The Trichromatic Strong Lottery Ticket Hypothesis: Neural Compression With Three Primary Supermasks

Ángel López García-Arias[1,*] **Yasuyuki Okoshi**[2] **Hikari Otsuka**[2]
Daiki Chijiwa[1] **Yasuhiro Fujiwara**[1] **Takeuchi Susumu**[1] **Masato Motomura**[2]

[1]NTT Corporation, Japan. [2]Institute of Science Tokyo, Japan.

Abstract

The Strong Lottery Ticket Hypothesis (SLTH) demonstrated that a high-performing model can be obtained just by pruning a randomly initialized dense neural network by optimizing a pruning mask, known as a supermask. Supermask accuracy has recently been enhanced by incorporating sign flipping or weight scaling. Furthermore, it has been demonstrated that supermask training can be extended to sparse random networks. This work proposes the Trichromatic Strong Lottery Hypothesis (T-SLTH), a generalization of the SLTH that (1) connects supermasks to quantization-aware training, (2) consolidates all existing supermasks into a single design framework based on three additive primary supermasks, and (3) contains novel supermask types that support arbitrary connectivity. In addition to sparsity and quantization, the partial randomness of supermask-based models provides specialized digital hardware accelerators with a unique opportunity for neural compression. The models offered by the T-SLTH set the SoTA for supermask-based models in accuracy-size tradeoff: a ResNet-50 scoring 78.43% on CIFAR-100 can be compressed $38\times$ to 2.51 MB, or even $144\times$ down to 0.66 MB while retaining 74.52% accuracy, and $25\times$ to 4.1 MB while scoring 75.28% on ImageNet.

1 Introduction

Deep neural networks (DNNs) have long been known to be overparametrized, as large portions of their weights can be pruned after training without affecting their accuracy [5, 17]. The sparsity introduced by pruning has been a popular approach for specialized neural engine designers, which have exploited it to compress model size via entropy coding [18, 19] and to reduce the computational cost via sparse computation [18, 24]. Another popular strategy for compressing neural size and reducing computation has been quantization. The traditional FP32 format used for GPU training is known to be unnecessarily large, encouraging efforts to transition to an FP8 format [34] and implementations with even more aggressive quantization [12, 19, 46, 49] down to binarization [30, 36, 48].

Recently, a series of specialized hardware accelerators have exploited the Strong Lottery Ticket Hypothesis (SLTH) [41] for an additional compression vector: *randomness*. The SLTH demonstrated that inference can be performed just by overlaying a binary pruning mask—a supermask—over a randomly initialized neural network, which uncovers a subnetwork that has "won the initialization lottery". Since the random initialization pattern can be simply reconstructed from seed using a random number generator (RNG), it is only necessary to store the sparse supermask [7, 22]. This approach has encouraged multiple hardware implementations [7, 38, 22], applications [10, 35, 47], training method improvements [1, 6, 8, 9, 11, 14, 26, 44, 45, 51], and enhancements of the supermasks with, e.g., quantization [37, 47], sign flipping [6, 27, 25], additional randomness [9, 16], or recurrence [32, 47].

This paper shows that some of these improvements have inadvertently eliminated the exploitable randomness and become a particular case of quantization-aware training (QAT). Through this analysis,

*Correspondence to: <lopez@ieee.org>.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

this work conjectures the *Trichromatic Strong Lottery Ticket Hypothesis* (T-SLTH), a generalization of the SLTH that questions the central role given to connectivity by previous work and clarifies the relation between the SLTH and standard QAT. Following, we propose a novel supermask construction method based on three additive primary supermasks that are combined to produce four secondary supermasks. This framework consolidates the existing supermask types and includes three novel types that support dense and random connectivity, thus extending supermasks to arbitrary connectivity. Furthermore, existing work on recurrent supermasks is applied to complete 14 types of SLTH models, offering hardware designers a flexible design space that makes supermask-based training and compression compatible with the needs of a broader range of computation designs and substrates.

2 Background

2.1 Supermask-Based Training

The Connectivity Supermask (*C*) proposed by the original work on the SLTH finds the "winning" subnetworks—strong lottery tickets (SLTs)—by training a binary supermask [41, 50] using the `Edge-Popup` algorithm [41], based on backpropagation. Each weight is assigned a *score*, updated with the gradients corresponding to the weight. The supermask is built after every update by including the top-$k\%$ positions with the largest score magnitudes. Sparsity may be enforced per layer [41], globally [51], or following a distribution [13, 16]. Alternatively, a fixed score threshold may be set instead of a target sparsity [8, 27]. Supermasks are applied during inference to uncover a subnetwork but not during the backpropagation stage, when straight-through estimation [4] is used. This process is equivalent to quantizing score magnitudes into binary values (i.e., $\{0, 1\}$).

The Signed Supermask (*SSup*) [27] enhanced SLT accuracy and sparsity by copying the sign of each learned score to their corresponding element in the *C* supermask. Effectively, this is equivalent to quantizing the scores into balanced ternary values (i.e., $\{-1, 0, +1\}$). A similar approach has been employed where only the signs are learned, and the sparsity is set by unlearned pruning [6, 25].

The Multicoated Supermask (*MSup*) [37] showed that when `Edge-Popup` finishes pruning edges in the *C* supermask, there is no further way of reducing loss, even though gradients still carry valuable optimization information. *MSup* solved this by obtaining from the same scores a set of N supermasks (N coats) of different sparsity (by defining a list of N top-$k\%$) and bundling them into a supermask of unsigned scalars. As a result, SLT accuracy and sparsity were enhanced. This process can be interpreted as quantizing score magnitudes into unsigned scalars (i.e., $[0..N]$).[2]

The Folded Supermask (*F*) [32] demonstrated that transforming the feed-forward DNN architecture into a recurrent neural network (RNN) enhances the accuracy and size of SLTs. This transformation is performed via folding [29], a neuro-inspired method that transforms repeated computational blocks into iterative computational blocks via weight-sharing (i.e., the reverse operation of RNN unrolling). Supermask sections corresponding to folded blocks are then iterated accordingly.

2.2 Supermask-Based Compression

Storing SLT **random** weights is unnecessary since they can be generated on the fly from the random seed [22]. Since these models are typically trained without biases and with non-affine BatchNorm, the only model data to be read from off-chip memory—the operations with the most significant power and time overheads [23]—is the supermask, which in the case of the *C* supermask it is just 1 bit per weight. This makes it possible to train models small enough to fit entirely in on-chip memory [31].

