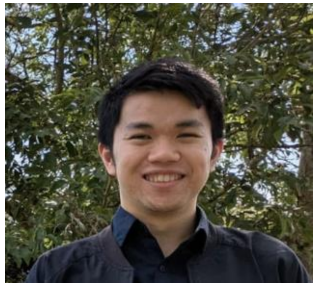
CISPA
HELMHOLTZ CENTER FOR
INFORMATION SECURITY

Language Models as Zero-shot Lossless Gradient Compressors: Towards General Neural Parameter Prior Models

Hui-Po Wang *Mario Fritz*



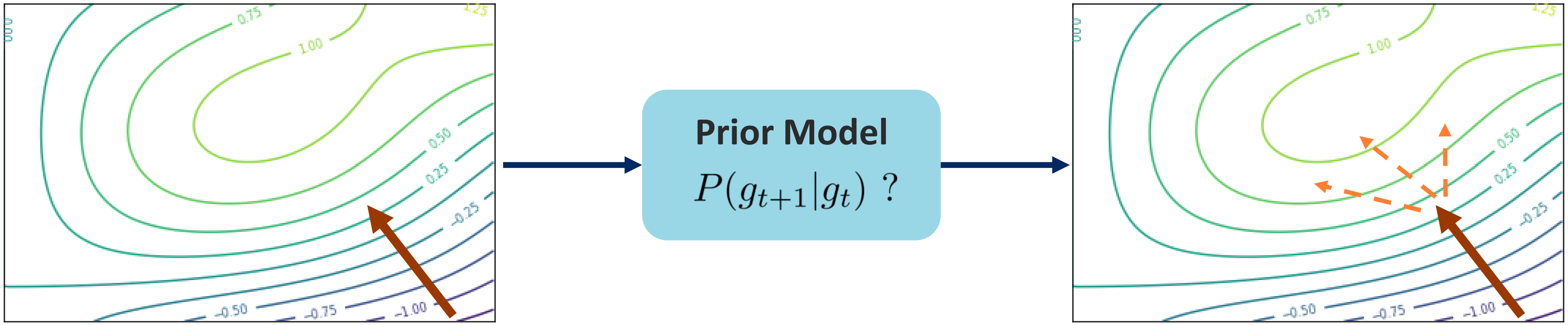
CISPA Helmholtz Center for Information Security, Germany

Presented @ NeurIPS 2024

Motivation

- Statistical prior models have been widely used in many applications
  - Such as image super-resolution and signal denoising

- They have been missing from gradients for a long time because:
  - ***High-dimensional and complex structures*** within gradients
  - ***Generalizability***



Contributions

- We showcase that *Large Language Models (LLMs) can potentially serve as gradient priors even in a zero-shot setting*

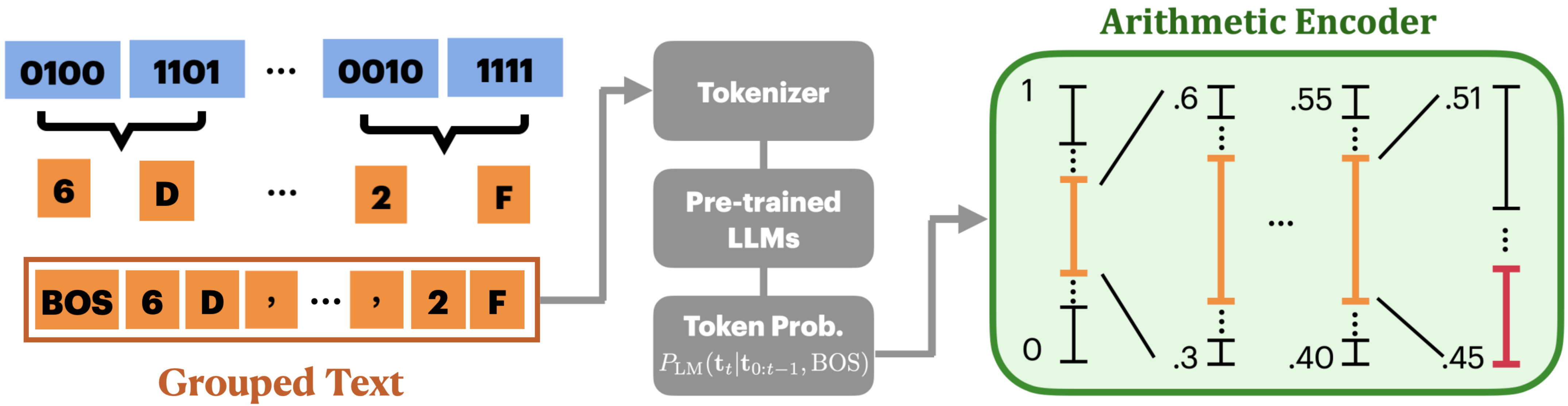
- We verify the property via lossless gradient compression

$$P_{\mathrm{LLM}}(\boldsymbol{g}) \approx P(\boldsymbol{g}) \to \text{compression efficiency} \uparrow$$

