HW3 - Symbolic Music Generation

Name: 譚至斌
ID: d12942015

Novelty

- Speed up inference with k-v cache
- Inference with sink-attention-like pattern

Methodology

- The methodology can be described by these steps:

    1. Convert all midi data to REMI tokens
    2. Train a decoder-only transformer on the dataset
    3. Inference from scratch

Methodology (cont.)

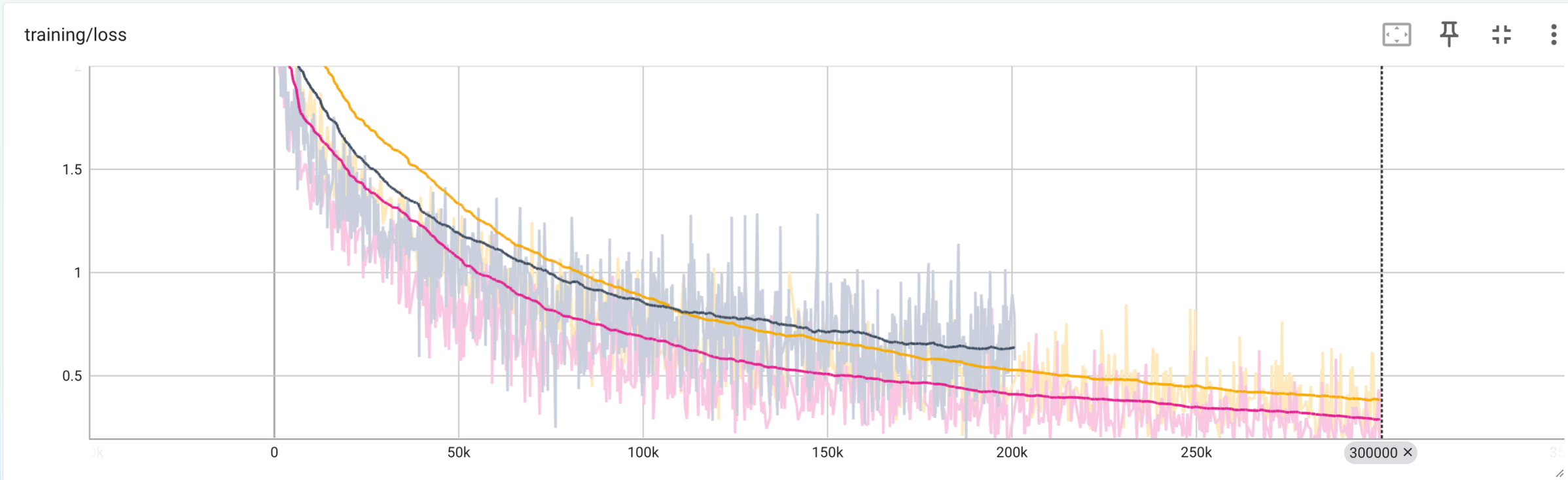
- Sink-attention claims that Transformer often attends to first few tokens
  - It always fails while I'm trying to use cache without recomputing
  - I keeps the 'bos' token at the start of the input sequence in the sliding window on inference
- To enable cache mechanism for Transformer, the positional encoding should be chosen carefully
  - Rotary Position Embedding is used due to its linear property
  - However, in the final setting, absolute PE still works

Experiment Setup

- The models are built with these configurations:
  - 512-dim latent embeddings
  - 8 heads
  - 2048-dim dense layers.
  - 12 attention layers