Furthermore, the **sparsity** of supermasks can be exploited for lossless entropy coding, e.g., using zero run-length encoding [22]. This approach was extended to the multi-coat **quantization** of *MSup* by using a *nested representation*: coat m_n only encodes data corresponding to non-pruned edges in coat m_{n-1}. This representation can also be used for *SSup* (i.e., store only signs of non-pruned weights). Alternatively, integer supermasks may be compressed with Huffman coding [38].

Supermask sparsity can also be exploited to compress **computation** in specialized digital hardware. Zero-masking can be employed as data-gating for power reduction in a dense architecture [22]. Alternatively, sparsity can be exploited in a sparse architecture to skip entire portions of computation [18, 24]. In the case of scalar supermasks, multiplication may be decomposed into a series of binary shifts, replacing power-hungry multiplication hardware with simple shift-adders [38].

[2]The notation $[a..b]$ denotes the interval of all integers between a and b, both included.

Pruning is also being explored in the field of emerging silicon photonic neural accelerators [3]. However, due to analog noise and fabrication uncertainties, simulating sparsity as zeroes is ineffective in analog substrates. Banerjee et al. [3] found that physically eliminating the hardware corresponding to pruned edges enhanced accuracy, power, and robustness. Nonetheless, since a network topology determined at fabrication time completely sacrifices the device's versatility, a method for training supermasks with a dense or a pruned device's arbitrary connectivity would be of interest.

3 The *Trichromatic* Strong Lottery Ticket Hypothesis (T-SLTH)

After a reinterpretation of the SLTH, this section proposes a novel supermask construction method.

3.1 A Reinterpretation of the SLTH

Here, a new supermask type is proposed by combining *SSup* and *MSup*:

The Signed-Multicoated Supermask (*SMSup*) integrates the enhancements of *SSup* and *MSup* by first computing the *MSup* and then applying the score signs in the same manner as *SSup*. This supermask quantizes scores into signed integers (i.e., $[-N..0..{+N}]$, where N is the number of coats), thus optimizing weight connectivity, signs, and magnitudes (through scalars). Therefore, it raises an important question: *does it leave any randomness in the model after training?* And if not, *what is the difference between the SLTH and standard weight quantization?*

Indeed, *SMSup* leaves no randomness, meaning it is equivalent to weight quantization-aware training (QAT). Table 1 collects the randomness left in weight connectivity, signs, and magnitudes by each supermask type, tallying the total as a degree of randomness R from 0 to 3, where $R=0$ corresponds to a fully learned model (i.e., equivalent to quantization), and $R=3$ corresponds to a completely random model. However, the proposed *SMSup* is not the only supermask with $R=0$: depending on the type of weight initialization, *some of the existing supermasks are equivalent to weight quantization.*

Weights are generally initialized from a continuous distribution, such as Kaiming normal (KN) weight initialization [20], which samples from the normal distribution $\mathcal{N}(0, \sigma^2)$ with average 0 and standard deviation σ. However, in the case of supermask-based models, higher accuracy has been reported when using the Signed Kaiming Constant (SK) weight initialization method [32, 41], which samples uniformly from $\{-\sigma, +\sigma\}$. Since σ can be viewed as a common scaling factor and thus be absorbed by the BatchNorm layer [22], SK can be seen as binary weight initialization (i.e., $\{-1, +1\}$). Since SK reduces the randomness of the initial weights to only their signs, *SSup* leaves *no randomness*: it is equivalent to balanced ternary (i.e., $\{-1, 0, +1\}$) weight quantization, where `Edge-Popup`'s *scores* are equivalent to the *latent weights* of standard QAT, and the *strong lottery ticket (SLT)* uncovered by the supermask is equivalent to QAT's *effective weights*. The same is true for *SMSup*.

Under this equivalence view, supermask-based training can be reinterpreted as partially random weight quantization, a particular type of QAT that targets part of the numerical properties of each effective weight (i.e., a subset of connectivity, sign, and magnitude) and leaves the rest with a fixed partial randomness pattern. This view considers pruning as a special case of quantization, placing no particular emphasis on subnetworks. Then, it should be possible to train supermasks where the part left random is the connectivity, i.e., to obtain SLTs with *random* or *dense connectivity*. Based on this, the SLTH can be expanded into the *Trichromatic Strong Lottery Ticket Hypothesis*:

> **The *Trichromatic* Strong Lottery Ticket Hypothesis (T-SLTH)**. *A randomly initialized neural network of arbitrary sparsity contains enough representational power such that—after only training its weight connectivity, signs, magnitudes, or any combination of them—it achieves competitive test accuracy.*

This generalization of the SLTH questions the central role that previous work has given to network topology: the SLTH is possible with or without subnetworks. Appendix B collects comparisons with other Lottery Ticket Hypotheses.

3.2 Trichromatic Supermasks: A Framework Based on Three Additive Primary Supermasks

While *SSup* expanded the *C* supermask's learned *connectivity* with learned *signs*, *MSup* expanded it with learned *magnitudes*, and the proposed *SMSup* learns all *connectivity*, *signs*, and *magnitudes*. This work explores the T-SLTH by detaching these three elements into a framework based on *three additive primary supermasks*, illustrated in Figure 1:

Table 1: Randomness of each trichromatic supermask.

Supermask Combination	W. Init.	Randomness CX	Randomness SIGN	Randomness MAG	R
C	KN	✗	✓	✓	2
	SK			✗	1
S	KN	✓	✗	✓	2
	SK			✗	1
M	KN	✓	✓	✗	2
	SK				2
CS (***SSup***)	KN	✗	✗	✓	1
	SK			✗	0
CM (***MSup***)	KN	✗	✓	✗	1
	SK				1
SM	KN	✓	✗	✗	1
	SK				1
CSM (***SMSup***)	KN	✗	✗	✗	0
	SK				0

CX: Connectivity; MAG: Magnitude.
R: Degrees of randomness.

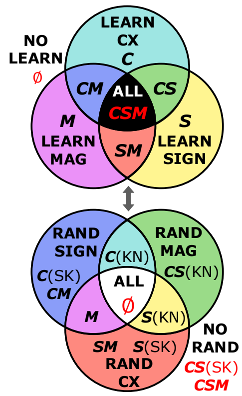


Figure 1: T-SLTH Venn diagrams.

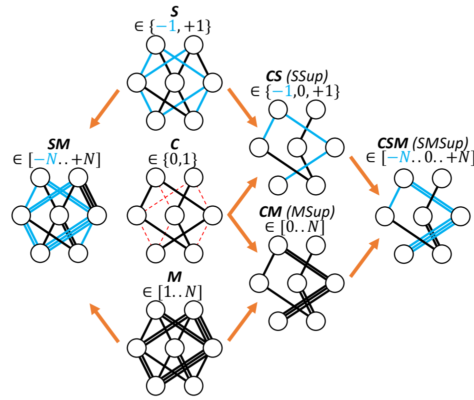


Figure 2: Trichromatic supermasks formed by combining three primary supermasks.

