Materials for LM-GC NeurIPS Rebuttal

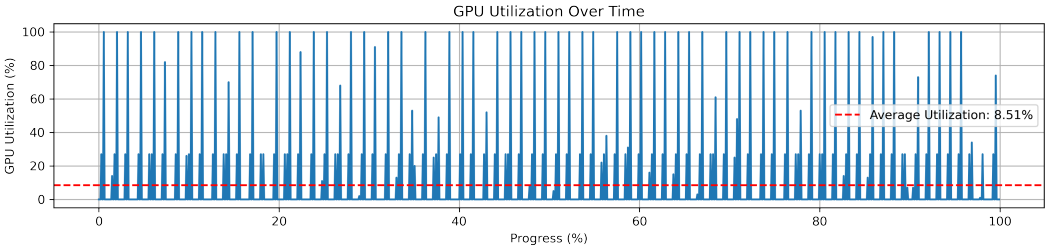

Figure A1: GPU utilization over time.

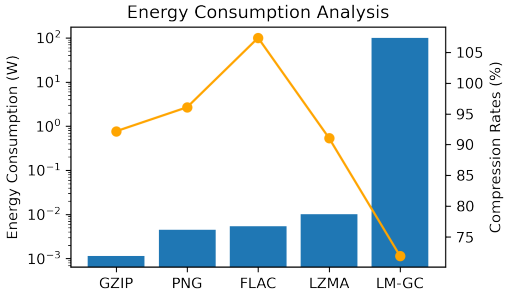

Figure A2: Energy consumption by different coding schemes.

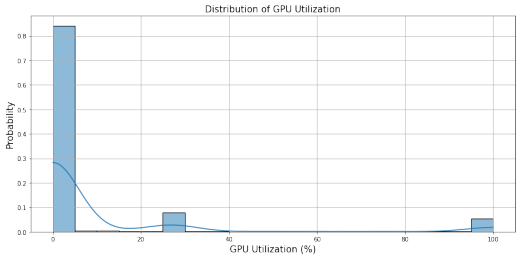

Figure A3: Distribution of GPU utilization by LM-GC. It is observed that GPUs are not the main bottleneck since they are idle most of the time.

	Traditional codec						
	PNG	FLAC	GZIP	LZMA	FPZIP	LM-GC (H_s)	Improvement
MNIST	50.05±4.3	55.20±1.7	45.05±5.2	43.19±1.3	44.62±0.6	**39.38±1.4**	8.8%
CIFAR-10	43.30±1.3	52.37±0.6	42.42±0.3	41.91±0.0	41.26±0.8	**38.83±0.4**	5.9%
TinyImageNet	96.08±0.1	107.36±0.0	92.18±0.0	91.06±0.1	86.88±0.1	**71.90±0.0**	17.2%

Table A1: Compression effectiveness on MNIST, CIFAR-10, and TinyImageNet datasets. We use a Tinyllama as the compressor to compress the gradients of ConvNets. The raw data are converted to hexadecimal numbers with spaces as the separator. The improvement over the best baseline highlights the capability of LM-GC in modeling complex gradients.

	Traditional codec					Ours (Tinyllama 1.1B)			
	PNG	FLAC	GZIP	LZMA	FPZIP	H_n	H_s	H_c	H_{c+s}
ConvNet	43.30±1.3	52.37±0.6	42.42±0.3	41.91±0.0	41.26±0.75	**36.30±0.8**	38.83±0.4	38.40±0.6	38.46±0.1
VGG16	95.61±0.2	-	91.91±0.0	91.27±0.1	89.15±0.17	83.23±0.0	**73.42±0.1**	75.32±0.2	73.97±0.1
ResNet18	97.22±0.1	-	92.47±0.0	91.72±0.1	90.72±0.07	83.20±0.3	**73.57±0.1**	75.55±0.3	73.95±0.2
ViT	94.50±0.4	-	89.20±1.2	87.98±1.2	89.77±0.48	78.65±3.3	**70.83±1.8**	72.60±2.0	71.62±1.7

Table A2: Gradient compression (%) for convolution neural networks (ConvNet), VGG-16, ResNet-18, and ViT trained on CIFAR-10.

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