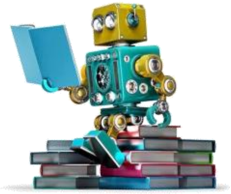
Attention Sinks and Spectral Filters

April 11, 2023



Efficient Streaming Language Models with Attention Sinks

https://arxiv.org/abs/2309.17453

Preprint

Efficient Streaming Language Models with Attention Sinks

Guangxuan Xiao[1*] Yuandong Tian[2] Beidi Chen[3] Song Han[1] Mike Lewis[2]

[1] Massachusetts Institute of Technology
[2] Meta AI
[3] Carnegie Mellon University
https://github.com/mit-han-lab/streaming-llm

arXiv:2309.17453v1 [cs.CL] 29 Sep 2023

Abstract

Deploying Large Language Models (LLMs) in streaming applications such as multi-round dialogue, where long interactions are expected, is urgently needed but poses two major challenges. Firstly, during the decoding stage, caching previous tokens' Key and Value states (KV) consumes extensive memory. Secondly, popular LLMs cannot generalize to longer texts than the training sequence length. Window attention, where only the most recent KVs are cached, is a natural approach — but we show that it fails when the text length surpasses the cache size. We observe an interesting phenomenon, namely *attention sink*, that keeping the KV of initial tokens will largely recover the performance of window attention. In this paper, we first demonstrate that the emergence of *attention sink* is due to the strong attention scores towards initial tokens as a "sink" even if they are not semantically important. Based on the above analysis, we introduce StreamingLLM, an efficient framework that enables LLMs trained with a *finite length* attention window to generalize to *infinite sequence length* without any fine-tuning. We show that StreamingLLM can enable Llama-2, MPT, Falcon, and Pythia to perform stable and efficient language modeling with up to 4 million tokens and more. In addition, we discover that adding a placeholder token as a dedicated attention sink during pre-training can further improve streaming deployment. In streaming settings, StreamingLLM outperforms the sliding window recomputation baseline by up to 22.2× speedup. Code and datasets are provided in the link.

1 Introduction

Large Language Models (LLMs) (Radford et al., 2018; Brown et al., 2020; Zhang et al., 2022; OpenAI, 2023; Touvron et al., 2023a;b) are becoming ubiquitous, powering many natural language processing applications such as dialog systems (Schulman et al., 2022; Taori et al., 2023; Chiang et al., 2023), document summarization (Goyal & Durrett, 2020; Zhang et al., 2023a), code completion (Chen et al., 2021; Rozière et al., 2023) and question answering (Kamalloo et al., 2023). To unleash the full potential of pretrained LLMs, they should be able to efficiently and accurately perform long sequence generation. For example, an ideal ChatBot assistant can stably work over the content of recent day-long conversations. However, it is very challenging for LLM to generalize to longer sequence lengths than they have been pretrained on, e.g., 4K for Llama-2 Touvron et al. (2023b).

The reason is that LLMs are constrained by the attention window during pre-training. Despite substantial efforts to expand this window size (Chen et al., 2023; kaiokendev, 2023; Peng et al., 2023) and improve training (Dao et al., 2022; Dao, 2023) and inference (Pope et al., 2022; Xiao et al., 2023; Anagnostidis et al., 2023; Zhang et al., 2023b) efficiency for lengthy inputs, the acceptable sequence length remains intrinsically *finite*, which doesn't allow persistent deployments.

In this paper, we first introduce the concept of LLM streaming applications and ask the question:

Can we deploy an LLM for infinite-length inputs without sacrificing efficiency and performance?

[*] Part of the work done during an internship at Meta AI.

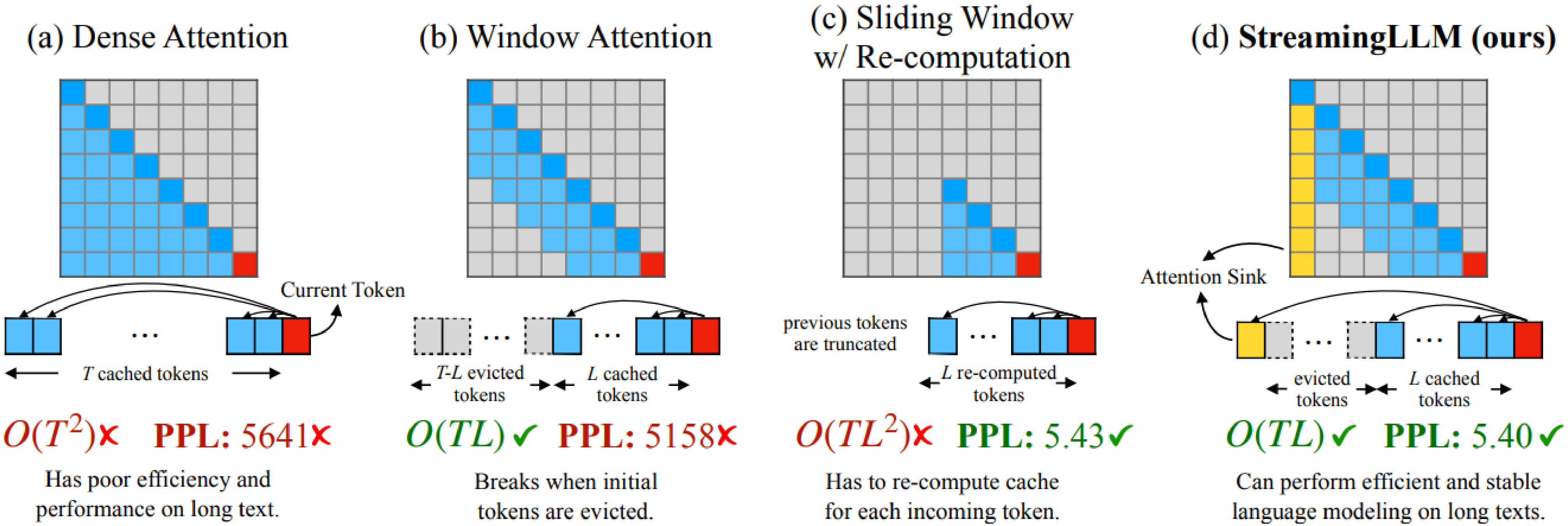
Attention sinks overview

- Efficient Streaming Language Models with Attention Sinks
- Guangxuan Xiao et al. from MIT+

- Challenge: continue generate text, such as in a multi-round chatbot, even after the history has exceeded your LLM’s max context length
- The obvious solution is to use a sliding window of most recent tokens, but this leads to severe performance degradation
- Prior “Heavy Hitter” paper showed some tokens get a lot of attention. But that paper didn’t identify which ones.
- Keeping an attention sink of first token(s) solves performance problem

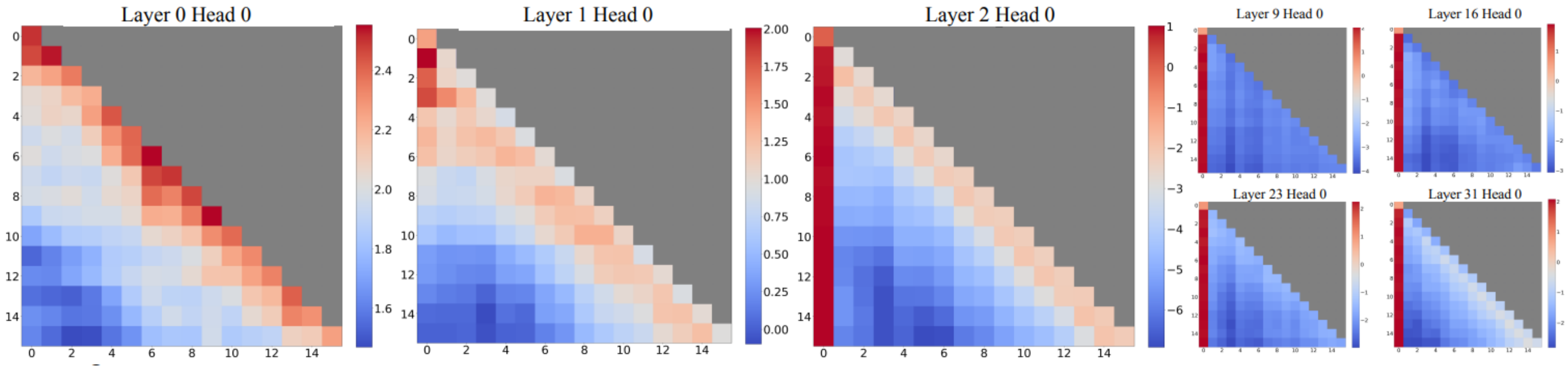
Long context approaches

- Regular sliding window with cached KV has bad quality (perplexity)
- Re-computation with sliding window doesn’t cache, so hurts speed
- Attention sink approach uses caching and maintains text quality



