ByteZip: Efficient Lossless Compression for Structured Byte Streams Using DNNs

Parvathy Ramakrishnan P and Satyajit Das

Department of Data Science, Indian Institute of Technology, Palakkad

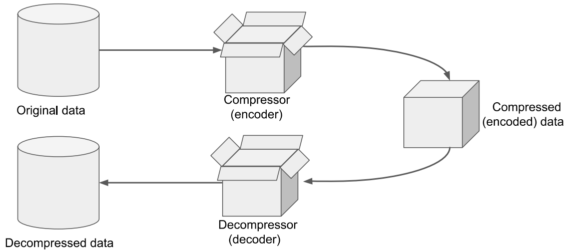
July 1, 2024

Outline of the Presentation

1. Introduction
   - Overview
   - Deep Learning in Lossless Data Compression
   - Related Works
   - Limitations
2. ByteZip
   - Proposed Method
   - Architecture
   - Experiments and Results
3. Conclusion
4. References

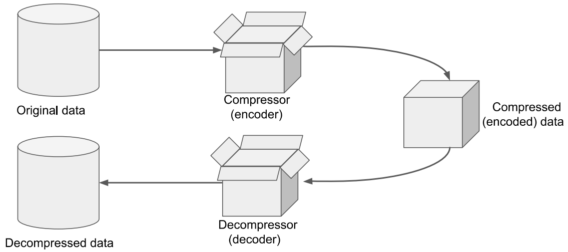
Data Compression

- Science of representing information in a compact form.



Data Compression

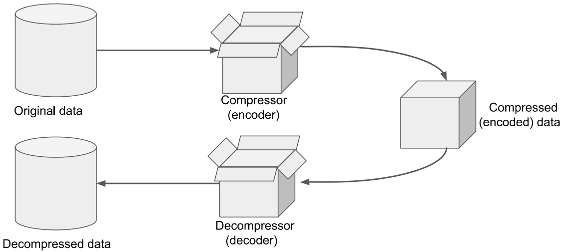
- Science of representing information in a compact form.



- Lossless and Lossy data compression

Data Compression

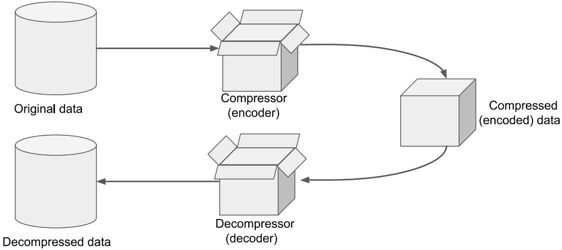
- Science of representing information in a compact form.



- Lossless and Lossy data compression
- Compression Ratio(size reduction) vs Compression Speed(runtime)

Data Compression

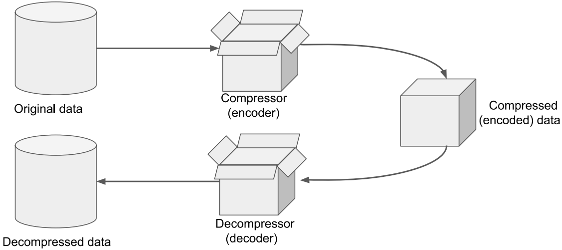
- Science of representing information in a compact form.



- Lossless and Lossy data compression
- Compression Ratio(size reduction) vs Compression Speed(runtime)
- Nature of the data being compressed, tolerance for loss, and requirements for speed and memory efficiency.

Data Compression

- Science of representing information in a compact form.



- Lossless and Lossy data compression
- Compression Ratio(size reduction) vs Compression Speed(runtime)
- Nature of the data being compressed, tolerance for loss, and requirements for speed and memory efficiency.
- Byte streams - Binary representation of information transmitted or stored as a continuous stream of bytes.

