Track 1:
Sparse Transfer Learning Accelerates and Enhances Certified Robustness: A Comprehensive Study

Anonymous Author(s)
Affiliation
Address
email

Abstract

Certified robustness is a critical measure for assessing the reliability of machine learning systems. Traditionally, the computational burden associated with certifying the robustness of machine learning models has posed a substantial challenge, particularly with the continuous expansion of model sizes. In this paper, we introduce an innovative approach to expedite the verification process for L_2-norm certified robustness through sparse transfer learning. Our approach is both efficient and effective. It leverages verification results obtained from pre-training tasks and applies sparse updates to these results. To enhance performance, we incorporate dynamic sparse mask selection and introduce a novel stability-based regularizer called DiffStab. Empirical results demonstrate that our method accelerates the verification process for downstream tasks by as much as **70-80%**, with only slight reductions in certified accuracy compared to dense parameter updates. We further validate that this performance improvement is even more pronounced in the few-shot transfer learning scenario.

1 Introduction

In recent years, ensuring the certified robustness of machine learning systems has emerged as a paramount research challenge. The primary objective is to guarantee consistent and resilient output predictions, impervious to perturbations spanning a defined range in any direction. Diverse verification techniques have been devised to quantify the certified robustness of neural networks. When confronting inputs perturbed within some L_{inf}-norm bound, the prevailing verification methods center around the branch-and-bound (BaB) technique [20, 16, 18]. In cases involving L_2-norm perturbations, randomized-smoothing approaches reign supreme [4, 10].

However, it is a widely recognized challenge that commonly used certified verification methods, such as the BaB methods [20, 18] and randomized smoothing [4] grapple with the inherent issue of computationally expensive verification for each sample. In fact, the computational cost of verification often surpasses that of inference for the same sample by *several orders of magnitude*. For instance, the BaB method exhibits exponential complexity, while randomized smoothing typically demands the sampling of approximately 1,000 noisy inputs for each individual verification. This predicament of resource-intensive verification is further exacerbated by the exponential growth in the sizes of state-of-the-art (SOTA) models across various benchmarks.

In this paper, we concentrate on developing novel streamlined techniques designed to expedite the verification processes based on randomized smoothing for L_2-norm certified robustness. Our approach begins by identifying ways to **efficiently reuse** the verification results from pre-training

Submitted to AdvML-Frontiers'24: The 3nd Workshop on New Frontiers in Adversarial Machine Learning@NeurIPS'24, Vancouver, CA. Do not distribute.

tasks to downstream tasks, and our innovation here is to introduce the tool of **sparse transfer learning** to update only a select subset of network parameters during the transfer. We then implement our novel *differential sparse verification* techniques to accelerate the verification process by leveraging specific patterns of sparsity. This is chiefly accomplished by hastening the forward propagation of noisy samples from the Monte-Carlo sampling of randomized smoothing-based verification, using (structured) sparse update vector multiplication. We further introduce two techniques to augment the certified robustness for sparse transfer learning, namely dynamic sparse mask selection and a novel stability-based regularizer. They result in significant enhancements in both the speed of the verification process and the robustness of the certified outcomes, when compared to the conventional approach of direct training and verification on downstream tasks.

Specifically, our contributions are outlined as follows:

- We for this first time investigate the use of sparse transfer learning to expedite the certified verification process, capitalizing on reusing the verification results from the upstream task and executing sparse weight updates. Specifically, we employ sparse transfer learning with three distinct sparsity patterns, thereby facilitating efficient transfer and accelerating the downstream verification process. This is achieved by propagating the intermediate verification results using the sparse convolutional operator.
- Recognizing that sparse transfer learning may affect the certified robustness of transferred models, we further propose to boost this process using dynamic mask selection and a novel stability-based regularizer. These measures significantly narrow the performance gap with the upper bound achieved by dense parameter updates.
- We empirically discover that our approach can hasten the verification process on downstream tasks by up to 70-80%, with only slight reductions in certified accuracy compared to dense parameter update. Furthermore, we find that the advantages of sparse transfer learning and acceleration can be further amplified in the context of few-shot transfer learning.

2 Related Work

2.1 L_2-norm Certified Robustness

Research into L_2-norm certified robustness aims to ensure stable machine learning system outputs when input perturbations lie within an L_2-norm ball. Cohen et al. [4] pioneered a verification method for L_2-norm certified robustness using randomized smoothing based on Monte-Carlo sampling [4]. Kumar and Goldstein [10] introduced a variant that facilitates L_2-robust training and verification for tasks with structured outputs, such as semantic segmentation [10]. An alternative approach verifies L_2-norm robustness by leveraging the diffusion model to denoise the input before making predictions with benign-trained models [1, 19]. However, this diffusion model-based method introduces an added denoising step during inference. This paper centers on accelerating the prevalent randomized smoothing techniques for L_2-norm certified robustness.

