversarial Training (AT): AT has been widely adopted to improve model robustness by explicitly optimizing for performance under worst-case input perturbations [1, 7, 14, 19, 28, 35, 43, 50]. A common approach is to use a hybrid loss $\mathcal{L}_{\text{Hybrid}}$ that combines standard classification loss $\mathcal{L}_{\text{CE}}$ and adversarial loss $\mathcal{L}_{\text{ADV}}$ [4, 37, 41, 48]:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{D}} \underbrace{[(1-\lambda)\mathcal{L}_{\text{CE}} + \lambda\mathcal{L}_{\text{ADV}}]}_{\mathcal{L}_{\text{Hybrid}}} \tag{3}$$

where the adversarial loss is defined as:

$$\mathcal{L}_{\text{ADV}} = \max_{\delta \in \mathcal{B}_\epsilon(\mathbf{x})} \mathcal{L}(f(\mathbf{x}+\delta; \theta), y) \tag{4}$$

and $\mathcal{B}_\epsilon(\mathbf{x}) = \{\delta \in \mathbb{R}^d \mid \|\delta\|_\infty \leq \epsilon\}$ is the L_∞-ball around $\mathbf{x}$, constraining the magnitude of adversarial perturbations. A widely used method for solving the inner maximization problem in adversarial training is Projected Gradient Descent (PGD) [4, 19, 28]. Given an input $\mathbf{x}$ and label y, PGD generates an adversarial example $\mathbf{x}'$ as:

$$\mathbf{x}' = \Pi_{\mathcal{B}_\epsilon(\mathbf{x})} \left(\mathbf{x} + \gamma \cdot \text{sign}\left(\nabla_{\mathbf{x}} \mathcal{L}(f(\mathbf{x}; \theta), y)\right)\right) \tag{5}$$

where γ is the step size and $\Pi_{\mathcal{B}_\epsilon(\mathbf{x})}(\cdot)$ denotes projection onto the L_∞-ball around $\mathbf{x}$. This procedure can be iterated multiple times to find stronger adversarial examples.

Switchable Dual Batch Normalization (SDBN): Dual Batch Normalization (DBN) [11, 20, 42] addresses the distribution mismatch between clean and adversarial data during adversarial training. DBN maintains two sets of BN parameters: one for clean (BN_{cln}) and one for adversarial data (BN_{adv}) [11, 12]. This normalization strategy prevents feature distortion and improves performance and robustness [2, 12, 39]. SDBN extends this idea to super-nets

containing multiple sub-nets and maintains DBN layer for each switch (or sub-net) in the backbone [38]. Each sub-net in the width adaptive backbones [38, 45], exhibits distinct feature statistics [38, 45]. SNNs use SBN to preserve performance across all widths, that maintains separate BN parameters $(\gamma_{\alpha_k}, \beta_{\alpha_k}, \mu_{\alpha_k}, \sigma^2_{\alpha_k})$, leading to the following parameter set.

$$\Theta = \{W, \{\gamma_{\alpha_k}, \beta_{\alpha_k}, \mu_{\alpha_k}, \sigma^2_{\alpha_k}\}_{k=1}^{K}\} \tag{6}$$

Where W represents the shared weights of Convolutional and Linear layers, $\gamma_{\alpha_k}, \beta_{\alpha_k}$ are the scale and shift parameters, $\mu_{\alpha_k}, \sigma^2_{\alpha_k}$ are the running mean and variance for each sub-net with width fraction α_k. The SBN parameters are not shared among the sub-nets. Beyond normalization, OATS [38] uses model-conditional learning to enable on-the-fly accuracy–robustness trade-offs by conditioning the model on structural or stochastic inputs (e.g. via FiLM layers on a hyperparameter like λ) [18, 21, 38, 40]. Despite these solutions, a bottleneck remains in the end: the **shared output layer** where all sub-nets share a single classification head, causing *logit interference* that limits learning capacity and calibration. To address this, we introduce the **Switchable Output Layer (SOL)**, which assigns a separate classification head to each sub-net in the backbone. Like SBN separates out the feature normalization for different sub-nets [45], the SOL isolates their logit learning processes, enabling overall better optimization for each sub-net.

4. Methodology

4.1. Bottleneck in OFA Training: Standard Output Layer (Shared Parameters)

OFA training aims to train a single over-parametrized super-net from which multiple sub-nets can be derived for diverse deployment requirements without retraining. Let $\mathcal{N}(x; \Theta)$ denote the super-net with input x and the parameter set Θ, as defined in (6). To enable flexible sub-net instantiation, a predefined set of width multipliers $\mathcal{W} = \{\alpha_1, \alpha_2, \ldots, \alpha_k\}$ where $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_k \leq 1$ is used to scale the number of active channels in each layer of $\mathcal{N}$ [38, 45]. The k^{th} sub-net, corresponding to width multiplier α_k, is denoted as $\mathcal{S}_{\alpha_k}(x) = \mathcal{N}(x; \Theta_{\alpha_k})$ where $\Theta_{\alpha_k} \subseteq \Theta$ represents the subset of parameters utilized by the sub-net. The full-width super-net corresponds to $\alpha_K = 1$, using the complete parameter set Θ. The goal of OFA training is to learn optimal parameters Θ that minimize a predefined objective as in (7).

$$\min_{\Theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(\mathcal{N}(x_i; \Theta), y_i) \tag{7}$$

During training, a width multiplier $\alpha_k \in \mathcal{W}$ is sampled to activate the corresponding sub-net $\mathcal{S}_{\alpha_k}$, which produces intermediate features $f_{\alpha_k} = \mathcal{S}_{\alpha_k}(x)$. All sub-nets share a common classification head C, used to compute logits: $z_{\alpha_k} = \mathrm{C}(f_{\alpha_k})$.

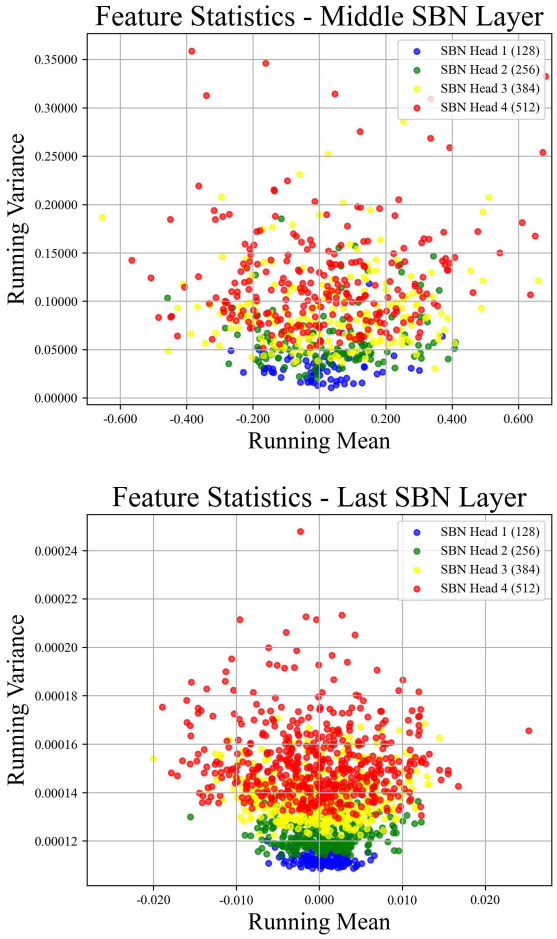

Figure 3. Visualizing the running means and variances of SBN layers for clean data in OATS-based ResNet-34 (with 4 sub-nets) during training. Top plot shows scattered statistics in the middle SBN, while bottom plot reveals tighter and distinct distributions in the last SBN. The shared output layer in OATS causes representational interference. SOL alleviates this by assigning separate classification heads to the sub-nets, thereby improving calibration and performance without increasing their parameter count.