Data Compression

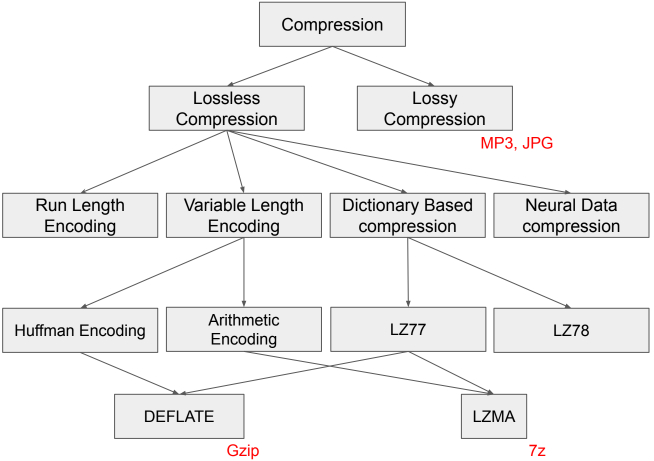


Figure: An overview of different data compression techniques

Deep Learning in Lossless Data Compression

- Neural Data Compression[1] - Application of neural networks and other machine learning methods to data compression

[1] Yang, Mandt, and Theis, "An introduction to neural data compression", 2022.

Deep Learning in Lossless Data Compression

- Neural Data Compression[1] - Application of neural networks and other machine learning methods to data compression
- First build a probabilistic model of the data, and then feed its probabilities into an entropy coding scheme that converts data into compact bit-strings

[1] Yang, Mandt, and Theis, "An introduction to neural data compression", 2022.

Deep Learning in Lossless Data Compression

- Neural Data Compression[1] - Application of neural networks and other machine learning methods to data compression
- First build a probabilistic model of the data, and then feed its probabilities into an entropy coding scheme that converts data into compact bit-strings

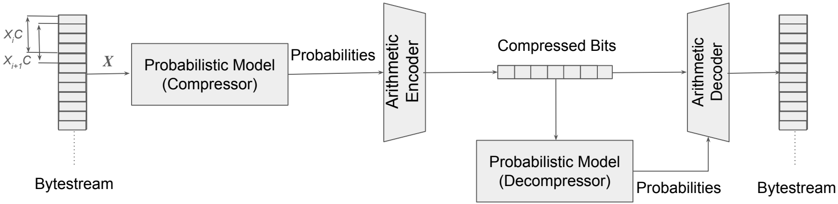


Figure: Overview of Neural Data Compression

[1] Yang, Mandt, and Theis, "An introduction to neural data compression", 2022.

State of The Art Deep Learning Compressors

- CMIX[2]
  - Bitwise predictions using an ensemble of independent models
  - Boosting over multiple compressor experts leads to improved performance
  - lstm-compress and tensorflow-compress
- NNCP[3]
  - Bytewise predictions
  - Improved compression ratio using multi-layer LSTMs and Transformer XL models

[2] Knoll, *Cmix*, 2014; Knoll, *Lstm-compress: data compression using LSTM*, 2019; Knoll, *Tensorflow-compress: lossless data compression using neural networks in TensorFlow*, 2020.

[3] Bellard, "Lossless data compression with neural networks", 2019; Bellard, "NNCP v2: Lossless Data Compression with Transformer", 2021.

State of The Art Deep Learning Compressors

- TRACE[4]
  - Addresses the execution time challenges associated with deep-learning-based compressors
  - Single layer transformer architecture with reduced number of parameters and better GPU utilization
  - 65% better compression ratio, but still is 1000x slower than Gzip

[4] Mao et al., "TRACE: A Fast Transformer-Based General-Purpose Lossless Compressor", 2022.