- **The *C* supermask** learns connectivity by quantizing score magnitudes to $\{0, 1\}$ according to a threshold or target sparsity, leaving weight magnitudes and signs untouched.
- **The *S* supermask** learns signs by quantizing scores to their sign $\{-1, +1\}$, leaving magnitudes and connectivity untouched.
- **The *M* supermask** learns magnitudes by quantizing scores to unsigned integer scalars $[1..N]$, leaving signs and connectivity untouched.

Then, *SSup* can be seen as a *secondary* ***CS* supermask** resulting of the superposition of the *C* and *S* supermasks (i.e., $C \cap S$); *MSup* as a ***CM* supermask** (i.e., $C \cap M$); and *SMSup* as a ***CSM* supermask** resulting of the superposition of the three (i.e., $C \cap S \cap M$). The novel *S* and *M* supermasks can be trained in isolation, leaving the arbitrary initial connectivity—whether dense, randomly pruned, or a specific pattern—untouched. Additionally, a third novel supermask can be defined by their superposition: **the *SM* supermask**. This reinterpretation of the supermasks is illustrated in Figure 2.

The construction of a trichromatic supermask $\boldsymbol{T}$, specified in Appendix C, is trivial: $\boldsymbol{T} = \boldsymbol{C} \odot \boldsymbol{M} \odot \boldsymbol{S}$, where $\boldsymbol{C}$ is the original supermask in [41], $\boldsymbol{M}$ is just a special case of *MSup* where the first coat is dense, $\boldsymbol{S}$ just contains score signs, and $\odot$ indicates the Hadamard product. Since random connectivity can be reconstructed from the seed in the same manner as random weights and reduces the number of edges from the beginning, it greatly reduces model size when using the nested representation (see sec. 2.2). The case of dense connectivity is also very compressible, as dense coats can be omitted.

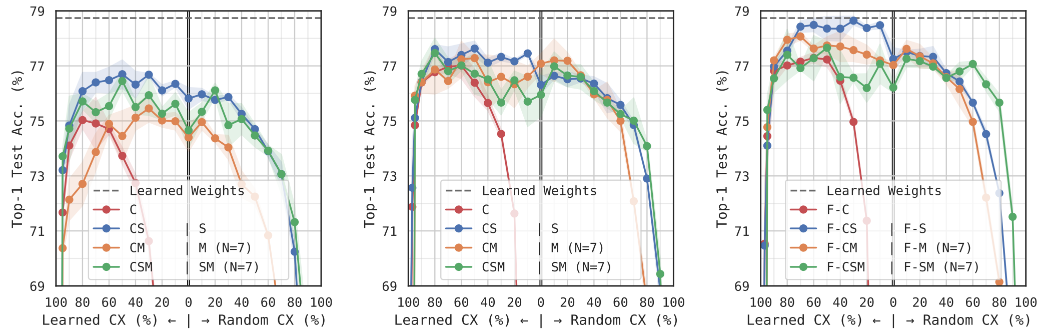
Table 1 collects the randomness analysis of the new three supermasks of arbitrary connectivity. This framework, which defines a total of 7 supermasks, is boosted with folding (*F*) to complete a design space totaling 14 possible SLTH models, summarized in Appendix C, Table 4.

4 Experiments and Results

Following previous work [32, 37, 41], this section evaluates the discussed models using ResNet-50 [21] and image classification datasets. All supermasks are trained using the original `Edge-Popup` [41] with a non-annealed global top-$k\%$ for comparison clarity. The top-$k\%$ list of *M* supermasks is determined using the Linear method in [37]. Following [32], folding is performed only in the last two stages of the model, using unshared BatchNorm parameters. Experiments on CIFAR-100 [28] show a 3-run average of top-1 test accuracy, with a shaded standard deviation, whereas ImageNet [42] results show single-run top-1 validation accuracy. The model size calculation considers an on-chip RNG for re-generating random weights and connectivity from the original seed. Supermasks are compressed using the nested representation, but entropy coding is not considered for ablation, as the compression ratio is very implementation-dependent. Learned BatchNorm parameters are counted as FP32. Training settings are detailed in Appendix A.

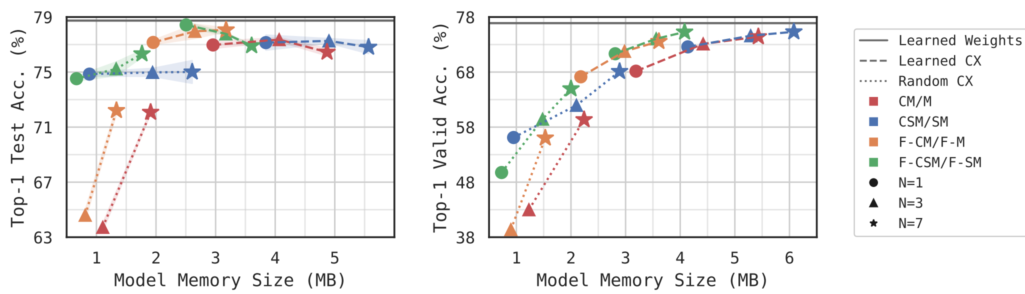
4.1 Supermasks With Learned, Dense, and Random Connectivity

This section examines all the discussed supermask types by testing them for a double range of sparsity: learned connectivity and random connectivity are set on the same axis, separated by the point of dense connectivity. The seven defined supermasks are compared for the KN and SK weight initialization



(a) KN initialization (b) SK initialization (c) Folded, SK initialization

Figure 3: Comparison using ResNet-50 and CIFAR-100 of all supermasks types with connectivity (CX) that is learned (left semiaxis), dense (0%), or random (right semiaxis).



(a) CIFAR-100 (Baseline: 95 MB) (b) ImageNet (Baseline: 102 MB)

Figure 4: Accuracy-model size tradeoff compared on a 70% sparse ResNet-50. When N=1, *CM* and *CSM* are equivalent to *C* and *CS*, respectively. CX: Connectivity; F: folded.

methods in Figure 3a and Figure 3b, respectively. As commonly reported by previous work, SK initialization results in higher classification accuracy for all models.

The results on the learned connectivity semiaxis show that adding *S* and *M* supermasks to the *C* supermask (i.e., *CS* and *CM*) have a similar effect of boosting accuracy and extending the effective sparsity range. From the view of the T-SLTH, this is natural since *S* and *M* supermasks do not require learned connectivity. Indeed, at the point of dense connectivity, they only suffer a slight drop in accuracy, demonstrating that learned connectivity only makes a minor contribution to their efficacy.