Since sub-nets of different widths produce different feature distributions [38, 45], SNNs [45] used SBN layers and OATS [38] used SDBN layers to maintain separate normalization parameters for each sub-net. Although these layers decouple intermediate features, all sub-nets still converge at the shared output layer, which is commonly adopted in traditional OFA methods [3, 8, 25, 38, 44, 45]. This shared head creates a bottleneck by forcing incompatible features into a single representation space, leading to degraded performance. The design space becomes more complex [24, 38, 44, 45], when the number of sub-nets in-

creases. We analyzed the statistics of the middle and final $\mathrm{BN}_{\mathrm{cln}}$ layers of ResNet-34 super-net in Figure 3. The figure reveals significant variance across sub-nets: the narrowest sub-net (blue) exhibits compact, low-variance features, whereas the widest (red) shows high-variance, dispersed distribution. This indicates that, although SBN and SDBN preserve internal feature separation, the **shared output head** remains a bottleneck at the end of the pipeline.

4.2. Overcoming Bottleneck with Switchable Output Layer (Separate Parameters)

As established in previous section, the shared output layer in OFA frameworks introduces a critical bottleneck by forcing diverse sub-net features into a single classification space, leading to interference and degraded performance. The **Switchable Output Layer (SOL)** enhances OFA frameworks [3, 8, 25, 38, 44, 45] by addressing performance degradation caused by the shared output layer across sub-nets. SOL introduces a lightweight modification by assigning a separate classification head to each sub-net in the super-net backbone, thereby preserving the sub-net specific representations through to the output and eliminating cross sub-net conflict during training (see Figure 1).

Instead of sharing a classification head C for all sub-nets, SOL introduces a unique head C_{α_k} for each sub-net $\mathcal{S}_{\alpha_k}$. During training, a width multiplier $\alpha_k \in \mathcal{W}$ activates the corresponding sub-net to produce intermediate features:

$$z_{\alpha_k} = \mathrm{C}_{\alpha_k}(f_{\alpha_k}) \tag{8}$$

Only the active head C_{α_k} receives updates during backpropagation, while others remain inactive. This dynamic switching ensures that each sub-net learns independently, avoiding interference from incompatible gradients at the output layer (see Figure 1 and Figure 4). The full OATS-SOL pipeline is detailed in Algorithm 1. SOL is agnostic to the choice of loss function and supports a range of training objectives, including standard cross-entropy $\mathcal{L}_{CE}$ [23, 27] or distillation $\mathcal{L}_{KLD}$ [15, 17, 36], and robust objectives like TRADES [48], MMA [10], and adversarial distillation [13, 15, 32]. For hybrid training (as in OATS), the total loss across eight sub-nets is defined as:

$$\mathcal{L} = \sum_{\alpha \in \{1/8,\, 2/8,\, \ldots,\, 8/8\}} \left[(1-\lambda)\, \mathcal{L}_{\mathrm{CE}}^{(\alpha)} + \lambda\, \mathcal{L}_{\mathrm{ADV}}^{(\alpha)} \right] \tag{9}$$

5. Experiments & Results

5.1. Super-Net Architectures and Setup

We evaluate SOLAR on four super-net backbones: WideResNet-16-8 [46], ResNet-34 [16], WideResNet-40-2 [46], and MobileNetV2 [34], using five benchmark datasets: SVHN [29], CIFAR-10 [22], STL-10 [9], CIFAR-100 [22], and TinyImageNet [26]. Experiments have been conducted on a workstation with Intel Core i9-14900KF CPU, 36 MB L3 cache, and NVIDIA RTX 4090.

Algorithm 1 Once-for-All Adversarial Training and Slimming with Switchable Output Layer (OATS-SOL)

Require: Training set $\mathcal{D}$, set $\mathbb{S}$ with λ values, Super-Net $\mathcal{N}$, max iterations T, width multipliers list $\mathcal{W}$
Ensure: Network parameters Θ

```
for t = 1 to T do
    Sample batch (x, y) from D and λ from S
    Clear gradients: optimizer.zero_grad()
    for each α in W do
        Activate sub-net S_{α_k} with head C_{α_k} in N
        Generate adversarial example x_adv
        Perform forward pass
        Compute loss = L(x, y, λ)
        Accumulate gradients: loss.backward()
    end for
    Update parameters: optimizer.step()
end for
```

5.2. Hyperparameter Settings

We trained OATS-SOL on SVHN, CIFAR-10, STL-10, and CIFAR-100 for 40, 120, 120, and 120 epochs and SNN-SOL on CIFAR-10, CIFAR-100, and TinyImageNet for 120, 120, and 140 epochs, respectively. Batch sizes were set to 64 for STL-10 and TinyImageNet, and 128 for all the other datasets. We used SGD with momentum 0.9, cosine annealing scheduler, and learning rates: $\{0.1, 0.05, 0.01\}$. Reported results correspond to the best performance across multiple runs with different random seeds, selected based on optimal validation accuracy.

Adversarial Training and Evaluation: We use same settings as OATS [38] using 7-step PGD attack (L_∞ norm), $\epsilon = 8/255$, and step-size $= 2/255$. Models are evaluated on the basis of **Accuracy** and **Robustness**. Following OATS [38], we uniformly sample λ element-wise from the set $\mathbb{S} = \{0.0, 0.1, 0.2, 0.3, 0.4, 1.0\}$ during training and validation. For inference, λ can be any value in [0,1], as per requirements for the accuracy-robustness trade-off.

5.3. Baseline Methods: OATS and SNNs

We evaluated SOLAR on two baselines: Once-for-All Adversarial Training and Slimming (OATS) [38] and Slimmable Neural Networks (SNNs) [45]. After employing Switchable Output Layer (SOL), we denote them as OATS-SOL and SNN-SOL. For fair comparison, we evaluate the baseline and SOLAR methods using same hyperparameters.

