Appendix

A Discussion of Extraction with Number-Mapping Strategy

As we mentioned in the main paper (Section 3.3), extracting random numbers from sources with biased distribution will lead to low-precision training hard to converge to good accuracy. But the accuracy can be significantly improved by adopting a number-mapping strategy. For example, we map the original random number to a new representation, e.g., $\{0, 1, 2, 3, 4, 5, 6, 7\} \rightarrow \{3, 1, 2, 7, 4, 6, 5, 0\}$. In other words, the extracted '0' will be converted to '3' and so on during the stochastic rounding process. In this way, if '0' takes the majority of all extracted random numbers, by converting it to '3' (a middle number), it will not make the rounding decision significantly biased to rounding down.

Table A1. Comparison of low-precision training model accuracy using extracted random numbers from different sources. The "NM" stands for the results with number-mapping strategy. We use 6-bit quantization on weights, activations, errors, and gradeints. 6-bit random numbers are extracted for stochastic rounding.

Source	layer-2	layer-10	layer-19	layer-2	layer-18	layer-31
	CIFAR10 on ResNet20			CIFAR100 on ResNet32		
Activation	90.36	89.97	87.40	70.19	70.41	68.66
Activation w/ NM	90.47	90.12	87.73	71.95	72.36	72.74
Error	71.11	72.87	74.91	8.34	11.59	54.22
Error w/ NM	86.97	86.53	90.67	73.10	72.64	72.47
Weight	91.76	91.69	91.62	73.32	73.30	73.39
Weight w/ NM	91.78	91.67	91.64	73.37	73.57	73.63
Gradient	91.88	91.47	91.68	74.06	73.91	73.87
Gradient w/ NM	91.81	91.53	91.69	74.04	73.95	73.83

Table A1 shows the comparison between the accuracy with and without using number-mapping strategy. We can observed that using number-mapping strategy can significantly improve the model accuracy when using extracting random numbers from bad sources (i.e., activations and errors), while it cannot further improve the results from sources with good distribution (i.e., gradients).

B Visualization of Random Number Extraction on Different Sources

Figure A1 shows the visualization of LSB's changing trend of a specific source location along the entire training process on CIFAR-100 using ResNet32. In specific, we randomly select a location from a type of source (e.g., the 3rd weight of a layer) and we keep tracking the value of its LSB along the entire training process. We reshape the 1-D (row) view to the 2-D format for better visualization. Each pixel represents the LSB value at a certain training iteration. And the beige and black color represent value of '1' and '0', respectively. And we can see that the LSB in the error is '0' for the most of time, and the '1' and '0' in the gradient have randomly and evenly distributed. And the model accuracy results also show that using gradients as the source of random number extraction is the best choice.

