FaShapley: Fast and Approximated Shapley Based Model Pruning Towards Certifiably Robust DNNs

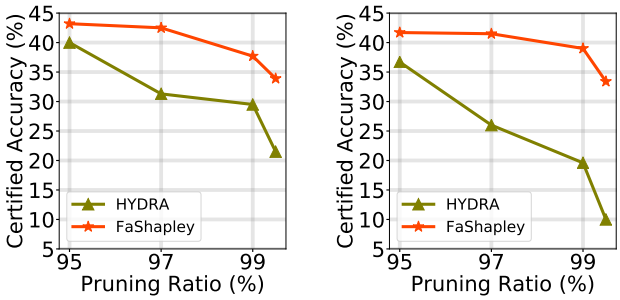
***Abstract*—Despite the great success achieved by deep neural networks (DNNs) recently, several concerns have been raised regarding their robustness against adversarial perturbations as well as large model size in resource-constrained environments. Recent studies on robust learning indicate that there is a tradeoff between robustness and model size. For instance, larger smoothed models would provide higher robustness certification. Recent works have tried to weaken such a tradeoff by training small models via optimized pruning. However, these methods usually do not directly take specific neuron properties such as their importance into account. In this paper, we focus on designing a quantitative criterion, neuron Shapley, to evaluate the neuron weight/filter importance within DNNs, leading to effective unstructured/structured pruning strategies to improve the certified robustness of the pruned models. However, directly computing Shapley value for neurons is of exponential computational complexity, and thus we propose a fast and approximated Shapley (FaShapley) method via gradient-based approximation and optimized sample-size. Theoretically, we analyze the desired properties (e.g, linearity and symmetry) and sample complexity of FaShapley. Empirically, we conduct extensive experiments on different datasets with both unstructured pruning and structured pruning. The results on several DNN architectures trained with different robust learning algorithms show that FaShapley achieves state-of-the-art certified robustness under different settings.**

***Index Terms*—model pruning, certified robustness**

I. Introduction

Deep neural networks (DNNs) have been deployed in a variety of real-word applications such as medical imaging analysis and language understanding [13, 57]. However, several studies have shown that DNNs are vulnerable to adversarial attacks, which are stealthy to humans but very effective in misleading machine learning models [3, 8, 14, 40, 41, 55, 63, 71, 80]. Therefore, it is critical to understand, evaluate, and most importantly, certify the **robustness** of DNNs before massive production and deployment of safety-critical applications. Meanwhile, an almost equally important factor before real-world deployment of DNNs is the **model size**, especially in resource-limited environments [5, 49].

Recent works show that there is a tradeoff implicitly between these two factors (i.e., robustness and model size): over-sparsified DNNs are vulnerable [23], and over-parameterization is important for achieving decent robustness [47]. For instance, empirically, models with larger size would achieve higher robustness after adversarial training [76]. Similarly, the certified robustness of DNNs trained via randomized smoothing also indicates that large models with $13\times$ more parameters would achieve 10% better certification robustness than the smaller counterpart [10, Section 4].



(a) CIFAR-10 (b) SVHN

Fig. 1. Comparison of our proposed FaShapley and the state-of-the-art algorithm HYDRA based on CROWN-IBP training ($\epsilon = 2/255$) via unstructured pruning. Certified robustness of pruned models based on FaShapley outperforms HYDRA significantly under different pruning ratios.

To weaken such a tradeoff in practice, several efforts have been made on model pruning for model robustness [23, 46, 56, 59, 76], but these methods do not directly take specific neuron properties such as their importance into account. Note that compared to empirical robustness [7, 21, 47], the *certified robustness* [10, 17, 79] can provide a lower bound of prediction accuracy under *arbitrary* attacks under certain conditions. In particular, deterministic verification [**?**] provides a lower bound to the certified accuracy with the guarantee of no false positive predictions and is more meaningful on pruned networks with bound propagation based certification approaches, and therefore we focus on deterministic verification and evaluate on standard model architectures following the literature [64, 73, 78]. In this paper, we aim to ask: *Can we design an efficient criterion based on neuron properties to evaluate the importance of neuron weights and filters for unstructured and structured pruning, respectively? Can we improve the certified robustness for the pruned models with smaller size to relax the tradeoff?*

Shapley value (SV) [60] has been successfully applied to a variety of ML tasks, including data summarization, data evaluation, and active data acquisition [18, 28, 29] due to its ability to evaluate the importance of data samples. In this work, we aim to leverage the power of SV to evaluate the neuron weight/filter importance towards the certified robustness. Concretely, we view the problem of model pruning as a cooperative game with the objective of achieving high certified robustness, and each weight or filter is viewed as a player in the game. Recent works [1, 2, 19, 20, 20] apply SV to neuron importance evaluation, but they only explore filter-level evaluation and do not consider the certified robustness

of models. In contrast, our proposed FaShapley enables SV to be applied to weight-level evaluation to prune models towards high certified robustness. In our work, the SV of each neuron weight/filter is calculated as the criterion to evaluate the "non-important" ones for pruning. The criterion has *four* advantages: (1) *desired properties (linearity, symmetry, nullity, efficiency),* (2) *awareness of the certified robustness objective by using specific utility functions,* (3) *effectiveness due to the consideration from a global view,* (4) *interpretability as a neuron selection criteria.*

However, there are additional challenges of directly utilizing SV for model pruning towards certified robustness, *1) exponential computational complexity* and *2) less effective without enough sampling*. Calculating the exact SV is a #P-complete problem, and therefore it is usually approximated with a sampling-based method [18]. Even so, when SV is applied to unstructured pruning for models with millions of neurons, it is impossible to calculate the value of every weight with feasible computation. To reduce the computational costs, we propose a fast and approximated SV method (FaShapley), leveraging the neuron properties by multiplying the norm of weights with gradients which is able to scale up to weight-level evaluation for unstructured pruning. Besides reducing computational complexity, FaShapley also demonstrates effectiveness on model pruning towards certifiably robust DNNs with our sample-size optimization. As shown in Figure 1, it is clear that FaShapley achieves much higher certified accuracy than the state-of-the-art algorithm HYDRA [59].

In addition, we theoretically analyze (1) desired properties of our FaShapley, and (2) sample complexity of our method. We prove that the sample complexity of FaShapley is a function of the sample-size (i.e., size of the sampled subgroup of neurons), and based on the empirical observations of model pruning literature, we conclude that our bound should be tight for general DNNs.

Technical Contributions. In this paper, we take the *first* step towards designing an effective neuron importance evaluation method for model pruning to achieve high certified robustness. We make the following contributions.

- We propose a fast and approximated Shapley method (FaShapley) for model pruning to achieve high certified robustness for small models (e.g., up to 99% pruning), which is applicable to both unstructured and structured pruning.
- Theoretically, we analyze the desired properties of FaShapley (e.g. linearity and symmetry) and its sample complexity. We also demonstrate the efficiency of FaShapley compared to existing Shapley approximation.
- Empirically, we conduct extensive experiments on *three* datasets with both unstructured pruning and structured pruning. The results on *three* DNN architectures trained with different robust learning algorithms show that FaShapley achieves the state-of-the-art certified robustness under different pruning ratios (e.g., FaShapley achieves up to 35.8% higher certified robustness than baselines under 99% pruning on CIFAR-10 via CROWN-IBP verification).

II. Related Work

Model pruning aims to compress neural networks for deployment in resource-constrained environments. Network pruning can generally be categorized into unstructured and structured methods. Unstructured methods [22, 24, 25] prune unimportant weights in the model, while structured pruning [34, 68] methods remove the whole filters or channels with low importance. There have been pioneering works [11, 12, 23, 38, 46, 54, 56, 59, 72, 76] that explore the potential of pruning on top of robust DNNs, but this line of research mainly focuses on the training strategy and does not directly take specific neuron properties such as their importance into account. In the literature of model pruning, there are different effective criteria designed to evaluate the importance of neurons using norm of weights, gradients, or BN scaling factors [26, 36, 39, 44, 62]. Compared to these criteria, FaShapley is motivated by powerful Shapley value in cooperative game theorey and is more explainable by using the expected marginal contributions of neuron weights/filters as the importance score. Therefore, FaShapley guarantees desired properties such as linearity and symmetry as an evaluation criterion and shows significant performance improvements over the heuristic criteria.

Shapley value (SV) [60] is a classic concept in cooperative game theory and demonstrates great power in a range of ML tasks including data summarization, data evaluation, and active data acquisition [18, 28, 29]. However, SV calculates the marginal contributions of players based on all combinations and requires exponential computational costs. To overcome the challenge of large computational costs of calculating the exact Shapley value, existing works approximate the value based on Monte Carlo sampling [18] or properties of utility functions [15]. Recent works [1, 2, 19, 20] leverage SV to evaluate the neuron importance, which differ from ours in two perspectives, (1) they do not explore weight-level importance evaluation, (2) they do not apply it to achieve certified robustness on pruned DNNs. In this paper, we propose a fast Shapley approximation based on the gradients and optimized sample-size for effectiveness and further computational costs reduction. We take the *first* step to leverage SV for weight-level evaluation and unleash its power for model pruning towards certified robustness.

Certified robustness provides provable robustness guarantees under a given perturbation budget. Certifiably robust approaches include robust training and verification methods [4, 10, 42, 43, 61, 67, 74, 75]. A recent survey categorizes verification approaches into deterministic and probabilistic verification [**?**]. Both lines of research are faced with the challenge of high verification costs due to millions of neurons in DNNs. In particular, deterministic verification [**?**] provides a lower bound to the certified accuracy with the guarantee of no false positive predictions and is more meaningful on pruned networks with bound propagation based certification approaches, which are the focus in this paper. Specifically, we mainly evaluate certified robustness against ℓ_∞-bounded perturbations, which is a particularly challenging [6, 32] and commonly used threat

model, and the methods for this threat model are extensible for handling other threat models [50, 69, 70]. Our proposed FaShapley pruning algorithms can be directly leveraged on certifiably robust DNNs and weaken the tradeoff between robustness and the model size.

III. FaShapley For Model Pruning

In this section, we introduce our proposed FaShapley and the complete algorithm, leveraging which we achieve high certified robustness after model pruning. Section III-A formulates the problem of model pruning to achieve certified robustness. Section III-B introduces our sample-size optimization and the fast approximation of Shapley based on weights and gradients. Section III-C illustrates the complete pruning algorithm to achieve certified robustness for unstructured and structured pruning.

A. Problem Formulation

Given a training dataset D and a network with weights θ, the goal of our model pruning is to prune the least important weights (or filters) and maintain high certified robustness. Similar with [59], we introduce a binary mask m_b in which each bit corresponds to a neuron weight or filter and all bits in m_b are initialized as 1. One weight (or filter) can be pruned by setting its mask to 0. Therefore, the objective of unstructured pruning can be formulated as follows:

$$\begin{aligned} \hat{m_b} = \underset{m_b \in \{0,1\}^N}{\arg\min} \underset{(x,y)\sim D}{E} & [L_{cer}(\theta \odot m_b, x, y)] \\ s.t.\ & |m_b| \leq (1-p)N \end{aligned} \tag{1}$$

where x and y denote the images and labels in the dataset D. $\theta \odot m$ refers to the element-wise multiplication of weights θ with the mask m_b, L_{cer} is the loss function of the certifiably robust training approach, p is the pruning ratio, N is the number of weights, and $|m_b|$ is the ℓ_0 norm of m_b. For structured pruning, the only difference is that m_b in Equation 1 has an extra constraint that the masks corresponding to one filter must be set to 0 or 1 simultaneously.

B. FaShapley Based Importance Evaluation

Shapley value (SV) has been applied to different domains given desirable properties (linearity, symmetry, nullity, efficiency) and interpretability. However, computing SV is of exponential computational complexity and thus calculating SV exactly is infeasible. Even with sampling-based approximation, we show that approximating the marginal contributions of millions of neurons weights still cannot be completed in reasonable time. Directly applying SV to importance evaluation also introduces effectiveness concerns especially for a small sample-size which is illustrated in detail in this section. To solve these challenges, we propose 1) *sample-size optimization to make the calculation effective*, and 2) *a gradient-based optimization strategy to approximate the marginal contribution of neurons efficiently*.

Sample-Size Optimization. To evaluate the importance of neuron weights or filters, we can view each weight or filter as a player in the cooperative game and calculate their SV as pruning criteria. Suppose that $\theta = \{\theta_i\}_{i=1}^N$ is the weights of a neuron network and $U(S)$ is the utility function, representing the value calculated based on the additive aggregation of $\{\theta_i\}_{i\in S}$ and $S \subseteq I = \{1, ..., N\}$. The SV $s^i(U)$ of weight/filter i can be formulated as follows:

$$s^i(U) = \frac{1}{N} \sum_{S \subseteq I/\{i\}} \frac{1}{\binom{N-1}{|S|}} [U(S \cup \{i\}) - U(S)] \tag{2}$$

To make the evaluation aware of the training objective to achieve certified robustness, we utilize the negative robustness loss $-L_{cer}(\theta, x, y)$ as the utility function U.