2.2 Transfer learning

Transfer learning facilitates knowledge transfer from a source to a target domain, particularly when data in the target domain is scarce. Many approaches adopt a pretrain-and-finetune framework, varying primarily in their pre-training objectives. This includes contrastive pre-training [3, 11], pretext tasks [7], and autoencoding [5]. Recently, Guo et al. [9] introduced DiffPruning, a parameter-efficient method that updates only a sparse subset of model parameters for each downstream task. In this paper, we harness DiffPruning for both sparse transfer learning and sparse differential verification.

2.3 Robustness Transfering

Recent studies have highlighted the robustness of neural networks pre-trained on large-scale datasets. Such networks tend to possess robust feature extractors that can be transferred to downstream tasks [4, 15, 10, 12]. Salman et al. [14] discovered that adversarially trained networks can enhance accuracy in these downstream tasks. Furthermore, Vaishnavi et al. [17] introduced a method using knowledge transfer to expedite the training for certified robustness. Nevertheless, to our knowledge, our work is pioneering in its approach to accelerate the certified robustness verification process via sparse transfer learning.

3 Methodology

We introduce the preliminaries of this work, including sparsity patterns and dynamic mask selection with RigL in appendix A. Certified verification techniques, such as randomized smoothing, grapple with substantial computational overhead—often eclipsing the inference time for the same sample. For the L_2-norm, this cost intensifies with randomized smoothing methods [4, 15, 10], which use Monte-Carlo sampling to certify each original sample by sampling multiple noisy inputs. In this section, we explore how sparse transfer learning can bolster the efficiency of the randomized smoothing-based verification and enhance the certified robustness for downstream tasks. We also discuss dynamic mask selection and our novel **stability regularizer**, both tailored to amplify certified robustness. The architecture of our framework is illustrated in fig. 1.

3.1 Sparse Transfer Learning for Certified Robustness

Sparse transfer learning involves updating a selected subset of parameters in pre-trained models during transfer. We pinpoint two benefits of employing sparse transfer learning for L_2-norm certified robustness: ❶ Transfer learning typically yields superior certified robustness in downstream tasks than training exclusively on those tasks. This is attributed to the foundational robustness instilled during the pre-training phase [15, 12]. ❷ Sparse transfer learning facilitates the acceleration of the randomized smoothing verification process across various sparsity patterns. We expedite the randomized smoothing certification by leveraging efficient Monte-Carlo sampling inference, informed by sparse update vectors derived from sparse transfer learning.

We further explore the potential of sparse transfer learning to expedite the verification process by examining various sparsity patterns. In our approach, we integrate DiffPruning, as presented in Guo et al. [9], with different types of sparsity: unstructured, structured (channel-wise), and group-wise, as detailed in Sec. A.1. This amalgamation allows for sparse transfer learning. Furthermore, we discuss the method of capitalizing on the certified verification outcomes from pre-training tasks. By employing the sparse update masks corresponding to the various sparsity patterns, we aim to speed up the verification procedure for the transferred tasks. It's crucial to mention that in order to benefit from the verification results of the pre-training tasks, consistency in input between pre-training and downstream tasks is imperative

In methods rooted in randomized smoothing, the predominant computational demand during the verification process stems from Monte-Carlo sampling. For each sample undergoing verification, it is commonplace for these techniques to draw upon 1,000 noisy inputs, forecast outcomes, and then gauge the verification conclusion from these forecasts. This underscores that bolstering the speed of forward propagation for each prediction can, in turn, hasten the overarching verification procedure. In this study, we venture to enhance the pace of the forward propagation. We achieve this by integrating the differential outcome of forward propagation, brought about by sparse update vectors, with the dense verification results inherited from pre-training tasks, as depicted in Fig. 1.

Subsequently, we materialize this acceleration across diverse sparsity patterns:

- **Unstructured sparsity** In CNNs, the convolutional operation is the primary source of computational complexity during inference. To achieve acceleration, our focus is on optimizing this convolutional operation. It's well-understood that convolutional operations can be translated into matrix multiplications. Therefore, the differential forward propagation can be conducted using a matrix multiplication between the dense input matrix and the sparse parameter update matrix for each convolutional layer. This process is further expedited using a sparse **coordinate list** operator tailored for matrix computations.