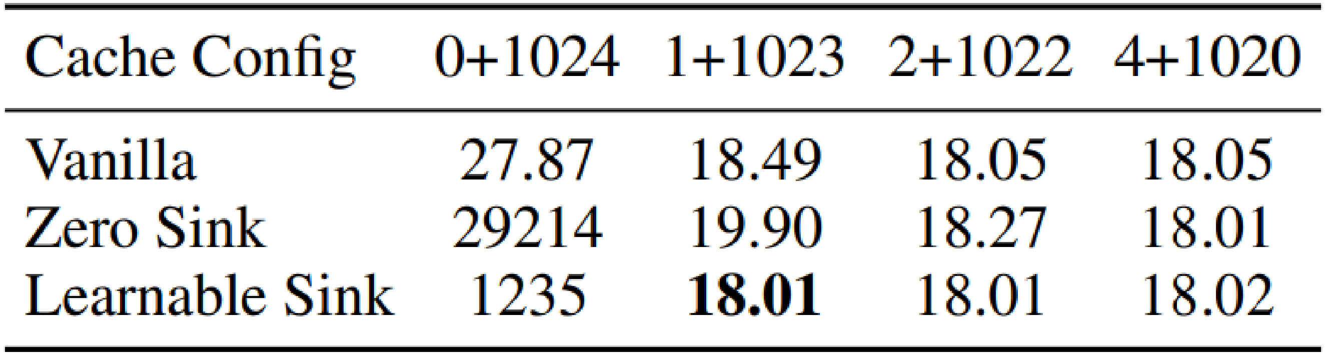
Why sliding window performance is impacted

- The attention maps in autoregressive LLMs put a lot of attention on the first token(s), which they name *attention sinks*
  - This was seen in the transformer circuits work by Anthropic https://transformer-circuits.pub/2021/framework/index.html
  - The softmax in attention must put attention somewhere – it sums to one
  - Test showed it is the position, not the content of first tokens, that matters



Attention sinks

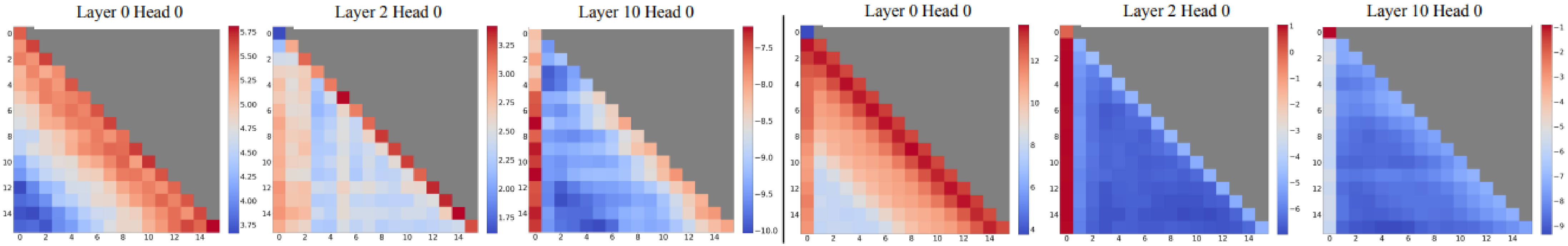
- Experiments showed most models require first four tokens to recover normal text quality (perplexity)
- Cached KV values use positions values 1 to L, regardless of if current tokens are well beyond the Lth token
- Using RoPE or ALiBi, the base token embeddings can be cached, then the positional information applied on top of the base embeddings
- Pre-training a model with a learned attention sink reduces requirement to a single attention sink token



Cache Config	0+1024	1+1023	2+1022	4+1020
Vanilla	27.87	18.49	18.05	18.05
Zero Sink	29214	19.90	18.27	18.01
Learnable Sink	1235	**18.01**	18.01	18.02

Pre-trained model attention patterns

- Attention on first token is more consistent in models pre-trained with an attention sink token



Layer 0 Head 0 · Layer 2 Head 0 · Layer 10 Head 0 · Layer 0 Head 0 · Layer 2 Head 0 · Layer 10 Head 0

Pre-Trained without Sink Token

Pre-Trained with Sink Token

- These pre-trained models had similar performance on benchmarks when compared to models without attention sinks

Attention sinks conclusion

- Using attention sinks during decoding solves text generation quality after the context exceeds the max sequence length
  - This approach works with KV caching when using additive, RoPE, or ALiBi position embeddings
- There’s almost no downside, since sparing a few tokens is negligible
- Pre-training models to expect attention sinks reduces cost to just a single attention sink token
- Context length is still a problem for LLMs. This work adds to the growing work to understand transformers.

References

- H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models
  Zhenyu Zhang et al. (2023)
  https://arxiv.org/abs/2306.14048
- RoFormer: Enhanced Transformer with Rotary Position Embedding
  Jianlin Su et al. (2021)
  https://arxiv.org/abs/2104.09864
- Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation
  Ofir Press et al. (2021)
  https://arxiv.org/abs/2108.12409

Spectral Filters, Dark Signals, and Attention Sinks

https://arxiv.org/abs/2402.09221

Spectral Filters, Dark Signals, and Attention Sinks

Nicola Cancedda

FAIR at Meta

Projecting intermediate representations onto the vocabulary is an increasingly popular interpretation tool for transformer-based LLMs, also known as the *logit lens* (Nostalgebraist). We propose a quantitative extension to this approach and define *spectral filters* on intermediate representations based on partitioning the singular vectors of the vocabulary embedding and unembedding matrices into bands. We find that the signals exchanged in the tail end of the spectrum are responsible for attention sinking (Xiao et al., 2023), of which we provide an explanation. We find that the loss of pretrained models can be kept low despite suppressing sizeable parts of the embedding spectrum in a layer-dependent way, as long as attention sinking is preserved. Finally, we discover that the representation of tokens that draw attention from many tokens have large projections on the tail end of the spectrum.

Date: February 15, 2024

Correspondence: Nicola Cancedda at ncan@meta.com

1 Introduction

Large foundation models dominate the state of the art in numerous AI tasks. While we understand how these models work in terms of elementary operations, and black-box evaluations help characterize observable behaviours, we lack a clear understanding of the connection between the two.

There is a growing body of work providing insights into properties of model components, e.g. (Voita et al., 2019; Pimentel et al., 2020; Voita and Titov, 2020; Geva et al., 2022; Meng et al., 2022; Voita et al., 2023), as well as identifying and explaining fundamental phenomena, often with the support of simple models (Elhage et al., 2021, 2022; Olsson et al., 2022; Todd et al., 2023).

Most recent works assign a central role to the model's *residual stream* (RS) as the shared communication channel between model components. In this perspective, the probability distribution of a token is initialised from the projection of the embedding of the previous token through the unembedding matrix, and receives additive updates from attention heads and MLP components, each reading from the residual stream of the same or previous tokens. The role played by components is interpreted projecting their contribution on the probability distribution over vocabulary items, in what is referred to as the *logit lens* (Nostalgebraist; Geva et al., 2020). We extend this approach and introduce *logit spectroscopy*, the spectral analysis of the content of the residual stream and of the parameter matrices interacting with it. Equipped with this tool, we look at the part of the residual stream spectrum that is most likely to be neglected by the logit lens: the linear subspace spanned by the right singular vectors of the unembedding matrix with the *smallest* singular values. Drawing an analogy with "dark matter" in astrophysics, that interacts with light only indirectly, we dub projections onto this subspace *dark* parameters, features, activations etc.

Figure 1 Spectral filters project signals exchanged between components onto selected subspaces as defined by the spectral decomposition of the vocabulary embedding and unembedding matrices of the model.