The results on the random pruning side confirm the proposed T-SLTH: high accuracy can be achieved by only learning part of the network elements and leaving the rest randomly initialized, and this is not limited to the learned connectivity and random weights of the SLTH. Although the accuracy with random connectivity is lower than with learned connectivity, it does not necessarily mean that there is some intrinsic advantage in finding subnetworks: while the randomness of weights uses the more sophisticated method of Kaiming initialization [20], the random pruning at initialization (RPaI) implemented here is naively random.

Figure 3c shows the same comparison using folded supermasks with SK initialization, demonstrating an additional accuracy boost across the entire connectivity spectrum. Remarkably, *F-CS*s of 10–70% sparsity reach almost the same accuracy as the learned FP32 weight baseline, and *F-SM* of up to 60% random sparsity matches the accuracy of the best performing original supermask (*F-C*).

4.2 Accuracy-Size Tradeoff

Figure 4a and Figure 4b compare the accuracy-size tradeoff of all the supermask types using 70% sparse ResNet-50 on CIFAR-100 and ImageNet, respectively. Results are connected by number of coats (N), following the convention that *C* is a special case of *MSup* (*CM*) where N=1 [37].

In the case of CIFAR-100, *F-SM* reaches 78.43% accuracy, close to the 78.74% of the dense baseline, in only 2.51 MB, a $38\times$ compression rate that sets the SoTA for supermask models in this dataset.

Even with 70% random connectivity, *F-CSM* reaches 74.52% in just 0.66 MB, a 144× reduction that guarantees fitting in on-chip memory even on modest implementations.

Although in the simpler task the gains provided by the *CS* and *CM* supermasks do not compound predictably when combined into *CSM*, on ImageNet the accuracy grows monotonically with the number of primary supermasks and coats. Signing a 7-coated *CM* boosts its 74.43% accuracy to 75.32%, setting the SoTA for an SLTH-based ResNet-50 on this dataset. Even for the folded version, which is known to suffer from lower accuracy on ImageNet [32], the compound benefits of each primary supermask raise its accuracy from 67.14% to 75.28%, almost matching its feedforward counterpart despite being 1.49× smaller. Compared to the 76.89% accuracy of the standard ResNet-50, the 7-coated *CSM* and *F-CSM* reduce the model size by 16.8× and 25.0×, respectively, while only suffering an accuracy drop of 1.5 percentage points. Although extending supermasks to arbitrary connectivity succeeds in producing smaller models (>1 MB), it fails to improve the accuracy-size tradeoff, thus acting as a lower-size extension of the tradeoff set by learned connectivity.

4.3 Comparison With Other Quantization-Aware Training (QAT) Methods

Table 2: QAT methods compared on ImageNet.

Method	Model	Top-1 Acc. (%)	W/A (bits)	Sparsity (%)	INT Size (MB)	Nested Size (MB)
EP (*C*) [41]∗	RN50	68.6	1/32	70	3.1	3.1
Hiddenite (*C*) [22]∗	RN50	70.09	1/BFP8	70	3.1	3.1
WD (*F-CS*) [38]∗	RN50	71.54	2/FP8	70	4.2	2.8
PaB (≈*S*) [6]∗	RN50	63.58	2/32	30	6.4	5.4
LCQ [46]	RN50	**75.1**	2/2	0	**6.4**	6.4
LCQ [46]	RN50	76.3	3/3	0	9.6	>6.4
LSQ [12]	RN50	76.7	4/4	0	12.8	>6.4
Dense Baseline [21]	RN50	76.89	32/32	0	102.2	—
F-C∗	RN50	67.14	1/32	70	2.2	2.2
F-CS∗	RN50	**71.33**	2/32	70	4.2	**2.8**
F-CSM (*N*=7)∗	RN50	**75.28**	4/32	70	8.3	**4.1**
SM (*N*=7)∗	RN50	68.12	4/32	70	9.6	2.9
ReAct [30]	RN18	69.4	1/1	0	1.4	1.4
DDQ [49]	MNV2	**71.8**	4/4	0	**1.8**	—
MNV2 [43]	MNV2	71.66	8/16	0	3.4	—
HFN (*F-C*) [32]∗	WRN50	73.08	1/32	70	5.3	5.3
EP (*C*) [41]∗	WRN50	73.3	1/32	70	8.6	8.6
MPT (*C*) [11]∗	WRN50	74.03	1/32	80	8.6	8.6
CM (*N*=7) [37]∗	RN101	76.5	3/32	70	16.7	9
LSQ [12]	RN101	77.5	3/3	0	16.7	—

W/A: Weights/Activations; ∗: Supermask-based
RNx: ResNet-x; WRNx: Wide-RNx; MNV2: MobileNetV2

Table 2 compares on ImageNet the T-SLTH models with other QAT methods, including some supermask-based methods. When comparing on the same ResNet-50, T-SLTH models offer a better size-accuracy tradeoff than standard QAT methods, which do not induce sparsity: *F-CSM* (N=7) reaches 75.28% accuracy in just 4.1 MB, where a similarly performing 75.1% accurate counterpart trained using LCQ [46] occupies 6.4 MB. This superior tradeoff is possible due to the nested representation offered by supermask sparsity, as a plain INT encoding would result in a poorer tradeoff.

A comparison with deeper or wider supermask-based models suggests that T-SLTH models will scale to higher accuracy while keeping a superior tradeoff. However, comparing with smaller models, such as a DDQ-quantized MobileNetV2 [49], reveals a need for further improvement in the ultra-small size range.

Although non-supermask QAT methods also quantize activations, which could explain their lower tradeoff, the robustness of supermasks to quantized activations has been demonstrated by implementations that quantize them to BFP8 [22], FP8 [38], or even binarize them [11]. A similar table with comparisons on CIFAR-100 is collected in Appendix D.

5 Discussion and Conclusion

Although the models resulting from the Strong LTH are not as accurate as the sparse models with trained weights produced by the *weak* LTH [15], this work aims to demonstrate that the competitive accuracy and high compressibility of partially-random SLTH models makes them a practical choice for efficient hardware implementations. Furthermore, by connecting the SLTH and standard weight quantization, this work hopes to stimulate the application of more sophisticated quantization techniques to partially random supermask-based training. Finally, by making it possible to train supermasks of arbitrary connectivity, this work plans to extend SLTH acceleration, already established in digital hardware, to analog substrates with a dense topology or one pruned at fabrication time without loss of computation versatility.

References

[1] Maxwell M Aladago and Lorenzo Torresani. Slot machines: Discovering winning combinations of random weights in neural networks. In *Proc. Int. Conf. Mach. Learn.*, pages 163–174, 2021.