Comparison with OATS: For comparing OATS and OATS-SOL, we used WideResNet-16-8, ResNet-34, WideResNet-40-2, and ResNet-34 as backbones on the

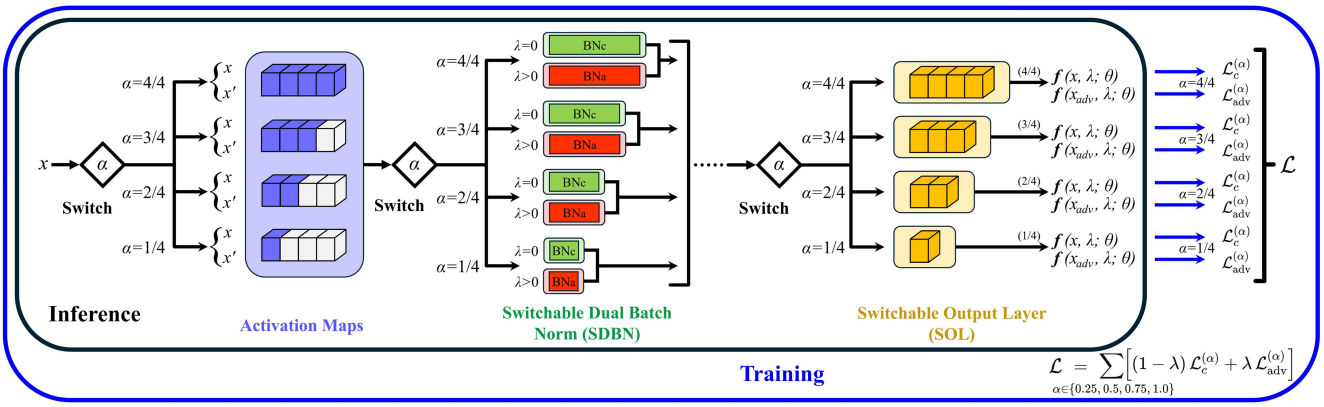

Figure 4. Illustration of OATS framework [38] with Switchable Dual Batch Norm (SDBN) layers and Switchable Output Layer (SOL). Value of α indicates the width fraction for the corresponding sub-net in the backbone that is activated during training or inference. Convolutional layers have shared parameters whereas SDBN and SOL have separate parameters for each sub-net.

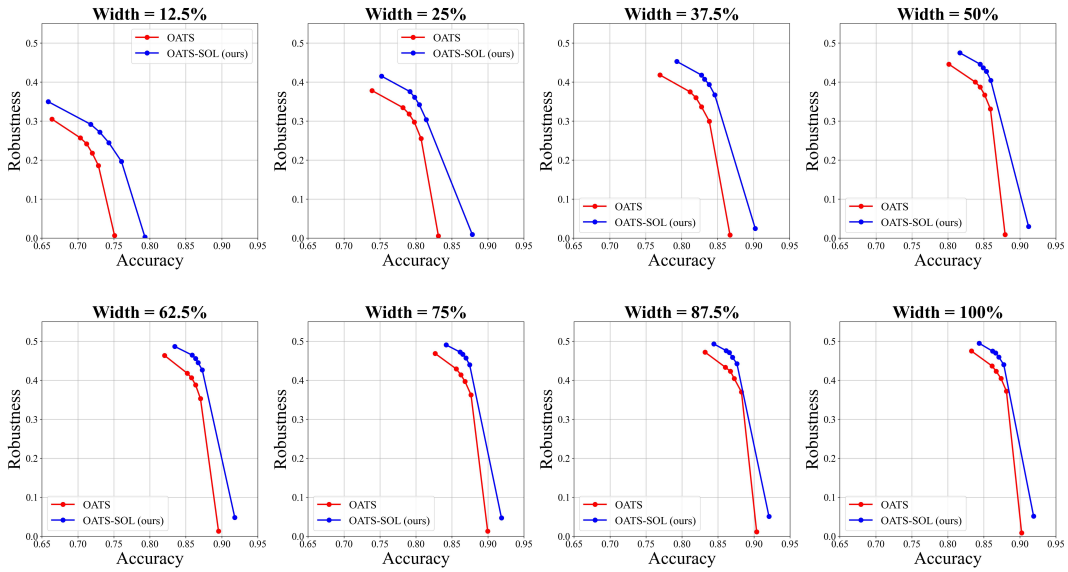

Figure 5. Comparison of OATS [38] and OATS-SOL on CIFAR-10 dataset using ResNet-34 backbone (with 8 sub-nets). OATS-SOL provides superior performance than OATS for all the sub-nets in terms of accuracy and PGD-7 based robustness.

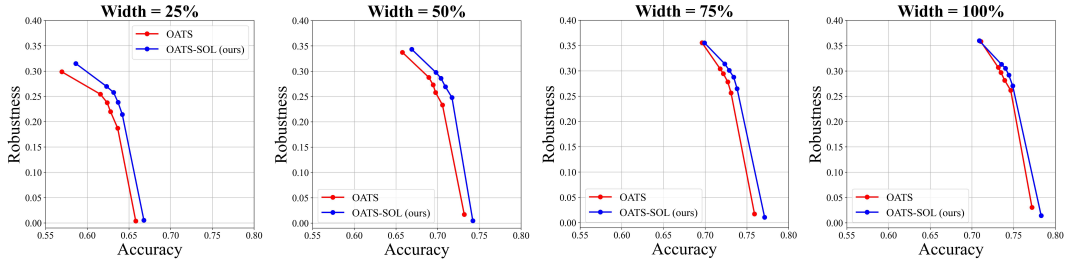

Figure 6. Comparison of OATS [38] and OATS-SOL on STL-10 dataset using WideResNet-40-2 as backbone with 4 sub-nets.

SVHN, CIFAR-10, STL-10, and CIFAR-100 datasets, respectively. The accuracy and robustness of OATS and OATS-SOL sub-nets can be tuned by changing the values of $\lambda \in [0.0, 1.0]$ for free during run-time [38]. To meet the tighter memory and storage constraints of devices, sub-nets with smaller widths are preferred. The comparison of OATS [38] and OATS-SOL on SVHN dataset using WideResNet-16-8 backbone (packed with 8 sub-nets)

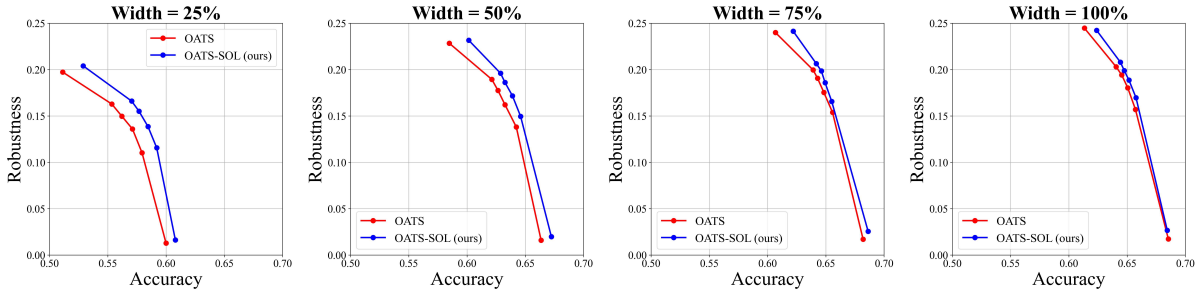

Figure 7. Comparison of OATS [38] and OATS-SOL on CIFAR-100 dataset using ResNet-34 as backbone with 4 sub-nets.

Table 1. Accuracy of SNN [45] and SNN-SOL based Sub-Nets on CIFAR-10 for backbones packed with 16 Sub-Nets.

