Linformer: Self-Attention with Linear Complexity

ASCII Paper Reading Group

Kilian Ulrichsohn

Motivation

- Transformer
  - Inference and training time $\sim n^2$
    n is sequence length

Prior Work: Sparse attention

- Each token attends to a subset of tokens
- $O(n \ * \sqrt{n})$
- However: significant performance decrease

Prior Work: Locally sensitive Hashing

- multi-round hashing scheme when computing dot-product attention
- $O(n \log(n))$
- - more sequal operations
- Speed gains only when n > 2048
    - This is in practice not really a limitation since n is greater than 2048 in most GPT applications anyway

Prio Work: Mentionings that have not a lot in common with this paper

- Knowlegde Distillation
  - Train large model -> use large model to train slow model
- Mixed Precision
  - Use fp16 where higher precision is not needed
- Micro Batching
  - separately runs forward and backward passes on microbatches with gradient accumulation
- Gradient Checkpointing

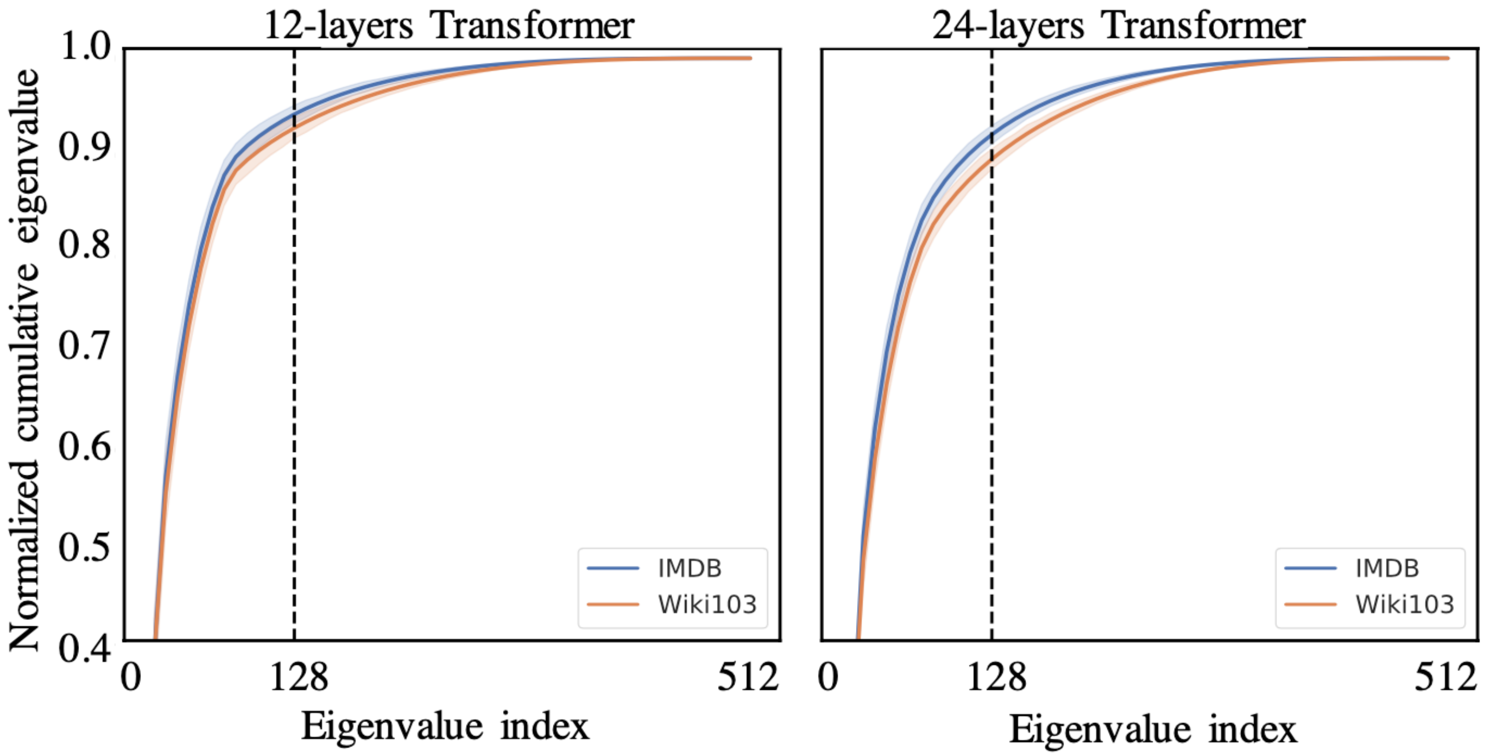
Self-Attention is Low Rank

- $head = \underbrace{softmax\left(\frac{QW^Q(KW^K)^T}{\sqrt{d_k}}\right)}_{P} VW^V$