- Our approach LM-GC:
  - Convert floating-point gradients into text-like formats, retaining all information and optimizing token efficiency.
  - Leverage large LLMs as priors to achieve **up to 17.2%** improvement over state-of-the-art methods in compressing gradients
  - Showcase the potential of LLMs to interpret **data modalities that are not fully understandable to humans.**

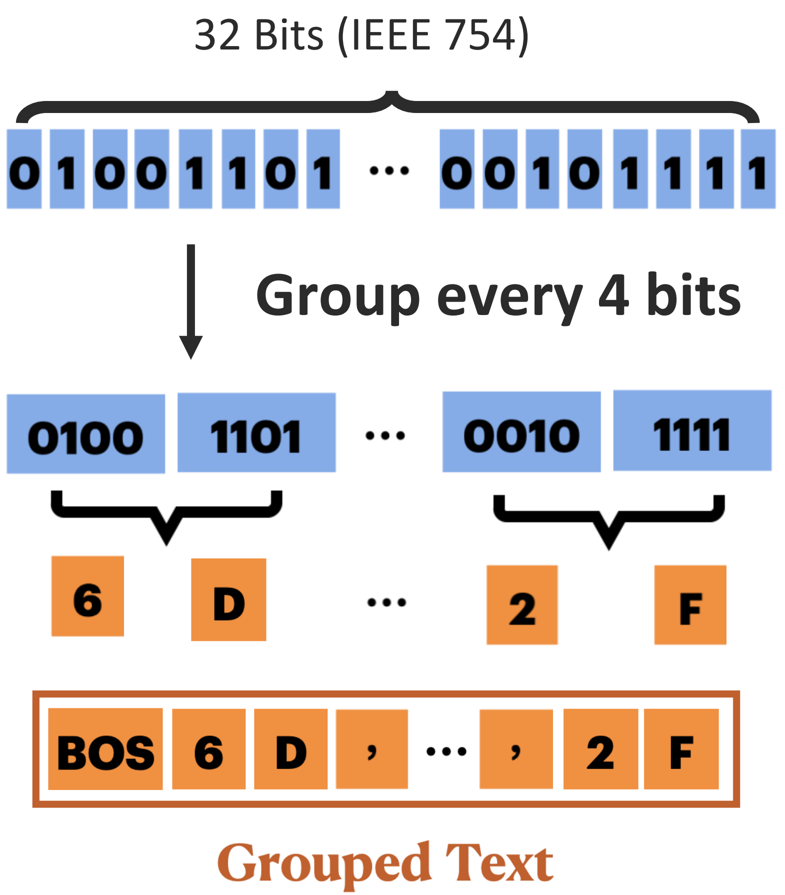
Method – LM-GC

- Our approach consists of three steps:
  - (1) serialization
  - (2) Inference
  - (3) arithmetic coding



LM-GC – (1) Serialization

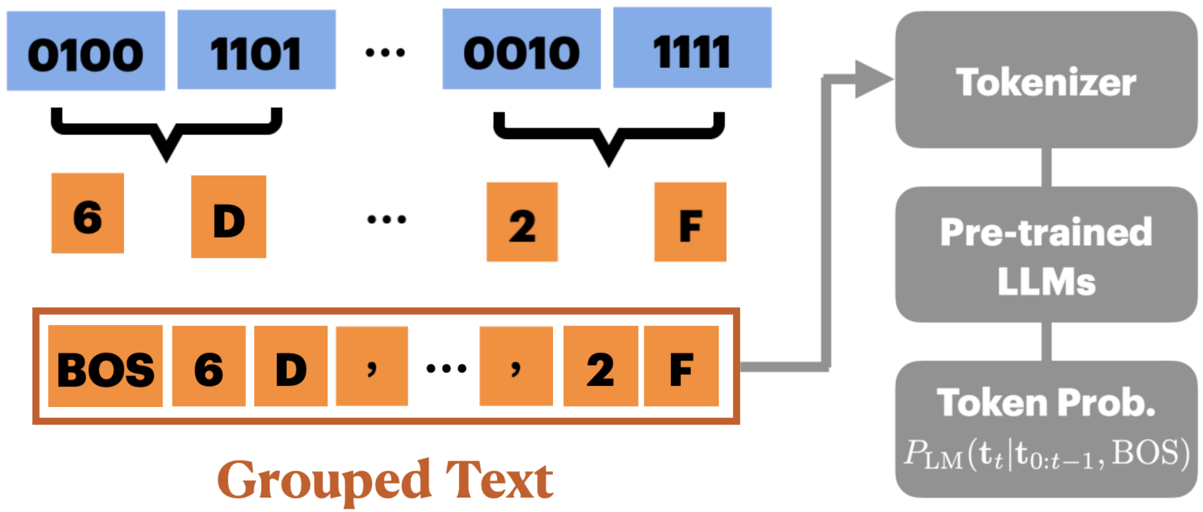
- **Goal:** convert floating-point gradients into ***hexadecimal*** numbers that LLMs know while retaining all information