Sub-Net Width	ResNet-34			WideResNet-16-8			MobileNetV2		
	SNN	SNN-SOL	Gain % ↑	SNN	SNN-SOL	Gain % ↑	SNN	SNN-SOL	Gain % ↑
Small Sub-Nets									
1/16	83.28	83.55	**0.27**	82.72	82.94	**0.22**	82.16	84.17	**2.01**
2/16	87.71	88.01	**0.30**	86.96	88.37	**1.41**	86.24	88.91	**2.67**
3/16	89.88	90.07	**0.19**	88.98	91.41	**2.43**	88.13	90.50	**2.37**
4/16	90.96	91.27	**0.31**	92.35	92.46	**0.11**	89.18	90.59	**1.41**
5/16	91.86	91.88	**0.02**	92.51	93.12	**0.61**	89.56	91.96	**2.40**
6/16	92.32	92.64	**0.32**	93.00	93.63	**0.63**	90.26	92.61	**2.35**
Medium Sub-Nets									
7/16	92.48	92.98	**0.50**	93.14	93.81	**0.67**	90.82	92.70	**1.88**
8/16	92.54	93.35	**0.81**	93.78	93.95	**0.17**	90.68	92.88	**2.20**
9/16	92.78	93.48	**0.70**	93.93	94.23	**0.30**	91.37	92.87	**1.50**
10/16	92.84	93.80	**0.96**	93.97	94.20	**0.23**	91.80	93.18	**1.38**
11/16	92.88	93.86	**0.98**	93.91	94.28	**0.37**	91.51	93.38	**1.87**
Large Sub-Nets									
12/16	93.07	93.84	**0.77**	94.12	94.46	**0.34**	91.91	93.21	**1.30**
13/16	93.13	93.89	**0.76**	94.07	94.55	**0.48**	91.93	93.29	**1.36**
14/16	93.27	93.98	**0.71**	94.11	94.62	**0.51**	92.12	93.64	**1.52**
15/16	93.32	93.94	**0.62**	94.10	94.57	**0.47**	92.03	93.65	**1.62**
16/16	93.38	94.04	**0.66**	94.09	94.67	**0.58**	92.15	93.45	**1.30**

Table 2. Accuracy of SNN [45] and SNN-SOL on CIFAR-100 for backbones with 8 sub-nets.

Sub-Net Width	ResNet-34			WideResNet-16-8			MobileNetV2		
	SNN	SNN-SOL	Gain % ↑	SNN	SNN-SOL	Gain % ↑	SNN	SNN-SOL	Gain % ↑
1/8	66.23	66.18	**-0.05**	64.54	64.33	**-0.21**	64.45	64.37	**-0.08**
2/8	71.64	72.07	**0.43**	71.64	71.96	**0.32**	68.94	68.91	**-0.03**
3/8	73.91	74.82	**0.91**	74.48	74.28	**-0.20**	71.16	71.59	**0.43**
4/8	75.33	76.26	**0.93**	75.86	75.97	**0.11**	72.15	72.38	**0.23**
5/8	76.26	77.28	**1.02**	76.71	76.93	**0.22**	73.04	73.54	**0.50**
6/8	76.79	78.10	**1.31**	76.91	77.27	**0.36**	73.26	73.78	**0.52**
7/8	76.74	78.08	**1.34**	77.37	77.83	**0.46**	73.51	73.92	**0.41**
8/8	76.78	78.43	**1.65**	77.97	78.22	**0.25**	73.23	74.19	**0.96**

Table 3. Accuracy of SNN [45] and SNN-SOL on TinyImageNet for backbones with 8 sub-nets.

Sub-Net Width	ResNet-34			WideResNet-16-8			MobileNetV2		
	SNN	SNN-SOL	Gain % ↑	SNN	SNN-SOL	Gain % ↑	SNN	SNN-SOL	Gain % ↑
1/8	51.22	52.33	**1.11**	45.81	46.54	**0.73**	48.64	49.57	**0.93**
2/8	57.79	59.65	**1.86**	56.30	56.22	**-0.08**	54.85	55.76	**0.91**
3/8	60.12	61.73	**1.61**	60.02	60.26	**0.24**	57.62	58.97	**1.35**
4/8	61.62	63.41	**1.79**	61.93	62.86	**0.93**	58.67	59.37	**0.70**
5/8	62.02	64.37	**2.35**	62.24	63.74	**1.50**	59.75	60.04	**0.29**
6/8	62.83	65.65	**2.82**	63.34	64.73	**1.39**	59.67	60.31	**0.64**
7/8	63.50	66.08	**2.58**	63.12	65.46	**2.34**	59.66	60.81	**1.15**
8/8	63.46	66.39	**2.93**	63.61	65.34	**1.73**	59.91	61.17	**1.26**

is shown in Figure 2. OATS-SOL is represented by **blue color** and provides superior performance than OATS for all the sub-nets in terms of accuracy and robustness. The performance gain is generally higher in smaller sub-nets as compared to larger ones. Similarly, Figure 5, Figure 6, and Figure 7 show the comparison of OATS and OATS-SOL on CIFAR-10 (ResNet-34 backbone with 8 sub-nets), STL-10 (WideResNet-40-2 backbone with 4 sub-nets), and CIFAR-100 (ResNet-34 backbone with 4 sub-nets), respectively. OATS-SOL provides better accuracy and robustness for all

Table 4. Parameter counts of SNN [45] vs. SNN-SOL backbones packed with 8, 32, and 64 sub-nets for CIFAR-10 dataset.

Super-Net	8 Sub-Nets			32 Sub-Nets			64 Sub-Nets		
	SNN	SNN-SOL	Increase	SNN	SNN-SOL	Increase	SNN	SNN-SOL	Increase
WRN-40-2	2.26M	2.28M	0.97%	2.33M	2.41M	3.61%	2.41M	2.58M	6.88%
MobileNetV2	2.35M	2.40M	1.91%	2.75M	2.96M	7.23%	3.28M	3.68M	12.32%
WRN-16-8	10.99M	11.00M	0.16%	11.07M	11.15M	0.72%	11.19M	11.35M	1.45%
ResNet-34	21.34M	21.36M	0.08%	21.55M	21.63M	0.37%	21.82M	21.98M	0.74%

the sub-nets across the datasets. We present a numerical comparison between OATS and OATS-SOL for CIFAR-10 dataset in the Appendix with additional insightful results.

Comparison with SNNs: We compare SNN [45] and SNN-SOL using ResNet-34, WideResNet-16-8, and MobileNetV2 backbones on CIFAR-10, CIFAR-100, and TinyImageNet. Table 1 reports results on CIFAR-10 with 16 sub-nets, where SNN-SOL achieves maximum gains of **0.32%**, **2.43%**, and **2.67%** for small sub-nets, **0.98%**, **0.67%**, and **2.20%** for medium sub-nets, and **0.77%**, **0.58%**, and **1.62%** for large sub-nets across the three backbones. On CIFAR-100 (Table 2), SNN-SOL improves accuracies by up to **1.65%**, **0.46%**, and **0.96%**, using the backbones with 8 sub-nets. On TinyImageNet (Table 3), improvements reach **2.93%**, **2.34%**, and **1.35%** for ResNet-34, WideResNet-16-8, and MobileNetV2, all packed with 8 sub-nets, respectively. The experimental results demonstrate that when the super-net backbones are packed with higher number of sub-nets, the Switchable Output Layer (SOL) provides notable performance gains for the sub-nets.