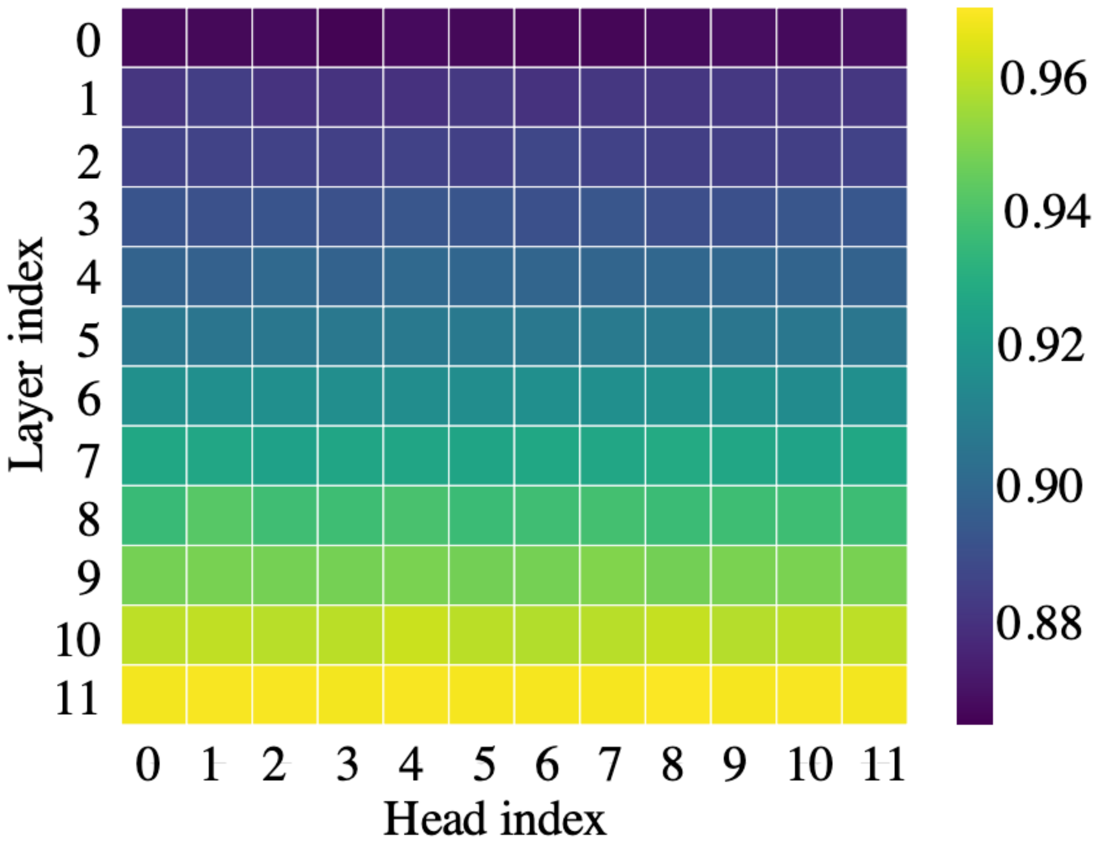
Claim: P is low rank

Observation

- Singular Value Decomposition of P
- $P = U \Sigma V^*$
- Σ is a diagonal matrix with zero padding
- implies that most of the information of matrix P can be recovered from the first few largest singular values



128th largest eigenvalue per layer



- Information in higher layer is mor concentrated in the largest singular value
- Rank P is lower for higher layers