  In the forward propagation, for each layer, we combine this differential output with the results previously obtained from the pre-training task. This integrated result is then propagated to the subsequent layer. By iteratively integrating the outcome of this sparse forward propagation for each convolutional layer, we can efficiently compute the output of the final convolutional layer.

- **Structured sparsity** The acceleration process for structured sparsity is notably straightforward. The sparse update vector in this context is channel-wise, functioning as a binary indicator for every layer. When this indicator has a value of 1, it signifies that the parameters of the related channel have undergone updates. We compute the differential output exclusively for the updated channels. Then, for each layer, we combine this differential output with the results derived from the pre-training task. The aggregated result is subsequently propagated to the following layer.
- **Group-wise sparsity** Group-wise sparsity can be conceptualized as an amalgamation of both unstructured and structured sparsity. Given that certain channels without selected dense blocks are omitted, we ultimately achieve a sparsity mask with an unstructured configuration. Consequently, we can employ a combined acceleration strategy, drawing from the methods above used for both structured and unstructured sparsity.

We empirically find that, for a given sparse ratio k, structured sparsity typically yields a more pronounced acceleration compared to unstructured sparsity, with group-wise sparsity falling somewhere in the middle. In contrast, when evaluating certifiable robustness (specifically, verified accuracy), the performance trend for each sparsity pattern is opposite to their respective acceleration effects. Notably, group-wise sparsity strikes a more balanced compromise between acceleration and certifiable robustness in comparison to the other two sparsity paradigms.

3.2 Regularizing Sparse Transfer Learning

As mentioned above, the proposed method with sparse transfer learning can help the model achieve better-verified accuracy than training directly on downstream tasks, but not as good as dense transfer learning, where we update all the parameters while transferring. We identify 2 reasons for this phenomenon: firstly, we originally expected that sparse transfer learning is possible to achieve better certified robustness than dense transfer learning since the network is already pre-trained to be robust and has stable intermediate outputs for its layers given the same input. However, we failed to observe this phenomenon and conclude that unconstrained sparse transfer learning is unable to preserve the robustness obtained from the pre-training task. Secondly, we believe that the domain gap between the pre-training task and the downstream task prevents sparse transfer learning to achieve better robustness than dense transfer learning.

To tackle these two challenges, we introduce a dual-method approach: **Firstly**, We advocate for a regularizer based on stability, which specifically targets the L_2 distance of the lower and upper bounds for each neuron. By ensuring these bounds remain consistent between the pre-training and downstream tasks—given identical input and perturbation ranges—we aim to maintain the inherent stability and robustness from the pre-training phase. To this end, we employ Interval Bound Propagation (IBP) [8]. The L_{inf}-norm bounds provided by IBP are not only efficient but also align with the computational complexity of a network's forward pass. It's worth highlighting that even though the L_{inf}-norm bound is distinct from the L_2-norm bound, our empirical findings suggest that the former is effective in regularizing L_2-norm robustness. Formally defined, if the lower and upper bounds of neuron i for the pre-training task are denoted by lb_i and ub_i respectively, and for the downstream task they are lb'_i and ub'_i, the regularization loss is computed as follows:

$$loss_{stab} = \frac{1}{N}\sum_{i}^{N}(lb'_i - lb_i)^2 + (ub'_i - ub_i)^2 \tag{1}$$

Where N is the number of neurons across the network. We call this regularizer as *DiffStab*. And the overall loss for transfer learning is:

$$loss = loss_{orig} + loss_{stab} \tag{2}$$

Where $loss_{orig}$ is the original loss of transfer learning. **Secondly**, to mitigate the domain gap challenge, we advocate for dynamic mask selection. Specifically, we implement the RigL approach as outlined in [6]. This method ensures enhanced mask flexibility during the transfer phase. Our empirical analysis confirms that dynamic mask selection markedly boosts certified robustness in sparse transfer learning, especially when confronted with a substantial domain gap.

Table 1: The comparison of verified accuracy and verification time of different transfer setting with different sparsity patterns.