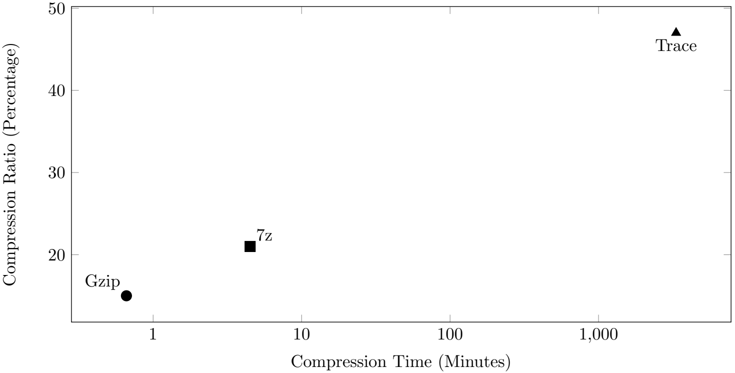


Figure: Compression Time vs Compression Ratio for compressing 1 GB data.

Autoregressive Compressors

- Key Limitation- Very high runtime

Autoregressive Compressors

- Key Limitation- Very high runtime
- 2 key factors
  - Complexity of the model used in probability modeling
  - Sequential nature of the overall compression process

Autoregressive Compressors

- Key Limitation- Very high runtime
- 2 key factors
  - Complexity of the model used in probability modeling
  - Sequential nature of the overall compression process
- Sequential GPU/CPU/IO operations for each byte

Autoregressive Compressors

- Key Limitation- Very high runtime
- 2 key factors
  - Complexity of the model used in probability modeling
  - Sequential nature of the overall compression process
- Sequential GPU/CPU/IO operations for each byte

Table: Average Compression and Decompression Speed

Model	Avg Comp Speed (MB/Min)	Avg Decomp Speed (MB/Min)
TRACE_8	2.01	1.39
FNN_8	2.92	1.73
Uniform AE	3.31	1.83

Overview

- Hierarchical autoencoder-based approach[5] to model probability for byte streams instead of existing autoregressive approaches.
- Instead of a forward pass per byte, a chunk of data is processed in one pass resulting in faster compression.
- Support for partial decompression

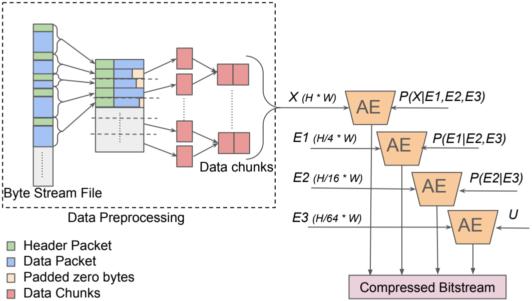
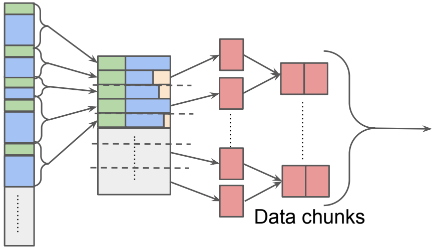


Figure: Overview of ByteZip compressor

[5] Mentzer et al., "Practical Full Resolution Learned Lossless Image Compression", 2019.

Data Preprocessing

Transforms the input data into a form that is more easily compressible



Byte Stream File

Data Preprocessing

- Header Packet
- Data Packet
- Padded zero bytes
- Data Chunks

Algorithm 1 Data Preprocessing

---

procedure PREPROCESSDATA(bytestream)
    for all *packet* in *bytestream* **do**
        if n repetitive packets **then**
            Join n packets to a single record
            $w \leftarrow w_1 + w_2 + \ldots + w_n$
        end if
    end for
    $W \leftarrow$ Width of longest record
    for all *record* in *bytestream* **do**
        $record \leftarrow record + [(W - w)$ zero bytes$]$
    end for
    $N \leftarrow len(bytestream)//(H * W)$
    $chunks \leftarrow bytestream.reshape(N, H, W)$
    if $W < 256$ **then**
        $k \leftarrow math.ceil(256/W)$
        $chunks \leftarrow chunks.reshape(N//k, H, k * W)$
    end if
end procedure

Probability Modelling

- An autoencoder with a mixed-density network is employed to model probability

Probability Modelling

- An autoencoder with a mixed-density network is employed to model probability
- Three key modules - Encoder, Decoder and Probability estimator