Proof. Based on the definition of the context mapping matrix P, we can write

$$P = \text{ softmax} \underbrace{\left[\frac{QW_i^Q (KW_i^K)^T}{\sqrt{d}} \right]}_{A} = \exp{(A)} \cdot D_A^{-1}, \tag{4}$$

where D_A is an $n \times n$ diagonal matrix. The main idea of this proof is based on the distributional Johnson–Lindenstrauss lemma (Lindenstrauss, 1984) (JL for short). We construct the approximate low rank matrix as $\tilde{P} = \exp{(A)} \cdot D_A^{-1} R^T R$, where $R \in \mathbb{R}^{k \times n}$ with i.i.d. entries from $N(0, 1/k)$. We can then use the JL lemma to show that, for any column vector $w \in \mathbb{R}^n$ of matrix VW_i^V, when $k = 5\log(n)/(\epsilon^2 - \epsilon^3)$, we have

$$\Pr\left(\|PR^T Rw^T - Pw^T\| \le \epsilon \|Pw^T\|\right) > 1 - o(1). \tag{5}$$

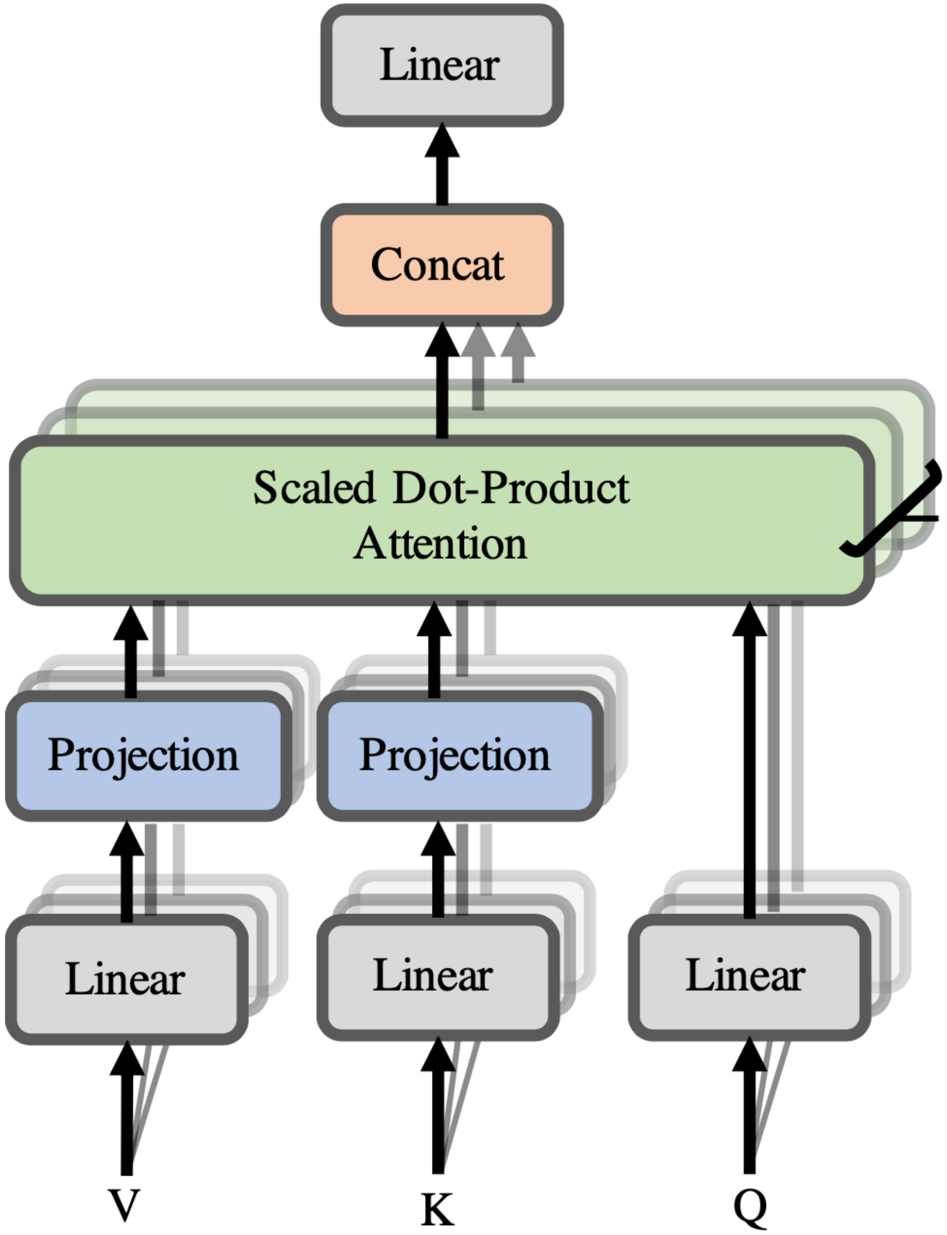
Given the low-rank property of the context mapping matrix P, one straightforward idea is to use singular value decomposition (SVD) to approximate P with a low-rank matrix P_{low}, as follows