		Direct Train	Dense Transfer	Sparse Transfer						
Sparsity Pattern	Updated Params(%)	100	100	1	2	4	8	16	32	64
Unstruct	Ver Acc(%)	60.4	61.2	55.1	56.2	57.9	59.1	60.3	60.6	61.1
	Time Saved(%)	0	0	77.6	63.5	46.6	29.1	10.2	3.2	1.2
Struct	Ver Acc(%)	60.4	61.2	53.4	54.7	56.2	57.6	58.8	59.7	60.3
	Time Saved(%)	0	0	92.1	88.5	82.2	72.8	55.4	33.1	15.8
Group-wise	Ver Acc(%)	60.4	61.2	54.0	55.3	56.8	57.8	59.1	59.8	60.5
	Time Saved(%)	0	0	87.2	81.8	75.2	61.9	49.3	28.7	12.5

4 Experiments

In this section, our objective is to address two primary inquiries via comprehensive experiments: (1) How effectively does the proposed method hasten the certified verification process and amplify the certified robustness for a downstream task under L_2-norm input perturbations? (2) How do DiffStab and RigL contribute to enhancing the certified robustness performance in the context of our proposed sparse transfer learning and verification methodology?

To address the posed questions, we carry out experiments in two distinct settings across two datasets, including CIFAR10 and CelebV-HQ. The details about experiment settings are highlighted in appendix B.

4.1 Sparse Transfer Learning Accelerates and Enhances Certified Robustness

Table 2: The comparison of verified accuracies before and after adding DiffStab regularizer and RigL(dynamic mask selection) of different sparsity patterns. The relative improvements in the brackets are obtained by comparing them with the baselines of different sparsities.

		Direct Train	Dense Transfer	Sparse Transfer						
Sparsity Pattern	UpdateParams	100	100	1	2	4	8	16	32	64
Unstruct	Ver Acc(%)	60.4	61.2	55.1	56.2	57.9	59.1	60.3	60.6	61.1
+DiffStab+RigL	Ver Acc(%)	60.4	62.2 (+1.0)	58.1 (+3.0)	59.0 (+2.8)	60.6 (+2.7)	61.2 (+2.1)	61.5 (+1.2)	62.2 (+1.6)	62.2 (+1.1)
Struct	Ver Acc(%)	60.4	61.2	53.4	54.7	56.2	57.6	58.8	59.7	60.3
+DiffStab+RigL	Ver Acc(%)	60.4	62.2 (+1.0)	57.0 (+3.6)	58.7 (+4.0)	59.4 (+3.2)	60.6 (+3.0)	60.9 (+2.1)	61.4 (+1.7)	61.6 (+1.3)
Group-wise	Ver Acc(%)	60.4	61.2	54.0	55.3	56.8	57.8	59.1	59.8	60.5
+DiffStab+RigL	Ver Acc(%)	60.4	62.2 (+1.0)	57.0 (+3.0)	58.6 (+3.3)	59.3 (+2.5)	60.3 (+2.5)	60.7 (+1.6)	61.2 (+1.4)	61.7 (+1.2)

4.1.1 CIFAR10 Results

In this subsection, we assess the effectiveness of combining sparse transfer learning with sparse differential verification to expedite the randomized smoothing-based verification process. Notably, when given an ample amount of pre-training data, sparse transfer learning not only facilitates faster performance but also achieves superior results.

To corroborate the acceleration effect, we implemented sparse transfer learning on the CIFAR10 dataset at predetermined sparsity ratios. We juxtaposed the outcomes from sparse differential verification with those obtained using the standard randomized smoothing. The results of this comparison are delineated in table 1. Although similar acceleration findings were empirically noted on the CelebV-HQ dataset (owing to the consistent network architecture and a dominant influence of sparsity ratio over input or network configuration), for the sake of brevity, we've confined our exposition to the CIFAR10 dataset. As evident from table 1, as the sparsity ratio increases, sparse differential verification can hasten the verification process by a staggering 77.6% to 92.1%. However, a trade-off is observed in the form of a reduced verified accuracy. While we employed contrastive learning for pre-training, aiming to harness robust self-supervision signals, the scale of the dataset remains a constraint, limiting the significant benefits of pre-training for subsequent tasks. In subsequent sections, we will discuss how augmenting the pre-training dataset size can alleviate this challenge. Additionally, by incorporating our novel DiffStab regularizer and dynamic mask selection, we demonstrate that performance can be further enhanced.

When we examine various sparsity patterns presented in table 1, it's evident that the acceleration effects increase in the order of unstructured, group-wise, and structured sparse differential verifications. This progression aligns with our expectations. Structured sparsity directly omits entire channels from the verification process, while group-wise sparsity can be perceived as an amalgamation of both unstructured and structured sparsity, as outlined in our methodology.