We were motivated by the thought that LLMs could learn to use signals in the dark linear subspace to maintain global features responsible for long-range dependencies while minimizing their interference with the next token prediction. We discovered instead that

Spectral filters overview

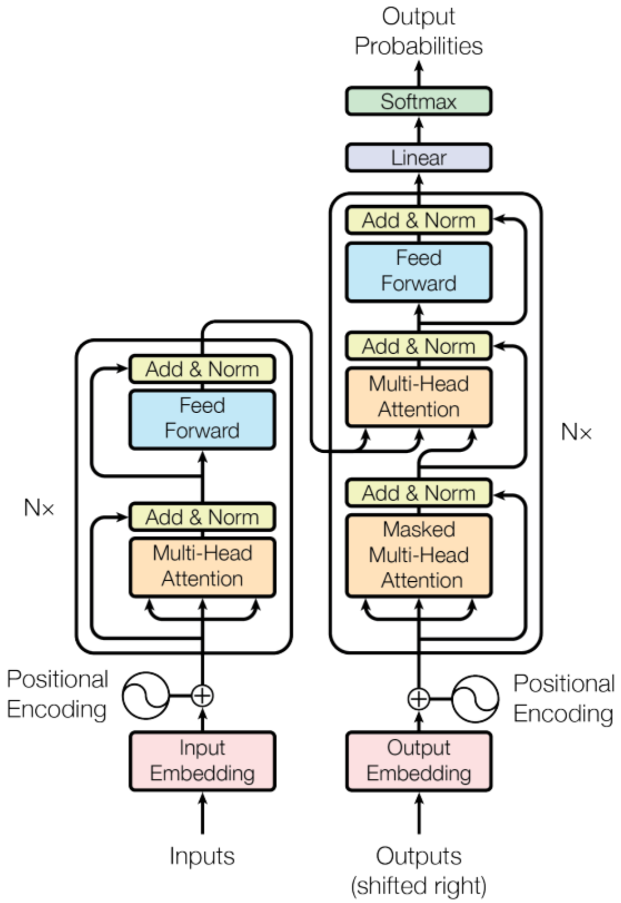
- Spectral Filters, Dark Signals, and Attention Sinks
- Nicola Cancedda from FAIR

- Studied Llama2, the 7B, 13B, and 70B sizes
- Used the SVD decomposition of the embedding and unembedding matrices to explore linear subspaces of the residual stream in $\mathbb{R}^d$
- Whereas in PCA you focus on the few largest singular values, this paper found interesting properties of the smallest 5%
  - Named the data in subspace of the smallest 5% *dark signals*
- Performed a few experiments to show that dark signals are instrumental to implementing attention sinks

Decoder-only LLM data flow

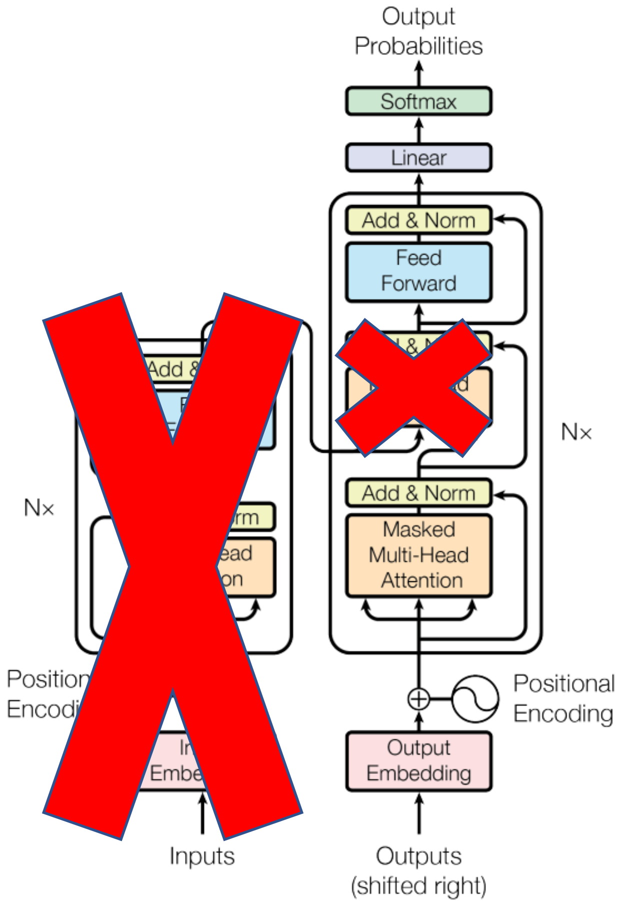
- We try to crystallize understanding of the flow of information on the residual stream from embedding, across layers, and to unembedding
- Starting with the original attention diagram, we step-by-step make small modifications until we have a diagram with explicit token positions and layers, eliminating the need for mental abstraction
- To understand spectral filtering, we think of the residual stream as a vector space
  - The logit lens used this concept to test unembedding at earlier layers
  - Note that the tuned lens considered that the residual stream might be an affine space with different bias at each layer. This paper does not address that explicitly, using the approximation that it is the same vector space the entire way from embedding to unembedding.