$$P \approx P_{\text{low}} = \sum_{i=1}^{k} \sigma_i u_i v_i^T = \underbrace{\begin{bmatrix} u_1, \cdots, u_k \end{bmatrix}}_{k} \operatorname{diag}\{\sigma_1, \cdots, \sigma_k\} \left.\begin{bmatrix} v_1 \\ \vdots \\ v_k \end{bmatrix}\right\} k \tag{6}$$

SVD is **too expensive** to do during training

Model

- Idea: project $V, Q \in R^{n \times d}$ to $R^{k \times d}$
  - So basically a fixed mapping from a sequence of length n to sequence of length k
- Where k is larger than the smaller of
  - $5\,\theta(\log(n)\,(\epsilon^2 - \epsilon^3))$
  - $\theta(9d\log(d)\ \epsilon^{-2})$
- n … sequence length
- d … embedding dimension



$$= \underbrace{\text{softmax}\left(\frac{QW_i^Q(E_iKW_i^K)^T}{\sqrt{d_k}}\right)}_{\bar{P}:n\times k} \cdot \underbrace{F_iVW_i^V}_{k\times d},$$

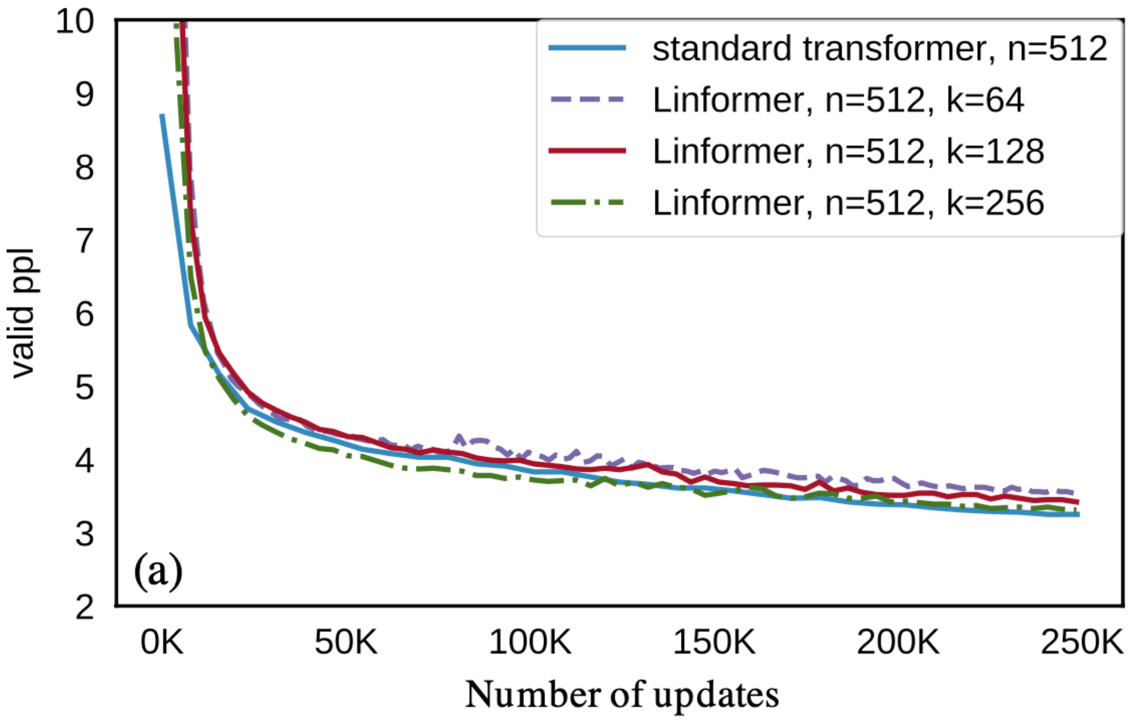
Since SVD is expensive, they E and F are learned matrices