However, when looking at verified accuracies, they tend to decrease in the order from unstructured to structured sparsity. This outcome is plausible since unstructured sparsity employs the most adaptive sparsity masks. This observation parallels findings in the model compression domain where unstructured pruning often surpasses structured pruning in terms of subnetwork performance.

Upon Comparing group-wise sparsity with the other two types, it becomes clear that group-wise sparsity aligns more closely with unstructured sparsity in terms of verified accuracy, while resembling structured sparsity in acceleration outcomes. Therefore, we can infer that group-wise sparsity strikes an optimal balance, presenting a commendable trade-off between performance and acceleration, particularly when certifying robustness.

To demonstrate the broad applicability of our method across various network architectures, we further evaluated its acceleration performance on both ResNet-18 and VGG-16. The results are presented in table 3 in the Appendix.

4.1.2 CelebV-HQ Results

For the CelebV-HQ dataset, we commenced with analogous experiments involving unstructured sparsity, both under a standard transfer setting utilizing 100% of the downstream data and a few-shot transfer setting with just 1% of downstream data. For detailed results, see the 1st and 5th rows in table 4.

By comparing these outcomes with those in table 1, it becomes evident that the expansive scale of the pre-training dataset in CelebV-HQ markedly bolsters the certified robustness achieved through sparse transfer learning. Let's remember that for our pre-training, we utilized 40 attributes, while only 1 attribute was used for each downstream task. Notably, even when a mere 8% of network parameters are updated during sparse transfer learning, the enhanced network showcases a performance that's on par with direct training that involves dense parameter updates. This comes with the added advantage of a 29.1% acceleration for unstructured sparsity.

The advantages of sparse transfer learning become even more pronounced in a few-shot transfer learning environment. Here, sparse transfer learning significantly outperforms direct training. This can be attributed to the fact that the extensive multi-attribute classification pre-training infuses the network with substantial robustness. In contrast, direct training is limited by its access to a smaller dataset, curtailing its robust training capabilities. Interestingly, the performance disparity between dense transfer learning and sparse transfer learning narrows in the few-shot setting. This can be explained by the limited data available for finetuning in the few-shot scenario. Consequently, the performance is less adversely impacted by the 'lazy' update strategy, that is, the sparse parameter update. More study is provided in appendix C and appendix D.

5 Conclusion

In this paper, we introduce sparse differential verification to accelerate the L_2-norm robustness verification process based on randomized smoothing. Building on sparse differential forward propagation, our approach hastens the Monte-Carlo Sampling inherent to randomized smoothing. We explore three sparsity patterns for transfer learning, discussing their pros and cons. To bridge the gap between dense and sparse transferring, we employ dynamic mask selection and our new DiffStab regularizer. Empirically, our method achieves up to 80% acceleration while maintaining verified accuracies comparable to dense transfer methods. One constraint is the need for consistent input between pre-training and downstream tasks, limiting our model's breadth. Still, our work offers a promising step towards leveraging transfer learning for faster, reliable machine learning verification.

References

[1] Nicholas Carlini, Florian Tramer, J Zico Kolter, et al. (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*, 2022.

[2] Tianlong Chen, Xuxi Chen, Xiaolong Ma, Yanzhi Wang, and Zhangyang Wang. Coarsening the granularity: Towards structurally sparse lottery tickets. *arXiv preprint arXiv:2202.04736*, 2022.

[3] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[4] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[5] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28, 2015.

[6] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *International Conference on Machine Learning*, pages 2943–2952. PMLR, 2020.

[7] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

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

[9] Demi Guo, Alexander M Rush, and Yoon Kim. Parameter-efficient transfer learning with diff pruning. *arXiv preprint arXiv:2012.07463*, 2020.

[10] Aounon Kumar and Tom Goldstein. Center smoothing: Certified robustness for networks with structured outputs. *Advances in Neural Information Processing Systems*, 34:5560–5575, 2021.

[11] Fuli Luo, Pengcheng Yang, Shicheng Li, Xuancheng Ren, and Xu Sun. Capt: contrastive pre-training for learning denoised sequence representations. *arXiv preprint arXiv:2010.06351*, 2020.

[12] Laura Fee Nern and Yash Sharma. How adversarial robustness transfers from pre-training to downstream tasks. *arXiv preprint arXiv:2208.03835*, 2022.

[13] Masuma Akter Rumi, Xiaolong Ma, Yanzhi Wang, and Peng Jiang. Accelerating sparse cnn inference on gpus with performance-aware weight pruning. In *Proceedings of the ACM International Conference on Parallel Architectures and Compilation Techniques*, pages 267–278, 2020.