Note that the power of SV as an importance evaluation criterion lies in the fact that it aggregates the marginal contribution considering all combinations for the rest of the players (i.e., weights/filters), which also leads to the high computational complexity. Let $k = |S|$ be the **sample-size**, the calculation of SV views the importance scores of players from a global view by traversing all possible sample-sizes (i.e., all integers in the range $[0, N-1]$):

$$\begin{aligned} s^i(U) &= \frac{1}{N} \sum_{k=0}^{N-1} \sum_{|S|=k, S \subseteq I \setminus \{i\}} \frac{1}{\binom{N-1}{k}} [U(S \cup \{i\}) - U(S)] \\ &= \frac{1}{N} \sum_{k=0}^{N-1} s_k^i(U) \end{aligned} \tag{3}$$

where $s_k^i(U)$ denotes the marginal contribution of player i given a fixed sample-size k based on utility function U. Note that the sample-size k belongs to the set $\{i\}_{i=0}^{N-1}$.

From Equation 3, we can see that the SV calculation takes the average of the terms $s_k^i(U)$ with different sample-sizes. In addition, we observe that the term $s_k^i(U)$ with a relatively large sample-size k plays a more effective role in the evaluation. The results that confirm our hypothesis are provided in Table XVII. There are *two* possible reasons for such observation: (1) $s_k^i(U)$ *with a large sample-size k introduces more fair and stable comparisons since this way the expectation of the size of the intersection set of sampled subsets S is larger*; (2) *the utility of a small number of players is usually low, thus making the marginal contribution calculated based on it less useful.*

In light of this observation, we calculate $s_k^i(U)$ with a large sample-size k to evaluate the importance of neuron weights/filters. Note that if the sample-size k is set N (the number of neurons), it is reduced to Leave-One-Out (LOO) [29]. LOO is shown to lose the advantage of considering the importance from multiple combinations, which enables a much better evaluation from a global view (more detailed evaluations provided in Table IX and Table X). Without special specification in the experiments, we fix the sample-size k as $\lceil 0.9N \rceil$ in this work.

Fast and Approximated Shapley Calculation. The challenge in calculating SV lies in its exponential computational costs. Calculating the exact SV in Equation 2 involves computing the marginal utility of every neuron for each subsample, which is $\mathcal{O}(2^N)$ with N the total number of neurons. To overcome this, existing works calculate the approximation of

the value based on Monte Carlo sampling [18], or design algorithms for the special properties of utility functions [15].

In this work, we propose to leverage the norm of the multiplication of weights and gradients to approximate the marginal contribution of weights/filters as follows:

$$\begin{aligned}|U(S \cup i) - U(S)| &= |-L(\theta + d\theta_i, x, y) + L(\theta, x, y)| \\ &\approx |\frac{\partial L}{\partial \theta_i} d\theta_i| = |\frac{\partial L}{\partial \theta_i}||d\theta_i| = |\frac{\partial L}{\partial \theta_i}||\theta_i|\end{aligned} \quad (4)$$

In the Taylor expansion of the loss function L, we approximate by only taking the first-order terms into consideration and such approximation is also leveraged in [35]. Since the marginal contribution of a neuron weight is the difference of utility before pruning it and utility after pruning it, we have: $|d\theta_i| = |\theta_i - 0| = |\theta_i|$. In this way, we can calculate the approximation of the marginal contribution of all neurons in **one backward pass**, in contrast to the standard SV calculation where the marginal contribution of one neuron requires two forward passes. Thus, FaShapley is applicable to the weight-level importance evaluation.

C. FaShapley Based Model Pruning

We design a three-step model pruning pipeline based on FaShapley to achieve certified robustness: pre-training a network with robust training methods, model pruning, and fine-tuning (FT). The pre-training step generates DNNs based on specific loss functions. The pruning step masks out a particular ratio of weights or filters and attempts to maintain the certified robustness to the largest extent. The fine-tuning step refines the remaining weights/filters to further minimize the loss. Our proposed FaShapley serves as an effective criterion to be applied in the pruning step. Concretely, FaShapley assigns each weight an importance score and next the weights with small importance scores are pruned.

In algorithm 1, we provide the detailed steps to calculate the importance of each weight and prune the model with a predefined pruning ratio via unstructured pruning. V stores FaShapley values. In one iteration, it is updated by the norm of the multiplication of weights and gradients with a randomly selected group of weights being masked out. Finally, the weights with low FaShapley values are pruned. For structured pruning, the only difference is that each filter is viewed as a unit for evaluation.

Algorithm 1 FaShapley Based Pruning

Input: model *M*, data *loader*, sample complexity *m*, sample-size k, neuron weights *players* of size N, loss function L_{cer}, number of pruning units *p*
Output: model M_o
Initialize $V[\ ] = 0$ {store FaShapley values of neuron weights}
for *x*,*y* in *loader* **do**
  Initialize *times* $= 0$
  repeat
    times = *times* +1
    S = sample(*players*, $N - k$) {randomly sample *(N-k)* neuron weights to be pruned}
    M' = prune(*M*, *S*) {prune the sampled neuron weights *S* in model *M*}
    pred = *M'*(*x*)
    loss = L_{cer}(*pred*, *y*)
    loss.backward( )
    for *player* **in** *players* **do**
      V[*player*] += ‖*player*.gradient ∗ *player*.weight‖
    end for
  until *times* = *m*
end for
M_o = prune(*M*,argsort(*V*)[:*p*]) {prune based on *V*}

IV. Theoretical Analysis for FaShapley

Shapley value's power of contribution evaluation benefits from its four desired properties (linearity, symmetry, nullity, efficiency). However, SV fails to be exactly calculated for millions of neuron weights due to large computational costs, thus inducing extensive consideration of sample complexity for SV approximation methods. In this section, we provide theoretical analysis on these desired properties of our proposed FaShapley approximation as well as its sample complexity. From the analysis, we conclude that (1) FaShapley can reserve four desired properties of SV (Section IV-A), (2) FaShapley demonstrates lower sample complexity than existing sampling-based SV (Section IV-B).

A. Desired Properties of FaShapley

It is proved that SV (including its continuous variants like Aumann-Shapley and Serial Cost methods) is the *only* way of assigning attributions to players that satisfies four properties [60]: linearity, symmetry, nullity, and efficiency. These four desired properties are defined as follows: (1) linearity: if the utility function can be seen as a linear combination of two other utility functions, then any SV should also be a linear combination of the SVs computed with other utility functions; (2) symmetry: if the utility function value depends on two players but not their order, then they receive the same SV; (3) nullity: if the utility function value does not depend on a particular player, then its SV is zero; (4) efficiency: the overall reward can be assigned to all players. We theoretically prove that **expectation of FaShapley value** can preserve these properties. In other words, we do not consider the negligible error induced by the gradient-based approximation.

Formally, we present the FaShapley of player i with sample-size k as

$$F^i(U, k) = \mathbb{E}_{S \subseteq I/\{i\},\ |S|=k}[U(S \cup \{i\}) - U(S)] \quad (5)$$

where S is the randomly sampled group from all players $I = \{i\}_{i=1}^N$ and $U(S)$ denotes the utility function.

Theorem 1. *FaShapley $F^i(U, k)$ of player i with sample-size k satisfies four desired properties as follows:*
(1) linearity: if $U(S) = t_1 U_1(S) + t_2 U_2(S)$, then $F^i(U, k) = t_1 F^i(U_1, k) + t_2 F^i(U_2, k)$.
(2) symmetry: if $\forall S \subseteq I \backslash \{i, j\}$ and $|S| = k$, $U(S \cup \{i\}) = U(S \cup \{j\})$, then $F^i(U, k) = F^j(U, k)$.

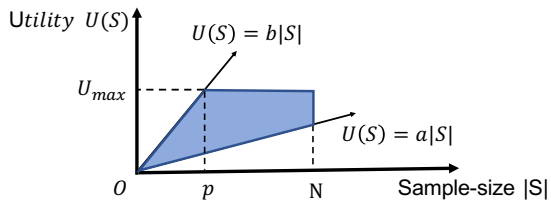


Fig. 2. An illustration for the bounds of utility $U(S)$. When $|S|$ increases, the interval between upper and lower bounds first increases and then decreases.

(3) nullity: if $\forall S \subseteq I \backslash \{i\}$ *and* $|S| = k$, $U(S \cup \{i\}) = U(S) + U(\{i\})$, *then* $F^i(U,k) = U(\{i\}) - U(\{\phi\})$.
(4) efficiency: $\sum_{i=1}^{N} F^i(U,k) = N[\mathbb{E}_{S \subseteq I, |S|=k+1} U(S) - \mathbb{E}_{S \subseteq I, |S|=k} U(S)]$.

Proof Sketch. For the proof of linearity, we can decompose the utility function in the definition of FaShapley with the linear combination of two utility functions and reorder the expression to compose FaShapley regarding the new utility functions. For the proof of symmetry, we only need to apply the identity of utility functions regarding player i,j to the definition and directly get the identity of FaShapley. For the proof of nullity, $U(S \cup \{i\})$ can be equalized to the addition of utility and the point is that the utility of the empty set ϕ is 0. For the proof of nullity, the key observation is that the expectation of the utility $U(S \cup \{i\})$ where S is sampled from $I/\{i\}$ with size k connects to $U(S)$ where S is sampled from I with size $k+1$ with a scaling factor N. The complete proof is provided in Appendix B.

Remark. These properties of FaShapley benefit neuron importance evaluation towards certified robustness. Linearity indicates that the utility can be viewed as a linear combination of classification utility (e.g., CE loss) and robustness utility (e.g., robust loss). Symmetry suggests that neurons with the same utility receive the same FaShapley value. Nullity indicates neurons with no contribution receive zero value. Efficiency guarantees that the total utility is distributed to all neurons.

B. Analysis of Sample Complexity

Direct Shapley calculation fails to be scaled up to weight-level evaluation on a large architecture and the sample complexity of approximating Shapley value is of great importance. Given the nice structure of FaShapley, here we present the theoretical analysis of its sample complexity and demonstrate the efficiency of FaShapley compared to the naive sampling-based Shapley calculation.

When applying FaShapley to neuron evaluation on a model, we use the utility function $-L_{cer}(\theta, x, y)$ where $L_{cer}(\theta, x, y)$ is loss function of robust learning. Let $L_{max} = \max(L_{cer}(\theta, x, y))$ be the maximum value of the loss function. For simplicity, we denote $U(\theta) = -L_{cer}(\theta, x, y) + L_{max}$ as the utility function to make the range of utility U be $[0, +\infty)$. N denotes the number of neuron weights/filters. S is the weights/filters sampled from $I = \{i\}_{i=1}^{N}$ and $|S|$ is the sample-size of set S. $U_{max} = \max(U(S))$, $\forall S \subseteq I$ refers to maximum utility. We view FaShapley value $F^i(U,k)$ as a continuous function given sample-size k ($k \in [0, N-1]$).

Fact IV.1. *There exist numbers* a, b $(0 < a < b)$, *which satisfy the following inequality,*

$$a|S| \leq U(S) \leq \min(b|S|, U_{max}) \tag{6}$$

Remark. We illustrate the bound of FaShapley in Figure 2. First, the utility $U(S)$ is positively correlated with $|S|$, and therefore can be bounded using $a|S| \leq U(S) \leq b|S|$ (Equation (6) in [48]). Second, $U(S)$ has an upper bound as U_{max} and the inflection point $p = \lfloor U_{max}/b \rfloor$ in Figure 2 is relatively small since existing neural network pruning techniques have shown that it is possible to reduce the parameters of trained networks by over 90% without compromising the accuracy [16]. In other words, Fact IV.1 is aligned with the existing empirical observations in literature.

Theorem 2. *With probability at least* δ *and estimation error* ϵ, *the sample complexity* m *for FaShapley is lower bounded as follows:*

$$m \geq \begin{cases} \lceil \frac{[(b-a)k+b]^2 \ln \frac{2}{\delta}}{2\epsilon^2} \rceil & 0 \leq k \leq \lfloor U_{max}/b \rfloor - 1 \\ \lceil \frac{(-ak+U_{max})^2 \ln \frac{2}{\delta}}{2\epsilon^2} \rceil & \lfloor U_{max}/b \rfloor - 1 < k \leq N-1 \end{cases} \tag{7}$$

With probability at least δ *and estimation error* ϵ, *the sample complexity* m_{sv} *for Shapley value approximation has the lower bound as follows:*

$$m_{sv} \geq \lceil \frac{U_{max}^2 \ln \frac{2}{\delta}}{2\epsilon^2} \rceil \tag{8}$$

Proof Sketch. We first derive the range of the utility function $U(S)$ regarding the sample-size $|S|$ with a piecewise function according to Fact IV.1 and the illustratetion in Figure 2. Then we can leverage Hoeffding's inequality and plug in the range of the utility function $U(S)$, sample complexity m, and the estimation error ϵ. Solving the inequality, we can finally get the lower bound of sample complexity m for FaShapley. For direct Shapley value appriximation, the range of the utility function is U_{max} and we can similarly leverage Hoeffding's inequality to derive the lower bound of sample complexity m_{sv}. The complete proof is provided in Appendix B.