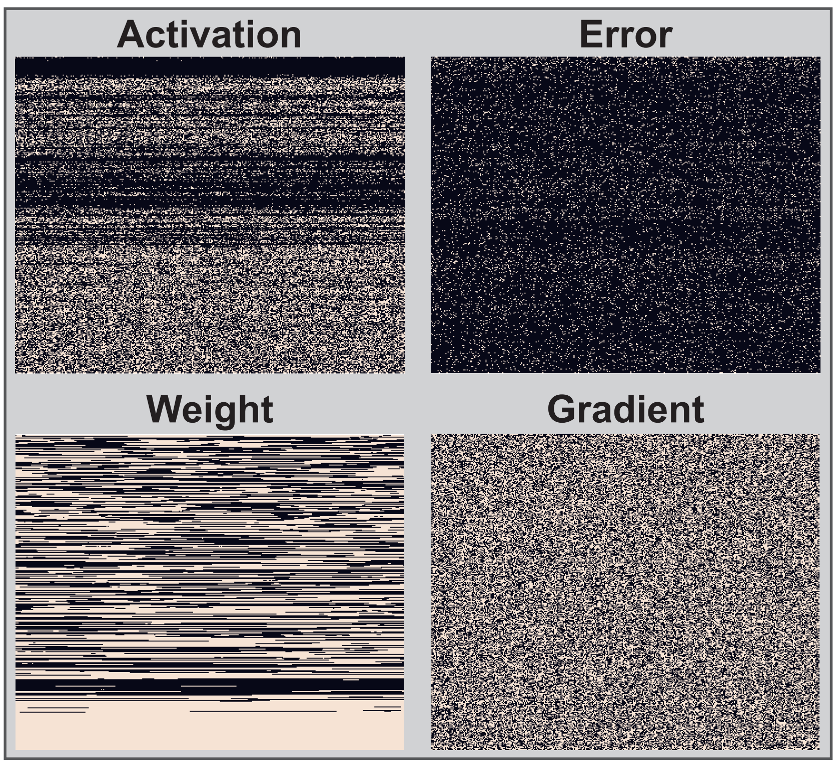

Fig. A1. Visualization of LSB's changing trend of a specific source location along the entire training process.

C Checking the Quality of Random Number using the NIST Test Suite

We use the NIST test suite SP800-22 to evaluate the quality of the random numbers extracted from different sources. Figure A2 shows the report of the NIST test when we are using the 2nd layer's gradients as the source for random number extraction throughout an entire training process. Our extracted random number can pass all the tests, indicating a very high random number quality.

Test Name	P-Value	Result
monobit_test	0.8512318716652107	PASS
frequency_within_block_test	0.6550591135121175	PASS
runs_test	0.6663240502419868	PASS
longest_run_ones_in_a_block_test	0.5667348815869525	PASS
binary_matrix_rank_test	0.6842085773521752	PASS
dft_test	0.736032069994816	PASS
non_overlapping_template_matching_test	0.9999988863312715	PASS
overlapping_template_matching_test	0.6467261043438572	PASS
maurers_universal_test	0.1044507167463431	PASS
linear_complexity_test	0.98989840879283	PASS
serial_test	0.37045666806568434	PASS
approximate_entropy_test	0.7872286033166492	PASS
cumulative_sums_test	0.9058052629683142	PASS
random_excursion_test	0.19019732893198868	PASS
random_excursion_variant_test	0.20123449731469215	PASS

Fig. A2. The report of the test on the NIST test suite SP800–22. We use the 2nd layer's gradients as the source for random number extraction.

Figure A3 shows the testing results on the NIST test suite SP800–22 using extracted random numbers from different types and layers of sources. In the figure, we denote the source used for random number extraction in a $source\_type-layer\_num$ format(e.g., g-10 stands for the gradients in the 10th layer). We show the testing results of the first, middle, and last layers of different types of sources. From the figure we can see that even though extracting random numbers from weights can achieve comparable model accuracy as using gradient as the source, they still fail the most of the randomness tests. Both the relatively lower information entropy and the quite low bit-flipping frequency lead to this failure.

And we can also observe that, generally, the gradients have better randomness than the other sources. However, the last layer fails the majority of the tests, where the first layer fails some of them but is better than the last layer. The gradient of the middle layer can pass all the tests and is the best source for random number extraction. It is worth noting that besides the middle layer shown in the figure, we also find that the gradients from other layers (except the first and last layer) can generally pass all or the majority of the tests. This indicates that there are a lot of good sources in the DNN available for high-quality random number extraction.