[14] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

[15] Ali Shafahi, Parsa Saadatpanah, Chen Zhu, Amin Ghiasi, Christoph Studer, David Jacobs, and Tom Goldstein. Adversarially robust transfer learning. *arXiv preprint arXiv:1905.08232*, 2019.

[16] Zhouxing Shi, Yihan Wang, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Fast certified robust training with short warmup. *Advances in Neural Information Processing Systems*, 34, 2021.

[17] Pratik Vaishnavi, Kevin Eykholt, and Amir Rahmati. Accelerating certified robustness training via knowledge transfer. *Advances in Neural Information Processing Systems*, 35:5269–5281, 2022.

[18] Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *arXiv preprint arXiv:2103.06624*, 2021.

[19] Quanlin Wu, Hang Ye, Yuntian Gu, Huishuai Zhang, Liwei Wang, and Di He. Denoising masked autoencoders are certifiable robust vision learners. *arXiv preprint arXiv:2210.06983*, 2022.

[20] Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. *arXiv preprint arXiv:2011.13824*, 2020.

[21] Hao Zhu, Wayne Wu, Wentao Zhu, Liming Jiang, Siwei Tang, Li Zhang, Ziwei Liu, and Chen Change Loy. Celebv-hq: A large-scale video facial attributes dataset. *arXiv preprint arXiv:2207.12393*, 2022.

A Preliminary

A.1 Sparsity Patterns

In this paper, we explore three distinct types of sparsity: unstructured sparsity, structured sparsity, and group-wise sparsity[2]. Given a sparse ratio k $(0 < k < 1)$, the network's sparsity can be depicted by a binary mask M, where each element corresponds to a single parameter of the model and the ratio of non-zero elements equals k. In a conventional neural network with convolutional layers, unstructured sparsity implies no constraints on the mask M other than its sparse ratio being k, whereas structured sparsity ensures channel-wise uniformity in the mask, i.e., all parameters in the same channel or kernel must share the same mask value. Lastly, group-wise sparsity [2] combines both unstructured and structured sparsity. Here, the mask M is first generated in an unstructured manner, and a hypergraph partitioning algorithm [13] identifies dense blocks of activated parameters, reactivating any deactivated parameters within these blocks. The remaining activated parameters not in these dense blocks are deactivated, with the ratio of chosen blocks controlled so that the sparse ratio still equals k after determining the group-wise mask.

A.2 Dynamic Mask Selection with RigL

We adapt the RigL method [6] for dynamic mask selection across three sparsity patterns. The RigL method dynamically activates and deactivates network parameters based on gradient magnitudes and magnitudes of parameter values, respectively, during training. Originally designed for unstructured sparsity, we modify RigL for structured sparsity by shifting from parameter-level to channel-level activation and deactivation, guided by the magnitudes and gradient magnitudes of the channel weight γ in BatchNorm layers. We control the overall sparsity ratio using a strategy similar to [6]. For group-wise sparsity, we simply follow the unstructured version of RigL, allowing the group-wise sparsity to determine the structured sparsity.

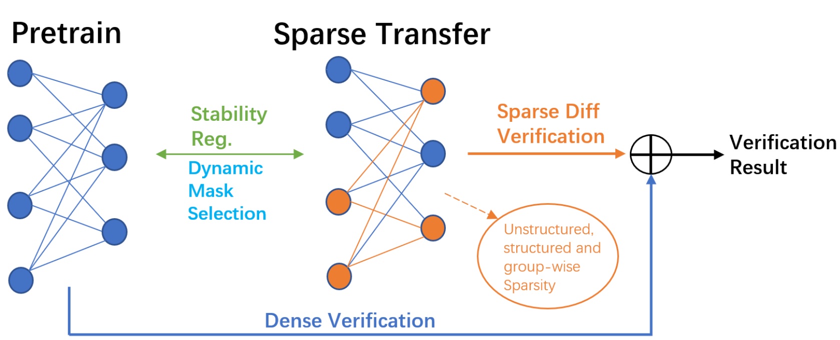

Figure 1: Our novel framework integrating differential verification with sparse transfer learning. For downstream tasks, the pre-trained network is refined using DiffPruning[9] coupled with dynamic mask selection. To ensure network robustness during sparse parameter updates, we introduce a neuron stability-based regularizer. For verification, we synergize sparse differential verification techniques with reusable dense verification results to yield a conclusive verification outcome.

Table 3: The acceleration results for a certified verification of different architectures with sparse transfer learning under different sparsity patterns.