Original Transformer

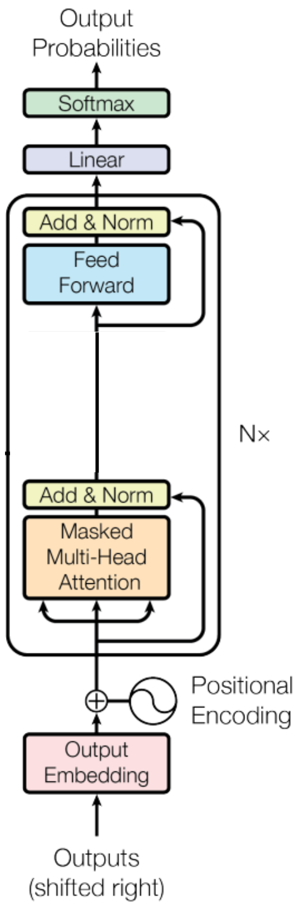


"Attention Is All You Need," Vaswani et al., https://arxiv.org/abs/1706.03762

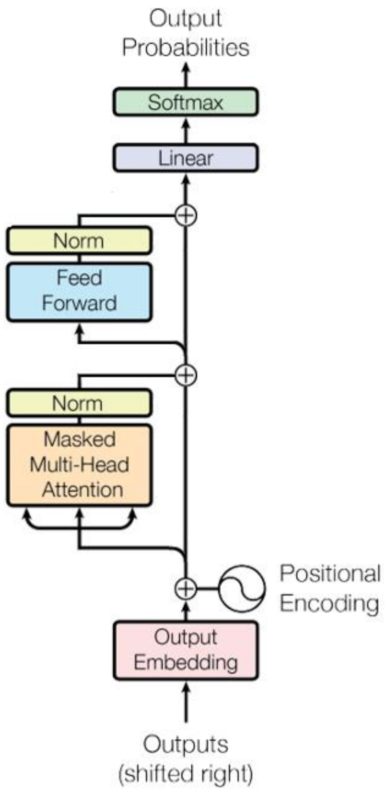
Decoder-only Transformer – 1



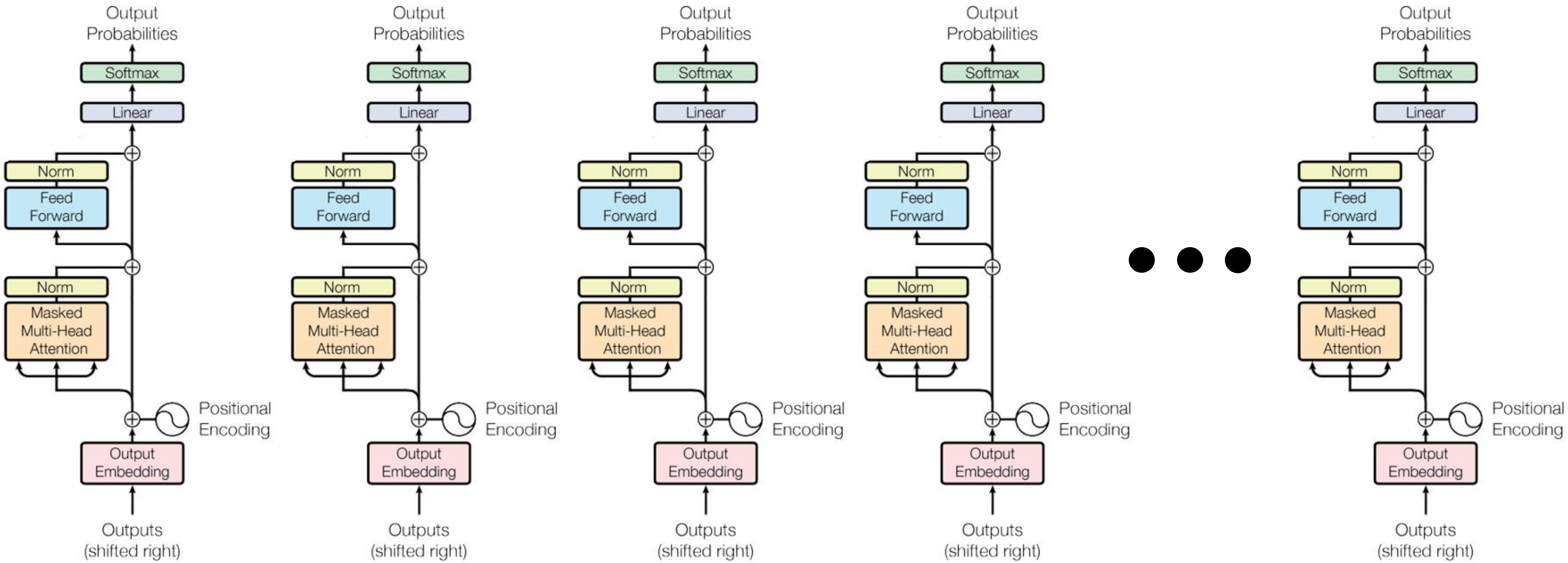
Decoder-only Transformer – 2



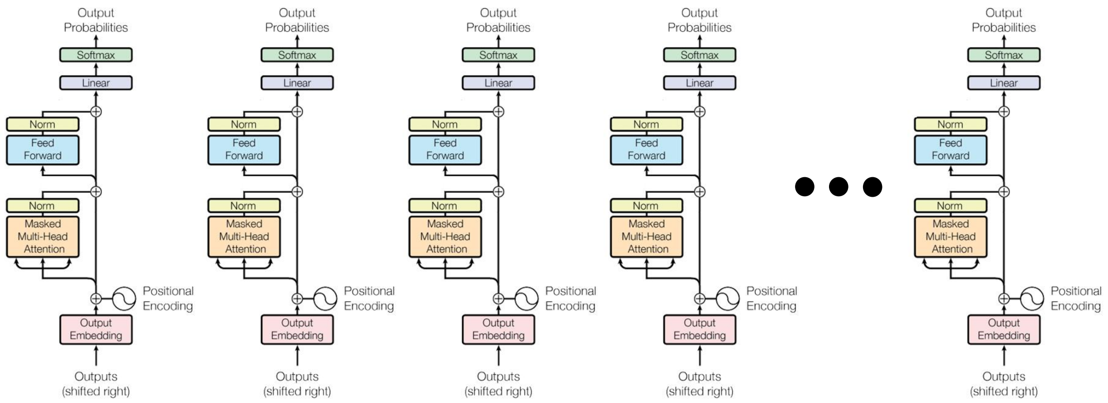
Decoder-only Transformer, Straight Through