5.4. Impact of SOL on Parameter Count and FLOPs

The training and inference overhead using SOL is negligible: only the active head is used for each sub-net, containing the same number of parameters as in the baseline. The shared output layer performs **"slimming"** whereas SOL performs **"switching"**. SOL increases parameter storage due to use of multiple heads in the super-net but does not increase the parameters for any sub-net. The performance gains come from the **"unshared"** nature of parameters. Table 4 shows the parameter increase due to SOL in the super-nets. FLOPs during forward pass for a SOL based sub-net with width fraction α_k is the sum of the FLOPs of feature encoder $\mathcal{S}_{\alpha_k}$ and the FLOPs of its head C_{α_k}, as in (10).

$$\mathcal{F}(\mathcal{S}_{\alpha_k}) + \mathcal{F}(\mathcal{C}_{\alpha_k}) = \mathcal{F}(\mathcal{S}_{\alpha_k}) + \mathcal{F}(\mathcal{C}) \tag{10}$$

Since, the number of parameters is same in the output layer after slimming or switching, FLOPs of the SOL sub-nets are identical to the corresponding baseline sub-nets.

5.5. Post-Search Fine-Tuning of Sub-Nets

We perform post-search fine-tuning on randomly selected sub-nets from the SNN and SNN-SOL backbones. As shown in Table 5, SOL sub-nets consistently achieve higher performance even after fine-tuning. This indicates that the unshared output heads in SOL enable sub-nets to learn stronger representations from the beginning. Fine-tuning refines these representations but does not fundamentally alter them, so well-optimized sub-nets continue to outperform. In contrast, sub-nets from the backbones with shared output layer cannot close this gap, as their representational limitations persist despite fine-tuning. Overall, SOL facilitates better optimization across all sub-nets by guiding them toward more generalizable minima, ensuring their dominance is preserved even after fine-tuning.

Table 5. Performance of different sub-nets after fine-tuning.

Dataset	Sub-Net	SNN	SNN-SOL	Gain % ↑
CIFAR-10	MobileNetV2$_{2/16}$	87.17	89.59	**2.42**
	WRN-16-8$_{2/16}$	87.62	89.20	**1.58**
CIFAR-100	MobileNetV2$_{8/8}$	73.41	74.38	**0.97**
	ResNet-34$_{8/8}$	77.29	78.86	**1.57**
TinyImageNet	WRN-16-8$_{5/8}$	62.66	63.84	**1.18**
	ResNet-34$_{5/8}$	62.34	64.53	**2.19**

6. Conclusion and Future Perspectives

We introduce **Switchable Output Layer (SOL)** to enhance the performance and robustness of Once-for-All (OFA) training frameworks. SOL assigns independent classification heads to the sub-nets in a super-net backbone, which decouple their logit learning processes, mitigating the competition at the *"shared output layer"*—a bottleneck limiting the sub-net accuracy, robustness, and optimization. Extensive experiments using two baseline methods: Once-for-All Adversarial Training and Slimming (OATS) and Slimmable Neural Networks (SNNs), across multiple datasets and diverse super-net architectures, demonstrate that incorporation of SOL consistently improves performance of sub-nets without introducing additional training overhead or complexity. SOL generalizes well, which highlights its potential as an effective enhancement for the OFA frameworks, encouraging flexible, scalable, and reliable deployment of specialized models across a wider range of devices and constraints. For future, we aim to extend SOL to more OFA frameworks (e.g. [5], [44], [49]) and conduct large-scale evaluations on the ImageNet-1K dataset. We also intend to study the impact of layer normalization on reducing representational interference across the sub-nets.

Acknowledgment

This work was supported by the U.S. National Science Foundation (NSF) under the Grant #2245055.

References

[1] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021. 3

[2] Nils Bjorck, Carla P Gomes, Bart Selman, and Kilian Q Weinberger. Understanding batch normalization. *Advances in neural information processing systems*, 31, 2018. 3

[3] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-All: Train One Network and Specialize it for Efficient Deployment, 2020. arXiv:1908.09791 [cs, stat]. 1, 2, 4, 5

[4] Qi-Zhi Cai, Min Du, Chang Liu, and Dawn Song. Curriculum adversarial training. *arXiv preprint arXiv:1805.04807*, 2018. 3

[5] Yun-Hao Cao, Peiqin Sun, and Shuchang Zhou. Three guidelines you should know for universally slimmable self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15742–15751, 2023. 8

[6] Nicholas Carlini and David Wagner. Towards Evaluating the Robustness of Neural Networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, San Jose, CA, USA, 2017. IEEE. 1

[7] Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 699–708, 2020. 1, 3

[8] 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. *Advances in Neural Information Processing Systems*, 34:19637–19651, 2021. 1, 2, 4, 5

[9] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 2, 5

[10] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018. 5

[11] Gavin Weiguang Ding, Kry Yik Chau Lui, Xiaomeng Jin, Luyu Wang, and Ruitong Huang. On the Sensitivity of Adversarial Robustness to Input Data Distributions, 2019. arXiv:1902.08336 [cs, stat]. 3

[12] Angus Galloway, Anna Golubeva, Thomas Tanay, Medhat Moussa, and Graham W. Taylor. Batch Normalization is a Cause of Adversarial Vulnerability, 2019. arXiv:1905.02161 [cs, stat]. 2, 3

[13] Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 3996–4003, 2020. 5

[14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples, 2015. arXiv:1412.6572 [cs, stat]. 1, 3

[15] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021. 5

[16] 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*, pages 770–778, 2016. 2, 5

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5

[18] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens Van Der Maaten, and Kilian Q Weinberger. Multi-scale dense networks for resource efficient image classification. *arXiv preprint arXiv:1703.09844*, 2017. 4

[19] Tianjin Huang, Vlado Menkovski, Yulong Pei, and Mykola Pechenizkiy. Bridging the performance gap between fgsm and pgd adversarial training. *arXiv preprint arXiv:2011.05157*, 2020. 3

[20] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 2, 3

[21] Yigitcan Kaya, Sanghyun Hong, and Tudor Dumitras. Shallow-deep networks: Understanding and mitigating network overthinking. In *International conference on machine learning*, pages 3301–3310. PMLR, 2019. 4

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. 2, 5

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 5

[24] Souvik Kundu, Sairam Sundaresan, Massoud Pedram, and Peter A. Beerel. FLOAT: Fast Learnable Once-for-All Adversarial Training for Tunable Trade-off between Accuracy and Robustness. In *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2348–2357, Waikoloa, HI, USA, 2023. IEEE. 1, 3, 4

[25] Souvik Kundu, Sairam Sundaresan, Sharath Nittur Sridhar, Shunlin Lu, Han Tang, and Peter A Beerel. Sparse mixture once-for-all adversarial training for efficient in-situ trade-off between accuracy and robustness of dnns. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 1, 3, 4, 5

[26] Yann Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 2, 5

[27] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. 5

[28] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards Deep Learning Models Resistant to Adversarial Attacks, 2019. arXiv:1706.06083 [cs, stat]. 1, 3

[29] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, Andrew Y Ng, et al. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, page 4. Granada, 2011. 2, 5

[30] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, Boston, MA, USA, 2015. IEEE. 1

