Papers Track

World Model in LLMs

Othello GPT:
[2210.13382] Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task
Actually, Othello-GPT Has A Linear Emergent World Representation — Neel Nanda
[2310.07582] Linear Latent World Models in Simple Transformers: A Case Study on Othello-GPT

Chess GPT:
[2403.15498] Emergent World Models and Latent Variable Estimation in Chess-Playing Language Models

Attention Alternatives

SSM:
[2008.07669] HiPPO: Recurrent Memory with Optimal Polynomial Projections
[2111.00396] Efficiently Modeling Long Sequences with Structured State Spaces
[2212.14052] Hungry Hungry Hippos: Towards Language Modeling with State Space Models
[2312.00752] Mamba: Linear-Time Sequence Modeling with Selective State Spaces
[2403.01590] The Hidden Attention of Mamba Models
[2401.09417] Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model
[2403.19887] Jamba: A Hybrid Transformer-Mamba Language Model

Others:
[2402.19427] Griffin: Mixing Gated Linear Recurrences with Local Attention for Efficient Language Models

Efficient LLMS:
(PDF) Efficient Large Language Models: A Survey

GPT Image Generation

[2010.11929] An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (ViT)
Generative Pretraining from Pixels (ImageGPT by OpenAI)

Parameter Efficient Fine-tuning

[2106.09685] LoRA: Low-Rank Adaptation of Large Language Models
[2305.14314] QLoRA: Efficient Finetuning of Quantized LLMs
[2312.05677] Batched Low-Rank Adaptation of Foundation Models (FLoRA)

Increasing LLM Context

[2212.10947] Parallel Context Windows for Large Language Models
[2305.15262] Revisiting Parallel Context Windows: A Frustratingly Simple Alternative and Chain-of-Thought Deterioration

[2212.06713] Structured Prompting: Scaling In-Context Learning to 1,000 Examples

Papers to Read (unclassified)

Parallel Context Windows for LLMs

[2212.10947] Parallel Context Windows for Large Language Models

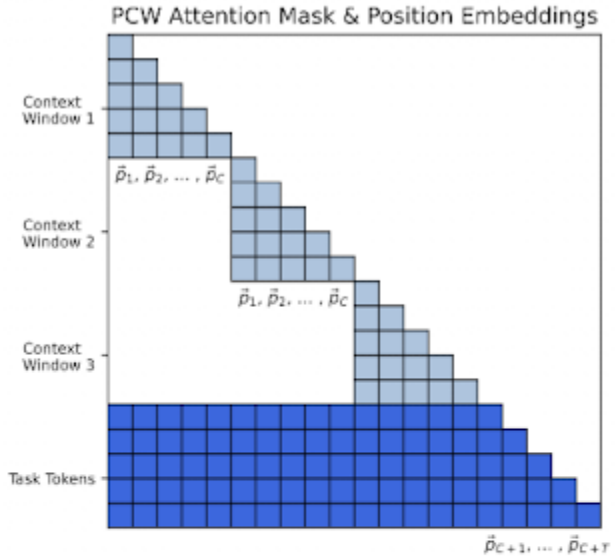


Figure 3: Attention pattern and positional embeddings assignment in PCW. The (i, j) cell in the matrix is colored iff the i^{th} token can attend to the j^{th} token. Each context window (in grey) attends to itself and is assigned positional embeddings ($\vec{p}_i$) independently, thus re-using the positional vectors. Task tokens (in blue) attend to all the windows. PCW makes the attention matrix sparser, effectively parallelizing the processing of multiple windows.

$$a_{ii'}^{b,b'} = \begin{cases} 1, & \text{if } 1 \le i' \le i \le C \text{ and } b = b' \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The following implies that the T task tokens attend to all tokens in all B context windows (for $i > C$):

$$a_{ii'}^{B+1,b'} = \begin{cases} 1, & \text{if } 1 \le i' \le i \le N, \ b' \in [B+1] \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

The above attention masks allow the model to attend to B times more context when decoding the output, while keeping the computational cost linear in the number of parallel contexts B. Overall, for both the above PCW modifications, assigning $B = 1$ corresponds to the vanilla LLM mechanism.