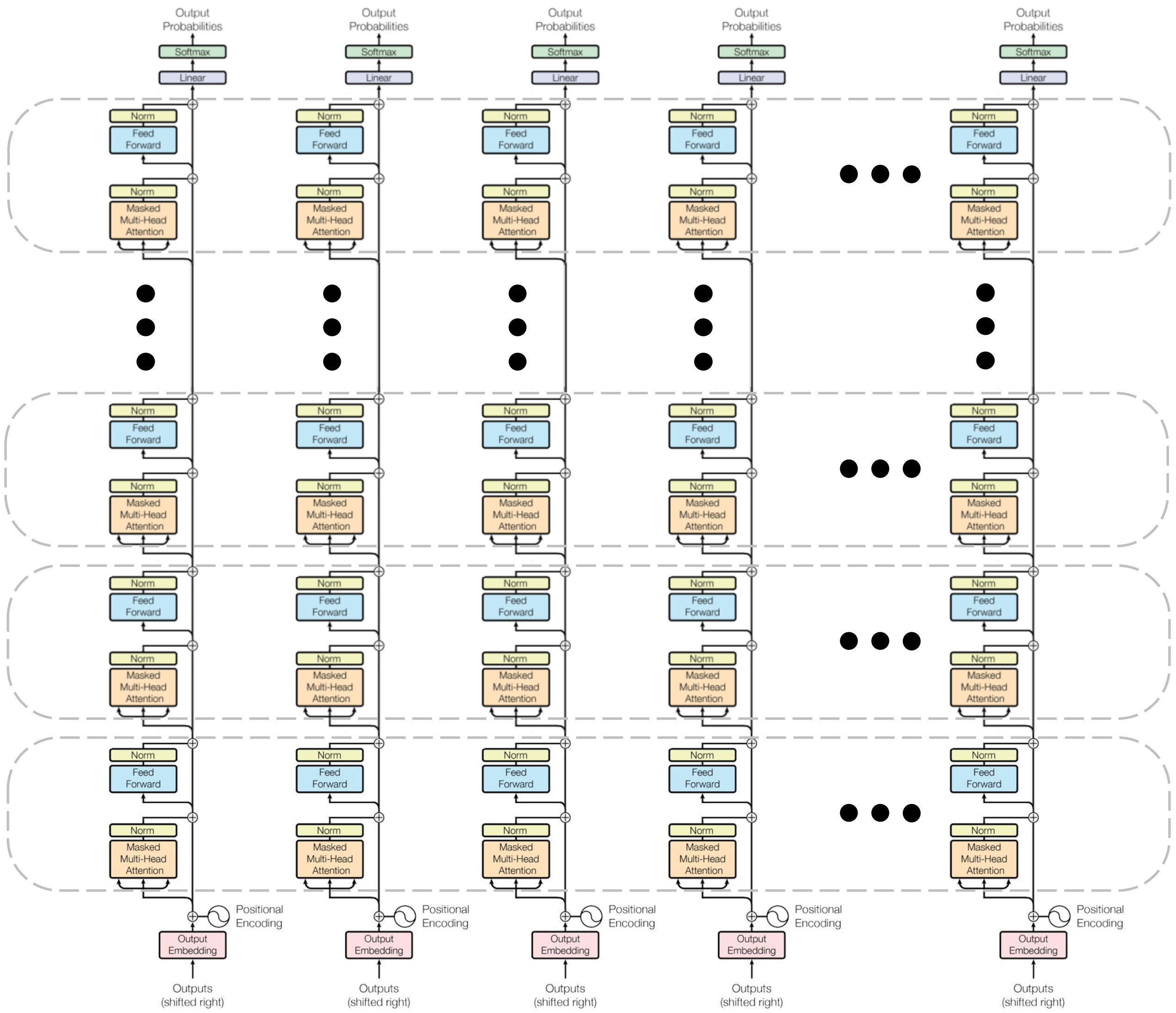
Decoder, Multiple Tokens



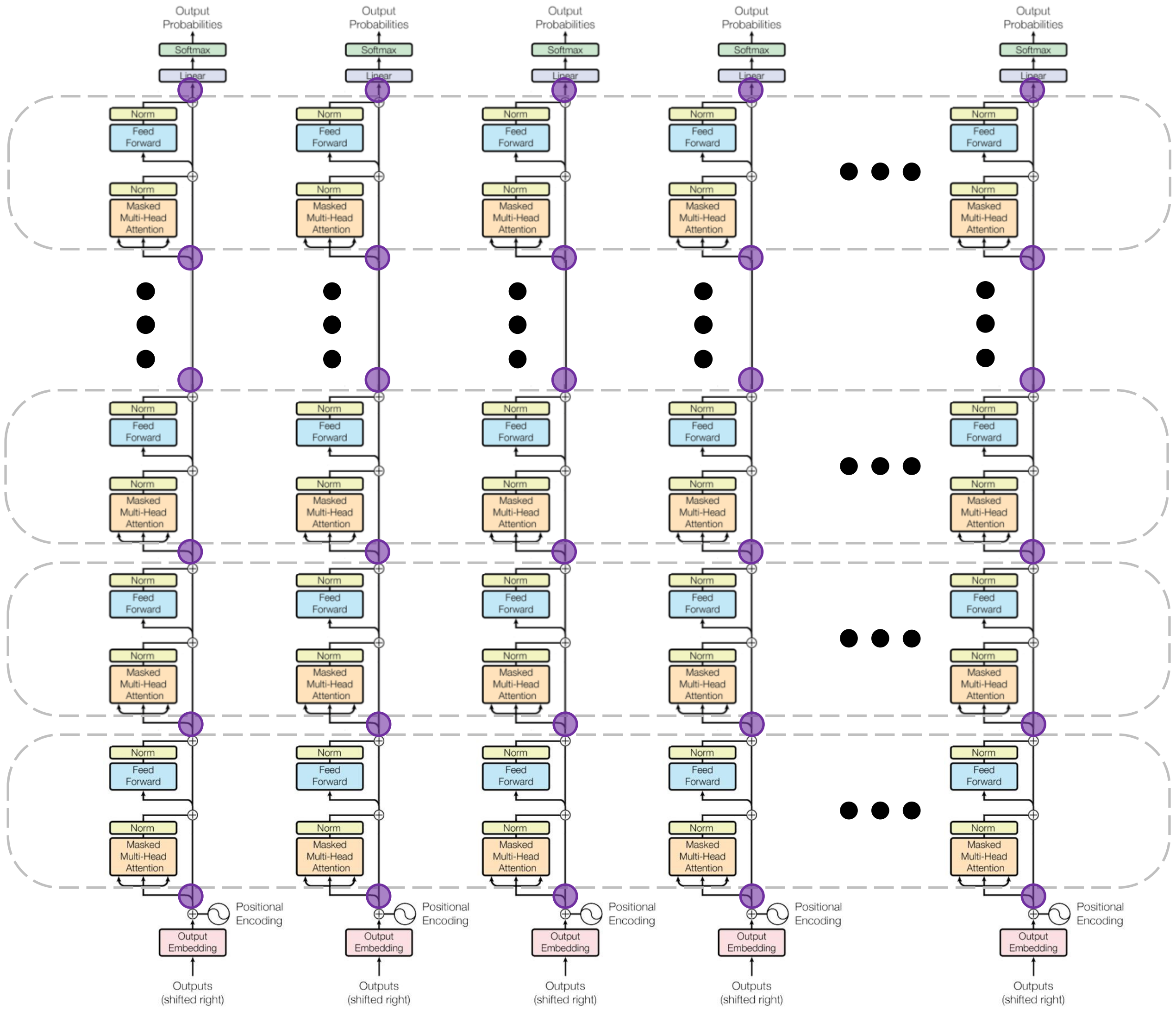
Decoder, Multiple Tokens, Small



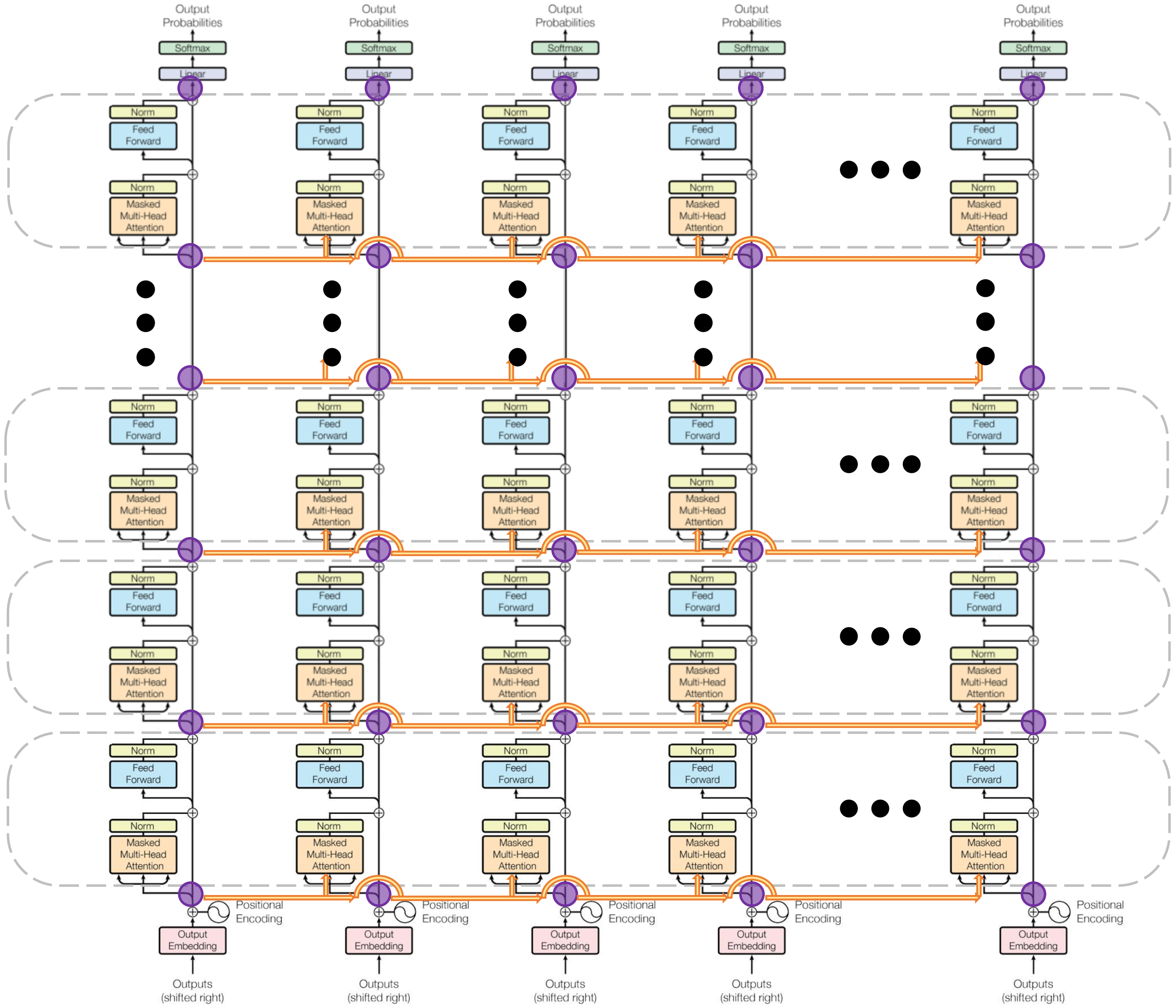
Decoder, Adding Multiple Layers



Decoder, Adding Hidden States



Decoder, Adding Attention Paths

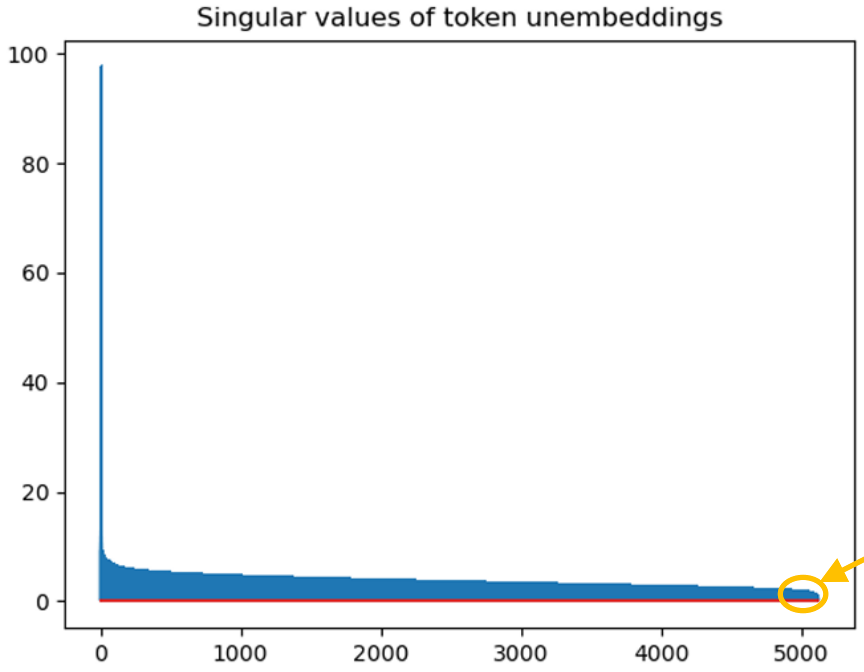


Llama2 and notation

- Llama2 7B, 13B, and 70B have model dimensions, d, of size 4096, 5120, and 8192 respectively
- Attention uses weights for Q, K, V, and O
- The MLP uses SwiGLU activation, with weights W_1, W_2, and W_3
  - For input x, output is $(\text{Swish}_1(xW_1) \otimes xW_3)W_2$
- This means that Q, K, V, W_1, and W_3 read from the residual stream and O and W_2 write to the residual stream
- Note also that Llama2 always begins with a special beginning of sentence `BoS` token, so we know that token 0 will always be `BoS`