[31] Jianhong Pan, Siyuan Yang, Lin Geng Foo, Qiuhong Ke, Hossein Rahmani, Zhipeng Fan, and Jun Liu. Progressive channel-shrinking network. *IEEE Transactions on Multimedia*, 26:2016–2026, 2023. 2

[32] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks, 2016. arXiv:1511.04508 [cs, stat]. 5

[33] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 3

[34] 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*, pages 4510–4520, 2018. 2, 5

[35] Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in neural information processing systems*, 32, 2019. 3

[36] Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15731–15740, 2024. 5

[37] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 3

[38] Haotao Wang, Tianlong Chen, Shupeng Gui, Ting-Kuei Hu, Ji Liu, and Zhangyang Wang. Once-for-All Adversarial Training: In-Situ Tradeoff between Robustness and Accuracy for Free, 2020. arXiv:2010.11828 [cs]. 1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14

[39] Haotao Wang, Aston Zhang, Shuai Zheng, Xingjian Shi, Mu Li, and Zhangyang Wang. Removing batch normalization boosts adversarial training. In *International Conference on Machine Learning*, pages 23433–23445. PMLR, 2022. 3

[40] Xin Wang, Fisher Yu, Zi-Yi Dou, Trevor Darrell, and Joseph E Gonzalez. Skipnet: Learning dynamic routing in convolutional networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 409–424, 2018. 4

[41] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International conference on learning representations*, 2019. 3

[42] Cihang Xie and Alan Yuille. Intriguing properties of adversarial training at scale, 2019. arXiv:1906.03787 [cs]. 3

[43] Ahmeed Yinusa and Misagh Faezipour. Evaluating artificial intelligence robustness against fgsm and pgd adversarial attacks with l-norms perturbations. In *World Congress in Computer Science, Computer Engineering & Applied Computing*, pages 315–328. Springer, 2024. 3

[44] Jiahui Yu and Thomas S Huang. Universally slimmable networks and improved training techniques. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1803–1811, 2019. 1, 4, 5, 8

[45] Jiahui Yu, Linjie Yang, Ning Xu, Jianchao Yang, and Thomas Huang. Slimmable neural networks. *arXiv preprint arXiv:1812.08928*, 2018. 1, 2, 4, 5, 7, 8

[46] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 2, 5

[47] Bohang Zhang, Tianle Cai, Zhou Lu, Di He, and Liwei Wang. Towards Certifying L-infinity Robustness using Neural Networks with L-inf-dist Neurons, 2021. arXiv:2102.05363 [cs, stat]. 1

[48] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically Principled Trade-off between Robustness and Accuracy, 2019. arXiv:1901.08573 [cs, stat]. 1, 3, 5

[49] Shuai Zhao, Linchao Zhu, Xiaohan Wang, and Yi Yang. Slimmable networks for contrastive self-supervised learning. *International Journal of Computer Vision*, 133(3):1222–1237, 2025. 8

[50] Weimin Zhao, Sanaa Alwidian, and Qusay H Mahmoud. Adversarial training methods for deep learning: A systematic review. *Algorithms*, 15(8):283, 2022. 3

APPENDIX

Comparison of OATS and OATS-SOL on CIFAR-10 Dataset (8 Sub-Nets)

This section provides the numerical results of comparison between OATS [38] and OATS-SOL for ResNet-34 backbone, packed with 8 sub-nets, on CIFAR-10 dataset (as per Figure 5). Table 6 presents a quantitative comparison of accuracy and robustness between the sub-nets of OATS [38] and OATS-SOL for 25%, 50%, 75%, and 100% widths. For the sub-nets with these widths, the maximum improvement of **4.71%**, **3.26%**, **1.92%**, **1.66%** in accuracy is observed for $\lambda = 0.0$, respectively. Additionally, OATS-SOL provides robustness gains of up to **4.84**, **7.32%**, **7.71%**, and **6.80%** across these sub-nets, respectively, for multiple values of λ. It is evident that OATS-SOL consistently outperforms OATS. The performance gap is generally more pronounced for smaller sub-nets and gradually narrows as the sub-net size increases.

Table 6. Accuracy and Robustness of OATS [38] vs. OATS-SOL on CIFAR-10 using ResNet-34 backbone with 8 sub-nets.

λ	Accuracy OATS [38]	Accuracy OATS-SOL	Gain (%) ↑	Robustness OATS [38]	Robustness OATS-SOL	Gain (%) ↑
Sub-Net 2/8 ; Width = 25 %						
0.0	83.12	87.83	**4.71**	0.55	0.91	**0.36**
0.1	80.71	81.43	**0.72**	25.51	30.35	**4.84**
0.2	79.77	80.47	**0.70**	29.74	34.18	**4.44**
0.3	79.05	79.82	**0.77**	31.80	36.08	**4.28**
0.4	78.19	79.17	**0.98**	33.44	37.54	**4.10**
1.0	73.90	75.23	**1.33**	37.81	41.50	**3.69**
Sub-Net 4/8 ; Width = 50 %						
0.0	87.94	91.20	**3.26**	0.89	2.96	**2.07**
0.1	85.90	85.95	**0.05**	33.10	40.42	**7.32**
0.2	85.10	85.34	**0.24**	36.68	42.71	**6.03**
0.3	84.48	84.90	**0.42**	38.72	43.64	**4.92**
0.4	83.79	84.48	**0.69**	40.00	44.57	**4.57**
1.0	80.12	81.65	**1.53**	44.58	47.48	**2.90**
Sub-Net 6/8 ; Width = 75 %						
0.0	89.96	91.88	**1.92**	1.31	4.7	**3.39**
0.1	87.65	87.46	**-0.19**	36.23	43.94	**7.71**
0.2	86.82	86.94	**0.12**	39.65	45.68	**6.03**
0.3	86.24	86.50	**0.26**	41.38	46.65	**5.27**
0.4	85.60	86.13	**0.53**	42.89	47.20	**4.31**
1.0	82.66	84.20	**1.54**	46.85	49.04	**2.19**
Sub-Net 8/8 ; Width = 100 %						
0.0	90.24	91.90	**1.66**	0.87	5.14	**4.27**
0.1	88.13	87.76	**-0.37**	37.18	43.98	**6.80**
0.2	87.38	87.08	**-0.30**	40.44	45.91	**5.47**
0.3	86.71	86.63	**-0.08**	42.27	46.99	**4.72**
0.4	86.14	86.20	**0.06**	43.63	47.44	**3.81**
1.0	83.27	84.34	**1.07**	47.48	49.47	**1.99**

Comparison of OATS and OATS-SOL w.r.t λ on SVHN Dataset

This section provides the plots for comparison of OATS [38] and OATS-SOL from WideResNet-16-8 backbone (packed with 8 sub-nets). The results are shown in Figure 8 and Figure 9 from a different perspective, focusing accuracy and robustness w.r.t λ values for SVHN dataset.

Comparison of OATS and OATS-SOL w.r.t λ on CIFAR-10 Dataset

This section provides the plots for comparison of OATS [38] and OATS-SOL from ResNet-34 backbone (packed with 8 sub-nets). The results are shown from a different perspective in Figure 10 and Figure 11, focusing accuracy and robustness w.r.t λ values for CIFAR-10 dataset.