			Sparse Transfer						
Architecture	Sparsity Pattern	Updated Params(%)	1	2	4	8	16	32	64
ResNet-18	Unstruct	Time Saved(%)	92.5	88.4	82.4	72.1	55.8	33.2	16.2
	Struct	Time Saved(%)	77.8	63.8	47.2	29.0	11.0	3.1	1.4
	Group-wise	Time Saved(%)	87.2	81.8	75.2	61.9	49.3	28.7	12.5
ResNet-50	Unstruct	Time Saved(%)	92.1	88.5	82.2	72.8	55.4	33.1	15.8
	Struct	Time Saved(%)	77.6	63.5	46.6	29.1	10.2	3.2	1.2
	Group-wise	Time Saved(%)	87.2	81.8	75.2	61.9	49.3	28.7	12.5
VGG-16	Unstruct	Time Saved(%)	92.5	89.1	82.5	73.1	55.4	33.2	15.6
	Struct	Time Saved(%)	77.8	63.6	46.8	29.3	10.3	3.5	1.4
	Group-wise	Time Saved(%)	89.4	84.8	77.5	63.5	50.3	30.1	13.7

B Datasets

Our experiment setting is elaborated as below:

CIFAR10 Initially, we deploy CIFAR10 to gauge the efficacy of our methodologies. Given the comparatively modest scale of CIFAR10, we employ contrastive learning during the pre-training phase to extract a rich self-supervision signal and subsequently use image classification for the downstream task. It's noteworthy that contrastive learning predicts based on a dense feature map rather than a singular scalar probability. Consequently, randomized smoothing is unsuitable for pre-training this model. As an alternative, we utilize Center Smoothing [10]—a variant of randomized smoothing designed to secure L_2-norm robustness for dense outputs—in tandem with contrastive learning to pre-train our network. Following this, randomized smoothing is incorporated for transfer learning within the image classification task.

CelebV-HQ Introduced in [21], CelebV-HQ is a contemporary benchmark tailored for multi-attribute classification tasks. It offers classifications for 83 facial attributes bifurcated into two categories: appearance and action attributes. Given that CelebV-HQ is rooted in video classification, we extract five disparate frames at random from each video, resize them to 64x64 dimensions, and utilize them as the input for every sample. This approach morphs the multi-attribute video classification task into a multi-attribute image classification challenge. Our strategy then encompasses random sampling of 40 attributes for pre-training, with the remaining attributes earmarked for downstream transfer. It's pivotal to understand that, in this dataset, the pre-training endeavor involves multi-attribute classification using the 40 selectively sampled attributes. Each subsequent downstream task revolves around binary classification, leveraging each of the residual attributes. To ascertain comprehensive results, evaluations across all downstream tasks are averaged. In the context of this dataset, we contemplate three distinct transfer settings: standard transfer, and a "few-shot" transfer, wherein the downstream tasks have access to merely 1% of randomly sampled data for their training.

We employ DiffPruning, as previously discussed, for our sparse transfer learning approach. Our evaluation criteria are bifurcated: first, we gauge the time saved in verification through our proposed methodologies in contrast to direct verification of samples in downstream tasks. Second, we assess certified robustness, which equates to the verified accuracies, as confirmed by randomized smoothing in the subsequent tasks. Pertaining to the model architecture, unless stated otherwise, we consistently utilize ResNet-50 as the foundational network for our experiments. Only the fully connected layers of the network undergo reinitialization, with sparse transfer learning executed on the convolutional layers. We've earmarked the perturbation radius of the input L_2-norm ball at 0.25, considering a normalized image input.

C Acceleration Results on More Architectures

In order to validate the consistent acceleration performance of our proposed techniques, we substituted ResNet-50 with both ResNet-18 and VGG-16 in our CIFAR10 experiments. Our objective was to compare the acceleration outcomes of these three architectures when subject to randomized smoothing-based verification. These findings are documented in table 3.

Table 4: The comparison of under normal/few-shot transfer setting. **+Reg** means applying DiffStab and RigL. The relative improvements in the brackets are obtained by comparing them with the unstructured baseline.