[2] Yue Bai, Huan Wang, Zhiqiang Tao, Kunpeng Li, and Yun Fu. Dual lottery ticket hypothesis. *arXiv preprint arXiv:2203.04248*, 2022.

[3] Sanmitra Banerjee, Mahdi Nikdast, Sudeep Pasricha, and Krishnendu Chakrabarty. Pruning coherent integrated photonic neural networks. *IEEE J. Sel. Topics Quantum Electron.*, 29(2: Optical Computing):1–13, 2023.

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

[5] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

[6] Xiaohan Chen, Jason Zhang, and Zhangyang Wang. Peek-a-Boo: What (more) is disguised in a randomly weighted neural network, and how to find it efficiently. In *Proc. Int. Conf. Learn. Repr.*, 2022.

[7] Yung-Chin Chen, Shimpei Ando, Daichi Fujiki, Shinya Takamaeda-Yamazaki, and Kentaro Yoshioka. HALO-CAT: A hidden network processor with activation-localized CIM architecture and layer-penetrative tiling. *arXiv preprint arXiv:2312.06086*, 2023.

[8] Hao Cheng, Pu Zhao, Yize Li, Xue Lin, James Diffenderfer, Ryan Goldhahn, and Bhavya Kailkhura. Efficient multi-prize lottery tickets: Enhanced accuracy, training, and inference speed. *arXiv preprint arXiv:2209.12839*, 2022.

[9] Daiki Chijiwa, Shin'ya Yamaguchi, Yasutoshi Ida, Kenji Umakoshi, and Tomohiro Inoue. Pruning randomly initialized neural networks with iterative randomization. In *Proc. Adv. Neural Inform. Process. Syst.*, 2021.

[10] Hee Min Choi, Hyoa Kang, and Dokwan Oh. Is overfitting necessary for implicit video representation? In *Proc. Int. Conf. Mach. Learn.*, pages 5748–5770. PMLR, 2023.

[11] James Diffenderfer and Bhavya Kailkhura. Multi-prize lottery ticket hypothesis: Finding accurate binary neural networks by pruning a randomly weighted network. In *Proc. Int. Conf. Learn. Repr.*, 2021.

[12] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. In *Proc. Int. Conf. Learn. Repr.*, 2020.

[13] Utku Evci, Trevor Gale, Jacob Menick, Pablo Samuel Castro, and Erich Elsen. Rigging the lottery: Making all tickets winners. In *Proc. Int. Conf. Mach. Learn.*, pages 2943–2952, 2020.

[14] Jonas Fischer and Rebekka Burkholz. Lottery tickets with nonzero biases. *arXiv preprint arXiv:2110.11150*, 2021.

[15] Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *Proc. Int. Conf. Learn. Repr.*, 2019.

[16] Advait Harshal Gadhikar, Sohom Mukherjee, and Rebekka Burkholz. Why random pruning is all we need to start sparse. In *Proc. Int. Conf. Mach. Learn.*, pages 10542–10570. PMLR, 2023.

[17] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Proc. Adv. Neural Inform. Process. Syst.*, pages 1135–1143, 2015.

[18] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A Horowitz, and William J Dally. EIE: Efficient inference engine on compressed deep neural network. *Proc. Int. Symp. Comp. Archit.*, 2016.

[19] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding. *Proc. Int. Conf. Learn. Repr.*, 2016.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 1026–1034, 2015.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. and Pattern Recognit.*, pages 770–778, 2016.

[22] Kazutoshi Hirose, Jaehoon Yu, Kota Ando, Yasuyuki Okoshi, Ángel López García-Arias, Junnosuke Suzuki, Thiem Van Chu, Kazushi Kawamura, and Masato Motomura. Hiddenite: 4K-PE hidden network inference 4D-tensor engine exploiting on-chip model construction achieving 34.8-to-16.0TOPS/W for CIFAR-100 and ImageNet. In *Proc. IEEE Int. Solid-State Circuits Conf.*, volume 65, pages 1–3, 2022.

[23] Mark Horowitz. 1.1 computing's energy problem (and what we can do about it). In *Proc. IEEE Int. Solid-State Circuits Conf.*, pages 10–14, 2014.

[24] Yu hsin Chen, Tien-Ju Yang, Joel S. Emer, and Vivienne Sze. Eyeriss v2: A flexible accelerator for emerging deep neural networks on mobile devices. *IEEE J. Emerg. Sel. Top. Circuits Syst.*, 9(2):292–308, 2019.

[25] Cristian Ivan and Razvan Florian. Training highly effective connectivities within neural networks with randomly initialized, fixed weights. *arXiv preprint arXiv:2006.16627*, 2020.

[26] Yusuke Iwasawa, Masato Hirakawa, and Yutaka Matsuo. Soft iEP: On the exploration inefficacy of gradient based strong lottery exploration, 2024. URL `https://openreview.net/forum?id=0XBsK3GsL6`.

[27] Nils Koster, Oliver Grothe, and Achim Rettinger. Signing the supermask: Keep, hide, invert. In *Proc. Int. Conf. Learn. Repr.*, 2022.

[28] Alex Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, Toronto, 2009.

[29] Qianli Liao and Tomaso Poggio. Bridging the gaps between residual learning, recurrent neural networks and visual cortex. *arXiv preprint arXiv:1604.03640*, 2016.

[30] Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. ReActNet: Towards precise binary neural network with generalized activation functions. In *Proc. European Conf. Comput. Vis.*, pages 143–159. Springer, 2020.

[31] Ángel López García-Arias, Masanori Hashimoto, Masato Motomura, and Jaehoon Yu. Hidden-Fold Networks: Random recurrent residuals using sparse supermasks. In *Proc. British Mach. Vis. Conf.*, 2021.

[32] Ángel López García-Arias, Yasuyuki Okoshi, Masanori Hashimoto, Masato Motomura, and Jaehoon Yu. Recurrent residual networks contain stronger lottery tickets. *IEEE Access*, 11: 16588–16604, 2023. doi: 10.1109/ACCESS.2023.3245808.

[33] Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. Proving the lottery ticket hypothesis: Pruning is all you need. In *Proc. Int. Conf. Mach. Learn.*, pages 6682–6691, 2020.

[34] Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, et al. FP8 formats for deep learning. *arXiv preprint arXiv:2209.05433*, 2022.

[35] Hamid Mozaffari, Virat Shejwalkar, and Amir Houmansadr. FSL: Federated supermask learning. *arXiv preprint arXiv:2110.04350*, 2021.