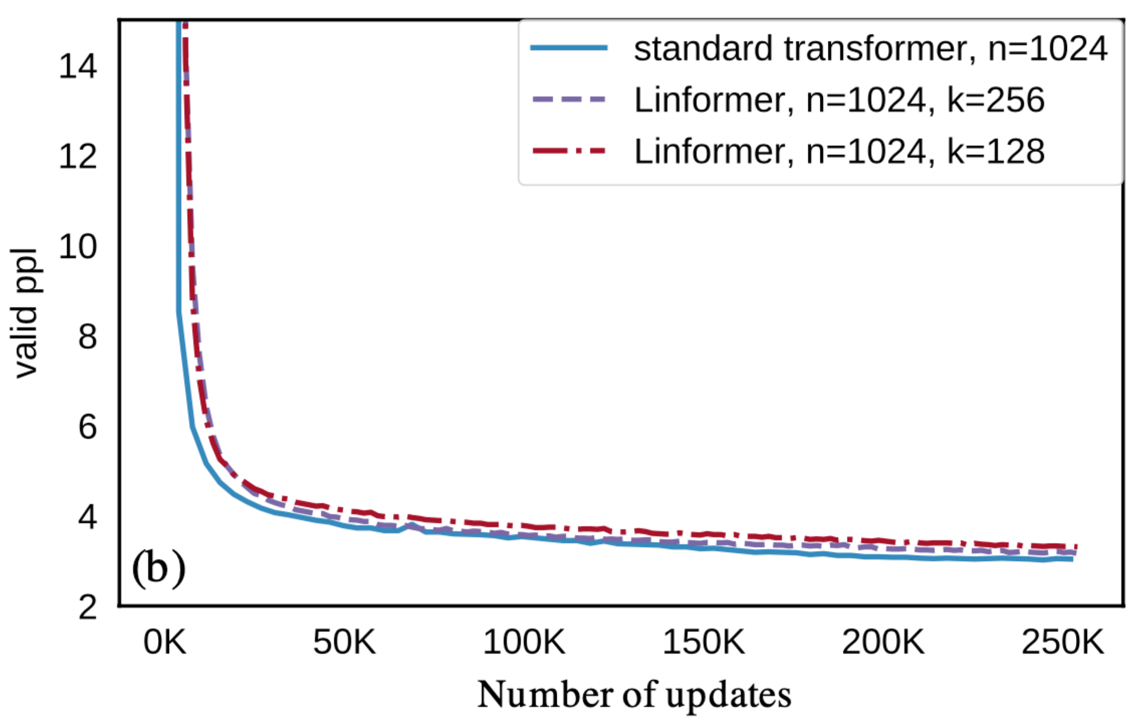
Pre-Training

- Data
  - BookCorpus
  - English Wikipedia
- Task
  - masked-language-modeling
- GPUs
  - 64x Nvidia V100

