TensorCodec: Compact Lossy Compression of Tensors without Strong Assumptions on Data Properties (Appendix)

A PROOF OF MEMORY REQUIREMENTS

Theorem A.1 (Memory Requirements for Compression). *TensorCodec (Algorithm 1) requires* $O(\prod_{i=1}^{d} N_i + h(2^d + (h+R^2)\log N_{\max}) + \sum_{i=1}^{d} N_i \log N_i)$ *memory space.*

Proof. Storing $\mathcal{X}$ requires memory of size $O(\prod_{i=1}^{d} N_i)$, and storing the current compressed output requires memory of $O(h(2^d + h + R^2) + \sum_{i=1}^{d} N_i \log N_i)$ by Theorem 4.2. The operations in LSTM, the embedding layer, and the fully connected layer consume $O(h(h + R^2)\log N_{\max})$ memory during the inference and backpropagation of the model if we set the size of min-batches to a constant. The other parts are dominated by the parts above. In conclusion, the memory required for compression is $O(\prod_{i=1}^{d} N_i + h(2^d + (h+R^2)\log N_{\max}) + \sum_{i=1}^{d} N_i \log N_i)$. □

B DETAILED EXPERIMENTAL SETTING

For real-world tensors used in Section 5, we initially assign 2 to $n_{k,l}$ for all $k \in \{1, \cdots, d\}$ and $l \in \{1, \cdots, d'\}$ and modify some of them using integers at most 5 to ensure that both the input and folded tensors possess similar numbers of entries. The assigned values in the form of a d by d' matrix for each tensor are as follows:

- `Uber`:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 3 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 5 \end{bmatrix}$$

- `Airquality`:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- `Action`:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 5 \end{bmatrix}$$

- `PEMS-SF`:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 5 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 \end{bmatrix}$$

- `Activity`:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 5 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 \end{bmatrix}$$

- `Stock`:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 3 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

Table 1: Sizes of tensors used in Figure 5 of Section 5.4 of the main paper.

Order	Size of Tensor	Order	Size of Tensor
3	$256 \times 256 \times 128$	4	$64 \times 64 \times 64 \times 64$
	$256 \times 256 \times 256$		$128 \times 64 \times 64 \times 64$
	$512 \times 256 \times 256$		$128 \times 128 \times 64 \times 64$
	$512 \times 512 \times 256$		$128 \times 128 \times 128 \times 64$
	$512 \times 512 \times 512$		$128 \times 128 \times 128 \times 128$

Table 2: The modes of real-world tensors that are reordered by the order initialization method of TensorCodec.

Uber	Airquality	Action	PEMS-SF	Activity	Stock	NYC	Absorb
1	2, 3	2, 3	1, 3	2, 3	1, 2, 3	1, 2, 4	4

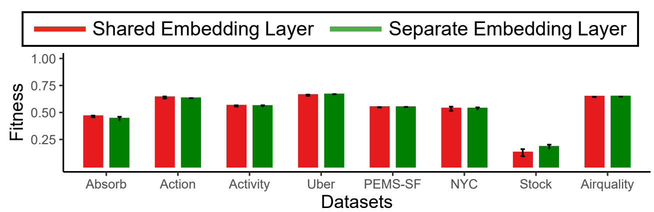

Figure 1: The running time comparison between our models with different embedding layers. The fitness of our model does not significantly change when the design of the embedding layers is changed.

- `NYC`:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 3 & 3 & 1 & 1 & 1 \end{bmatrix}$$

- `Absorb`:

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 & 3 \\ 2 & 2 & 2 & 2 & 2 & 3 & 3 \\ 2 & 2 & 2 & 2 & 2 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{bmatrix}$$

The hyperparameter settings of the algorithms used in Section 5.2 are provided in Table 3. The sizes of tensors used in Section 5.4 are given in Table 1. The modes of the tensor where the order of indices is initialized by our method are provided in Table 2.

C EXPERIMENTS ON THE DESIGN OF EMBEDDING LAYERS

The comparison of the Fitness of the model when using shared embedding layers and separate embedding layers is in Figure 1. We set the TT rank R and hidden size h to the settings of Section 5.3 for `Action`, `Airquality`, `PEMS-SF`, and `Uber` datasets. For the remaining datasets, R and h is set to the settings of the smallest compressed size in Section 5.2 (i.e. the first row of Table 3).