[36] Marina Neseem, Conor McCullough, Randy Hsin, Chas Leichner, Shan Li, In Suk Chong, Andrew Howard, Lukasz Lew, Sherief Reda, Ville-Mikko Rautio, et al. PikeLPN: Mitigating overlooked inefficiencies of low-precision neural networks. 2024.

[37] Yasuyuki Okoshi, Ángel López García-Arias, Kazutoshi Hirose, Kota Ando, Kazushi Kawamura, Thiem Van Chu, Masato Motomura, and Jaehoon Yu. Multicoated supermasks enhance hidden networks. In *Proc. Int. Conf. Mach. Learn.*, pages 17045–17055, 2022.

[38] Yasuyuki Okoshi, Ángel López García-Arias, Jaehoon Yu, Junnnosuke Suzuki, Hikari Otsuka, Thiem Van Chu, Kazushi Kawamura, Daichi Fujiki, and Masato Motomura. WhiteDwarf: 12.24 TFLOPS/W 40 nm versatile neural inference engine for ultra-compact execution of CNNs and MLPs through triple unstructured sparsity exploitation and triple model compression. In *Proc. IEEE Asian Solid-State Circuits Conf.*, 2024. In press.

[39] Laurent Orseau, Marcus Hutter, and Omar Rivasplata. Logarithmic pruning is all you need. In *Proc. Adv. Neural Inform. Process. Syst.*, pages 2925–2934, 2020.

[40] Ankit Pensia, Shashank Rajput, Alliot Nagle, Harit Vishwakarma, and Dimitris S. Papailiopoulos. Optimal lottery tickets via subset sum: Logarithmic over-parameterization is sufficient. In *Proc. Adv. Neural Inform. Process. Syst.*, 2020.

[41] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. and Pattern Recognit.*, pages 11893–11902, 2020.

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

[43] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobilenetV2: Inverted residuals and linear bottlenecks. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. and Pattern Recognit.*, pages 4510–4520, 2018.

[44] Kartik Sreenivasan, Jy-yong Sohn, Liu Yang, Matthew Grinde, Alliot Nagle, Hongyi Wang, Kangwook Lee, and Dimitris Papailiopoulos. Rare gems: Finding lottery tickets at initialization. 2022.

[45] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. volume 33, 2020.

[46] Kohei Yamamoto. Learnable companding quantization for accurate low-bit neural networks. pages 5029–5038, 2021.

[47] Jiale Yan, Hiroaki Ito, Ángel López García-Arias, Yasuyuki Okoshi, Hikari Otsuka, Kazushi Kawamura, Thiem Van Chu, and Masato Motomura. Multicoated and folded graph neural networks with strong lottery tickets. In *Learning on Graphs Conference*, pages 11–1. PMLR, 2024.

[48] Yichi Zhang, Zhiru Zhang, and Lukasz Lew. PokeBNN: A binary pursuit of lightweight accuracy. pages 12475–12485, 2022.

[49] Zhaoyang Zhang, Wenqi Shao, Jinwei Gu, Xiaogang Wang, and Ping Luo. Differentiable dynamic quantization with mixed precision and adaptive resolution. In *Proc. Int. Conf. Mach. Learn.*, pages 12546–12556. PMLR, 2021.

[50] Hattie Zhou, Janice Lan, Rosanne Liu, and Jason Yosinski. Deconstructing lottery tickets: Zeros, signs, and the supermask. In *Proc. Adv. Neural Inform. Process. Syst.*, pages 3597–3607, 2019.

[51] Xiao Zhou, Weizhong Zhang, Hang Xu, and Tong Zhang. Effective sparsification of neural networks with global sparsity constraint. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. and Pattern Recognit.*, 2021.

A Training Settings

Table 3: Training settings for each dataset.

Setting	CIFAR-100	ImageNet
Epochs	200	200
Train/Valid. split	90%/10%	—
Label smoothing	0	0.1
Score init.	Kaiming Uniform	
Optimizer	SGD	
Momentum	0.9	0.875
Weight decay	5E−4	3.05E−5
Batch size	128	256
LR schedule	Cosine Annealing	
Start LR	0.1	0.256
Warmup	0	5

LR: learning rate.

B Comparison With Other Lottery Ticket Hypotheses

- **The (Weak) Lottery Ticket Hypothesis (LTH) [15].**
  A randomly initialized, dense neural network contains a subnetwork (a winning ticket) that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations.

Unlike the proposed T-SLTH, the LTH iterates pruning and training, resulting in a model with higher accuracy, but no compressible randomness.

- **The *Dual* (Weak) Lottery Ticket Hypothesis (DLTH) [2].**
  A randomly selected subnetwork from a randomly initialized dense network can be transformed into a trainable condition, where the transformed subnetwork can be trained in isolation and achieve better at least comparable performance to LTH and other strong baselines.

Although similar to the proposed T-SLTH in that it extends the LTH to arbitrary initial connectivity, the DLTH is also weak (i.e., trains weights).

- **The *Strong* Lottery Ticket Hypothesis (SLTH) [33, 41].**
  A randomly initialized, dense neural network contains a subnetwork (a strong ticket) that is initialized such that—without any weight training—it achieves competitive test accuracy.

The original Strong LTH only considers dense parent networks and is focused on the concept of subnetworks (i.e., pruning), whereas the proposed T-SLTH considers parent networks of arbitrary connectivity and 7 ways of uncovering tickets, including 3 that do not perform pruning.

- **The *Multi-Prize* (Strong) Lottery Ticket Hypothesis (M-SLTH) [11].**
  A sufficiently over-parameterized neural network with random weights contains several subnetworks (winning tickets) that (a) have comparable accuracy to a dense target network with learned weights (prize 1), (b) do not require any further training to achieve prize 1 (prize 2), and (c) is robust to extreme forms of quantization (i.e., binary weights and/or activation) (prize 3).

Although the M-SLTH covers quantization robustness, it is not considered a method for uncovering tickets. Furthermore, its paper only discusses binarization. The T-SLTH shows that quantization itself of different types can uncover strong tickets.

- **The *Disguised* (Strong) Lottery Ticket Hypothesis (D-SLTH) [6].**
  A randomly initialized, dense neural network contains a sparse subnetwork (a disguised strong ticket) that is initialized such that—after unmasking it with some simple transformations—it achieves competitive test accuracy.

Like the *S* supermask proposed in this work, the D-SLTH first prunes the parent network and then trains a sign-flipping mask. It can be considered a special case of the proposed T-SLTH, but since it uses untrained (nor random) pruning, the resulting tickets are less compressible.