Remark. Theorem 2 indicates that (1) FaShapley is more computationally efficient than direct Shapley approximation because the lower bound of m_{sv} in Inequality 8 is larger than that of m in Inequality 7, (2) when sample-size k is smaller than $p = \lfloor U_{max}/b \rfloor$, the sample complexity is positively correlated with the sample-size k; while if the sample-size k exceeds p, the sample complexity is negatively correlated to the sample-size k. This phenomenon is very interesting, since intuitively, when the sample-size grows up to p, there is more and more information involved, while after the sample-size exceeds p, there is a large overlap between samples and therefore the information gain is limited among samples. We believe our theoretical analysis provides another novel perspective for the meaning of p, which represents the standard empirical "optimal" pruning ratio in model pruning literature [22, 24, 25].

V. Experiments

A. Experimental Setup

Dataset. We conduct extensive experiments on *three* datasets: CIFAR-10 [31], SVHN [53], and Tiny-ImageNet [33]. We use CIFAR-10 and SVHN for fair comparisons with the SOTA certified robustness pruning method [59]. We further evaluate on Tiny-ImageNet to perform evaluation on a large-scale dataset (notice that existing deterministic robustness verification methods cannot scale up to ImageNet). CIFAR-10 dataset consists of 60000 32x32 color images in 10 classes including transportation and animals, with 6000 images per class. SVHN dataset is used for digit recognition in the real-world scenario and incorporates extensive labeled data (over 600,000 digit images). Tiny-ImageNet is a subset of the ImageNet dataset and contains 100,000 images of 200 classes (500 for each class) downsized to 64×64 colored images.

Model Architecture. Our proposed FaShapley based pruning algorithm is evaluated for *three* verifiable robust training methods: CROWN-IBP [78], MixTrain [64], and Auto-LiRPA [73]. For CROWN-IBP and MixTrain, we select CNN-4 (0.3M params) and CNN-7 (2.2M params) [70] as the backbone, while for Tiny-ImageNet, WideResNet 28-10 [77] (36.5M params) is selected. The results on three architectures under multiple pruning budgets demonstrate the effectiveness of FaShapley towards achieving certified robustness on pruned networks. For CIFAR-10 and SVHN, we evaluate on CNN-4 and CNN-7 following HYDRA [59] for fair comparisons. For Tiny-ImageNet, we evaluate on WideResNet 28-10 network mainly because the SOTA verification LiRPA achieves the highest certified robustness with the type of network (Table 2 in [73]), and thus we use the same architecture for fair comparison purposes. Note that Tiny-ImageNet with WideResNet 28-10 is the large-scale evaluation we can achieve with deterministic verification, which requires the bound propagation in neurons and is extremely time-consuming. In this paper, our aim is to reduce the computation costs and memory overhead through pruning on different model architectures viable with deterministic verification. The details of the **model structures** are formulated as follows:

- CNN-4: 2 Conv-ReLU layers with $\{16, 32\}$ filters, 2 linear layers with $\{2048, 100\}$ neurons.
- CNN-7: 4 Conv-ReLU layers with $\{32, 32, 64, 64\}$ filters, 3 linear layers with $\{4096, 512, 512\}$ neurons.
- WideResNet 28-10: 3 wide basic blocks (6 Conv-ReLU-BN layers) with widen factor = 10 and an additional linear layer with 512 neurons.

Evaluation Metrics. The ℓ_∞ perturbation budget is $2/255$ for CIFAR-10, SVHN, and $1/255$ for Tiny-ImageNet. Both unstructured pruning and structured pruning are explored in all three datasets. A larger perturbation $\ell_\infty = 8/255$ in CIFAR-10 is also explored for better comparisons. We use **clean accuracy (Cl-Acc)** and **certified accuracy (Cr-Acc)** to evaluate the performance of the network. Clean accuracy represents the percentage of correctly classified images without perturbation, while certified robust accuracy is the lower bound of the clean accuracy under given perturbation budgets.

Robust Training and Certification Methods. For verified robust training, we leverage three state-of-the-art approaches: (1) CROWN-IBP [78] which combines the fast IBP bounds in a forward pass and a tight linear relaxation bound CROWN in a backward pass, (2) MixTrain [64] which balances standard accuracy and certified robustness by applying robustness-oriented regularization on selected training inputs, and (3) Auto-LiRPA training [73] which optimizes efficiently computable bounds of the training loss. For robust verification, deterministic verification provides a lower bound to the certified accuracy with the guarantee of no false positive predictions and is more meaningful on pruned networks with bound propagation based certification approaches, and therefore we focus on **deterministic verification** and evaluate on standard model architectures following the literature [64, 73, 78]. Concretely, we use **IBP**, **symbolic interval analysis**, and **Auto-LiRPA** verification to perform verification of models trained with CROWN-IBP, MixTrain, and Auto-LiRPA, respectively.

Selection of Pruning Ratios for Fair Comparisons. To demonstrate the effectiveness of FaShapley compared with SOTA robust pruning methods and different criteria, we aim to evaluate with a wide range of pruning ratios. In different settings, we follow the criteria that the comparison is done under an arithmetic sequence of pruning ratios such that FaShapley achieves comparable certified accuracy and clean accuracy as the pre-trained model for the smallest pruning ratio and the baselines perform poorly for the largest pruning ratio. Therefore, different methods can be compared in a wide and meaningful range of pruning ratios. The clean / certified accuracy of **pre-trained models** before pruning is summarized as follows:

- CNN-4 on CIFAR-10 based on CROWN-IBP: $54.0\%/42.5\%$; CNN-7 on CIFAR-10 based on CROWN-IBP: $60.3\%/46.8\%$
- CNN-4 on SVHN based on CROWN-IBP: $61.5\%/41.5\%$; CNN-7 on SVHN based on CROWN-IBP: $69.5\%/47.9\%$
- CNN-4 on CIFAR-10 based on MixTrain: $62.5\%/46.8\%$; CNN-7 on CIFAR-10 based on MixTrain: $63.8\%/47.7\%$
- CNN-4 on SVHN based on MixTrain: $72.5\%/48.4\%$; CNN-7 on SVHN based on MixTrain: $77.5\%/56.5\%$
- WideResNet on Tiny-ImageNet based on Auto-LiRPA: $27.0\%/15.1\%$

The certified accuracy is computed through IBP, symbolic interval analysis, and Auto-LiRPA verification for CROWN-IBP training, MixTrain, and Auto-LiRPA, respectively.

Baselines. We compare FaShapley with *four* types of baselines, 1) *state-of-the-art model pruning method to achieve certified robustness*, 2) *different neuron selection criteria* in the literature for model pruning, 3) *SOTA model pruning method*, and 4) *different Shapley based pruning approaches*. Concretely, **HYDRA** [59] is the state-of-the-art algorithm for pruning certifiably robust neural networks. HYDRA optimizes a binary mask of weights as parameters based on its robustness

TABLE I

CLEAN AND CERTIFIED ACCURACY (%) OF MODELS AFTER UNSTRUCTURED PRUNING ON CIFAR-10 AND SVHN BASED ON CROWN-IBP TRAINING. FASHAPLEY OUTPERFORMS THE STATE-OF-THE-ART (HYDRA) BY A LARGE MARGIN BOTH W/ OR W/O FINE-TUNING (FT).

		Pruning Ratio		99%		97%		95%	
			Method	**Cl-Acc** (%)	**Cr-Acc** (%)	**Cl-Acc** (%)	**Cr-Acc** (%)	**Cl-Acc** (%)	**Cr-Acc** (%)
CIFAR-10	CNN-4	w/o FT	HYDRA	16.3	15.2	31.7	24.4	39.8	31.0
			FaShapley	33.3 (+17.0)	26.9 (+11.7)	51.5 (+19.8)	40.7 (+16.3)	53.7 (+13.9)	42.5 (+11.5)
		w/ FT	HYDRA	34.6	29.5	36.3	31.3	49.5	40.0
			FaShapley	47.0 (+12.4)	37.7 (+8.2)	52.4 (+16.1)	42.5 (+11.2)	53.6 (+4.1)	43.2 (+3.2)
	CNN-7	w/o FT	HYDRA	10.0	10.0	35.6	28.5	46.0	33.1
			FaShapley	59.8 (+49.8)	45.8 (+35.8)	60.3 (+24.7)	46.7 (+18.2)	60.3 (+14.3)	46.7 (+13.6)
		w/ FT	HYDRA	47.7	39.4	48.4	43.7	57.8	46.2
			FaShapley	60.0 (+12.3)	46.6 (+7.2)	60.8 (+12.4)	48.0 (+4.3)	60.3 (+2.5)	47.5 (+1.3)
SVHN	CNN-4	w/o FT	HYDRA	19.6	19.6	28.7	24.0	44.6	30.2
			FaShapley	57.4 (+37.8)	38.7 (+19.1)	61.6 (+32.9)	41.7 (+17.7)	61.6 (+17.0)	41.7 (+11.5)
		w/ FT	HYDRA	19.6	19.6	32.5	26.0	53.0	36.7
			FaShapley	53.2 (+33.6)	39.0 (+19.4)	60.7 (+28.2)	42.0 (+16.0)	61.6 (+8.6)	41.7 (+5.0)
	CNN-7	w/o FT	HYDRA	45.9	33.4	67.7	45.3	68.8	46.8
			FaShapley	69.8 (+23.9)	47.5 (+14.1)	69.8 (+2.1)	48.0 (+2.7)	69.8 (+1.0)	48.0 (+1.2)
		w/ FT	HYDRA	56.3	42.8	68.3	46.2	69.2	47.6
			FaShapley	67.2 (+10.9)	47.8 (+5.0)	69.8 (+1.5)	48.0 (+1.8)	69.8 (+0.6)	48.0 (+0.4)

TABLE II

CLEAN AND CERTIFIED ACCURACY (%) OF MODELS AFTER UNSTRUCTURED PRUNING ON CIFAR-10 AND SVHN BASED ON MIXTRAIN. FASHAPLEY OUTPERFORMS THE STATE-OF-THE-ART (HYDRA) BY A LARGE MARGIN BOTH W/ OR W/O FINE-TUNING (FT).

		Pruning Ratio		99%		97%		95%	
			Method	**Cl-Acc** (%)	**Cr-Acc** (%)	**Cl-Acc** (%)	**Cr-Acc** (%)	**Cl-Acc** (%)	**Cr-Acc** (%)
CIFAR-10	CNN-4	w/o FT	HYDRA	10.0	9.9	19.0	16.4	22.8	18.9
			FaShapley	11.3 (+1.3)	12.7 (+2.8)	33.3 (+14.3)	25.2 (+8.8)	44.5 (+21.7)	32.8 (+13.9)
		w/ FT	HYDRA	27.0	24.9	45.2	27.2	50.7	38.3
			FaShapley	38.1 (+11.1)	29.3 (+4.4)	52.0 (+6.8)	40.2 (+13.0)	56.6 (+5.9)	42.8 (+4.5)
	CNN-7	w/o FT	HYDRA	31.7	21.9	44.4	34.5	55.4	40.0
			FaShapley	45.0 (+13.3)	35.2 (+13.3)	58.5 (+14.1)	43.5 (+9.0)	61.3 (+5.9)	45.0 (+5.0)
		w/ FT	HYDRA	42.7	35.3	59.2	44.8	60.2	45.3
			FaShapley	61.0 (+18.3)	44.7 (+9.4)	64.2 (+5.0)	46.4 (+1.6)	64.4 (+4.2)	46.8 (+1.5)
SVHN	CNN-4	w/o FT	HYDRA	19.6	19.6	19.6	19.6	19.6	19.6
			FaShapley	21.7 (+2.1)	14.6 (−5.0)	57.1 (+37.5)	33.4 (+13.8)	65.9 (+46.3)	42.1 (+22.5)
		w/ FT	HYDRA	19.6	19.6	53.7	24.2	52.5	33.7
			FaShapley	58.3 (+38.7)	32.7 (+13.1)	71.0 (+17.3)	41.2 (+17.0)	76.7 (+24.2)	40.6 (+6.9)
	CNN-7	w/o FT	HYDRA	19.6	19.6	20.6	19.3	19.6	19.6
			FaShapley	68.1 (+48.5)	45.9 (+26.3)	76.9 (+56.3)	54.8 (+35.5)	77.3 (+57.7)	55.2 (+35.6)
		w/ FT	HYDRA	19.6	19.6	66.2	44.9	74.8	53.7
			FaShapley	75.0 (+55.4)	51.1 (+31.5)	78.2 (+12.0)	54.0 (+9.1)	78.6 (+3.8)	55.1 (+1.4)

objective. Through extensive experiments on three datasets with different training methods, we demonstrate that our FaShapley based pruning algorithm outperforms HYDRA significantly. We also compare FaShapley with other criteria used to evaluate the importance of weights/filters in DNNs: **weight-based protocol (LWM)** [76], **gradient-based protocol (Grad)** [51], **Taylor-based protocol (Taylor)** [35, 52, 58], and **stability-based protocol (Stab)**. The weight-based protocol selects the importance score as the magnitude of the neuron weights. The gradient-based protocol selects the importance score as the gradients of weights. The Taylor-based protocol selects the importance score as the magnitude of the multiplication of the neuron weights and the gradients of weights. Note that existing studies show that neuron stability is highly correlated with the DNN certified robustness [78], and thus we consider the stability-based protocol, which prunes unstable neurons, as another strong baseline. Notice that all the importance scores including our FaShapley are applied the criteria of one-shot pruning after convergence for fair comparisons. Furthermore, SOTA model pruning methods **Flying Bird (FB)** [9] and **BAR** [37] are the state-of-the-art unstructured pruning and structured pruning methods, respectively. We demonstrate that our FaShapley can ourperform them towards achieving certified robustness with both unstructured pruning and structured pruning. We also compare our FaShapley with different Shapley based pruning approaches, **SVPrune** [2] and **ShapPrune** [20], from which we show that only FaShapley can scale up to unstructured pruning and FaShapley also demonstrates significant performance improvements for structured pruning.