Pre-Training Observations

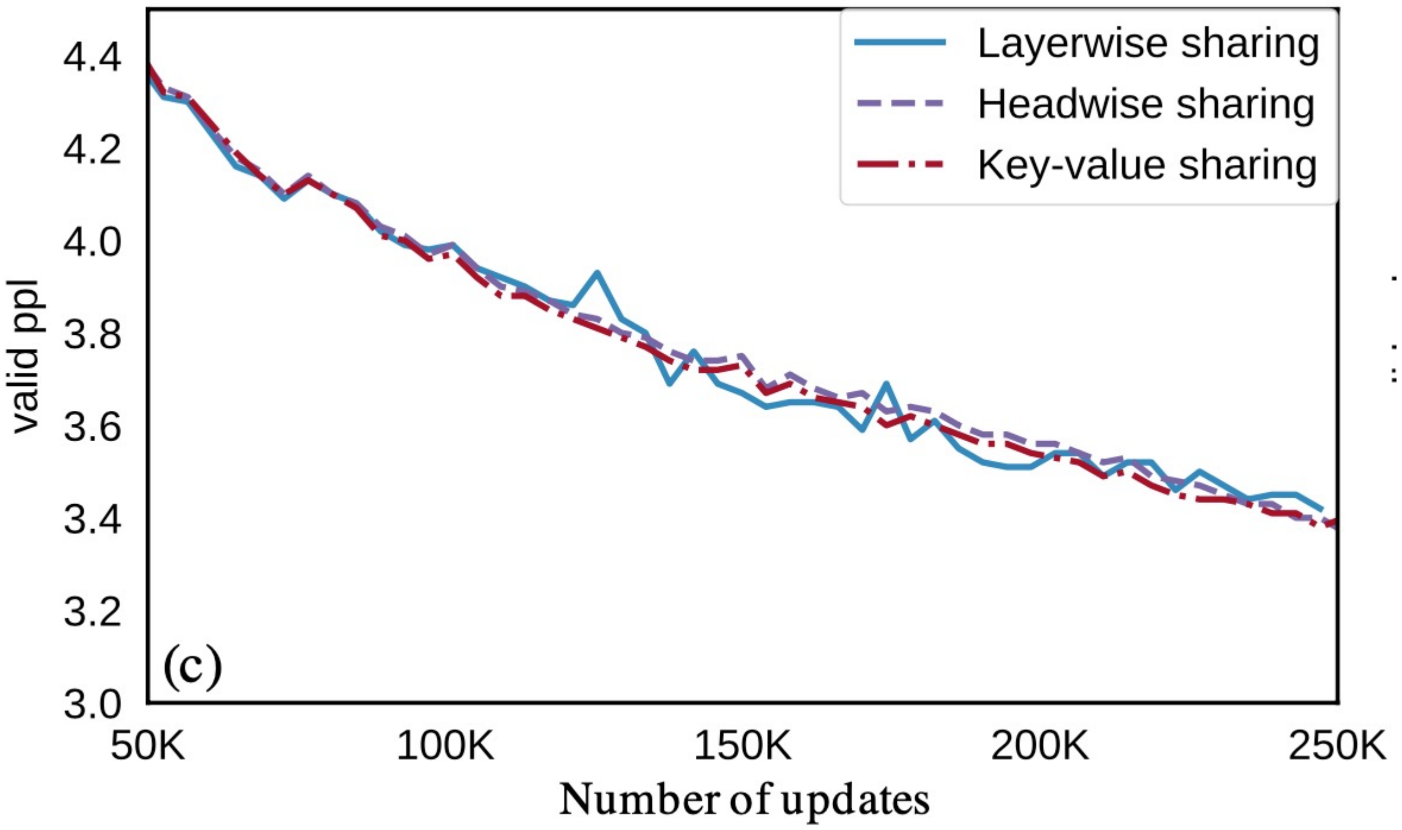
- Effect of k (projected dimension)