			Direct Train	Dense Transfer	Sparse Transfer						
		Updated Params(%)	100	100	1	2	4	8	16	32	64
Normal transfer	Unstruct (Baseline)	Ver Acc(%)	55.8	59.1	52.8	53.8	54.2	55.3	56.6	57.2	57.9
	Unstruct +Reg	Ver Acc(%)	55.8	59.1	56.2 (+3.4)	56.9 (+3.1)	57.5 (+3.2)	57.9 (+2.6)	58.6 (+2.0)	58.9 (+1.7)	59.0 (+1.1)
	Struct +Reg	Ver Acc(%)	55.8	59.1	54.7 (+1.9)	56.3 (+2.5)	56.6 (+2.4)	57.1 (+1.8)	57.8 (+1.2)	58.4 (+1.2)	58.8 (+0.9)
	Group-wise +Reg	Ver Acc(%)	55.8	59.1	55.6 (+2.8)	57.0 (+3.2)	57.2 (+3.0)	57.7 (+2.4)	58.3 (+1.7)	58.7 (+1.5)	58.8 (+0.9)
Few-shot transfer	Unstruct (Baseline)	Ver Acc(%)	39.2	54.2	48.6	50.2	51.3	52.4	53.2	53.6	53.8
	Unstruct +Reg	Ver Acc(%)	39.2	54.2	52.5 (+3.9)	53.1 (+2.9)	53.5 (+2.4)	53.8 (+1.4)	54.0 (+0.8)	54.1 (+0.5)	54.1 (+0.3)
	Struct +Reg	Ver Acc(%)	39.2	54.2	49.3 (+0.7)	51.9 (+1.7)	52.6 (+1.3)	53.1 (+0.8)	53.6 (+0.4)	53.8 (+0.2)	53.9 (+0.1)
	Group-wise +Reg	Ver Acc(%)	39.2	54.2	51.6 (+3.0)	52.7 (+2.5)	53.2 (+0.9)	53.5 (+1.1)	53.7 (+0.5)	53.9 (+0.3)	53.9 (+0.1)

Remarkably, the proportion of verification time saved remains relatively stable across the diverse architectures. For instance, the discrepancy between ResNet-18 and ResNet-50 is negligible, remaining within a 1% margin in most scenarios. VGG-16 presents marginally superior acceleration outcomes. This can potentially be attributed to VGG-16's relatively straightforward architecture when contrasted with residual networks. Consequently, it encompasses fewer operations, which wouldn't benefit from acceleration during forward propagation.

D DiffStab and Dynamic Mask Selection Boost Certified Robustness of Sparse Transfer Learning

From the results presented in the preceding subsection, a discernible performance discrepancy between dense transfer learning and sparse transfer learning remains evident. We delved into potential reasons for this in Sec. 3.2, subsequently proposing two remedies: the DiffStab regularizer and dynamic mask selection using RigL. Upon applying these methodologies to three distinct sparsity patterns on both the CIFAR10 and CelebV-HQ datasets, the outcomes, as depicted in table 2 and table 4 respectively, show a marked enhancement in verified accuracy for sparse transfer learning. The effectiveness of these strategies is directly proportional to the degree of sparsity, with higher sparsity ratios benefiting more significantly.

Interestingly, while these techniques are tailored for sparse transfer learning, they appear to have no discernible impact on dense transfer learning. This aligns with our expectations, given that there's inherently no room for dynamic mask selection in dense parameter updates. Moreover, it can be inferred that the DiffStab regularizer truly shines in environments where updated parameters are sufficiently sparse. This enables the regularizer to more effectively modulate network stability and robustness, without inadvertently hindering model training.

With the implementation of the two techniques, we are now equipped to pinpoint hyperparameter configurations that strike a balance between impressive verified accuracies and commendable acceleration outcomes for sparse transfer learning. ❶ Taking the CIFAR10 dataset as an example: Among configurations that surpass the verified accuracy of direct training, structured sparsity with an 8% sparse ratio stands out, yielding a remarkable 72.8% reduction in verification time when juxtaposed with traditional verification. When filtering for configurations that achieve over 80% verification acceleration, the same structured sparsity setting with an 8% sparse ratio boasts the pinnacle of verified accuracy, only trailing direct training by a slim 1% in accuracy. ❷ Turning our attention to the CelebV-HQ dataset under the standard transfer setting: We discern that nearly all sparse transfer configurations armed with regularizers outperform direct training. Notably, group-wise sparsity at a 16% sparse ratio demonstrates a negligible performance dip, less than 1% compared to dense transfer, while simultaneously realizing a 49.3% acceleration. ❸ In the few-shot setting: Some of the most aggressive sparse configurations, updating a mere 2% of parameters with both unstructured and group-wise sparsities, exhibit a performance delta of under 2% accuracy loss. This is paired with remarkable acceleration gains of 63.5% and 81.8%, respectively.