Experiment Details. We leverage a three-step model pruning pipeline involving pre-training a network, pruning it, and later fine-tuning it. In the pre-training step of CROWN-IBP, we follow the standard setting [78]. We set the scheduling length to 60 epochs, during which we gradually decrease the

TABLE III
CLEAN / CERTIFIED ACCURACY (%) OF MODELS AFTER STRUCTURED PRUNING ON CIFAR-10 AND SVHN BASED ON CROWN-IBP TRAINING. FASHAPLEY OUTPERFORMS THE STATE-OF-THE-ART (HYDRA) SIGNIFICANTLY BOTH W/ OR W/O FINE-TUNING (FT).

			Pruning Ratio	40%	50%	60%
CIFAR-10	CNN-4	w/o FT	HYDRA	40.8 / 32.3	18.0 / 15.4	15.5 / 13.8
			FaShapley	**52.9 / 41.7**	**38.4 / 30.7**	**15.7 / 14.3**
		w/ FT	HYDRA	51.2 / 38.9	44.2 / 32.5	**32.5** / 22.0
			FaShapley	**53.1 / 41.8**	**46.7 / 36.9**	32.1 / **26.8**
			Pruning Ratio	80%	85%	90%
	CNN-7	w/o FT	HYDRA	44.7 / 34.9	12.1 / 11.2	10.0 / 10.0
			FaShapley	**55.3 / 44.2**	**41.2 / 34.9**	**15.7 / 15.1**
		w/ FT	HYDRA	52.1 / 39.8	28.5 / 22.0	10.0 / 10.0
			FaShapley	**55.5 / 43.9**	**45.6 / 38.0**	**19.5 / 14.2**
SVHN			**Pruning Ratio**	50%	60%	70%
	CNN-4	w/o FT	HYDRA	32.8 / **28.7**	17.1 / 12.9	15.9 / 15.9
			FaShapley	**41.0 / 28.7**	**31.7 / 22.1**	**30.5 / 21.5**
		w/ FT	HYDRA	**49.8** / 33.9	40.8 / 28.6	19.6 / 19.6
			FaShapley	49.5 / **37.2**	**45.1 / 34.6**	**41.9 / 32.3**
			Pruning Ratio	75%	80%	85%
	CNN-7	w/o FT	HYDRA	41.1 / 25.0	25.0 / 17.7	15.9 / 15.9
			FaShapley	**68.8 / 48.2**	**57.0 / 42.6**	**26.6 / 25.4**
		w/ FT	HYDRA	63.4 / 42.2	49.2 / 34.9	15.9 / 15.9
			FaShapley	**69.1 / 48.3**	**60.1 / 43.6**	**39.0 / 32.0**

TABLE IV
CLEAN / CERTIFIED ACCURACY (%) OF MODELS (CNN-4) AFTER UNSTRUCTURED PRUNING ON CIFAR-10 BASED ON CROWN-IBP UNDER A LARGER PERTURBATION BUDGET ($\epsilon = 8/255$). FASHAPLEY OUTPERFORMS THE STATE-OF-THE-ART (HYDRA) BY A LARGE MARGIN BOTH W/ OR W/O FINE-TUNING (FT). THE PERFORMANCE BEFORE PRUNING IS 36.6/26.0.

Pruning Ratio		90%	95%	99%
w/o FT	HYDRA	29.0/21.7	33.0/24.3	16.1/14.4
	FaShapley	**36.7/26.2**	**36.7/26.2**	**36.2/26.1**
w/ FT	HYDRA	31.7/24.3	30.3/24.3	25.9/20.3
	FaShapley	**36.8/26.2**	**36.8/26.2**	**36.1/26.1**

portion of verifiable robust loss obtained by CROWN-IBP while increasing the portion obtained by IBP for each training batch. For the rest of the epochs after the scheduling epochs, only IBP contributes to the verifiable robust loss. IBP is used for evaluation. In the pre-training step of MixTrain, we use the best training setup [64] for both CIFAR-10 and SVHN. Concretely, we use sampling numbers 5 and 1 for CNN-4 and CNN-7, respectively. We select the balance factor $\alpha = 0.8$ to balance between regular loss and verifiable robust loss. The trained networks are evaluated with symbolic interval analysis [65, 66]. In the pre-training step of Auto-LiRPA, we follow the best training setup [73]. The networks were trained using the Adam [30] optimizer with an initial learning rate of 5×10^{-4}. Also, gradient clipping with a maximum ℓ_2 norm of 8 is applied. We gradually increase ϵ within a fixed epoch length of 400. We uniformly divide the epoch length with a factor 0.4, and exponentially increase ϵ during the former interval and linearly increase ϵ during the latter interval, to avoid a sudden growth of ϵ at the beginning stage. A hyperparameter β to balance LiRPA bounds and IBP bounds for the output layer is set and gradually decreases from 1 to 0 (1 for only using LiRPA bounds and 0 for only using IBP bounds), as per the same schedule of ϵ, and the end ϵ for training is set to 10% higher than the one in test.

Our proposed FaShapley serves as an effective criterion to be applied in the pruning step. $1,000$ randomly selected images are used for the calculation of FaShapley values due to computational cost consideration. We set the sample-size k (normalized to $[0, 1]$ based on the number of neuron weights/filters) to be 0.9 and the sample complexity m to be 30 for experiments on CIFAR-10 and SVHN. The sample-size k is selected as 0.999 for Tiny-ImageNet. The utility function is set to be the opposite of the loss function for CROWN-IBP and MixTrain. For the model trained with Auto-LiRPA, we simply use the opposite of Cross-Entropy (CE) loss as the utility function for efficiency concern. To further boost the performance of the pruned model via structured pruning on Tiny-ImageNet, we set a threshold 0.35 for all layers to avoid over-pruning in some layers under the pruning ratios of 20%, 25%, and 30%. In the fine-tuning step, we train the pruned model on CIFAR-10 and SVHN based on CROWN-IBP training for 60 epochs using the learning rate 1×10^{-5}. We train 30 epochs on the pruned models based on MixTrain training using the learning rate 1×10^{-5}. We train 10 epochs with the learning rate 5×10^{-4} on Tiny-ImageNet based on Auto-LiRPA training. The codes we used follow the MIT license. All experiments are conducted on a 1080 Ti GPU with 11,178 memory.

B. Evaluation Results

Certified Robustness via Unstructured Pruning. Through extensive experiments for *two* architectures (CNN-4 and CNN-7) on *two* datasets (CIFAR-10 and SVHN), we demonstrate that our algorithm FaShapley outperforms the state-of-the-art method HYDRA significantly with different robust training methods. We provide the results of CROWN-IBP and MixTrain robust training methods in Table I and Table II, respectively.

TABLE V
CLEAN / CERTIFIED ACCURACY (%) OF MODELS AFTER STRUCTURED PRUNING ON CIFAR-10 AND SVHN BASED ON MIXTRAIN TRAINING. FASHAPLEY OUTPERFORMS THE STATE-OF-THE-ART (HYDRA) SIGNIFICANTLY WITHOUT FINE-TUNING.

		Pruning Ratio	40%	50%	60%
CIFAR-10	CNN-4	HYDRA	41.2/21.4	16.4/12.3	**13.0**/11.2
		FaShapley	**51.5/38.5**	**35.2/24.6**	11.7/**11.5**
		Pruning Ratio	75%	80%	85%
	CNN-7	HYDRA	28.5/20.9	10.0/10.0	10.0/10.0
		FaShapley	**61.4/46.1**	**59.3/44.4**	**46.9/34.1**
		Pruning Ratio	50%	55%	60%
SVHN	CNN-4	HYDRA	25.6/12.7	9.8/9.4	11.2/9.3
		FaShapley	**32.7/20.1**	**19.8/17.1**	**18.5/16.6**
		Pruning Ratio	75%	80%	85%
	CNN-7	HYDRA	55.7/40.2	29.7/25.3	19.6/19.6
		FaShapley	**76.1/54.6**	**72.9/50.9**	**43.3/28.7**

The certified accuracy are verified through IBP and symbolic interval analysis, respectively. To directly compare the effectiveness of the pruning step with HYDRA, we also present the results before fine-tuning, from which we can see that our algorithm can maintain the certified robustness of the pruned model even without fine-tuning for a large pruning ratio such as 95%, while HYDRA relies more on the fine-tuning step to recover the performance (still lower than FaShapley). Results of pruning on a model trained with CROWN-IBP with a larger perturbation budget ($\epsilon = 8/255$) are given in Table IV. From the results of extensive evaluation via unstructured pruning, we can conclude that 1) FaShapley outperforms the SOTA certified robustness pruning method HYDRA in terms of both certified accuracy and clean accuracy via different certification methods (CROWN-IBP and MixTrain) under different pruning ratios, 2) FaShapley can maintain the certified robustness of the pruned model even without fine-tuning for a large pruning ratio such as 95% via CROWN-IBP, while HYDRA relies more on the fine-tuning step to recover the performance (still lower than FaShapley), 3) FaShapley enables the compressed network to achieve comparable or higher certified robustness than the pre-trained model (e.g., 95% pruning via CROWN-IBP), and 4) the effectiveness of FaShapley over HYDRA is demonstrated under larger perturbations ($\ell_\infty = 8/255$).

Certified Robustness via Structured Pruning. FaShapley can be easily applied to structured pruning for which we only need to treat the filters/channels as the players and evaluate their importance scores. Then we perform filter and channel pruning based on the importance scores. To the best of our knowledge, we are the *first* to provide results of the certified robustness via structured pruning. As the state-of-the-art algorithm for unstructured pruning, HYDRA also provides the empirical robust accuracy for structured pruning. Therefore, we choose it as a strong baseline and conduct extensive comparisons on CIFAR-10 and SVHN based on *two* architectures (CNN-4 and CNN-7) and *three* pruning ratios in each setting. The results in Table III and Table V show that for structured pruning, our proposed FaShapley can achieve much higher certified robustness compared to HYDRA with different robust training methods (CROWN-IBP and MixTrain) both with and without finue-tuning the weights after pruning.

TABLE VI
CLEAN / CERTIFIED ACCURACY (%) OF FASHAPLEY COMPARED WITH HYDRA ON TINY-IMAGENET VIA UNSTRUCTURED PRUNING (UNSTRUCT) AND STRUCTURED PRUNING (STRUCT). THE PERFORMANCE BEFORE PRUNING IS 27.0/15.1

		Pruning Ratio	60%	70%
Unstruct	w/o Fine-tuning	HYDRA	23.8/13.5	17.5/9.1
		FaShapley	**25.8/14.3**	**25.2/13.8**
	w Fine-tuning	HYDRA	27.2/16.3	26.8/16.1
		FaShapley	**27.5/16.6**	**27.2/16.5**
		Pruning Ratio	20%	30%
Struct	w/o Fine-tuning	HYDRA	15.5/13.7	0.5/0.1
		FaShapley	**22.1/12.3**	**3.4/1.4**
	w Fine-tuning	HYDRA	26.4/15.8	22.3/13.5
		FaShapley	**26.5/16.1**	**26.0/15.8**

TABLE VII
CLEAN / CERTIFIED ACCURACY (%) OF FASHAPLEY COMPARED WITH DIFFERENT NEURON SELECTION CRITERIA ON TINY-IMAGENET VIA UNSTRUCTURED PRUNING. THE PERFORMANCE BEFORE PRUNING IS 27.0/15.1

	Pruning Ratio	60%	65%	70%
w/o FT	LWM	21.7/11.5	11.7/4.3	1.5/0.1
	Grad	1.1/0.1	1.0/0.1	1.0/0.0
	Taylor	24.5/13.4	24.1/13.3	24.5/13.4
	FaShapley	**25.8/14.3**	**25.6/14.0**	**25.2/13.8**
w/ FT	LWM	24.7/15.4	24.4/15.5	23.9/15.3
	Grad	25.1/15.5	25.0/15.4	25.0/15.2
	Taylor	26.5/16.3	26.5/16.2	26.3/16.2
	FaShapley	**27.5/16.6**	**27.3/16.4**	**27.2/16.5**

TABLE VIII
CLEAN / CERTIFIED ACCURACY (%) OF FASHAPLEY COMPARED WITH DIFFERENT NEURON SELECTION CRITERIA ON TINY-IMAGENET VIA STRUCTURED PRUNING. THE ORIGINAL MODEL IS TRAINED WITH AUTO-LIRPA AND THE PERFORMANCE BEFORE PRUNING IS 27.0/15.1.