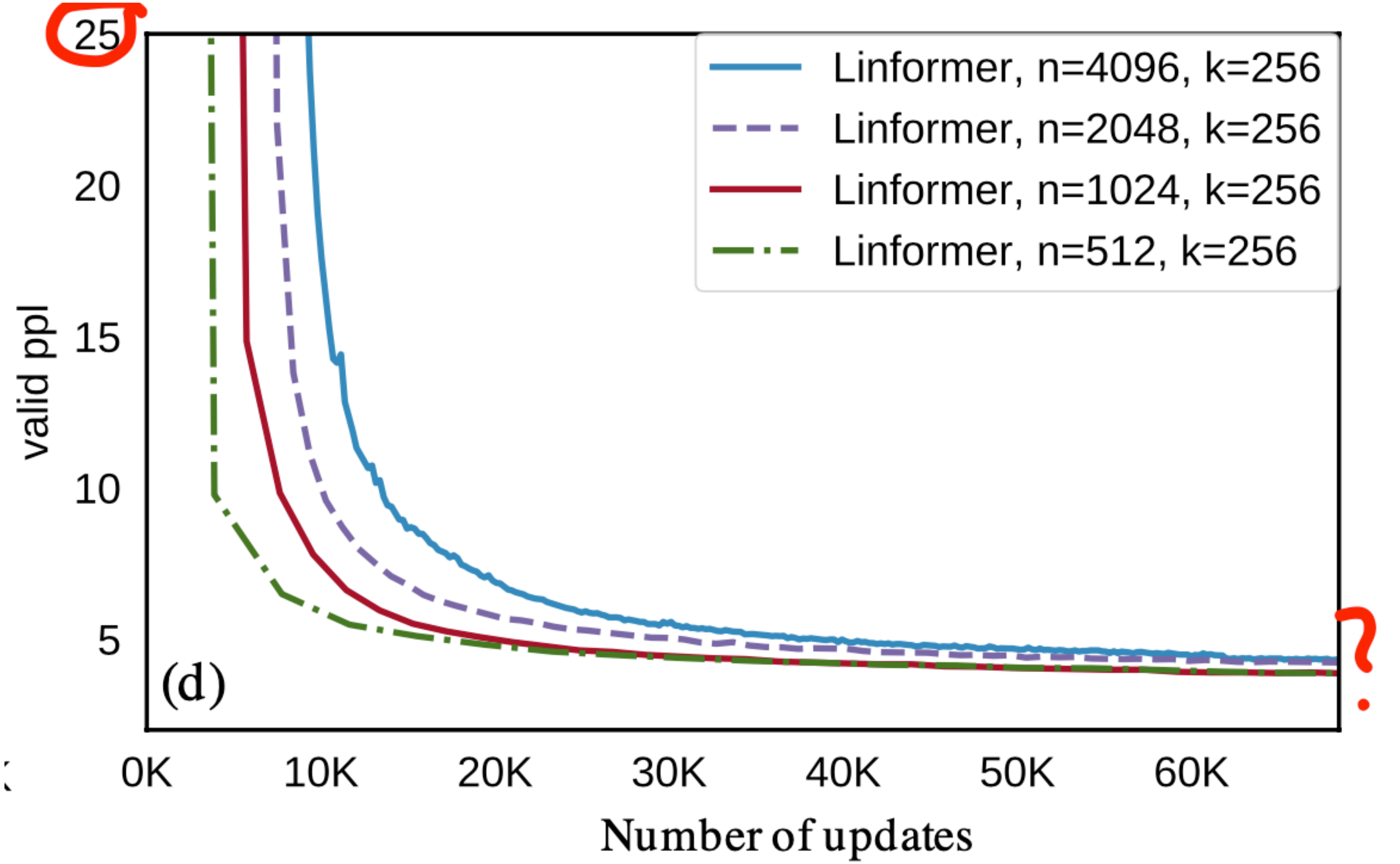
Pre-Training Observations

- Effect of sharing E,F



Pre-Training Observations

- Effect of sequence length n
- Missing baseline
- Scewed scaling
- **Claim**: "performance is equivalent for longer distances"



Downstream tasks

- Claim: equal to slightly better performance

Inference Time - speedup

length n	projected dimensions k 128	256	512	1024	2048
512	1.5x	1.3x	-	-	-
1024	1.7x	1.6x	1.3x	-	-
2048	2.6x	2.4x	2.1x	1.3x	-
4096	3.4x	3.2x	2.8x	2.2x	1.3x
8192	5.5x	5.0x	4.4x	3.5x	2.1x
16384	8.6x	7.8x	7.0x	5.6x	3.3x
32768	13x	12x	11x	8.8x	5.0x
65536	20x	18x	16x	14x	7.9x

~~Lin~~Logformer - Critic

- Claimed complexity O(nk) but k = something*log(n)
- Fixed Sequence length
  - Useless for GPT
- What is learned by E/F?
  - -> Fixed attention/compression matrix
    - Bestcase: averaging of information
    - Worstcase: concentration on fixed positions within the input

Bonus Slide: Long Range Arena

- LRA Score based on multiple scores of different tasks
- Input length: up to 4K
- Source "Long Range Arena: A Benchmark for Efficient Transformers" (2020)