Probability Modelling

- An autoencoder with a mixed-density network is employed to model probability
- Three key modules - Encoder, Decoder and Probability estimator
- Leverage the lower dimensional representation during decompression to calculate probability parameters

Probability Modelling

- An autoencoder with a mixed-density network is employed to model probability
- Three key modules - Encoder, Decoder and Probability estimator
- Leverage the lower dimensional representation during decompression to calculate probability parameters
- Multiscale hierarchical Autoencoders- Reduce the size of latent variables, Capture complex dependencies

Probability Modelling

- An autoencoder with a mixed-density network is employed to model probability
- Three key modules - Encoder, Decoder and Probability estimator
- Leverage the lower dimensional representation during decompression to calculate probability parameters
- Multiscale hierarchical Autoencoders- Reduce the size of latent variables, Capture complex dependencies
- Trained to minimize the sum of negative log-likelihood in all scales

Neural Network Architecture

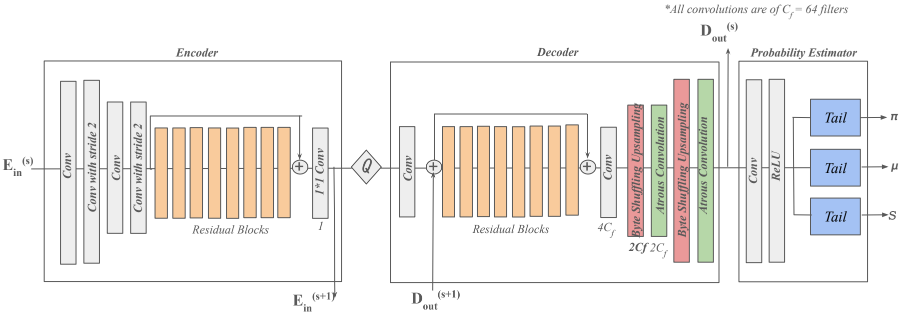


Figure: Architecture details for a single scale

Neural Network Architecture

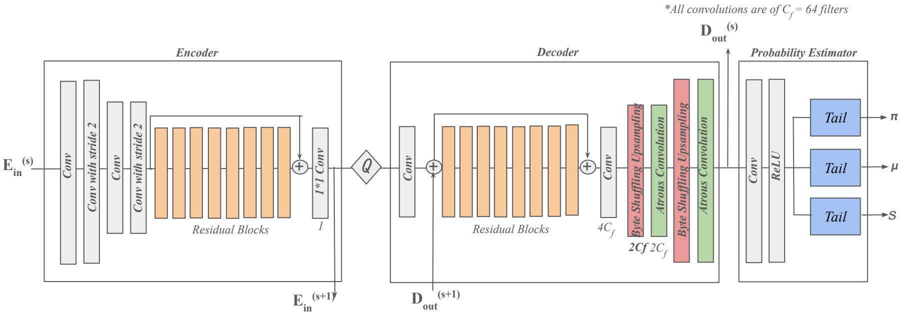


Figure: Architecture details for a single scale

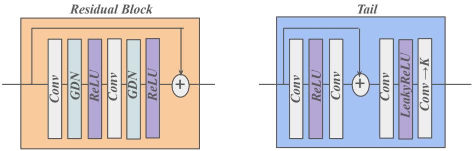


Figure: Residual and Tail Blocks

Neural Network Architecture

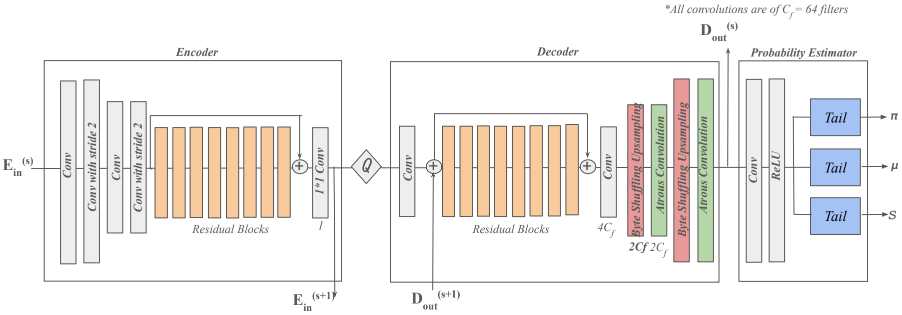


Figure: Architecture details for a single scale

$$P(x|\pi, \mu, s) \sim \sum_{i=1}^{K} \pi_i \text{logistic}(\mu_i, s_i)$$

Neural Network Architecture

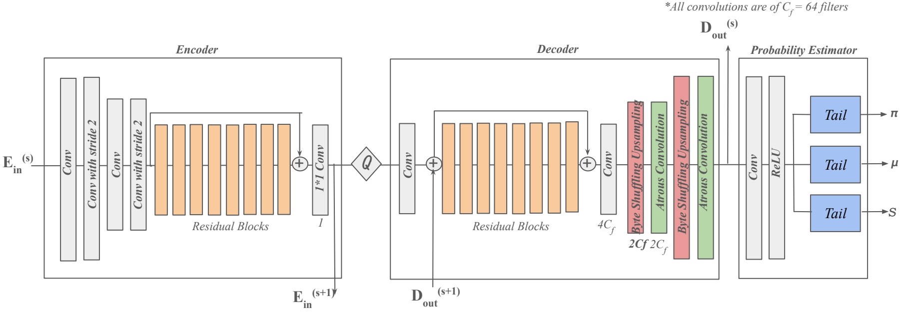


Figure: Architecture details for a single scale

$$P(x|\pi,\mu,s) \sim \sum_{i=1}^{K} \pi_i \text{logistic}(\mu_i, s_i)$$

$$P(x|\pi,\mu,s) \sim \sum_{i=1}^{K} \pi_i \left[\sigma\left((x+b/2-\mu_i)/s_i\right) - \sigma\left((x-b/2-\mu_i)/s_i\right)\right]$$