Positional Embeddings Modification:

- Positional Embeddings and Context Window Size: The paper explains that Transformer-based LLMs typically rely on positional embeddings to represent the order of tokens within the input sequence. The number of these embeddings is fixed and corresponds to the LLM's context window size (N).
- Limitations with Standard Positional Embeddings: With a fixed set of positional embeddings, the number of context tokens that can be processed is limited to C = N - T, where T represents the

number of tokens required to formulate the task. This restricts the number of training examples or relevant documents the LLM can access during ICL or open-book QA tasks.
- PCW Approach: PCW expands the number of processable context tokens by a factor of B by replicating the original set of positional embeddings. This allows the LLM to handle an input sequence of B * C + T tokens.
- Positional Embedding Mapping: The paper defines a mapping function that assigns one of the original N positional embedding vectors to each position in the expanded input sequence. This ensures that each context window replica receives a unique set of positional embeddings, signaling to the model that each window is equally "close" to the task tokens.

Attention Mask Modification:

- Attention Mask and Its Role: The attention mechanism in LLMs determines which tokens can attend to which other tokens within the input sequence. This is controlled by an attention mask, represented as a matrix of scores where a score of 1 indicates that attention is allowed and 0 indicates it is not.
- Standard Attention Mask: Typical autoregressive LLMs use a causal attention mask where a token can only attend to tokens that precede it in the sequence. This ensures that predictions for a given position depend only on past information.
- PCW Attention Mask: PCW modifies the attention mask to restrict attention within each context window replica. This means tokens within a replica can only attend to other tokens in the same replica, not to tokens in other replicas. However, the task tokens can attend to all context tokens across all replicas.

[2305.15262] Revisiting Parallel Context Windows: A Frustratingly Simple Alternative and Chain-of-Thought Deterioration

They propose that Parallel Ensemble (PE), i.e. using multiple LLMs in parallel over the context and taking an ensemble over the results, gives comparable or better performance than PCW.

Focusing on PCW:

- The paper specifically targets the PCW method proposed by Ratner et al. (2023) as a solution to the context window limitation. PCW utilizes window-wise attention and positional embedding techniques to expand the context window size of LLMs.
- The authors acknowledge PCW's reported success in improving in-context learning (ICL) performance, particularly on classification tasks with large label spaces. However, they identify two major limitations in PCW's evaluation:
    - Unfair Comparison: PCW's gains are demonstrated by comparing it to a single-sequence ICL baseline with fewer examples. This creates an unfair advantage for PCW, as it has access to more information during inference.
    - Unchallenging Tasks: The evaluation of PCW focuses primarily on simpler classification and generation tasks, leaving its effectiveness on more complex reasoning tasks involving CoT unexplored.

PCW is equivalent to Weighted Sum Ensemble for classification:

- Comparable Performance: The results on classification tasks reveal that PCW and PE (Parallel Ensemble) achieve similar performance across most datasets. In some cases, PE even slightly outperforms PCW.
- Explanation: This observation suggests that PCW's improvements might be attributed to a simple ensemble effect, where the model effectively averages the predictions from each context window.

- Questioning Practicality: The authors argue that PCW might not offer significant advantages over simpler ensemble methods like PE, especially considering the potential downsides of PCW, such as increased computational complexity and potential negative impact on reasoning abilities.

Concerns about PCW: The authors reiterate their concerns about the effectiveness and practicality of PCW, highlighting two main issues:

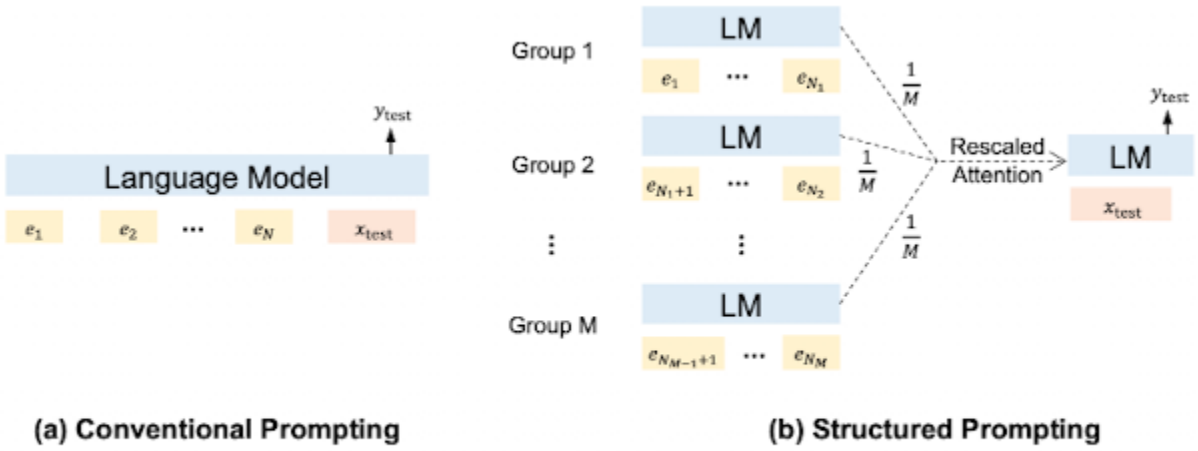
- Functionally Equivalent to Ensemble: The results from the experiments suggest that PCW's improvements are largely due to a simple ensemble effect across multiple context windows, rather than a genuine improvement in understanding long contexts. This raises questions about the necessity of the complex architectural modifications introduced by PCW.
- Degradation of Reasoning Abilities: PCW's negative impact on CoT reasoning abilities is a significant concern, especially considering the growing importance of complex reasoning tasks in LLM applications. The observed issues with false reasoning, question misinterpretation, and lack of CoT reasoning indicate that PCW might hinder the model's ability to effectively process and utilize information from multiple sources.

Critique of Parallel-Integrated Methods:

- The authors extend their critique to other parallel-integrated methods similar to PCW, arguing that while they may show improvements on specific tasks like multi-class classification, they primarily achieve this through brute-force ensembling of information from individual windows. This approach, they argue, can weaken the model's logical reasoning and knowledge comprehension abilities.

Structured Prompting: Scaling ICL

This is also similar in idea to PCW.



(a) Conventional Prompting **(b) Structured Prompting**

Figure 1: The illustrations of conventional prompting and structured prompting. We first encode group-structured exemplars independently. Then all exemplars are incorporated into the test input through the rescaled attention mechanism.

It does ICL in these 2 way:

- Grouped Context Encoding:
  - Using Right-Aligned Positional Embedding.This is to make sure all the contexts have the same maximum positional index (hence, all groups have the same relative distance wrt the test input).
- Structured Prompting:
  - Rescaled Attention

Rescaled Attention We use x instead of $\mathcal{T}(x_{\text{test}})$ for brevity. In each layer, we concatenate the keys and values of all exemplars and the test input, i.e., $\hat{K} = [K_{\mathcal{Z}_1}, ..., K_{\mathcal{Z}_M}, K_x]$, $\hat{V} = [V_{\mathcal{Z}_1}, ..., V_{\mathcal{Z}_M}, V_x]$. The test input x attends both demonstrations and itself with causal masks. Then the attention output is computed via:

$$\text{Attention}(Q_x, \hat{K}, \hat{V}) = A\hat{V} \tag{1}$$

$$A_{ij} \propto \begin{cases} M\exp(\frac{\boldsymbol{q}_i \cdot \boldsymbol{k}_j}{\sqrt{d}}) & j \in x \\ \exp(\frac{\boldsymbol{q}_i \cdot \boldsymbol{k}_j}{\sqrt{d}}) & j \in \mathcal{Z}_1, \ldots, \mathcal{Z}_M \end{cases} \tag{2}$$

where $\sum_j A_{ij} = 1$, the query vector $\boldsymbol{q}_i \in Q_x$, the key vector $\boldsymbol{k}_j \in \hat{K}^\intercal$, and d is dimension of queries and keys.