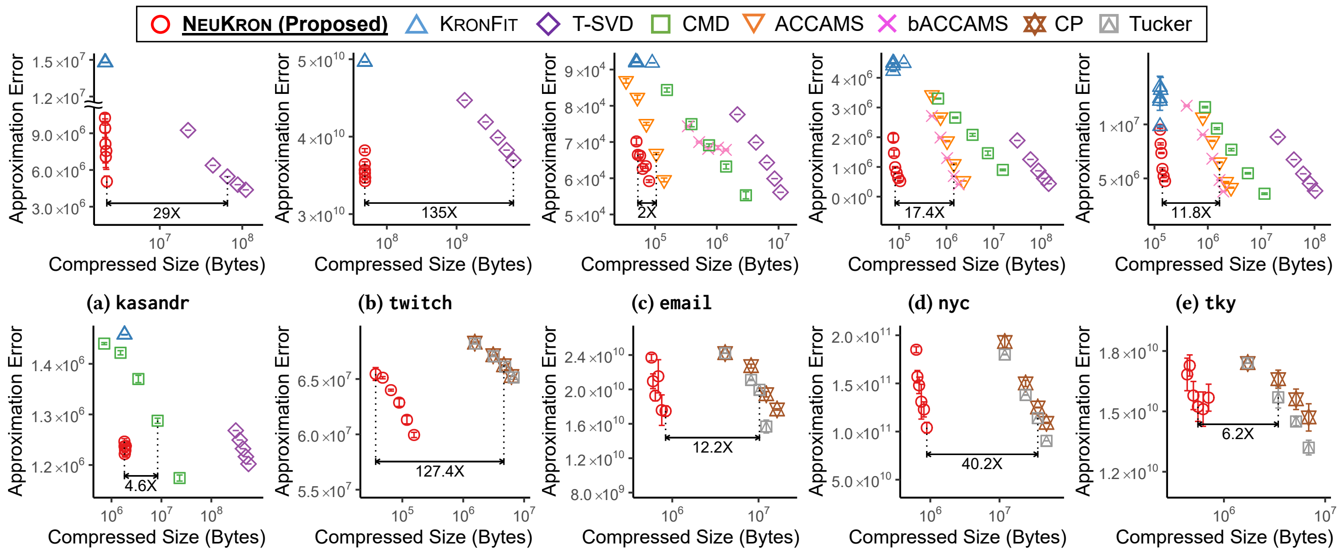
NeuKron: Constant-Size Lossy Compression of Sparse Reorderable Matrices and Tensors (Supplementary Document)

Anonymous Author(s)



(a) kasandr (b) twitch (c) email (d) nyc (e) tky

(f) threads (g) nips (h) enron (i) 3-gram (j) 4-gram

Figure 1: <u>NeuKron outperforms the competitors even in non-reorderable matrices.</u> NeuKron compactly compresses the matrices and the tensors up to two orders of magnitude times when saving orders of all dimensions, compared to the competitors with similar approximation error.

Table 1: Values used for γ and corresponding errors on email, nyc, tky, and kasandr datasets. We report the averages and standard errors of approximation errors.

Dataset	γ	Approximation error
email	1	90561.25 ± 467.996
	10	**58691.88** ± 335.143
	∞	59113.75 ± 891.544
nyc	1	421451.2 ± 4842.068
	10	402673.6 ± 17291.959
	∞	**397947.5** ± 2393.016
tky	1	4166292.3 ± 143013.605
	10	**3981669.6** ± 91907.201
	∞	4034389.1 ± 48117.964
kasandr	1	6315784.36 ± 140974.6535
	10	**4300280.71** ± 488804.599
	∞	4385800.32 ± 496004.629

1 EFFECTIVENESS OF NEUKRON ON NON-REORDERABLE SETTINGS (RELATED TO SECTION 1)

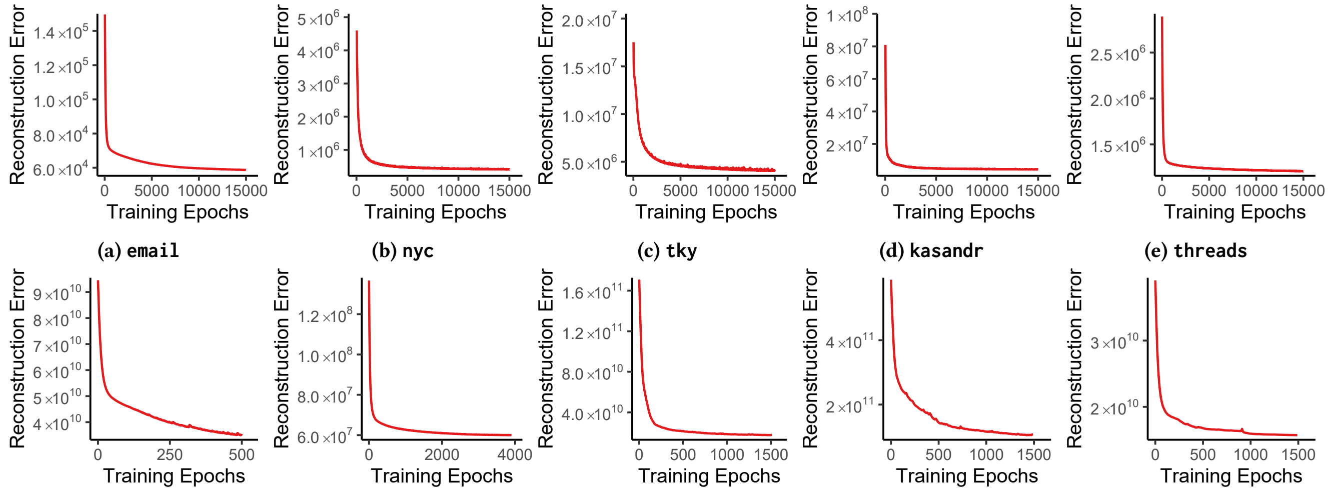
To consider non-reorderable settings, we added additional spaces for storing the orders of indices to the compressed size of NeuKron and that of KronFit. Even with this extra space requirement, NeuKron still reveals the best trade-off between the approximation error and the number of the parameters, as seen in Figure 1. Especially, the compressed size was 2 orders of magnitude smaller than those of the competitors with similar approximation error on twitch and nips dataset.