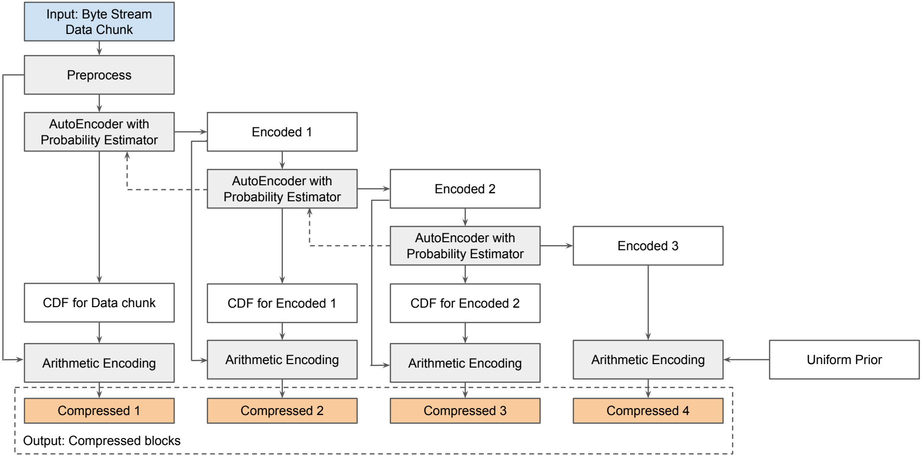


Figure: Hierarchical Compression Framework

Decompression

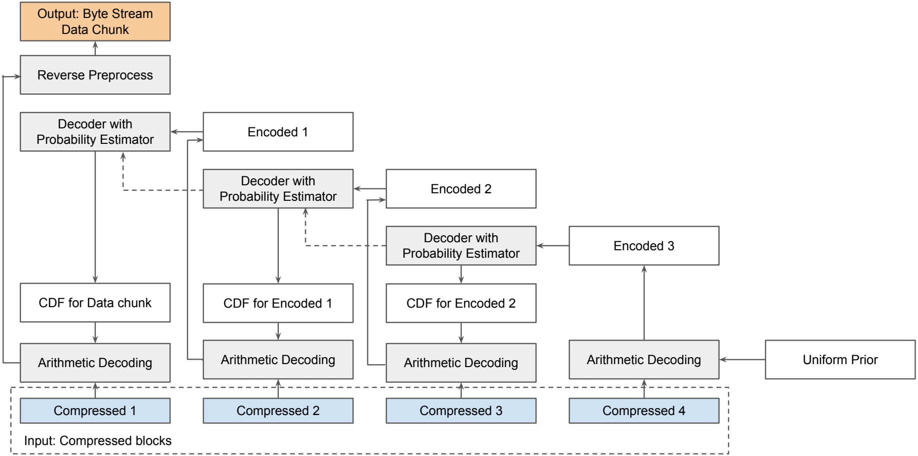


Figure: Hierarchical Decompression Framework

Experimental Setup

Models considered for performance comparison:

Model	Type	Pretraining Required	Parameter Updation During Compression	Support Partial Decompression	CPU/GPU
Gzip[6]	Traditional	No	No	No	CPU
7z[7]	Traditional	No	No	No	CPU
TRACE[8]	Autoregressive	No	Yes	No	GPU
FNN	Autoregressive	No	Yes	No	GPU
ByteZip	Non Autoregressive	Yes	No	Yes	GPU

[6] Deutsch, *GZIP file format specification version 4.3*, 1996.

[7] *7-Zip*.

[8] Mao et al., "TRACE: A Fast Transformer-Based General-Purpose Lossless Compressor", 2022.

Experimental Setup

Models considered for performance comparison:

Model	Type	Pretraining Required	Parameter Updation During Compression	Support Partial Decompression	CPU/GPU
Gzip[6]	Traditional	No	No	No	CPU
7z[7]	Traditional	No	No	No	CPU
TRACE[8]	Autoregressive	No	Yes	No	GPU
FNN	Autoregressive	No	Yes	No	GPU
ByteZip	Non Autoregressive	Yes	No	Yes	GPU

Evaluation metrics:

- $$\text{Average Compression Speed} = \frac{\text{Data Size (MB)}}{\text{Total Compression Time (Min)}}$$
- $$\text{Compression Ratio} = \frac{\text{Original Size}}{\text{Compressed Size}}$$
- $$\text{Compression Percentage} = \left(1 - \frac{\text{Compressed Size}}{\text{Original Size}}\right) \times 100$$