	Pruning Ratio	20%	25%	30%
w/o FT	LWM	3.3/0.6	0.5/0.1	0.5/0.1
	Grad	20.0/11.3	14.4/8.1	2.5/1.0
	Taylor	0.7/0.3	0.5/0.0	0.4/0.0
	FaShapley	**22.1/12.3**	**20.3/11.0**	**3.4/1.4**
w/ FT	LWM	**26.5**/15.5	20.0/10.1	13.6/8.6
	Grad	25.4/15.5	24.8/15.1	17.1/10.8
	Taylor	24.2/14.8	24.9/15.2	24.1/14.9
	FaShapley	**26.5/16.1**	**26.3/16.0**	**26.0/15.8**

Certified Robustness on Tiny-ImageNet. We also evaluate on Tiny-ImageNet [33], which is known challenging to be certified under the ℓ_∞ norm bounded perturbations. In this paper, we focus on *deterministic verification* which provides a lower bound to the certified accuracy and is more meaningful on pruned networks with bound propagation based certification approaches; while HYDRA applies randomized smoothing to large models to get the probabilistic robustness certification. To the best of our knowledge, existing deterministic verification methods *cannot scale up to large-scale datasets such as ImageNet*, so besides CIFAR-10 and SVHN, we evaluate

TABLE IX
CLEAN / CERTIFIED ACCURACY (%) OF FASHAPLEY COMPARED WITH DIFFERENT NEURON SELECTION CRITERIA ON CIFAR-10 VIA UNSTRUCTURED PRUNING. THE ORIGINAL MODEL (CNN-4) IS TRAINED WITH CROWN-IBP AND THE PERFORMANCE BEFORE PRUNING IS 54.0/42.5.

	Pruning Ratio	99.5%	99%	97%
w/o FT	LWM	14.9/13.1	22.4/18.9	48.4/37.2
	Grad	10.0/10.0	10.0/10.0	10.2/10.1
	Taylor	16.0/13.1	24.2/17.2	**51.7**/40.1
	FaShapley	**23.1/20.4**	**33.3/26.9**	51.5/**40.7**
w/ FT	LWM	15.9/15.9	24.9/20.6	50.6/39.4
	Grad	10.0/10.0	11.5/11.3	13.2/12.3
	Taylor	20.8/15.7	33.2/22.0	**52.7**/41.0
	FaShapley	**40.8/33.9**	**47.0/37.7**	52.4/**42.5**

TABLE X
CLEAN / CERTIFIED ACCURACY (%) OF FASHAPLEY COMPARED WITH DIFFERENT NEURON SELECTION CRITERIA ON CIFAR-10 VIA STRUCTURED PRUNING. THE ORIGINAL MODEL (CNN-7) IS TRAINED WITH CROWN-IBP AND THE PERFORMANCE BEFORE PRUNING IS 60.3/46.8.

	Pruning Ratio	80%	85%	90%
w/o FT	LWM	44.7/34.9	12.1/11.2	10.0/10.0
	L2-norm	48.4/40.0	15.5/14.1	10.0/10.0
	Grad	54.4/43.1	39.9/33.0	10.0/10.0
	Taylor	54.7/43.8	18.4/15.6	10.0/10.0
	Stab	32.7/28.5	28.5/25.7	10.0/10.0
	FaShapley	**55.3/44.2**	**41.2/34.9**	**15.7/15.1**
w/ FT	LWM	52.1/39.8	28.5/22.0	10.0/10.0
	L2-norm	53.9/41.2	36.1/25.2	10.0/10.0
	Grad	54.7/42.9	**48.1**/37.2	10.0/10.0
	Taylor	55.3/43.8	45.7/36.0	10.0/10.0
	Stab	49.8/39.0	40.0/32.8	12.5/11.2
	FaShapley	**55.5/43.9**	45.6/**38.0**	**19.5/14.2**

on Tiny-ImageNet to compare with the SOTA verification results. Auto-LiRPA is the first work to provide the ℓ_∞-certified robustness scaling up to Tiny-ImageNet. We leverage Auto-LiRPA as the certification method and take the *first* step to provide certified robustness for the pruned DNNs on Tiny-ImageNet. For better comparisons with HYDRA on a larger model and dataset, we conduct experiments using HYDRA and FaShapley by pruning a pre-trained WideResNet 28-10 network which is the most effective network in this setting. The results in Table VI show that for both structured pruning and unstructured pruning, FaShapley outperforms HYDRA to achieve high certified robustness on pruned networks on Tiny-ImageNet with and without fine-tuning via auto-LiRPA verification. In addition, we compare FaShapley with a set of neuron importance selection baselines, including the weight-based protocol (LWM), gradient-based protocol (Grad), and Taylor-based protocol (Taylor). The results in Table VII demonstrate that FaShapley can scale up to Tiny-ImageNet and outperform the strong baselines for unstructured pruning. The effectiveness of FaShapley on Tiny-ImageNet for structured pruning is demonstrated in Table VIII.

Comparison with Different Neuron Selection Criteria. In the literature of model pruning, there are many competitive criteria which aim to evaluate the importance of weights/filters in DNNs: weight-based (LWM) [76], L2 norm-based [26], gradient-based (Grad) [51], and Taylor-based (Taylor) [35, 52, 58] protocols. The weight-based protocol selects the importance score as the magnitude of the neuron weights. The gradient-based protocol selects the importance score as the gradients of weights. The Taylor-based protocol selects the importance score as the magnitude of the multiplication of the neuron weights and the gradients of weights. To demonstrate the effectiveness of FaShapley and differentiate it from score-based neuron evaluation methods, we compare FaShapley with these criteria. The evaluations of neuron importance with different criteria are all performed after convergence for fair comparisons. For structured pruning, we further propose a new baseline to consider the **stability of neurons**, which is an important factor for certified robustness of pruned models. The new criterion prunes unstable neurons (a neuron is unstable when the post-ReLU value can be both zero and positive for a specific perturbation budget of the input), which do harm to the certified robustness since this type of neurons can loosen the bound in the robustness verification.

TABLE XI
CLEAN / CERTIFIED ACCURACY (%) OF FASHAPLEY COMPARED WITH DIFFERENT NEURON SELECTION CRITERIA ON SVHN VIA UNSTRUCTURED PRUNING. THE ORIGINAL MODEL (CNN-4) IS TRAINED WITH CROWN-IBP AND THE PERFORMANCE BEFORE PRUNING IS 61.5/41.5.

	Pruning Ratio	99.5%	99%	97%
w/o FT	LWM	21.1/18.4	31.5/23.5	60.8/41.1
	Grad	15.9/15.9	15.9/15.9	13.2/9.7
	Taylor	45.0/26.2	57.0/37.2	61.0/41.0
	FaShapley	**45.5/30.7**	**57.4/38.7**	**61.6/41.7**
w/ FT	LWM	22.0/18.9	39.6/27.1	60.5/41.1
	Grad	15.9/15.9	16.7/15.6	20.0/16.8
	Taylor	44.4/28.9	**53.5**/38.0	60.1/41.5
	FaShapley	**44.7/33.4**	53.2/**39.0**	**60.7/42.0**

The results in Table IX and Table X show that our proposed FaShapley is much more effective for model pruning toward high certified robustness than these baselines for both unstructured pruning and structured pruning. We think that stability does not bring about expected results due to these two reasons: (1) *it does not benefits the clean accuracy of the model since APoZ [27] prunes the neurons with large probability of being zero based on the assumption that these neurons influence little on the final prediction;* (2) *some unstable neurons could improve the lower bound but are removed in this case.* Note that LWM and L2 norm give the same importance ranking for unstructured pruning, so we only report the results of LWM in Table IX. Comparisons of different neuron selection criteria on SVHN are provided in Table XI. Our extensive evaluation on different datasets shows that the effectiveness of FaShapley is significant under different pruning ratios.

We analyze the advantage of FaShapley over Taylor-based criteria in detail as follows. The Taylor-based pruning criterion is based on the leave-one-out (LOO) strategy and calculates the saliency scores (i.e., the influence of removing one neuron) without sampling. LOO can be viewed as a special case of Shapley value and thus the Taylor-based criterion is actually a special case of our proposed FaShapley. However, with a

TABLE XII
CLEAN / CERTIFIED ACCURACY COMPARED WITH FLYING BIRD (FB) ON CIFAR-10 VIA UNSTRUCTURED PRUNING. THE ORIGINAL MODEL (CNN-7) IS TRAINED WITH CROWN-IBP AND THE PERFORMANCE BEFORE PRUNING IS 60.3/46.8.

Pruning Ratio	99%	97%	95%
FB	48.4 / 38.6	53.9 / 42.8	60.0 / 46.3
FaShapley	**60.0 / 46.6**	**60.8 / 48.0**	**60.3 / 47.5**

TABLE XIII
CLEAN / CERTIFIED ACCURACY COMPARED WITH BAR ON CIFAR-10 VIA STRUCTURED PRUNING. THE ORIGINAL MODEL (CNN-7) IS TRAINED WITH CROWN-IBP AND THE PERFORMANCE BEFORE PRUNING IS 60.3/46.8.

Pruning Ratio	80%	90%
BAR	52.3 / 40.7	10.0 / 10.0
FaShapley	**55.5 / 43.9**	**19.5 / 14.2**

TABLE XIV
COMPARISON WITH OTHER SHAPLEY BASED PRUNING APPROACHES. THE MODELS (CNN-7) ARE TRAINED WITH CROWN-IBP ON CIFAR-10 AND PRUNED W/O FINE-TUNING UNDER DIFFERENT PRUNING RATIOS. THE PERFORMANCE BEFORE PRUNING IS 60.3/46.8.

	Pruning Ratio	99%	97%	95%	Runtime
Unstru	SVPrune	-	-	-	> 1000h
	ShapPrune	-	-	-	> 300h
	FaShapley	45.0/35.2	58.5/43.5	61.3/45.0	251s
	Pruning Ratio	85%	80%	75%	Runtime
Struct	SVPrune	24.2/20.1	26.3/22.3	26.5/22.7	$1,283$s
	ShapPrune	**41.5**/34.7	47.0/38.1	47.1/38.5	374s
	FaShapley	41.2/**34.9**	**55.3/44.2**	**57.4/45.1**	6s

reasonable sample-size, our FaShapley computes more effective marginal contributions, which consider multiple combinations and enables a more effective evaluation from a global view [29]. We also have a related discussion about the advantages of Shapley value over LOO in Section III-B, from which we can understand why calculating the saliency scores from multiple combinations is much more effective.

Comparison with the SOTA Model Pruning Algorithm. We also compare FaShapley with SOTA algorithms for both unstructured pruning and structured pruning in the general pruning field. Concretely, we select the SOTA unstructured pruning method Flying Bird (FB) [9] and SOTA structured pruning method BAR [37] as our baselines. Flying bird introduces a dynamic way to perform model pruning during training the masks which allows pruned parameters to be grown back and engaged in the next round of training or pruning. To make Flying Bird pruning adapt to the scenario of achieving high certified robustness, we plug in the certified training loss during the training. The results in Table XII demonstrate that FaShapley outperforms Flying Bird towards high certified robustness under different pruning ratios for unstructured pruning. BAR performs structured pruning via training the masks of channels and filters. Concretely, BAR adds the budget aware regularization and knowledge distillation loss to optimize the learnable mask. Similarly, we also additionally consider corresponding certified training loss for fair comparisons. The results in Table XIII suggest that FaShapley outperforms BAR in terms of both certified robustness and clean accuracy under different pruning ratios for structured pruning. We only report the results of FaShapley after fine-tuning here since both FB and BAR perform pruning and fine-tuning simultaneously. We deem that the reason why FaShapley as a score-based criterion can outperform SOTA pruning methods is that leveraging the power of Shapley value for importance evaluation, our FaShapley-based neuron importance metric is stable and indicative of the location of good subnetworks, whereas in most SOTA pruning methods, the training of the learnable masks that only take binary values often suffers from bad initialization, which is also demonstrated by HYDRA (details in Figure 2 of [59]).

Comparison with SV-Based Pruning Approaches. In the literature of model pruning, there exist works [1, 2, 19, 20] which leverage Shapley value approximation to evaluate the importance of filters/channels and perform structured pruning according to it. To demonstrate the effectiveness of our proposed sample-size optimization and the computational efficiency of our gradient-based approximation, we compare FaShapley with two strong baselines, SVPrune [2] and ShapPrune [20]. SVPrune analyzes the problem of estimating the contribution of hidden units with Shapley values as a principled ranking metric for this task. It leverages the Monte-Carlo methods to approximate the Shapley values and utilizes the addition of Shapley value and its deviation as the importance scores based on empirical observations. ShapePrune utilizes a discarding threshold to ignore marginal contributions based on subgroups with low utility and ϵ-greedy selection which samples useful players more times to estimate Shapley in backdoor defense. We adapt these SV-based neuron importance evaluation methods to our setting by using the same utility function (i.e., the negative of the certified robustness objective) as ours. We perform comparisons without fine-tuning after the pruning step for direct and fair comparisons. The results in Table XIV suggest that 1) for unstructured pruning, SVPrune and ShapPrune cannot scale up to large architectures such as CNN-7, while FaShapley requires significantly less runtime and shows the effectiveness of high certified robustness of pruned networks, and 2) for structured pruning, FaShapley not only demonstrates significant performance improvements but also requires less runtime.