C Trichromatic Supermask Construction

After optionally folding the neural structure into a recurrent architecture following [32], the random weight tensor $\boldsymbol{W}^{(l)}_{\text{rand}}$ in each layer l of the model is initialized with a given with initialization method (e.g., KN or SK). Here, arbitrary connectivity may be implemented using RPaI or a given connectivity pattern (e.g., the one corresponding to a particular hardware topology). Then, a score tensor $\boldsymbol{Z}^{(l)}$ and a trichromatic supermask $\boldsymbol{T}^{(l)}$ are defined for each $\boldsymbol{W}^{(l)}_{\text{rand}}$, with the same dimensionality. During inference, forward weights are calculated as

$$\boldsymbol{W}^{(l)} = \boldsymbol{W}^{(l)}_{\text{rand}} \odot \boldsymbol{T}^{(l)}, \tag{1}$$

where $\boldsymbol{T}^{(l)}$ is the trichromatic supermask. During the backward pass, STE [4] is used instead of applying the supermask, and the gradient of each weight $w_{uv} \in \boldsymbol{W}^{(l)}_{\text{rand}}$ is used to update its corresponding score $z_{uv} \in \boldsymbol{Z}^{(l)}$, analogously to `Edge-Popup` [41]. The trichromatic supermask is then reconstructed after each score update as

$$\boldsymbol{T}^{(l)} = \boldsymbol{C}^{(l)} \odot \boldsymbol{M}^{(l)} \odot \boldsymbol{S}^{(l)}, \tag{2}$$

where $\boldsymbol{C}^{(l)}$, $\boldsymbol{M}^{(l)}$, and $\boldsymbol{S}^{(l)}$ are the three primary supermasks, and $\odot$ is the Hadamard product operator. The construction settings are formed by c, m, and s, booleans that determine if the corresponding primary supermask is used or not, and $\mathcal{K}$, a set of N supermask densities (top-k%).

The *C* supermask is generated as

$$\boldsymbol{C}^{(l)}(\boldsymbol{Z}^{(l)}, \mathcal{K}) = \begin{cases} \mathcal{H}(\boldsymbol{Z}^{(l)}, k_0) & c \\ \mathbf{1} & \neg c \end{cases}, \tag{3}$$

in which $k_0 \in \mathcal{K}$ is the first defined top-k%, and $\mathcal{H}$ is the original supermask generating function, defined as

$$\mathcal{H}(\boldsymbol{Z}^{(l)}, k) = \begin{bmatrix} h(z^{(l)}_{1,1}, k) & \cdots & h(z^{(l)}_{U,1}, k) \\ \vdots & \ddots & \vdots \\ h(z^{(l)}_{1,V}, k) & \cdots & h(z^{(l)}_{UV}, k) \end{bmatrix}, \tag{4}$$

where $h(z_{uv}, k)$ is a step function that prunes weight $w_{uv} \in \boldsymbol{W}^{(l)}_{\text{rand}}$ based its corresponding score $z_{uv} \in \boldsymbol{Z}^{(l)}$ and a threshold score z_t calculated from sparsity k:

$$h(z_{uv}, k) = \begin{cases} 1 & |z_{uv}| \geq z_t \\ 0 & |z_{uv}| < z_t \end{cases}. \tag{5}$$

Similar to [37], the *M* supermask is generated by iterating $\mathcal{H}$ over $\mathcal{K}$:

$$\boldsymbol{M}^{(l)}(\boldsymbol{Z}^{(l)}, \mathcal{K}) = \begin{cases} \mathbf{1} + \sum_{n=i}^{N-1} \mathcal{H}(\boldsymbol{Z}^{(l)}, k_n) & m \\ \mathbf{1} & \neg m \end{cases}, \tag{6}$$

where the first density $k_n \in \mathcal{K}$ used in *M* is set to be the second one (k_1) if the first one (k_0) is used in *C* by defining

$$i = \begin{cases} 0 & c \\ 1 & \neg c \end{cases}. \tag{7}$$

Lastly, the *S* supermask is generated as

$$\boldsymbol{S}^{(l)}(\boldsymbol{Z}^{(l)}) = \begin{cases} SGN(\boldsymbol{Z}^{(l)}) & s \\ \mathbf{1} & \neg s \end{cases}, \tag{8}$$

where $SGN(\boldsymbol{Z}^{(l)})$ applies an element-wise modified sign step function defined as

$$sgn(z_{uv}) = \begin{cases} -1 & z_{uv} < 0 \\ +1 & z_{uv} \geq 0 \end{cases}. \tag{9}$$

The three primary supermasks can be combined in different ways to generate 7 possible supermask types, which, combined with the optional folding, totals 14 possible models, collected in Table 4.

Table 4: Models supported by the proposed design framework.

Model	Randomness CX	Randomness SIGN	Randomness MAG	Recurrent
C [41]	✗	✓	▲	✗
CS [27]	✗	✗	▲	✗
CM [37]	✗	✓	✗	✗
CSM	✗	✗	✗	✗
S [25, 6]	▲	✗	▲	✗
M	▲	✓	✗	✗
SM	▲	✗	✗	✗
F-C [32]	✗	✓	▲	✓
F-CS	✗	✗	▲	✓
F-CM [47]	✗	✓	✗	✓
F-CSM	✗	✗	✗	✓
F-S	▲	✗	▲	✓
F-M	▲	✓	✗	✓
F-SM	▲	✗	✗	✓

▲: depends on initialization.

C.1 Trichromatic Supermask Compression

The memory size in bits of a trichromatic supermask $\boldsymbol{T}^{(l)}$ using the nested representation is given by

$$|\boldsymbol{T}^{(l)}| = (|\boldsymbol{C}^{(l)}| + |\boldsymbol{M}^{(l)}| + |\boldsymbol{S}^{(l)}|) \cdot k_r, \tag{10}$$

where k_r is the density after random pruning at initialization, and $|\mathbf{1}| = 0$ (the case where a primary supermask is not used), as it makes no contribution in Eq. (2). The size of each primary supermask (when included) is described below.

Since $\boldsymbol{C}^{(l)}$ includes one bit per weight, its size is given by

$$|\boldsymbol{C}^{(l)}| = |\boldsymbol{W}^{(l)}_{\text{rand}}|, \tag{11}$$

where $|\boldsymbol{W}^{(l)}_{\text{rand}}|$ is the number of weights in layer l.