[6] Deutsch, *GZIP file format specification version 4.3*, 1996.

[7] *7-Zip*.

[8] Mao et al., "TRACE: A Fast Transformer-Based General-Purpose Lossless Compressor", 2022.

Experimental Setup

Hardware:

Single workstation with Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz, NVIDIA GeForce RTX3080 Ti GPU with 10240 cores and 12GB RAM

[9] Burtscher and Ratanaworabhan, "FPC: A High-Speed Compressor for Double-Precision Floating-Point Data", 2009.
[10] Burtscher and Ratanaworabhan, "FPC: A High-Speed Compressor for Double-Precision Floating-Point Data", 2009.
[11] Choi et al., "Children's song dataset for singing voice research", 2020.

Experimental Setup

Hardware:

Single workstation with Intel(R) Core(TM) i9-10900K CPU @ 3.70GHz, NVIDIA GeForce RTX3080 Ti GPU with 10240 cores and 12GB RAM

Dataset:

Table: Description of datasets along with the parameters used

Name	Dataset Size	Description	chunk size	k	scale
Sonar	25 GB	Ocean acoustic data collected using sonar arrays	64 * 1504	3	3
obs_info[9]	500 MB	Latitude and Longitude information of the observation points of a weather satellite	256 * 256	3	4
obs_spitzer[10]	1.1 GB	Spitzer Space Telescope data with 64-Bit Double-Precision Floating-Point values	256 * 256	2	4
CSD[11]	1.5 GB	Children's Song Dataset for Singing Voice Research	256 * 256	3	4

[9] Burtscher and Ratanaworabhan, "FPC: A High-Speed Compressor for Double-Precision Floating-Point Data", 2009.