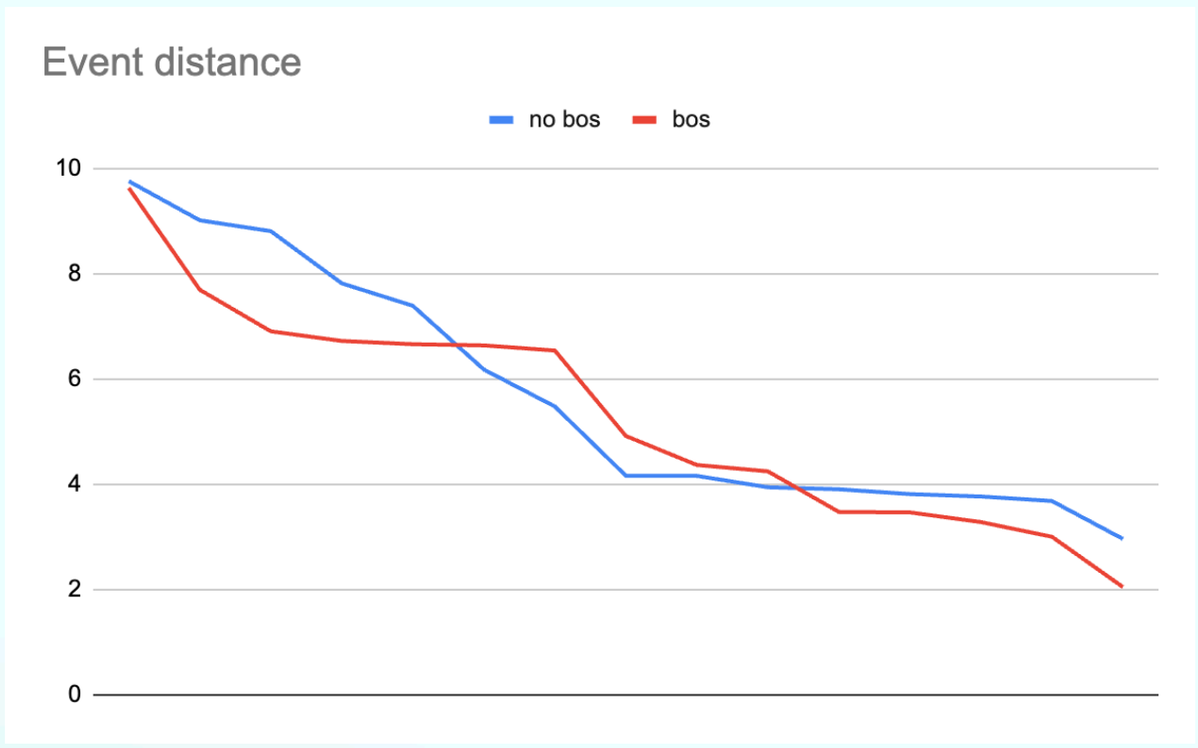
Result

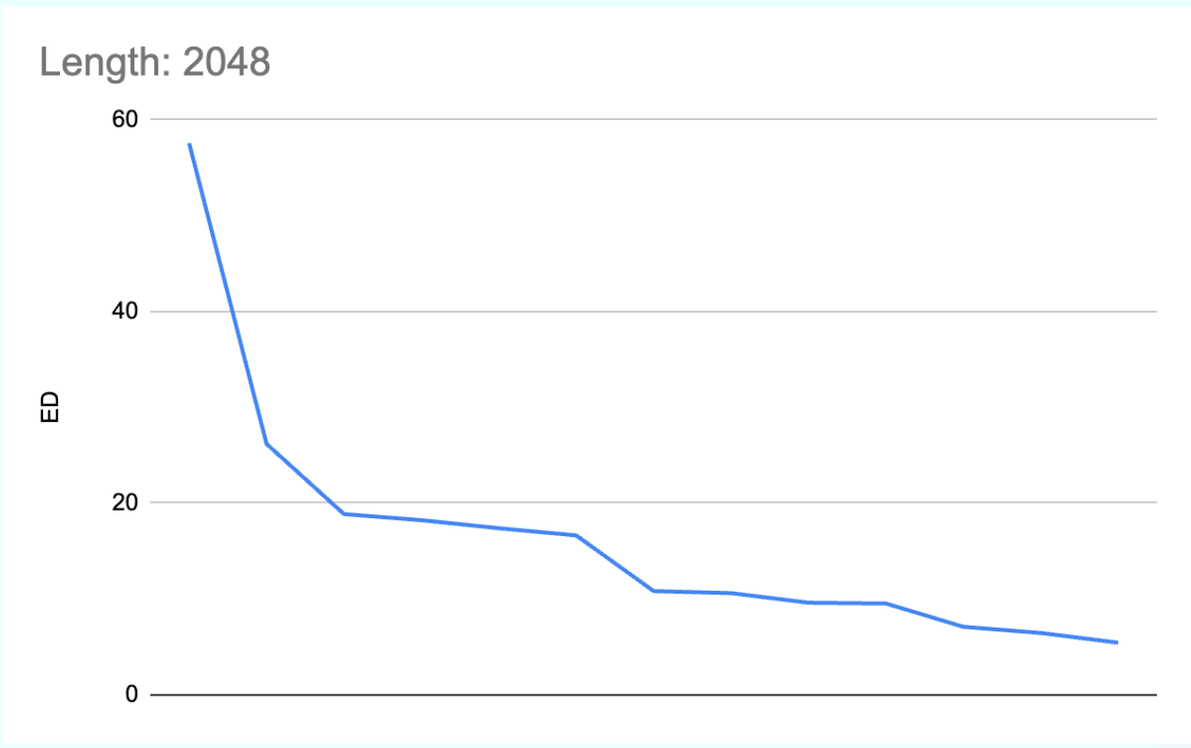
- I choose the model with 200k training steps



Run ↑	Smoothed	Value	Step	Relative
v1/logs	0.6384	0.5765	200,900	8.006 hr
v3/logs	0.2919	0.4357	300,000	14.31 hr
v4/logs	0.3843	0.3764	300,000	14.86 hr

Result





tag	H	GS	ED
v3_00200000_top-p_1.1_0.9_stride_1024	2.4696	0.85	9.6272
v3_00200000_top-p_1.05_0.85_stride_1024	2.3293	0.8639	7.693
v3_00200000_top-p_1.1_0.85_stride_1024	2.326	0.8983	6.9041
v3_00200000_top-p_0.9_0.85_stride_1024	2.3716	0.8736	6.7215
v3_00200000_top-p_0.95_0.85_stride_1024	2.3736	0.8839	6.6595
v3_00200000_top-p_1.0_0.9_stride_1024	2.3927	0.9104	6.6371
v3_00200000_top-p_1.0_0.85_stride_1024	2.3376	0.8807	6.5393
v3_00200000_top-p_1.0_0.95_stride_1024	2.4079	0.8488	4.9172
v3_00200000_top-p_0.9_0.95_stride_1024	2.4474	0.8902	4.3661
v3_00200000_top-p_0.95_0.9_stride_1024	2.4103	0.8383	4.2454
v3_00200000_top-p_1.1_0.95_stride_1024	2.4621	0.8518	3.4743
v3_00200000_top-p_1.05_0.9_stride_1024	2.3833	0.8121	3.466
v3_00200000_top-p_1.05_0.95_stride_1024	2.352	0.8458	3.2817
v3_00200000_top-p_0.9_0.9_stride_1024	2.3367	0.8665	3.0032
v3_00200000_top-p_0.95_0.95_stride_1024	2.4035	0.8102	2.0479
dataset_cache_bd_4	2.5207	0.7789	0

Findings

- I was surprised that even though the dataset is small (~1700 pop song), the small model (with 8 layers) couldn't overfit this dataset (the loss stopped decreasing).
- Extrapolate of RoPE is bad
- Unlike LLM, cache without recomputing is bad

Appendix

- source code: https://github.com/tanchihpin0517/NTU/tree/master/deepmir/hw3
- model: https://www.dropbox.com/scl/fi/9v11rzsip8wjq0r4wy9gc/model_00200000?rlkey=41i4ar5ezj7n1fpo5517ggc6p&dl=0