Comparison of OATS and OATS-SOL w.r.t λ on STL-10 Dataset

This section provides the plots for comparison of OATS and OATS-SOL from WideResNet-40-2 backbone (packed with 4 sub-nets). In Figure 12 and Figure 13, the results are shown for STL-10 dataset from a different perspective for accuracy and robustness w.r.t λ values.

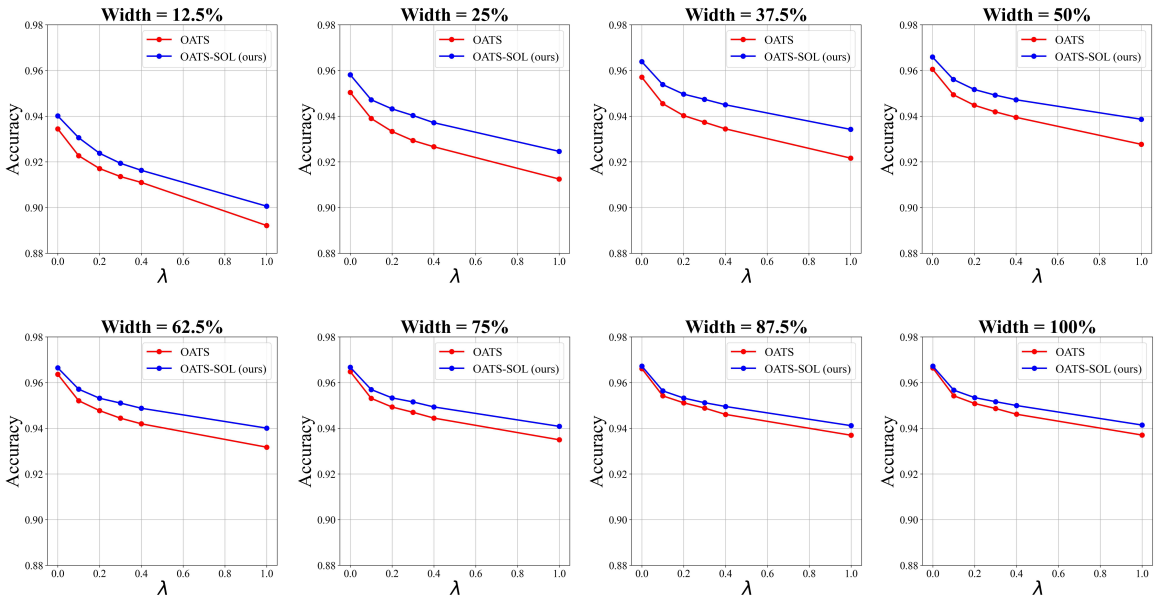

Figure 8. Comparison of OATS [38] and OATS-SOL on SVHN dataset using WideResNet-16-8 backbone (packed with 8 sub-nets) from the perspective of accuracy w.r.t λ.

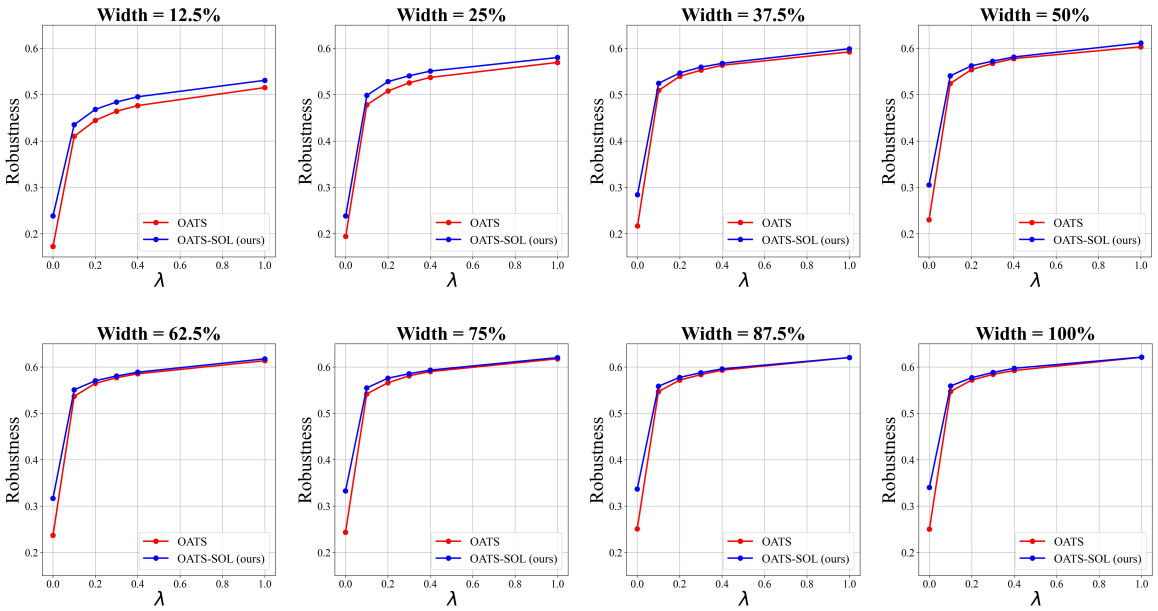

Figure 9. Comparison of OATS [38] and OATS-SOL on SVHN dataset using WideResNet-16-8 backbone (packed with 8 sub-nets) from the perspective of PGD-7 robustness w.r.t λ.

Extraction of Sub-Networks

After training, the sub-nets of various sizes and performance levels can be extracted by copying the corresponding parameters from the backbone into newly instantiated, size-matched architectures. For OATS-based sub-nets, the accuracy–robustness trade-off can be adjusted at inference time by varying the value of $\boldsymbol{\lambda}$. However, meeting strict deployment constraints often necessitates smaller sub-nets, which typically involves sacrificing both accuracy and robustness for model-size. In contrast, for SNN-based sub-nets, the trade-off is only between model-size and accuracy: smaller sub-nets offer reduced memory and storage footprints but at the cost of lower performance.