[10] Burtscher and Ratanaworabhan, "FPC: A High-Speed Compressor for Double-Precision Floating-Point Data", 2009.

[11] Choi et al., "Children's song dataset for singing voice research", 2020.

Runtime Improvements

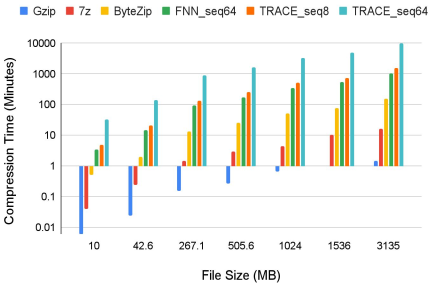


Figure: Compression Time

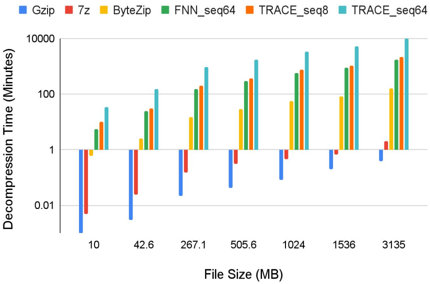


Figure: Decompression Time

Runtime Improvements

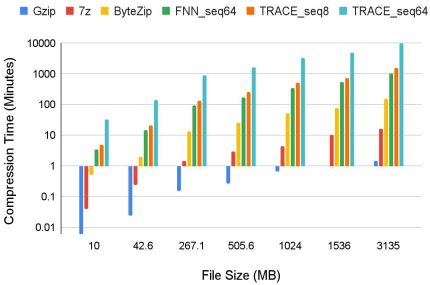


Figure: Compression Time

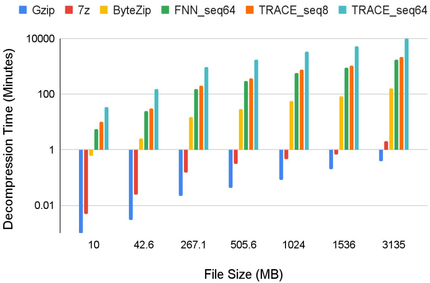


Figure: Decompression Time

Table: Average Compression and Decompression Speed

	Model	Avg Comp Speed (MB/Min)	Avg Decomp Speed (MB/Min)
Traditional	Gzip	1756	10974
	7z	191	1895
Non-Autoregressive	ByteZip(Ours)	20	18
Autoregressive	FNN_64	2.89	1.76
	TRACE_8	2.01	1.34
	TRACE_64	0.31	0.28