SVD of W_u

- Singular values of the unembedding matrix W_u have a huge first singular value, then a flat tail. (W_e is similar)
- The last 5% right singular vectors of W_u (or W_e) are called the *dark basis*, and the linear subspace they span contain what they called *dark signals*



Smallest 5% of singular values

Projections of weight matrices onto W_u

- Large norms of W_o and W_2 when multiplied by dark singular vectors (yellow ovals) indicate writing to the dark basis

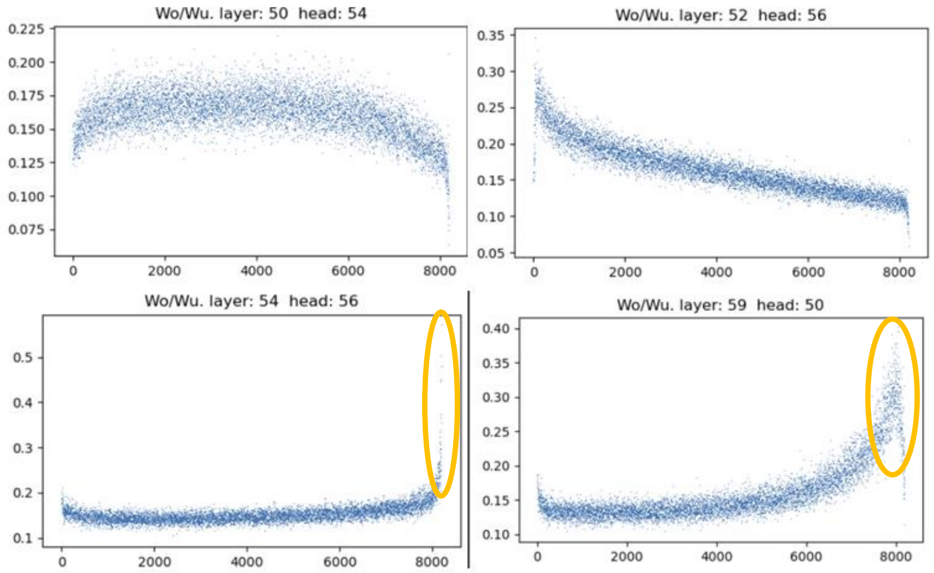


Figure 3 The projections of four W_o matrices of LLaMa2 70B on the RSVs of W_u. Different heads are equipped to write into different subspaces, with some targeting the dark subspace.

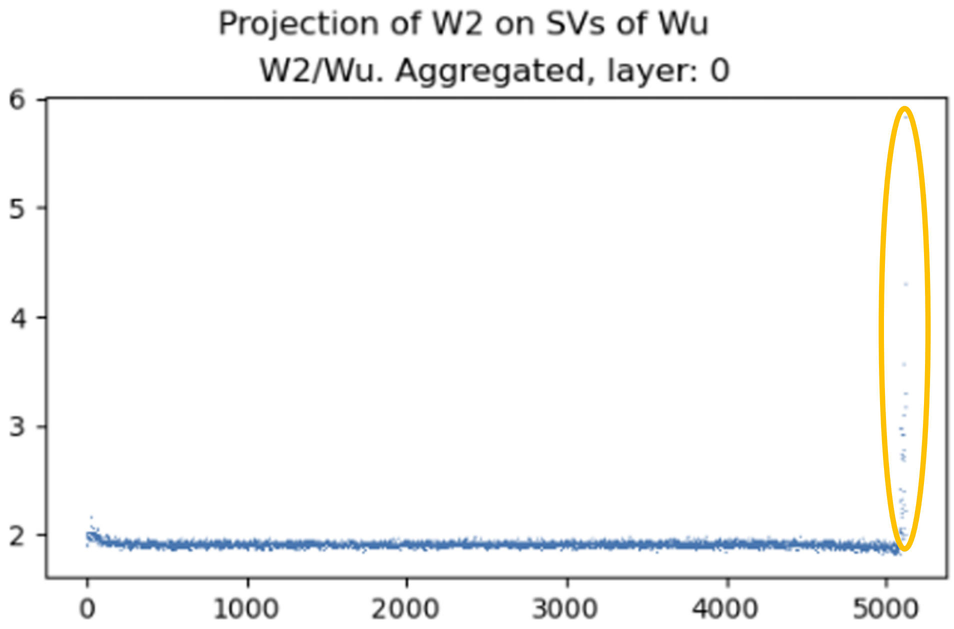


Figure 4 The projection of the rows of W_2 at L0 of LLaMa2 13B on the RSVs of W_u. Note the large values at the very right end of the spectrum, indicating the ability to write in the U-Dark space.

Spectral filtering

- For V, a subset of the right singular vectors, VV^{T} creates a matrix, named Φ, which can be used to project a vector onto their subspace
  - This is a spectral filter. Can be used with W_u or W_e. Applied in two locations.

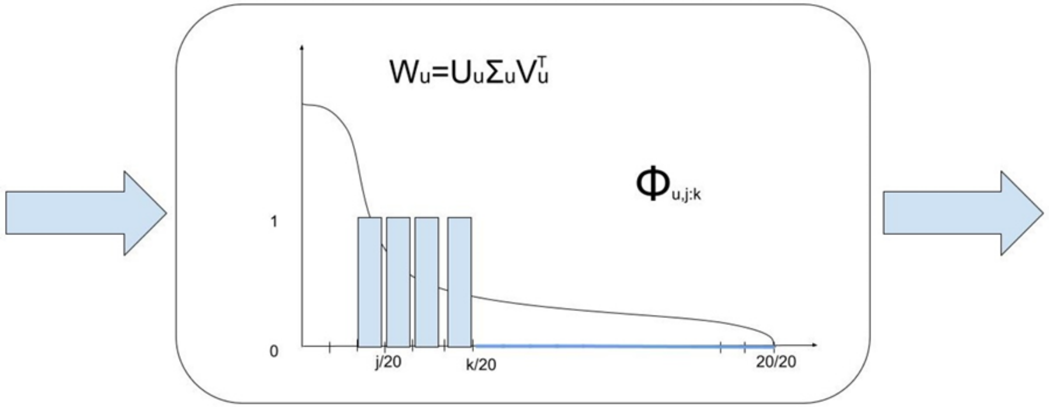


Figure 1 Spectral filters project signals exchanged between components onto selected subspaces as defined by the spectral decomposition of the vocabulary embedding and unembedding matrices of the model.

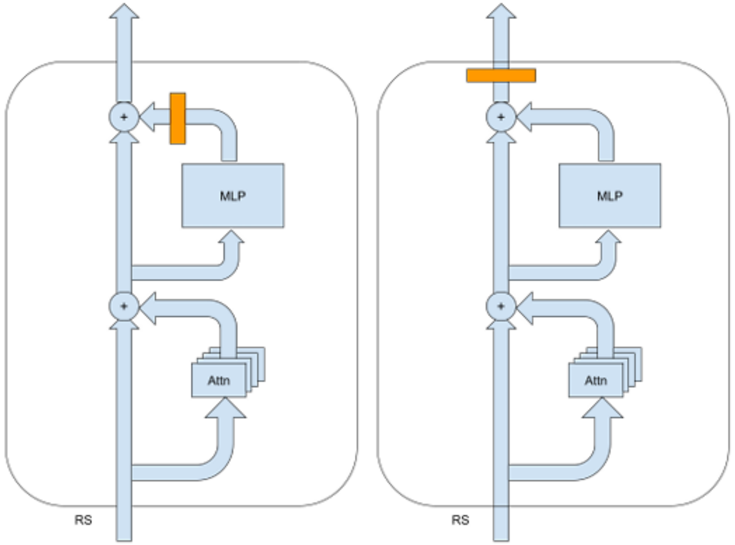


Figure 6 The two positions where we applied spectral filters: on the output of MLP layers, one at a time (left) or on the residual stream after a complete layer (right).