Table 2: The training time per epoch on all datasets. We report the averages and standard errors of training time.

Dataset (Hidden Dimension)	Training time
email (30)	0.19 ± 0.010
nyc (30)	0.21 ± 0.004
tky (30)	0.32 ± 0.005
kasandr (60)	1.93 ± 0.005
threads (60)	5.49 ± 0.012
twitch (90)	566.82 ± 3.308
nips (50)	6.31 ± 0.081
enron (90)	80.69 ± 0.266
3-gram (90)	27.19 ± 0.089
4-gram (90)	41.09 ± 0.785

2 HYPERPARAMETER ANALYSIS

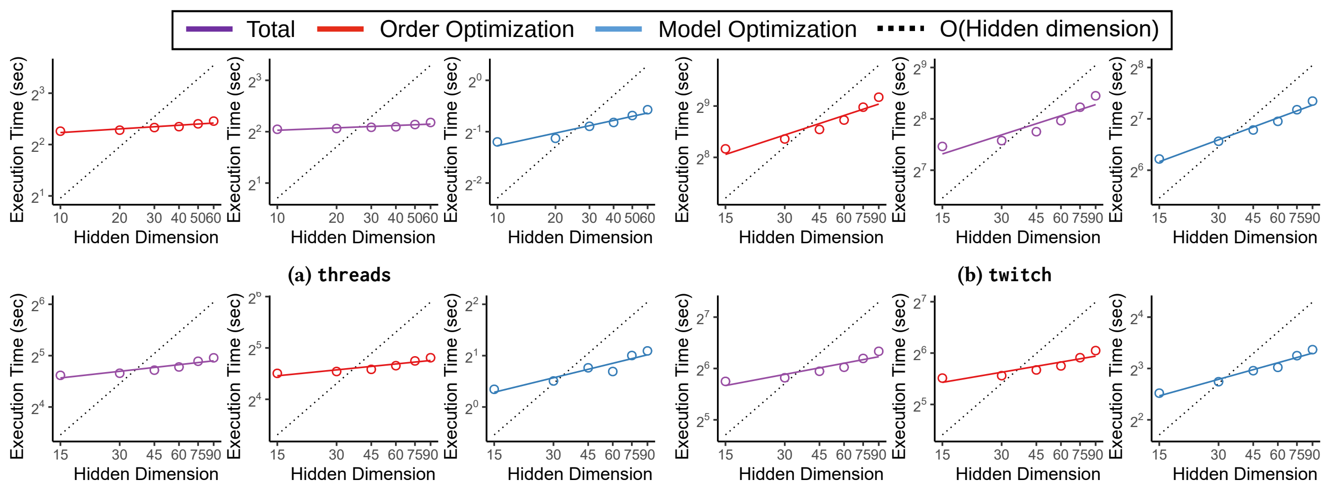
We investigate the approximation errors of NeuKron trained on three γ values and four datasets (email, nyc, tky, kasandr) and



(a) `email` (b) `nyc` (c) `tky` (d) `kasandr` (e) `threads`

(f) `twitch` (g) `nips` (h) `enron` (i) `3-gram` (j) `4-gram`

Figure 2: Reconstruction error of NeuKron after each epoch.



(a) `threads` (b) `twitch`

(c) `4-gram` (d) `enron`

Figure 3: <u>The training time of NeuKron is sub-linear to the hidden dimension of NeuKron.</u> Same as Figure 5 in the main paper, we verified the behavior in both the time for order optimization and the time for model optimization.

reported them in Table 1. Note that setting a γ as ∞ means a greedy algorithm which only switches rows/column in pairs when the approximation error decreases. The results show that the minimum approximation error is acquired when γ is set to 10 except for the nyc dataset.

3 SPEED AND SCALABILITY ON HIDDEN DIMENSION (RELATED TO SECTION 6.4)

We reported the average training time per epoch of NeuKron on the datasets considered in Table 2 of the main paper. The training time per epoch varies from less than 1 second to more than 9 minutes depending on the dataset. As seen in Figure 2, the training plots of all datasets drop dramatically within one third of total epochs. Thus, the model that works empirically well can be obtained earlier than when it converges.

We also analyzed the effect of the hidden dimension on training time per epoch. As seen in Figure 3, both the time for optimizing the orders and the time for model optimization were sublinear to the size of the hidden dimension of the model.