LM-GC – (2) Inference

- **Goal:** Feed serialized data into a pre-trained LLM pipeline to predict the next-token probability:

$$P_{\text{LM}}(\mathcal{T}) := \prod_{k=1}^{K} p(t_k | \texttt{BOS}, t_{<k})$$

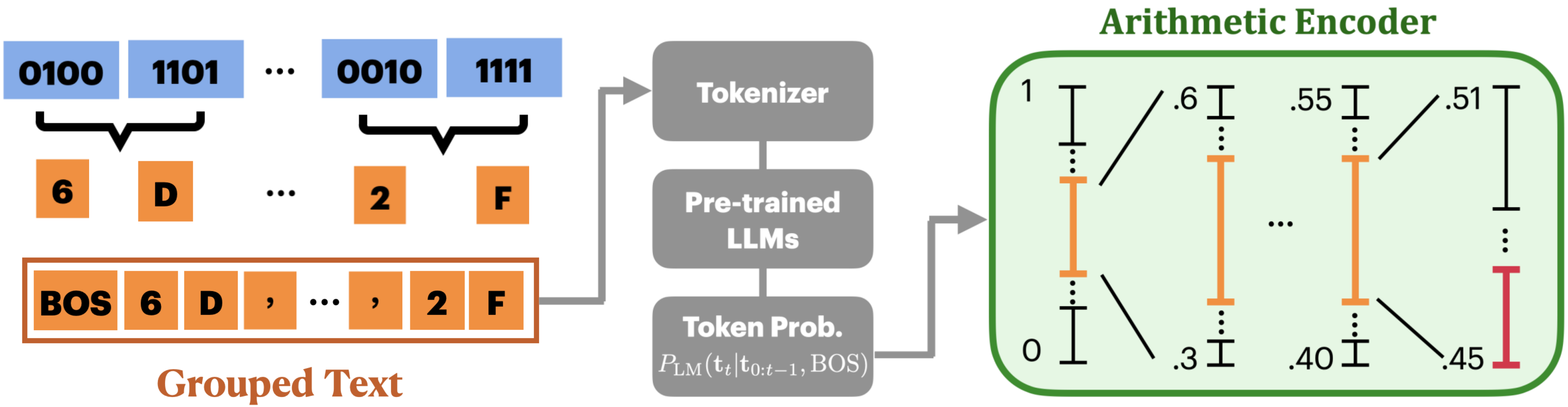


LM-GC – (3) Arithmetic Coding

- **Goal:** conduct the actual arithmetic coding using the probability predicted from the previous step

$$P_{\mathrm{LM}}(\mathcal{T}) := \prod_{k=1}^{K} p(t_k | \texttt{BOS}, t_{<k})$$

  – Note: the probability is used to partition the intervals



Results (1)

- Compare LM-GC to SOTA methods using 3 off-the-shelf language models
- LM-GC surpasses the SOTA by up to 17.2%, highlighting the potential of LLMs to serve as effective gradient priors.
- Metric:

$$\text{Compression Rate } (\%) = 100 \times \frac{\text{Compressed Data Size}}{\text{Original Data Size}}$$

	Traditional codec		LM-GC (Ours)					
	Unchunked	Chunked	ISO	H_n	H_s	H_c	H_{c+s}	H_{semi}
PNG	43.30±1.3	49.18±1.1						
FLAC	52.37±0.6	50.46±0.6						
GZIP	42.42±0.3	47.10±0.4						
LZMA	41.91±0.0	47.36±0.1						
FPZIP	41.26±0.8	49.27±0.3						
Tinyllama 1.1B			117.38±0.0	36.30±0.8	38.83±0.4	38.40±0.6	38.46±0.1	43.45±0.6
Openllama 3B			71.85±0.2	37.07±0.1	32.32±0.3	34.31±0.6	33.07±0.5	33.57±0.2
LLAMA 2 7B			109.07±0.2	72.10±0.5	32.26±0.5	32.96±0.3	**32.21±0.8**	32.78±0.4

Results (2)

- Compress gradients collected from 4 architectures trained on 3 datasets

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

	Traditional codec						
	PNG	FLAC	GZIP	LZMA	FPZIP	LM-GC (H_s)	Impr.
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 3: Compression effectiveness on MNIST, CIFAR-10, and TinyImageNet datasets.

Source Code

- Check our project page for more details!
- https://github.com/hui-po-wang/LM-GC