Table 3: Hyperparamters of TensorCodec and competitors used in Section 5.2 where $[\![1, N]\!]$ is the integer interval from 1 to N.

Method	Hyperparameter	Stock	Airquality	PEMS-SF	Activity	Action	Uber	Absorb	NYC
TensorCodec	(TT rank, hidden size of LSTM)	(7, 3), (5, 7), (8, 8), (8, 10)	(7, 11), (7, 20), (8, 25), (17, 21)	(6, 5), (5, 9), (7, 11), (14, 8)	(8, 6), (12, 8), (9, 13), (11, 15)	(6, 8), (11, 9) (13, 11), (13 ,14)	(7, 8), (7, 8), (7, 13), (12, 13), (12, 16)	(6, 4), (4, 8), (6, 10), (8, 11)	(2, 5), (5, 7), (6, 9), (10, 9)
	Learning Rate	1	0.1	1	1	1	0.1	1	0.1
	Tolerance	100	100	100	10	10	100	10	100
	Maximum Epochs	500	1400	1000	5000	5000	5000	5000	1500
NeuKron	hidden size of LSTM	5, 8, 11, 14	14, 23, 29, 35	7, 11, 14, 17	10, 15, 19, 23	10, 15, 19, 23	11, 16, 21, 24	5, 8, 11, 14	5, 8, 11, 14
	Learning Rate	10	0.1	0.1	0.1	0.1	0.1	0.1	0.1
	Tolerance	100	100	100	10	10	100	10	100
	Maximum Epochs	500	1400	1000	5000	5000	5000	5000	1500
CP	Rank	$[\![1, 5]\!]$	$[\![1, 5]\!]$	$[\![1, 4]\!]$	$[\![1, 7]\!]$	$[\![1, 9]\!]$	$[\![1, 6]\!]$	$[\![1, 10]\!]$	$[\![1, 8]\!]$
Tucker		$[\![1, 5]\!]$	$[\![1, 5]\!]$	$[\![1, 4]\!]$	$[\![1, 7]\!]$	$[\![1, 8]\!]$	$[\![1, 5]\!]$	$[\![1, 7]\!]$	$[\![1, 5]\!]$
TT		$[\![1, 5]\!]$	$[\![1, 4]\!]$	$[\![1, 4]\!]$	$[\![1, 5]\!]$	$[\![1, 5]\!]$	$[\![1, 5]\!]$	$[\![1, 4]\!]$	$[\![1, 4]\!]$
TR	Target Accuracy	1.4, 1.3, 1.2, 1.1, 1.05	0.6, 0.55, 0.5, 0.46	0.7, 0.65 0.6, 0.57	0.7, 0.6 0.5, 0.4	0.6, 0.47, 0.43, 0.395	0.6, 0.45, 0.41	0.9, 0.8 0.72, 0.6	0.8, 0.6, 0.5 0.42, 0.41
TTHRESH	Target Relative Error	0.94, 0.92, 0.9 0.85, 0.83, 0.8	0.4, 0.37, 0.35 0.32, 0.315, 0.3	0.48, 0.47, 0.46 0.45, 0.44, 0.43	0.57, 0.55, 0.54 0.53, 0.52	0.45, 0.43, 0.41 0.4, 0.39, 0.38	0.45, 0.43 0.4, 0.38	0.7, 0.65, 0.6 0.6, 0.55, 0.53	0.6, 0.5, 0.45 0.43, 0.42, 0.4
SZ3	Absolute Error Bound	2×10^{16}, 1.4×10^{16}, 1.15×10^{16}, 10^{16}	30, 28, 26, 24, 22	0.18	0.8, 0.7, 0.6	0.9, 0.8, 0.7, 0.6, 0.5	11, 10, 9, 8, 7	100, 50, 30	45, 40

The fitness when sharing embedding layers is higher in 5 datasets. There was no significant dependency of accuracy on the choice of the design of embedding layers. Thus, we choose to share embedding layers across different modes since the theoretical space complexity of this design is independent to the size of the input tensor.