Spectral filtering of Llama2 13B MLP outputs

- Spectral filtering of MLP in layers 0 and 3 “dwarf all others”
  - We have Φ_u and Φ_e which remove smallest singular vectors
  - $\Psi = (I - \Phi_e \Phi_u)$, intersect of Φ_u and Φ_e
- Filters incremented 5% at a time
- At layer 3, negative log likelihood (NLL) stays high for both Φ_u and Φ_e until the final increment which includes the dark basis
- Ψ always worse than random filter

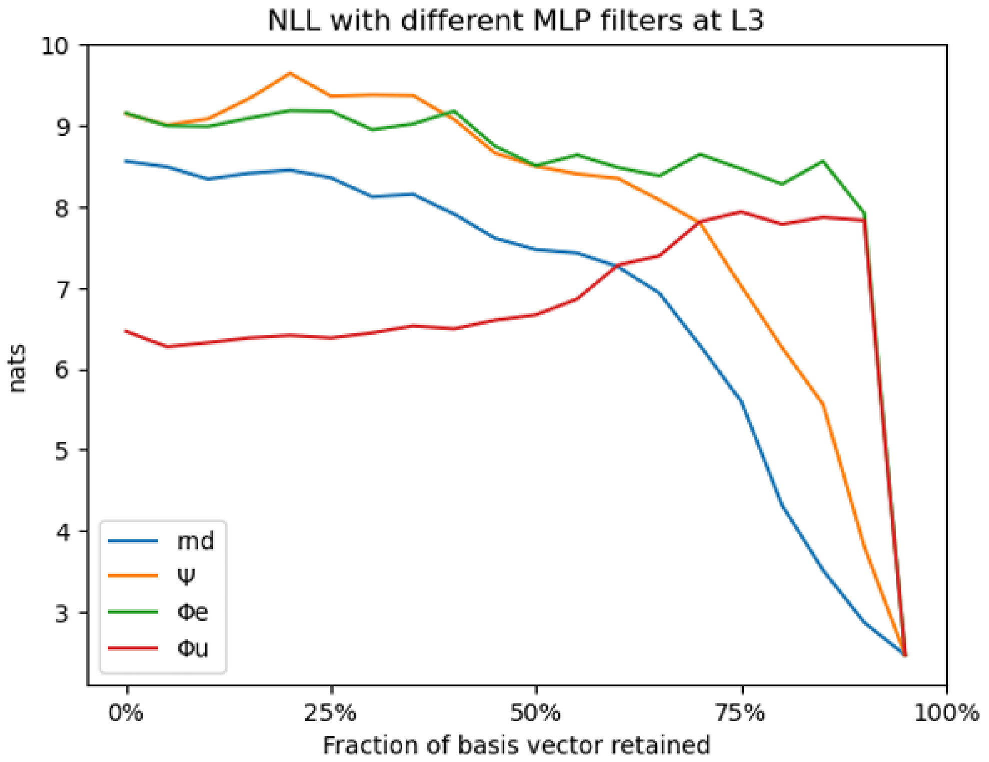


Figure 7 The effect of filtering 13B/L3/MLP with the filters defined in Section 4. ’Rnd’ indicates filtering projection on subsets of a random orthonormal basis, for reference.

Spectral filtering of Llama2 13B residual stream

- Spectral filtering of the residual stream causes much worse NLL until the dark basis is included
- Worse at early layers

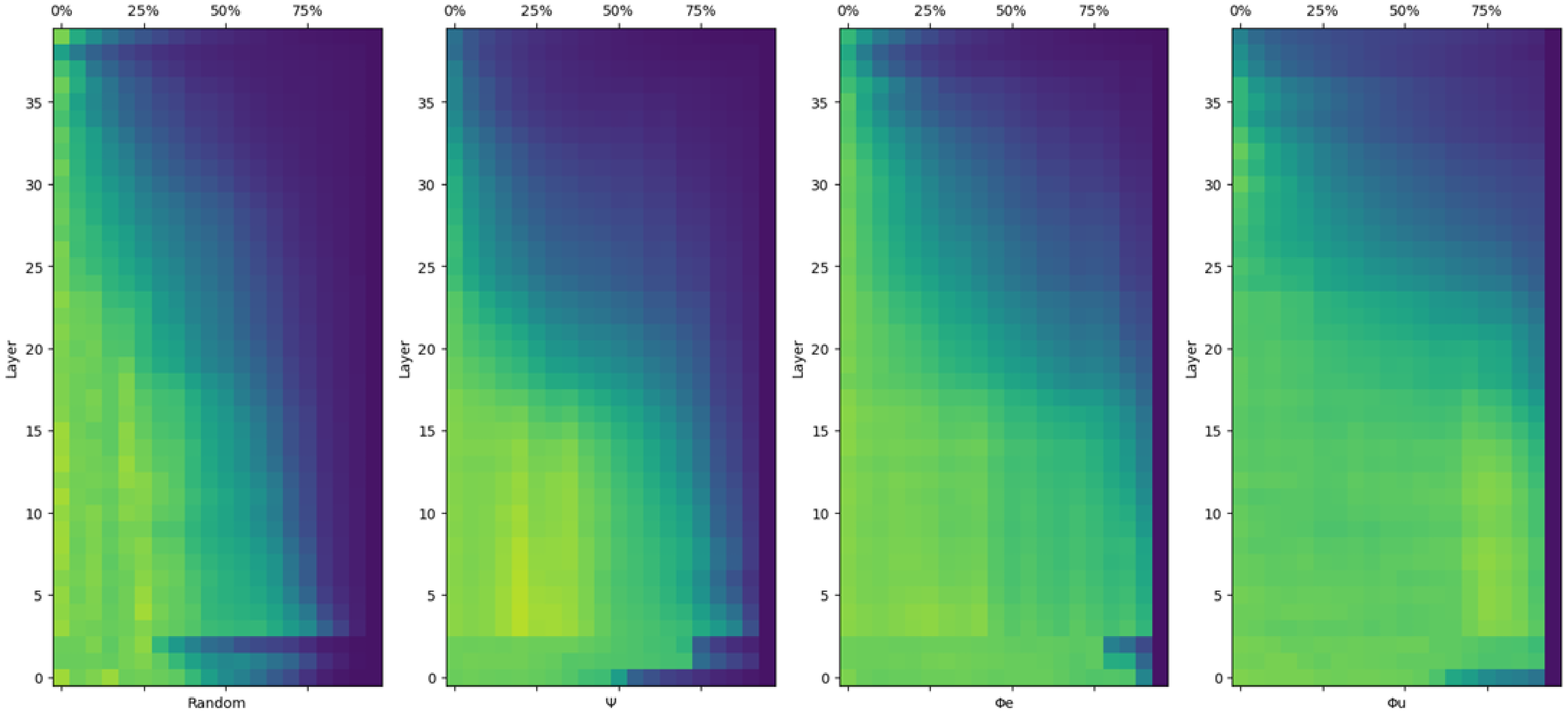


Figure 8 NLL of LLaMa2 13B on ccnet-405 when filtering the residual stream after a given layer (Y-axis), retaining an increasing number of SVs (X-axis) of (c) W_e or (d) W_u. (b) filters from the residual stream only its double projection onto both the W_e and W_u dark spaces. (a) shows, for comparison, the effect of adding more and more dimension in a random orthogonal base. See Fig.19 for similar heatmaps for LLaMa2 7B and 70B.

What happens to token 0

- Previous slides showed that dark signals are important, but didn't explain why
- Xiao et al. (2023) described *attention sinks*. We now examine what happens to the first token in Llama2.
- Plotting the norm of the residual stream for token 0 layer by layer shows a jump after layer 3, due to the MLP
    - Projecting on the dark basis shows it is the primary contribution to the norm
- Last few layers erase attention sink and prep next token prediction