In $\boldsymbol{M}^{(l)}$, binary coat m_n can be nested under coat m_{n-1}, i.e., coat m_n only encodes the elements that were not pruned in m_{n-1}. When used with $\boldsymbol{C}^{(l)}$, the first coat can be nested under it. Thus, the size is given by

$$|\boldsymbol{M}^{(l)}| = \begin{cases} |\boldsymbol{W}^{(l)}_{\text{rand}}| \cdot (\sum_{n=1}^{N-2} k_n) & c \\ |\boldsymbol{W}^{(l)}_{\text{rand}}| \cdot (1 + \sum_{n=1}^{N-2} k_n) & \neg c \end{cases}. \tag{12}$$

$\boldsymbol{S}^{(l)}$ includes one bit per weight (the sign bit), but when $\boldsymbol{C}^{(l)}$ or $\boldsymbol{M}^{(l)}$ are present, it can be nested under them, signing only the non-pruned elements:

$$|\boldsymbol{S}^{(l)}| = \begin{cases} |\boldsymbol{W}^{(l)}_{\text{rand}}| \cdot k_0 & c|m \\ |\boldsymbol{W}^{(l)}_{\text{rand}}| & \neg(c|m) \end{cases}. \tag{13}$$

In Eq. (10), all supermasks are nested under the random pruning pattern, which is regenerated from seed and thus provides the most compression. It shall be noted that these binary representations are sparse even after nesting—since k_n are set quasi-logarithmic [37] and $\boldsymbol{S}^{(l)}$ is expected to keep a normal distribution—meaning that they can be further compressed using one of the many available lossless entropy coding methods (e.g., run-length encoding or Huffman coding).

D Comparisons on CIFAR-100

Table 5: QAT methods compared on CIFAR-100.

Method	Model	Top-1 Acc. (%)	W/A (bits)	Spar sity (%)	INT Size (MB)	Nested Size (MB)
WD (*F-CS*) [38]∗	RN50	80.29	2/FP8	90	3.80	2.51
ERNet (*C*) [16]∗	RN50	67.67	1/32	99.5	2.96	2.96
Hiddenite (*C*) [22]∗	RN50	70.15	1/BFP8	70	2.96	2.96
Dense baseline [21]	RN50	78.74	32/32	0	94.82	—
S∗	RN50	74.85	1/32	70	2.96	0.88
M (N=7)∗	RN50	72.08	3/32	70	8.88	1.91
SM (N=7)∗	RN50	75.01	4/32	70	11.84	2.61
C∗	RN50	76.97	1/32	70	2.96	2.96
CS∗	RN50	77.14	2/32	70	5.92	3.84
CM (N=7)∗	RN50	76.45	3/32	70	8.88	4.87
CSM (N=7)∗	RN50	76.81	4/32	70	11.84	5.56
F-S∗	RN50	74.52	1/32	70	1.95	0.66
F-M (N=7)∗	RN50	72.21	3/32	70	7.80	1.34
F-SM (N=7)∗	RN50	76.33	4/32	70	7.80	1.76
F-C∗	RN50	77.15	1/32	70	1.95	1.95
F-CS∗	RN50	78.43	2/32	70	3.80	2.51
F-CM (N=7)∗	RN50	78.08	4/32	70	7.80	3.18
F-CSM (N=7)∗	RN50	76.92	4/32	70	7.80	3.60
F-S∗	RN50	77.25	1/32	0	1.95	1.95
F-M (N=7)∗	RN50	77.03	3/32	0	7.80	3.18
F-SM (N=7)∗	RN50	76.21	4/32	0	7.80	3.60
ERNet (*C*) [16]∗	RN101	71.16	1/32	50	5.56	5.56
HFN (*F-C*) [32]∗	RN200	78.90	1/32	70	3.02	3.02
PaB (≈*S*) [6]∗	WRN28	77.81	2/32	70	4.78	4.78
C [32]∗	WRN50	78.59	1/32	70	8.37	8.37
HFN (*F-C*) [32]∗	WRN50	79.16	1/32	70	5.11	5.11

W/A: Weights/Activations; ∗: Supermask-based
RNx: ResNet-x; WRNx: Wide-RNx

E Limitations

Although the claims presented in this work are supported with experimental evidence, no theoretical support is provided. Furthermore, experiments were only carried out on ResNet-50 and only on image classification datasets. Future work shall extend to the T-SLTH the work on the SLTH that offered theoretical proofs [33, 39, 40] and demonstrated its scalability to other models [32, 37, 38, 47] and tasks [47].

The benefits of the proposed models are only available to specialized hardware. When processed on standard hardware (e.g., CPU or GPU), SLTH models offer no computational cost benefit. Furthermore, this work presents no hardware design nor experimental results on specialized hardware. Nonetheless, it references publications describing similar hardware implementations [7, 22, 38].

The benefits of the proposed models are mainly focused on inference. In fact, `Edge-Popup` is slightly more computationally expensive than standard backpropagation. Although it is unclear how to exploit the simplicity of supermask-based training to reduce its training cost, some work on the SLTH has demonstrated that supermasks can be learned using low-precision gradients [6].

For clarity, this work does not employ any of the improvements that have been proposed for `Edge-Popup` [1, 6, 8, 9, 11, 14, 26, 44, 45, 51], nor modern augmentation-regularization strategies, nor distillation or pre-trained models. Future work apply these methods to this work for better results. Additionally, this work does not consider the *weak* LTH [15]—i.e., the trainability of the found tickets—as it focuses on the compressibility opportunity offered by partial randomness, which is not present in trained weak winning tickets.

F Societal Impact Statement

This work presents a framework for designing neural networks for efficient hardware acceleration. It has the potential to help reduce the high computational cost associated with AI applications, which are now quickly becoming ubiquitous, and its associated climatic impact. To the best of our knowledge, this work has no potential negative societal impacts.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: This paper claims to (1) connect the SLTH to weight quantization, (2) consolidate existing supermasks in a single framework, and (3) extend the SLTH to arbitrary connectivity. Section 3 describes claims (1) and (2). Claim (3) is demonstrated in section 4 demonstrate the T-SLTH for learned, dense, and random conectivity.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 4 mentions that there are more advanced SLTH training methods that we do not use. The results discussion mentions that the proposed method is still not the SoTA for the smallest of models, while the conclusion section mentions that weak SLTH models are more accurate. Appendix E covers some additional limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not present theoretical results, although it does present a hypothesis, for which experimental evidence is presented.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The presented method is based on three existing methods for which the code is publicly available ([27, 37, 31]). We present how to combine them, as well as the training settings. Datasets are also well-known and publicly available.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Although we have not released our code, it is built from three pieces of publicly available code, as described in the previous question. Datasets used are publicly available.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details are detailed in section 4 and appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: As explained in section 4, we perform 3 repetitions in the case of CIFAR-100 experiments, for which averages and standard deviations are provided, and a single repetition in the case of ImageNet.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The experiments in this paper use a well-known model (ResNet-50) with common datasets, requiring no special kind of computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

Justification: Appendix F includes a societal impact statement.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used assets are referenced and credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.