C. Ablation Studies

How to design the utility function for Shapley evaluation? There are two natural ways of designing the utility function, namely using the negative of the training objective and directly using the nagative of cross-entropy (CE) loss. Generally, selecting the negative of the robust training objective as the utility function can make the process of evaluation aware of the certified robustness objective and thus outperforming only using CE loss. Only in some extreme cases where the clean accuracy of the model decreases dramatically after pruning, directly using CE loss can help maintain the ability of the classifier, thus boosting the certified robustness. The results

TABLE XV
ABLATION STUDY RESULTS (CLEAN / CERTIFIED ACCURACY (%)) OF THE **UTILITY FUNCTION** ON CIFAR-10 AND SVHN BASED ON CROWN-IBP TRAINING VIA UNSTRUCTURED PRUNING.

		Pruning Ratio	99.5%	99%	97%	95%
CIFAR-10	CNN-4	Training Loss	10.0/10.0	33.3/26.9	**51.5/40.7**	**53.7/42.5**
		CE Loss	**23.1/20.5**	**35.3/28.3**	51.1/40.6	53.6/42.4
	CNN-7	**Pruning Ratio**	99.5%	99%	97%	95%
		Training Loss	**55.8/40.3**	**59.8/45.8**	**60.3/46.9**	**60.3/46.9**
		CE Loss	46.0/34.5	59.3/44.9	59.8/**46.9**	60.0/**46.9**
SVHN	CNN-4	**Pruning Ratio**	99.5%	99%	97%	95%
		Training Loss	45.5/30.7	**57.4/38.7**	**61.6/41.7**	**61.6/41.7**
		CE Loss	**47.5/31.5**	**57.4**/38.1	61.5/41.6	**61.6/41.7**
	CNN-7	**Pruning Ratio**	99.5%	99%	97%	95%
		Training Loss	**69.1/45.7**	**69.8/47.5**	**69.8/48.0**	**69.8/48.0**
		CE Loss	68.4/44.8	69.5/46.9	69.7/47.6	69.7/47.6

TABLE XVI
ABLATION STUDY RESULTS (CLEAN / CERTIFIED ACCURACY (%)) OF THE **SAMPLE COMPLEXITY** m TO CALCULATE FASHAPLEY VALUES OF NEURONS ON CIFAR-10 BASED ON CROWN-IBP TRAINING VIA UNSTRUCTURED PRUNING.

m	1	2	5	10	20	30	40	60	100
CNN-4	25.3/13.3	36.1/22.3	35.8/21.5	44.5/28.3	47.2/32.0	45.5/30.7	45.7/31.2	48.7/34.3	44.8/30.1
CNN-7	21.9/17.9	62.2/35.4	68.2/44.6	69.1/45.3	69.1/45.6	69.1/45.7	68.7/45.0	68.7/44.9	68.9/45.0

TABLE XVII
ABLATION STUDY RESULTS (CLEAN / CERTIFIED ACCURACY (%)) OF THE **SAMPLE-SIZE** k ON CIFAR-10 AND SVHN BASED ON CROWN-IBP TRAINING. WE SELECT THE MODEL ARCHITECTURE AS CNN-7 IN THE EXPERIMENT.

Sample-size k		0.2	0.4	0.6	0.7	0.8	0.9
CIFAR-10	CNN-4	14.9/12.4	26.5/20.5	30.8/24.9	30.9/26.0	31.2/26.1	**33.3/26.9**
	CNN-7	23.9/19.8	42.8/30.4	55.2/41.9	56.5/42.4	58.0/43.6	**59.8/45.8**
SVHN	CNN-4	24.3/10.6	37.0/18.1	51.5/35.2	51.7/35.8	54.4/37.2	**57.4/38.7**
	CNN-7	21.3/13.6	36.7/21.1	65.8/42.8	69.3/45.9	69.7/47.0	**69.8/47.5**

in Table XV indicate that using the negative of the robust training loss can generally outperform only using CE loss, except for extreme cases where the clean accuracy decreases much (99.5% pruning on CNN-4 and CNN-7). Therefore, for consistency, we leverage the negative of the robust training loss as the utility function.

The results of our FaShapley converge fast. To avoid the exponential computational complexity of exactly calculating marginal contributions, a sampling-based approximation is leveraged, therefore inducing concerns of large sample complexity. To verify our sample complexity analysis, we evaluate FaShapley with different sample-sizes and show the results in Table XVI. It is shown that for both CNN-4 and CNN-7, FaShapley can converge within a small sample-size (i.e., 30 samples in the experiment), and therefore we set sample-size to be 30 for all evaluation without specification. From the runtime results in Table XIV, we can see that with the reduction of sample complexity, FaShapley can scale up to unstructured pruning for which the importance of millions of neurons weights is evaluated.

A relatively larger sample-size is more effective. After fixing the sample complexity as 30, we further explore the results with different sample-sizes and show the corresponding results in Table XVII. We can see that a large sample-size benefits the evaluation of weights. The results align with our analysis from two perspectives: (1) *terms $s_k^i(U)$ with a large sample-size k introduces fairer comparisons among the players since the expectation of the size of the intersection set of sampled subsets S becomes larger,* and (2) *the utility function value of a small number of players is usually low, thus making the marginal contribution calculated based on it meaningless.* Without specification, we set the sample-size k (normalized to $[0, 1]$) to 0.9 for the experiments.

VI. CONCLUSION

In this paper, we design an efficient criterion based on neuron properties to weaken the tradeoff between the certified robustness and the model size. We propose a fast and approximated Shapley method (FaShapley) via gradient-based approximation and sample-size optimization. The method inherits desired properties from Shapley value and overcomes the challenge of expensive computational costs of Shapley approximation. Through theoretical analysis and extensive evaluation, we demonstrate that FaShapley is both computationally efficient and empirically effective to achieve high certified robustness for pruned DNNs.

References

[1] Kamil Adamczewski and Mijung Park. Neuron ranking–an informed way to condense convolutional neural networks architecture. *arXiv preprint arXiv:1907.02519*, 2019.

[2] Marco Ancona, Cengiz Öztireli, and Markus Gross. Shapley value as principled metric for structured network pruning. *arXiv preprint arXiv:2006.01795*, 2020.

[3] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[4] Stanley Bak, Changliu Liu, and Taylor Johnson. The second international verification of neural networks competition (vnn-comp 2021): Summary and results. *arXiv preprint arXiv:2109.00498*, 2021.

[5] Ayoub Ben-Ameur, Andrea Araldo, and Tijani Chahed. Cache allocation in multi-tenant edge computing via online reinforcement learning. *arXiv preprint arXiv:2201.09833*, 2022.

[6] Avrim Blum, Travis Dick, Naren Manoj, and Hongyang Zhang. Random smoothing might be unable to certify ℓ_∞ robustness for high-dimensional images. *Journal of Machine Learning Research*, 21(211):1–21, 2020.

[7] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. 2018.

[8] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo Li. Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks. *2021 IEEE Symposium on Security and Privacy (SP)*, May 2021.

[9] 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.

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

[11] Justin Cosentino, Federico Zaiter, Dan Pei, and Jun Zhu. The search for sparse, robust neural networks. *arXiv preprint arXiv:1912.02386*, 2019.

[12] Hassan Dbouk and Naresh Shanbhag. Generalized depthwise-separable convolutions for adversarially robust and efficient neural networks. *Advances in Neural Information Processing Systems*, 34:12027–12039, 2021.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[14] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1625–1634, 2018.

[15] Shaheen S Fatima, Michael Wooldridge, and Nicholas R Jennings. A linear approximation method for the shapley value. *Artificial Intelligence*, 172(14):1673–1699, 2008.

[16] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*, 2018.

[17] Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2018.

[18] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251. PMLR, 2019.

[19] Amirata Ghorbani and James Y Zou. Neuron shapley: Discovering the responsible neurons. *Advances in Neural Information Processing Systems*, 33:5922–5932, 2020.

[20] Jiyang Guan, Zhuozhuo Tu, Ran He, and Dacheng Tao. Few-shot backdoor defense using shapley estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13358–13367, 2022.

[21] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

[22] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. *arXiv preprint arXiv:1608.04493*, 2016.

[23] Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. *arXiv preprint arXiv:1810.09619*, 2018.

[24] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[25] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.

[26] Yang He, Guoliang Kang, Xuanyi Dong, Yanwei Fu, and Yi Yang. Soft filter pruning for accelerating deep convolutional neural networks. *arXiv preprint arXiv:1808.06866*, 2018.

[27] Hengyuan Hu, Rui Peng, Yu-Wing Tai, and Chi-Keung Tang. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250*, 2016.

[28] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gürel, Bo Li, Ce Zhang,

Dawn Song, and Costas J Spanos. Towards efficient data valuation based on the shapley value. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1167–1176. PMLR, 2019.

[29] Ruoxi Jia, Fan Wu, Xuehui Sun, Jiacen Xu, David Dao, Bhavya Kailkhura, Ce Zhang, Bo Li, and Dawn Song. Scalability vs. utility: Do we have to sacrifice one for the other in data importance quantification? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8239–8247, 2021.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

[32] Aounon Kumar, Alexander Levine, Tom Goldstein, and Soheil Feizi. Curse of dimensionality on randomized smoothing for certifiable robustness. In *International Conference on Machine Learning*, pages 5458–5467. PMLR, 2020.

[33] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[34] Vadim Lebedev and Victor Lempitsky. Fast convnets using group-wise brain damage. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2554–2564, 2016.

[35] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*, 2018.

[36] Carl Lemaire, Andrew Achkar, and Pierre-Marc Jodoin. Structured pruning of neural networks with budget-aware regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9108–9116, 2019.

[37] Carl Lemaire, Andrew Achkar, and Pierre-Marc Jodoin. Structured pruning of neural networks with budget-aware regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[38] Bai Li, Shiqi Wang, Yunhan Jia, Yantao Lu, Zhenyu Zhong, Lawrence Carin, and Suman Jana. Towards practical lottery ticket hypothesis for adversarial training. *arXiv preprint arXiv:2003.05733*, 2020.

[39] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[40] Huichen Li, Linyi Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Nonlinear gradient estimation for query efficient blackbox attack. In *International Conference on Artificial Intelligence and Statistics (AISTATS 2021)*, Proceedings of Machine Learning Research. PMLR, 13–15 Apr 2021.

[41] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1221–1230, 2020.

[42] Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. Tss: Transformation-specific smoothing for robustness certification. In *ACM Conference on Computer and Communications Security (CCS 2021)*, 2021.

[43] Linyi Li, Zexuan Zhong, Bo Li, and Tao Xie. Robustra: Training provable robust neural networks over reference adversarial space. In *IJCAI*, pages 4711–4717, 2019.

[44] 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*, pages 2736–2744, 2017.

[45] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[46] Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Adversarial neural pruning with latent vulnerability suppression. In *International Conference on Machine Learning*, pages 6575–6585. PMLR, 2020.

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

[48] Sasan Maleki, Long Tran-Thanh, Greg Hines, Talal Rahwan, and Alex Rogers. Bounding the estimation error of sampling-based shapley value approximation. *arXiv preprint arXiv:1306.4265*, 2013.

[49] Mana Masuda, Yusuke Sekikawa, Ryo Fujii, and Hideo Saito. Neural implicit event generator for motion tracking. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2200–2206. IEEE, 2022.

[50] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586. PMLR, 2018.

[51] Deepak Mittal, Shweta Bhardwaj, Mitesh M Khapra, and Balaraman Ravindran. Studying the plasticity in deep convolutional neural networks using random pruning. *Machine Vision and Applications*, 30(2):203–216, 2019.

[52] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

[53] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[54] Ozan Ozdenizci and Robert Legenstein. Training adversarially robust sparse networks via bayesian connectivity sampling open website.

[55] Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li. Semanticadv: Generating adversarial examples via attribute-conditioned image editing. In *European Conference on Computer Vision*,

pages 19–37. Springer, 2020.

[56] Adnan Siraj Rakin, Zhezhi He, Li Yang, Yanzhi Wang, Liqiang Wang, and Deliang Fan. Robust sparse regularization: Simultaneously optimizing neural network robustness and compactness. *arXiv preprint arXiv:1905.13074*, 2019.

[57] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[58] Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20378–20389. Curran Associates, Inc., 2020.

[59] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks. *Advances in Neural Information Processing Systems*, 33:19655–19666, 2020.

[60] Lloyd S. Shapley. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952.

[61] Gagandeep Singh, Rupanshu Ganvir, Markus Püschel, and Martin Vechev. Beyond the single neuron convex barrier for neural network certification. 2019.