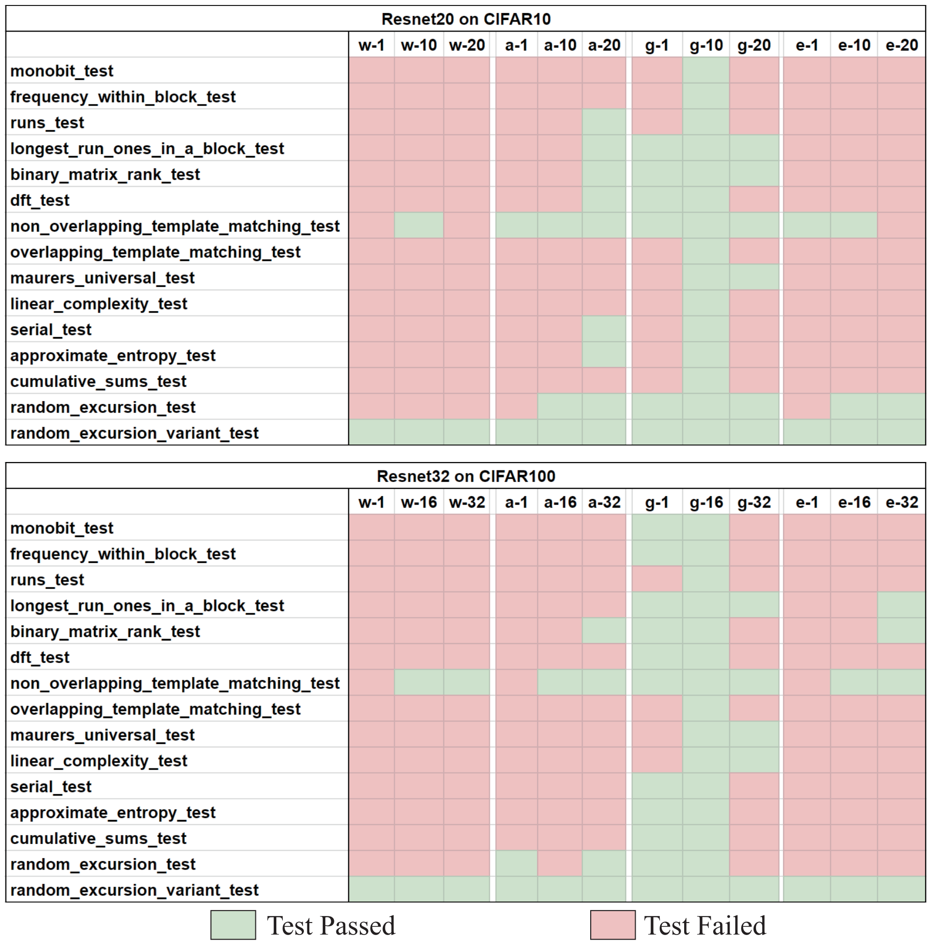

Fig. A3. Testing results on the NIST test suite SP800–22 using extracted random numbers from different types and layers of sources. We denote the source in a $source\_type - layer\_num$ format (e.g., g-10 stands for the gradients in the 10th layer). w: weight, a: activation, g: gradient, e: error.

D Random Numbers with Specified Distributions

We can also precisely control our image-pixel-based extraction method to obtain other specified distributions in high-quality. Figure A4 shows three different distributions obtained using the same input pixel location, which are (a) the uniform

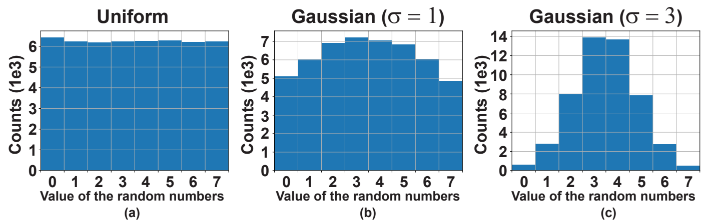

Fig. A4. Obtain high-quality specified distributions using our image-pixel-based random number extraction method.

distribution, (b) the Gaussian distribution with $\sigma = 1$, and (c) the Gaussian distribution with $\sigma = 3$. The distributions can be obtained by simply changing the corresponding threshold arrays. For a sanity check, we also randomly sampled several pixel locations to evaluate the quality of the distributions. High-quality results with desired distribution can always be obtained, which verifies the effectiveness of our proposed method.