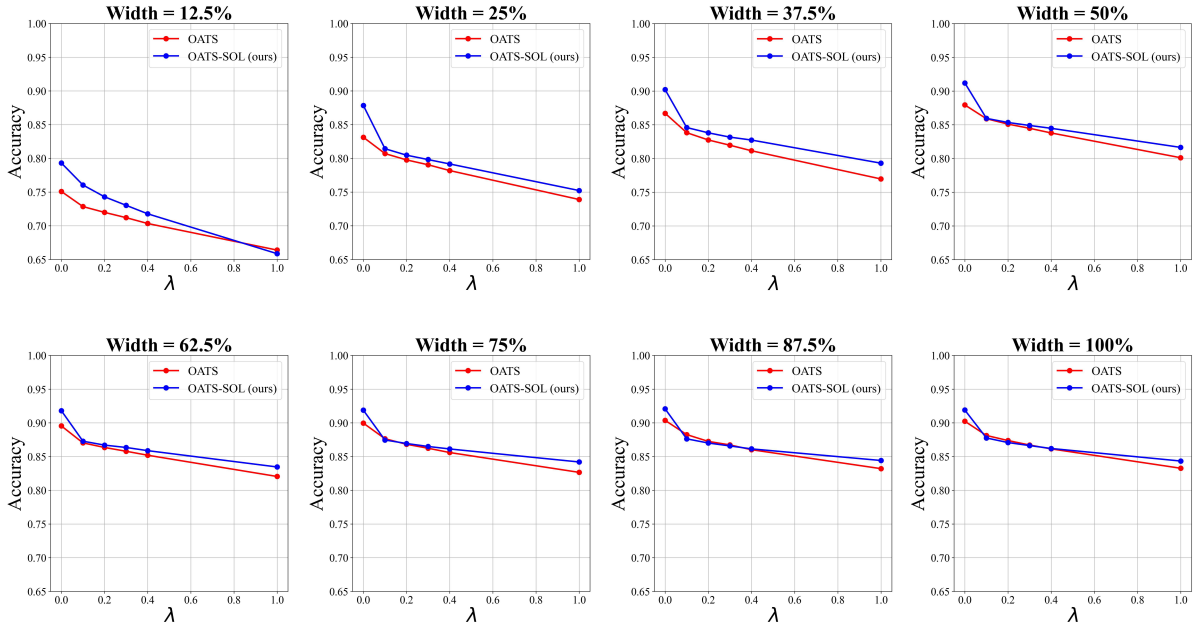

Figure 10. Comparison of OATS [38] and OATS-SOL on CIFAR-10 dataset using ResNet-34 backbone (packed with 8 sub-nets) from the perspective of accuracy w.r.t λ.

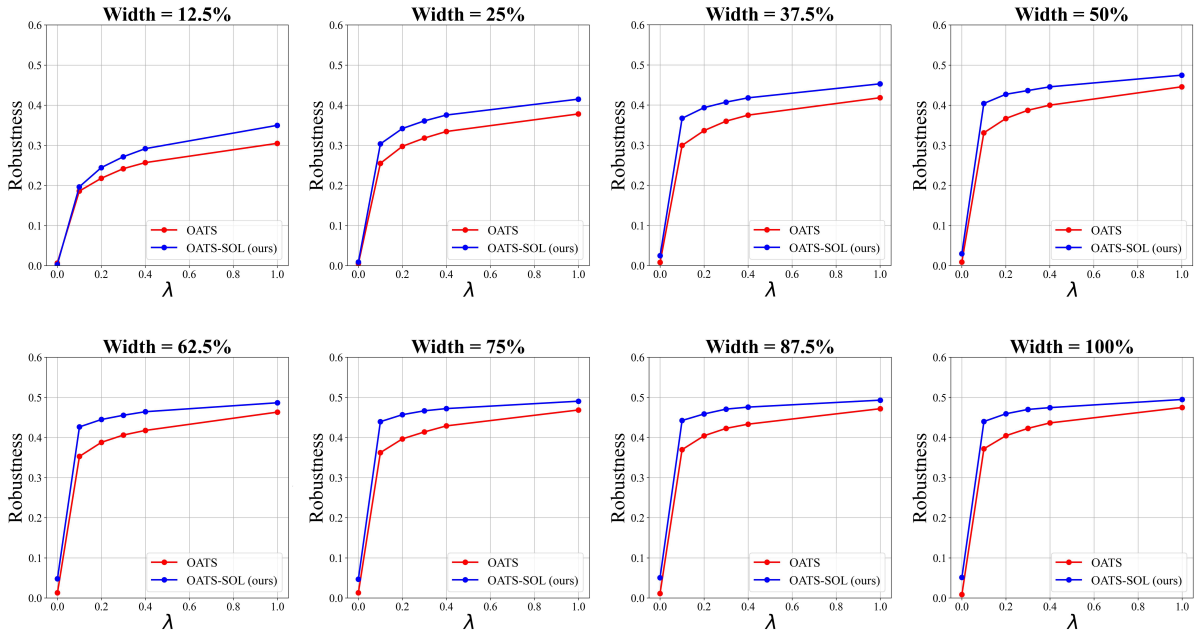

Figure 11. Comparison of OATS [38] and OATS-SOL on CIFAR-10 dataset using ResNet-34 backbone (packed with 8 sub-nets) from the perspective of PGD-7 robustness w.r.t λ.

Limitations

SOL increases the overall memory footprint due to the added output heads, which may be impractical for extremely large sub-net ensembles. Conversely, for super-nets with only a few sub-nets (e.g. two or three), its benefits are limited. Moreover, integrating SOL into pretrained OFA models requires architectural changes, which may hinder the direct reuse of the pretrained output layer weights.

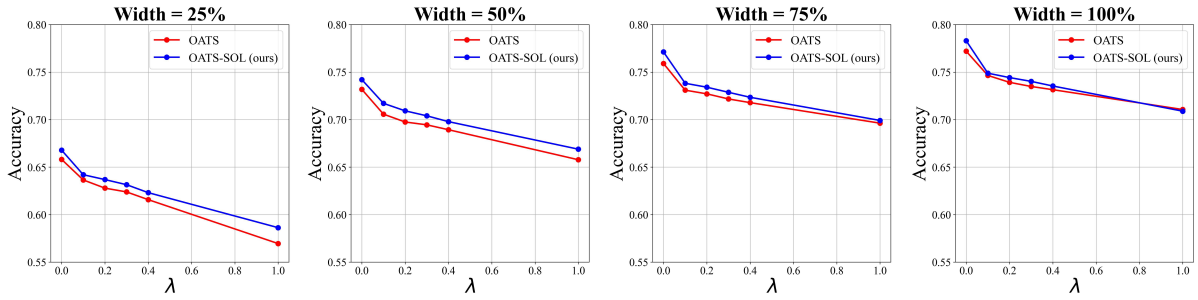

Figure 12. Comparison of OATS [38] and OATS-SOL on STL-10 dataset using WideResNet-40-2 backbone (packed with 4 sub-nets) from the perspective of accuracy w.r.t λ.

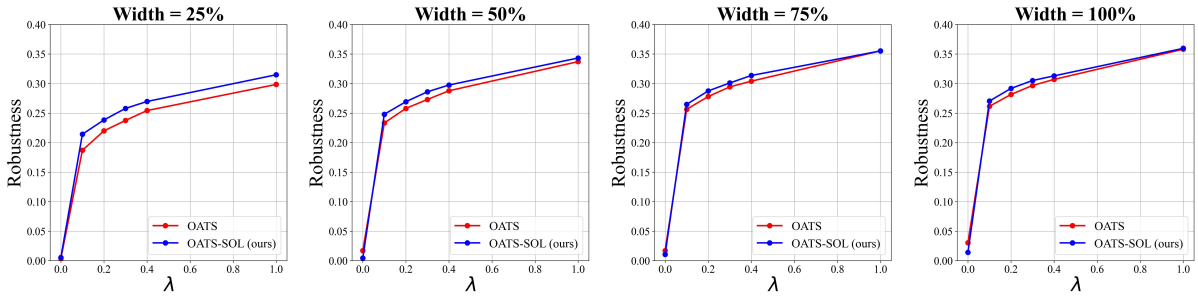

Figure 13. Comparison of OATS [38] and OATS-SOL on STL-10 dataset using WideResNet-40-2 backbone (packed with 4 sub-nets) from the perspective of PGD-7 robustness w.r.t λ.