[62] Rishabh Tiwari, Udbhav Bamba, Arnav Chavan, and Deepak K Gupta. Chipnet: Budget-aware pruning with heaviside continuous approximations. *arXiv preprint arXiv:2102.07156*, 2021.

[63] Boxin Wang, Chejian Xu, Shuohang Wang, Zhe Gan, Yu Cheng, Jianfeng Gao, Ahmed Hassan Awadallah, and Bo Li. Adversarial glue: A multi-task benchmark for robustness evaluation of language models. *NeurIPS*, 2022.

[64] Shiqi Wang, Yizheng Chen, Ahmed Abdou, and Suman Jana. Mixtrain: Scalable training of verifiably robust neural networks. *arXiv preprint arXiv:1811.02625*, 2018.

[65] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Efficient formal safety analysis of neural networks. *arXiv preprint arXiv:1809.08098*, 2018.

[66] Shiqi Wang, Kexin Pei, Justin Whitehouse, Junfeng Yang, and Suman Jana. Formal security analysis of neural networks using symbolic intervals. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 1599–1614, 2018.

[67] 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.

[68] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *Advances in neural information processing systems*, 29:2074–2082, 2016.

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

[70] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

[71] Chaowei Xiao, Ruizhi Deng, Bo Li, Fisher Yu, Mingyan Liu, and Dawn Song. Characterizing adversarial examples based on spatial consistency information for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 217–234, 2018.

[72] Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian McAuley, and Furu Wei. Beyond preserved accuracy: Evaluating loyalty and robustness of bert compression. *arXiv preprint arXiv:2109.03228*, 2021.

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

[74] 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. In *International Conference on Learning Representations*, 2021.

[75] Zhuolin Yang, Linyi Li, Xiaojun Xu, Bhavya Kailkhura, Tao Xie, and Bo Li. On the certified robustness for ensemble models and beyond. *ICLR*, 2021.

[76] Shaokai Ye, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 111–120, 2019.

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

[78] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019.

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

[80] Jiawei Zhang, Linyi Li, Huichen Li, Xiaolu Zhang, Shuang Yang, and Bo Li. Progressive-scale boundary blackbox attack via projective gradient estimation. *ICML*, 2022.

APPENDIX

A. Revision Summary

Feasibility with adversarial empirical training. We compare FaShapley with HYDRA in the empirical adversarial training setting. The models are pre-trained by adversarial training methods in [47] and during the pruning, we select the utility function as the minus of the training loss. The results in Table XVIII demonstrate that FaShapley also achieves higher empirical robustness than HYDRA on different architectures. The effectiveness of FaShapley is not restricted in one task, and as long as we define specific utility functions for different tasks we can perform effective importance evaluation for neurons.

TABLE XVIII
CLEAN / CERTIFIED ACCURACY (%) OF FASHAPLEY COMPARED WITH HYDRA ON CIFAR-10 AFTER PERFORMING 99% UNSTRUCTURED PRUNING ON MODELS ADVERSARIALLY TRAINED FROM [47].

	MobileNet-v2	WideResNet-28-2
HYDRA	39.7/26.4	54.2/34.1
FaShapley	**41.2/29.3**	**55.7/36.2**

Clarification of handling extremely large pruning ratios. For structured pruning with extremely large pruning ratios (e.g. 90%), we observe that constraining that each layer remains at least 5% channels benefits the pruning and thus we adopt the constraints. Since FaShapley can perform neuron importance evaluation from a global view, we do not observe that all the neuron weights in one layer are assigned low values and pruned, and thus we do not set any constraints for unstructured pruning.

Additional results of computational cost comparisons. We theoretically demonstrate that FaShapley has small sample complexity compared to direct Shapley calculation in Section IV.B and also empirically show the efficiency of FaShapley compared to SV-based pruning approaches in Table XIV. Following the suggestion, we further compare the runtime of FaShapley calculation with HYDRA optimization to show the effeciency of FaShapley in Table XXI. The results in Table XXI also suggest that the computation cost of pruning is much smaller compared with the standard training of the model. The evaluation is done on a single RTX 2080 Ti GPU.

TABLE XIX
CLEAN / CERTIFIED ACCURACY (%) AND RUNTIME OF FASHAPLEY COMPARED WITH HYDRA ON TINY-IMAGENET UNDER 70% UNSTRUCTURED PRUNING MOBILENET-V2.

	Clean / Certified accuracy (%)	Runtime
HYDRA	17.0/8.1	22,524s
FaShapley	**25.2/13.6**	1,462s

Visualization of the distribution of remaining weights in convolutional layers. We evaluate the distribution of remaining weights in different channels of the convolution layer. The results in Figure 3 suggest that a large fraction of channels are of low importance and most weights are pruned in them. The observation aligns with the structured pruning evaluation in which the performance only drops a little even with pruning 80% channels (see CNN-7 on CIFAR-10 in Table III).

TABLE XX
RUNTIME OF NEURON IMPORTANCE EVALUATION AND PRUNING ON CIFAR-10 FOR CNN-7 WITH CROWN-IBP. THE PERFORMANCE IS EVALUATED WITH CLEAN / CERTIFIED ACCURACY (%).

	Runtime	Performance
SVPrune	1,300h	-
+sample-size optimization (k=0.9)	700h	-
+gradient approximation	467s	50.2 / 32.7
+(k=0.9)+gradient approximation	251s	**59.8 / 45.8**

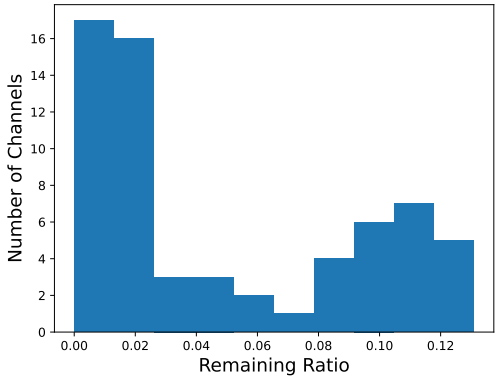


Fig. 3. The distribution of the remaining weights of channels in the last convolutional layer of CNN-7 under 95% unstructured pruning on CIFAR-10.

More explanations of Theorem 2. m in Equation 7 is the sample complexity of FaShapley while m_{sv} in Equation 8 is the sample complexity of Monte-Carlo Shapley approximation. As stated in the remark of Theorem 2, since m_{sv} in Equation 8 is larger than m, we theoretically prove that FaShapley is more computationally efficient than direct Shapley approximation.

Further results on the computational time over multiple instances and datasets. We evaluate FaShapley on CNN-4, CNN-7 following HYDRA for fair comparisons and also provide results of pruning WideResNet-28-10 on Tiny-ImageNet with SOTA LiRPA verification. We further report certified robustness and computational costs of pruning MobieNet-v2 in Table XIX to show that the effectiveness and efficiency of FaShapley generalizes to different architectures.

Additional evaluation of the speedup of FaShapley. We add evaluation to show the speedup of FaShapley from three perspectives: 1) influence of sample-size optimization and gradient approximation on runtime of calculation, 2) results of runtime for different sample-size k, and 3) runtime of pruning compared to HYDRA. The results in Table XX suggest the influence of sample-size optimization and gradient-approximation separately. We demonstrate that the gradient approximation benefits the calculation efficiency a lot and the sample-size optimization further improves the speed-up and the effectiveness. The results in Table XXII show runtime for

TABLE XXI
RUNTIME OF THE STANDARD TRAINING, HYDRA PRUNING, AND FASHAPLEY PRUNING FOR DIFFERENT ARCHITECTURES ON CIFAR-10 WITH CROWN-IBP.

	standard training	pruning without fine-tuning	pruning with fine-tuning
HYDRA (CNN-4)	23,451s	2,132s	5,583s
FaShapley (CNN-4)	23,451s	103s	3,562s
HYDRA (CNN-7)	3,826s	3,547s	7,927s
FaShapley (CNN-7)	3,826s	251s	4,821s
HYDRA (WideResNet-28-10)	273,632s	21,632s	48,982s
FaShapley (WideResNet-28-10)	273,632 s	1,648s	29,421s

TABLE XXII
RUNTIME AND RESULTS OF NEURON IMPORTANCE EVALUATION FOR DIFFERENT SAMPLE-SIZE K WITH FASHAPLEY ON CIFAR-10 FOR CNN-7 WITH CROWN-IBP.

k	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Runtime	857s	763s	677s	601s	529s	454s	390s	311s	251s	183s
Clean / Certified accuracy (%)	15.0/9.2	23.9/19.8	34.2/25.3	42.8/30.4	48.7/33.6	55.2/41.9	56.5/42.4	58.0/43.6	59.8/45.8	55.5/41.5

TABLE XXIII
RUNTIME OF HYDRA PRUNING, AND FASHAPLEY PRUNING FOR DIFFERENT ARCHITECTURES ON CIFAR-10 WITH CROWN-IBP.

	pruning without fine-tuning	pruning with fine-tuning
HYDRA (CNN-4)	2,132s	5,583s
FaShapley (CNN-4)	103s	3,562s
HYDRA (CNN-7)	3,547s	7,927s
FaShapley (CNN-7)	251s	4,821s
HYDRA (WideResNet-28-10)	21,632s	48,982s
FaShapley (WideResNet-28-10)	1,648s	29,421s

different sample-sizes k. We observe that a large sample-size benefits the efficiency because for large sample-sizes we only need to remove a small fraction of neurons which induces less computational cost. Notice that in baseline SVPrune with permutation-based Monte-Carlo approximation the expected sample size k is 0.5 and thus being less efficient that FaShapley. In Table XXIII, we further compare the runtime of FaShapley calculation with HYDRA optimization to show the effeciency of FaShapley.

Discussion of relations to SHAP [45]. There are several differences between SHAP and Shapley values. (1) SHAP [45] approximates the **feature** Shapley (importance) of a **trained** model under the assumption of feature independence and model linearity. Since it evaluates the feature importance during test time, meaning that the model doesn't need to be retrained, it's less computationally expensive. On the contrary, Shapley value by definition evaluates the importance of "training instances", meaning that for every combination of selected instances the model needs to be retrained which is much more expensive. (2) SHAP uses a linear explanation model to locally approximate the original model to satisfy the linearity assumption, and then approximates the importance of features regarding the linear model. However, our gradient-based Shapley approximation is directly derived from the Taylor expansion of the utility function and does not depend on any assumptions of the model. Therefore, SHAP and FaShapley approximate Shapley from two different perspectives, special function properties (i.e. linearity) and function value approximation. Since the gradients with respect to the features are computable, the efficiency of SHAP can be also improved using gradient expectation when computing the conditional expectation of the utility. With the model linearity assumption, the second-order error of Taylor expression is exactly zero, and thus the approximation error only comes from the sampling error.

Discussion of the motivation and application of results without fine-tuning and the influence of fine-tuning for FaShapley in more detail. Pruning without fine-tuning only requires access to a small validation set (i.e. 1000 images for CIFAR-10, SVHN, TIny-ImageNet) instead of the full training set in the fine-tuning step. Fine-tuning is also time-consuming compared to the pruning step. Furthermore, since we propose a novel neuron evaluation criteria in the pruning step, results without fine-tuning exclude the influence of fine-tuning and directly show the effectiveness of pruning. To conclude, we report the results without fine-tuning for their meaning in resource-limited scenarios and for fair comparisons with baselines which report the pruning w/o fine-tuning setting.

In general, fine-tuning can improve the performance of pruning in terms of clean accuracy and certified accuracy.

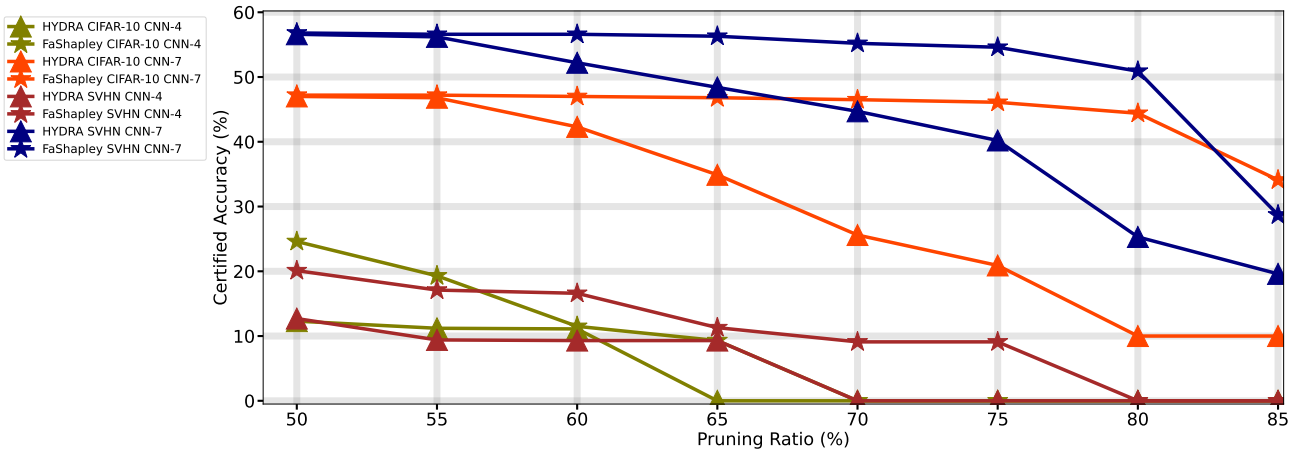


Fig. 4. Certified accuracy (%) after pruning with HYDRA and FaShapley under structured pruning with MixTrain training and IBP verification.