Size Reduction

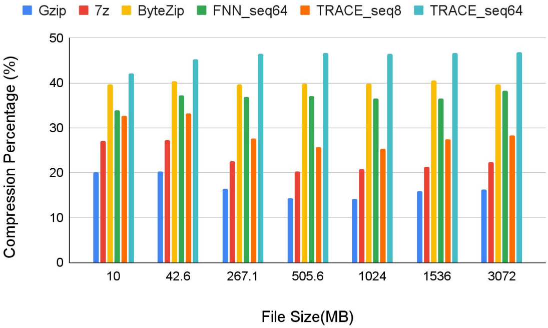


Figure: Compression percentage on different filesizes

Size Reduction

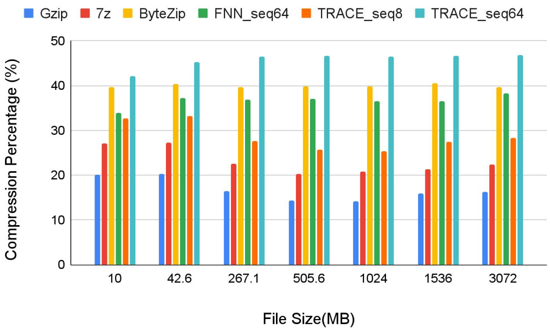


Figure: Compression percentage on different filesizes

Table: Average compression ratio

	Model	obs_info	obs_spitzer	CSD	Sonar
Traditional	Gzip	1.15	1.06	1.51	1.20
	7z	1.22	1.13	2.19	1.30
Non-Autoregressive	ByteZip(Ours)	2.17	1.21	3.01	1.65
Autoregressive	FNN_64	2.09	1.19	1.67	1.57
	TRACE_8	2.01	1.18	2.92	1.40
	TRACE_64	2.81	1.28	3.08	1.84

Compression Time vs Compression Ratio

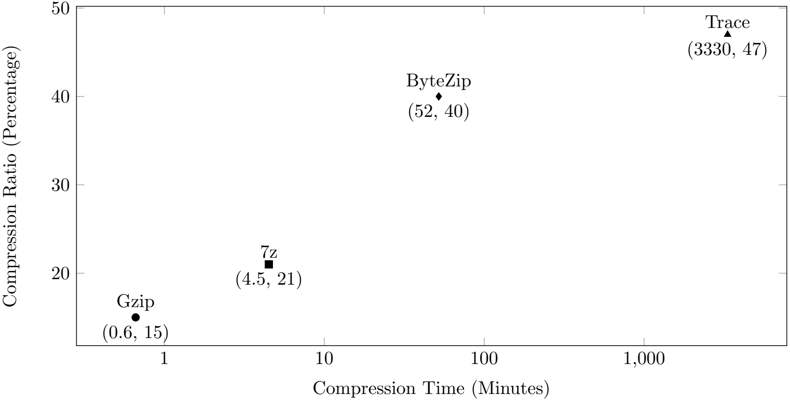


Figure: Compression Time vs Compression Ratio for compressing 1 GB bytestream data

Conclusion

- Autoregressive bytewise prediction-based lossless compression models have limitations in improving runtime
- Our proposed method ByteZip balances the higher compression ratio achieved by autoregressive neural network models with the practicality of attaining a reasonable compression speed
- Future Scope:
  - Reduce the computational requirements of the neural network compressor to create a more practical compressor.
  - Support the incorporation of domain-specific knowledge into the compressor when available.

Thank You.

References I

Yang, Yibo, Stephan Mandt, and Lucas Theis. "An introduction to neural data compression". In: *arXiv preprint arXiv:2202.06533* (2022).

Knoll, Bryon. *Cmix*. https://www.byronknoll.com/cmix.html. 2014.

— .*Lstm-compress: data compression using LSTM*. https://github.com/byronknoll/lstm-compress. 2019.

— .*Tensorflow-compress: lossless data compression using neural networks in TensorFlow*. https://github.com/byronknoll/tensorflow-compress. 2020.

Bellard, Fabrice. "Lossless data compression with neural networks". In: *URL: https://bellard. org/nncp/nncp. pdf* (2019).

— ."NNCP v2: Lossless Data Compression with Transformer". In: (2021).

Mao, Yu et al. "TRACE: A Fast Transformer-Based General-Purpose Lossless Compressor". In: *Proceedings of the ACM Web Conference 2022*. ACM, 2022, 1829–1838. ISBN: 9781450390965.

Mentzer, F. et al. "Practical Full Resolution Learned Lossless Image Compression". In: *CVPR*. 2019.

References II

- Deutsch, Peter. *GZIP file format specification version 4.3*. Tech. rep. 1996.
- *7-Zip*.
- Burtscher, Martin and Paruj Ratanaworabhan. "FPC: A High-Speed Compressor for Double-Precision Floating-Point Data". In: *IEEE Transactions on Computers* 58.1 (2009), pp. 18–31. DOI: 10.1109/TC.2008.131.
- Choi, Soonbeom et al. "Children's song dataset for singing voice research". In: *International Society for Music Information Retrieval Conference (ISMIR)*. 2020.