For low performance of pruning, fine-tuning brings large improvements (e.g. under 99% pruning on CIFAR-10 for CNN-4 in Table I), while for high performance close to the upper bound (i.e. performance without pruning), fine-tuning brings no improvements (e.g. under 99% pruning on SVHN for CNN-7 in Table I).

Explanations of certain results in Table I. For 97% and 95% unstructured pruning on CNN-7 on SVHN with CROWN-IBP, the clean and certified accuracy after pruning with FaShapley (i.e. 69.8/48.0) reaches the upper bound (i.e. performance of pre-trained model without pruning 69.5/47.9). It is reasonable and observable that for pruning ratios with which we can achieve the upper bound performance the results are of low variance (e.g. CNN-7 on CIFAR-10 in Table I).

Improved presentation of results. We present the results in Table V systematically in Figure 4 within a consistent range of pruning ratios. From Figure 4, we can see that FaShapley outperforms HYDRA in terms of certified accuracy for different architectures and pruning ratios on CIFAR-10 and SVHN.

B. Complete Proofs

Theorem 3. *FaShapley $F^i(U,k)$ of player i with sample-size k satisfies four desired properties as follows:*
(1) linearity: if $U(S) = t_1U_1(S) + t_2U_2(S)$, then $F^i(U,k) = t_1F^i(U_1,k) + t_2F^i(U_2,k)$.
(2) symmetry: if $\forall S \subseteq I\backslash\{i,j\}$ and $|S| = k$, $U(S\cup\{i\}) = U(S\cup\{j\})$, then $F^i(U,k) = F^j(U,k)$.
(3) nullity: if $\forall S \subseteq I\backslash\{i\}$ and $|S| = k$, $U(S\cup\{i\}) = U(S) + U(\{i\})$, then $F^i(U,k) = U(\{i\}) - U(\{\phi\})$.
(4) efficiency: $\sum_{i=1}^N F^i(U,k) = N[\mathbb{E}_{S\subseteq I,|S|=k+1}U(S) - \mathbb{E}_{S\subseteq I,|S|=k}U(S)]$.

Proof. We give the proof of linearity, symmetry, and nullity sequentially as follows: (1) Proof of linearity:

$$\begin{aligned}F^i(U,k) =&\mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[U(S\cup\{i\}) - U(S)]\\ =&\mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[(t_1U_1(S\cup\{i\}) - t_1U_1(S))]\\ &+\mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}(t_2U_2(S\cup\{i\}) - t_2U_2(S))]\\ =&t_1\mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[(U_1(S\cup\{i\}) - U_1(S))]\\ &+t_2\mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[(U_2(S\cup\{i\}) - U_2(S))]\\ =&t_1F^i(U_1,k) + t_2F^i(U_2,k)\end{aligned} \tag{9}$$

(2) Proof of symmetry:

$$\begin{aligned}F^i(U,k) =&\mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[U(S\cup\{i\}) - U(S)]\\ =&\mathbb{E}_{S\subseteq I/\{i,j\},\ |S|=k-1}[U(S\cup\{i,j\}) - U(S\cup\{j\})]\\ &+\mathbb{E}_{S\subseteq I/\{i,j\},\ |S|=k}[U(S\cup\{i\}) - U(S)]\\ =&\mathbb{E}_{S\subseteq I/\{i,j\},\ |S|=k-1}[U(S\cup\{i,j\}) - U(S\cup\{i\})]\\ &+\mathbb{E}_{S\subseteq I/\{i,j\},\ |S|=k}[U(S\cup\{j\}) - U(S)]\\ =&\mathbb{E}_{S\subseteq I/\{j\},\ |S|=k}[U(S\cup\{j\}) - U(S)]\\ =&F^j(U,k)\end{aligned} \tag{10}$$

(3) Proof of nullity:

$$\begin{aligned}F^i(U,k) &= \mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[U(S\cup\{i\}) - U(S)]\\ &= \mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[U(S) + U(\{i\}) - U(S)]\\ &= U(\{i\}) - U(\{\phi\})\end{aligned} \tag{11}$$

(4) Proof of efficiency:

$$\begin{aligned}\sum_{i=1}^N F^i(U,k) =&\sum_{i=1}^N \mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[U(S\cup\{i\}) - U(S)]\\ =&\sum_{i=1}^N \mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[U(S\cup\{i\})]\\ &-\sum_{i=1}^N \mathbb{E}_{S\subseteq I/\{i\},\ |S|=k}[U(S)]\\ =&N[\mathbb{E}_{S\subseteq I,|S|=k+1}U(S) - \mathbb{E}_{S\subseteq I,|S|=k}U(S)]\end{aligned} \tag{12}$$

□

Fact A.1. *There exist numbers a, b $(0 < a < b)$, which satisfy the following inequality,*

$$a|S| \le U(S) \le \min(b|S|, U_{max}) \tag{13}$$

Theorem 4. *With probability at least δ and estimation error ϵ, the sample complexity m for FaShapley is lower bounded as follows:*

$$m \ge \begin{cases} \lceil\frac{[(b-a)k+b]^2\ln\frac{2}{\delta}}{2\epsilon^2}\rceil & 0 \le k \le \lfloor U_{max}/b\rfloor - 1\\ \lceil\frac{(-ak+U_{max})^2\ln\frac{2}{\delta}}{2\epsilon^2}\rceil & \lfloor U_{max}/b\rfloor - 1 < k \le N-1\end{cases} \tag{14}$$

With probability at least δ and estimation error ϵ, the sample complexity m_{sv} for Shapley value approximation has the lower bound as follows:

$$m_{sv} \ge \lceil\frac{U_{max}^2\ln\frac{2}{\delta}}{2\epsilon^2}\rceil \tag{15}$$

Proof. Define $r_i^k = \max(U(S\cup\{i\}) - \min(U(S))$ for $S \subseteq I/\{i\}$ and $|S| = k$. Then according to Fact A.1, we have:

$$r_i^k = \begin{cases} (b-a)k + b & 0 \le k \le \lfloor U_{max}/b\rfloor - 1\\ -ak + U_{max} & \lfloor U_{max}/b\rfloor - 1 < k \le N-1\end{cases} \tag{16}$$

If X is the sum of l independent random variables $x_1, ..., x_l$, each of which is bounded by the lower bound a_i and the upper bound b_i. For all $t>0$, the hoeffding's inequality can be written as follows,

$$Pr(|X - E|X|| \ge t) \le 2\exp(-\frac{2t^2}{\sum_{i=1}^l(b_i-a_i)^2}) \tag{17}$$

Let $q_k^i(U) = U(S\cup\{i\}) - U(S)$ for $S \subseteq I$ and $|S| = k$ be a random variable of the marginal contribution. For simplicity, we denote $q_k^i(U)$ as q. Suppose that we have m samples $q_1, ..., q_m$ and $Q = \sum_{j=1}^m q_j$. Then the approximated FaShapley value is $F^i(U,k) = \frac{1}{m}\sum_{i=0}^m q_i$. We also denote $F^i(U,k)$ as F for simplicity. Let $X = Q$, $x_i = q_i$, and $l = m$ in Inequality 17, we have:

$$\begin{aligned}Pr(F - E(F) \ge \epsilon) &= Pr(Q - E(Q) \ge m\epsilon)\\ &\le 2\exp(-\frac{2m^2\epsilon^2}{m(b_i-a_i)^2}) \le \delta\end{aligned} \tag{18}$$

Equality $r_k^i = b_i - a_i$ holds by definition. Therefore, we have the upper bound of $Pr(F - E(F) \ge \epsilon)$ as follows :

$$\begin{cases} 2\exp(-\frac{2m^2\epsilon^2}{m[(b-a)k+b]^2}) \le \delta & 0 \le k \le \lfloor U_{max}/b\rfloor - 1\\ 2\exp(-\frac{2m^2\epsilon^2}{m(-ak+U_{max})^2}) \le \delta & \lfloor U_{max}/b\rfloor - 1 < k \le N-1\end{cases} \tag{19}$$

Solving the inequality, we can get the bound of sample complexity m as follows,

$$m \ge \begin{cases} \lceil\frac{[(b-a)k+b]^2\ln\frac{2}{\delta}}{2\epsilon^2}\rceil & 0 \le k \le \lfloor U_{max}/b\rfloor - 1\\ \lceil\frac{(-ak+U_{max})^2\ln\frac{2}{\delta}}{2\epsilon^2}\rceil & \lfloor U_{max}/b\rfloor - 1 < k \le N-1\end{cases} \tag{20}$$

For direct Shapley value approximation, define $r_{sv} = \max_{S\subseteq I}U(S) - \min_{S\subseteq I}U(S)$. According to Fact A.1, $r_{sv} = U_{max}$. Plugging $r_{sv} = U_{max}$ into Inequality 18 and solve it, we have:

$$m_{sv} \ge \lceil\frac{U_{max}^2\ln\frac{2}{\delta}}{2\epsilon^2}\rceil \tag{21}$$

□

C. Prior Reviews and Our Revisions

Review 1: Authors should present the motivation to choose WideResNet over ResNet in their experiments. WideResNet is overparameterized and it is of more interest to compress models such as ResNet. For such models, analyzing the stability of certified robustness at different levels of pruning budget would add value to this work. Further, results are demonstrated for a larger model of relatively small-scale datasets – TinyImagenet.

Our Revision: We illustrate the rationale of selecting WideResNet as the network for evaluation on Tiny-ImageNet in the model architecture paragraph of Section V-A. Concretely, we evaluate on WideResNet 28-10 network for Tiny-ImageNet mainly because the SOTA verification LiRPA achieves the highest certified robustness with the type of network (Table 2 in [73]), and thus we use the same architecture for fair comparison purposes.

Review 2: In the generic pruning literature, there already exist methods that could outperform Shapley in terms of network pruning. It is of interest to investigate the certified robustness of pruned models obtained with this method. I would like the authors to comment on why Shapely is needed for certified robustness and not one of the SOTA pruning methods (BAR, Chipnet or CHIP) be tested for it?

Our Revision: Besides the comparisons with the SOTA unstructured pruning method Flying Bird, we additionally compare with the SOTA pruning method BAR and provide the results in Table Table XIII. The results suggest that FaShapley outperforms BAR in terms of both certified robustness and clean accuracy under different pruning ratios. We also add more analysis about the advantage of FaShapley over SOTA pruning methods in that paragraph. We deem that the reason why FaShapley as a score-based criterion can outperform SOTA pruning methods is that our FaShapley-based neuron importance metric is stable and indicative of the location of good subnetworks; whereas the training of learnable mask in most SOTA methods suffers from bad initialization, which is also demonstrated by HYDRA (Figure 2 in [e]).

Review 3: For the Tiny ImageNet, the authors do not report the HYDRA results and only compare their method with different pruning criterions. More experiments should be conducted to show the author's method is better than HYDRA.

Our revision: We add comparisons with HYDRA on Tiny-ImageNet and provide the results in Tab VI. The results show that FaShapley outperforms HYDRA to achieve high certified robustness on pruned networks on Tiny-ImageNet via auto-LiRPA verification. Note that Tiny-ImageNet with WideResNet 28-10 is the large-scale evaluation we can achieve with deterministic verification, which requires the bound propagation in neurons and is extremely time-consuming. Therefore, the additional evaluation further demonstrates the effectiveness of FaShapley. We include the illustration with more details in Section V-B.

Review 4: I suggest adding the explanation for the selection of pruning ratio in the experiments.

Our Revision: We add a separate paragraph in Section V-A to illustrate how we select the pruning ratios in the experiments. Concretely, we follow the criterion that the comparison is done under an arithmetic sequence of pruning ratios such that FaShapley achieves comparable certified accuracy and clean accuracy as the pre-trained model for the smallest pruning ratio and the baselines perform poorly for the largest pruning ratio. Therefore, different methods can be compared in a wide and meaningful range of pruning ratios.

Review 5: Is the proposed gradient-based estimation for Shapley value better than other estimation methods, such as the sampling-based estimation used in "Few-shot Backdoor Defense Using Shapley Estimation"? Please provide comparison results on computational efficiency, certified robustness, and clean accuracy.

Our Revision: We add comparisons with the SV-based neuron importance evaluation method ShapPrune and show the results in Table XIV. The results suggest that 1) for unstructured pruning, ShapPrune cannot scale up to large architectures such as CNN-7, while FaShapley requires significantly less runtime and shows the effectiveness of high certified robustness of pruned networks, and 2) for structured pruning, FaShapley not only demonstrates significant performance improvements but also requires less runtime. All the analysis is added in Section V-B.