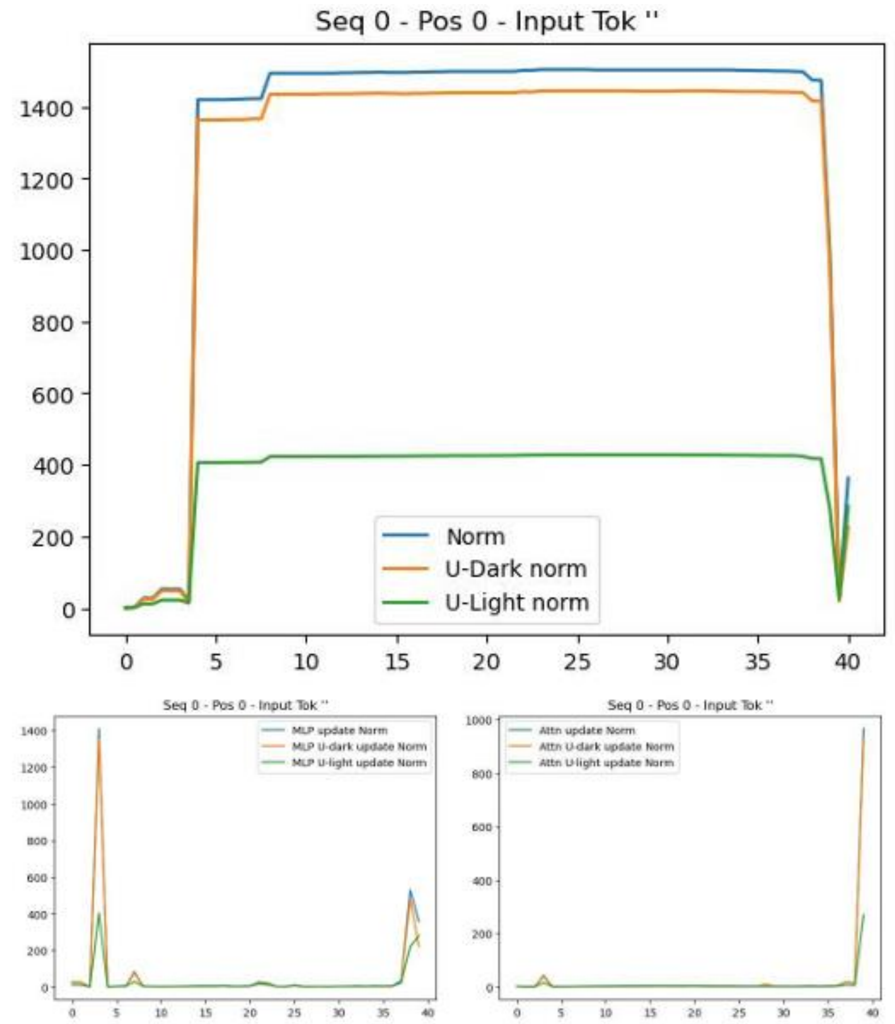


Figure 9 (Top) The composition of the RS of the BoS token for LLaMa2 13B as a function of the layer. (Bottom left) The norms of the contribution of MLP layers to the BoS RS. (Bottom right) The norms of the contribution of Multi-Head Attention components to the BoS RS.

Attention sink used by other tokens [1]

- Now that we know the attention sink is written by layer 3, we perform three causal experiments to impact attention sink usage
- First, we swap (not filter) Φ_u info between two different text inputs
  - This corrupts information for all tokens except token 0, because `BoS` is the same for every input
- Second, we do Φ_u spectral filtering, but only on token position 0
  - This preserves all information except the ability for token 0 to write to the dark basis
- Third, we modify Φ_u so it still filters smaller singular values, but now keeps the final 5% that form the dark basis (this is named Ω_u)

Attention sink used by other tokens [2]

- Left shows the first experiment (swap)
  - NLL is worse when more filtered, and worse when filtered in earlier layers, but no sharp drop for last 5%
- Middle show second experiment
  - Only filtering token 0 hurts NLL and has sharp drop in loss for keeping last 5%
- Right shows third experiment (Ω_u)
  - NLL not hurt much because dark basis is not filtered by Ω_u

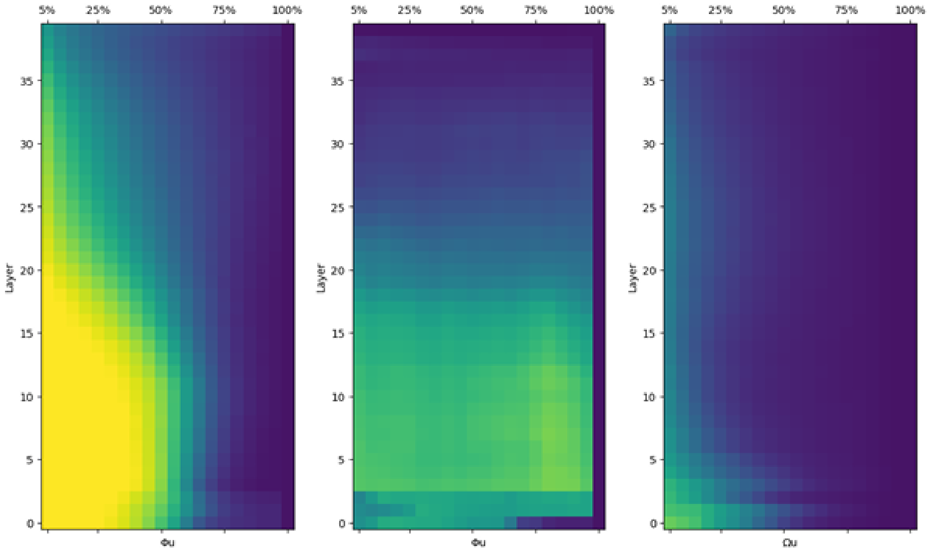


Figure 11 NLL of LLaMa2 13B on ccnet-405 when filtering the residual stream after a given layer (Y-axis). (Left:) Swapping the filtered vector components between RS at the same layer and token position, but in a different sample, perturbing all RSs except the BoS one. (Middle:) Filtering only the residual stream of the BoS token. (Right:) Applying the sink-preserving spectral filters $\Omega_{u,k}$ to a section of the residual stream of LLaMa2 13B right after a layer.

Attention sink used by other tokens [3]

- When we plot the NLL of all of the filters (including random), we see the huge improvement in loss that preserving the dark basis with Ω_u makes over all other filters
- Sample generations show that Ω_u filtering remains coherent up to 20% of singular vectors filtered, but other filters fall into repetitive patterns

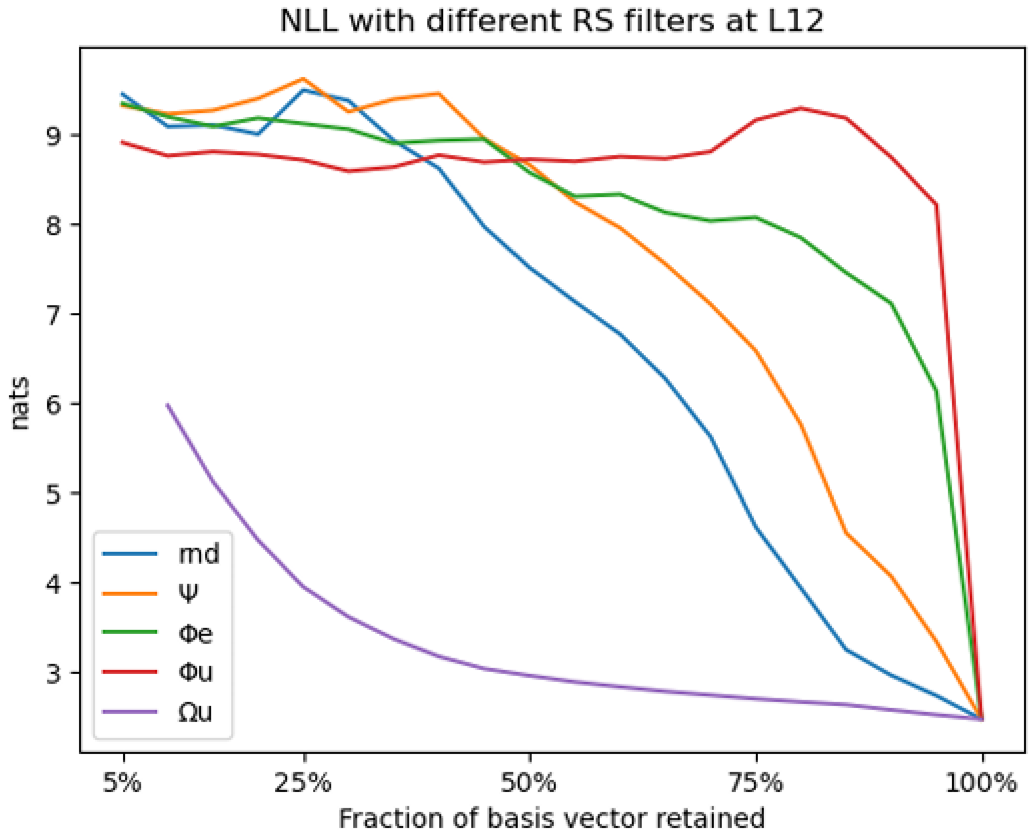


Figure 13 NLL by number of retained dimensions when applying different spectral filters at L12.

Attention bars

- An attempt was made to see if similarity to token 0 in the dark basis is the cause for attention bars at other token positions
- First, locate tokens with steady attention (High-Mean, Low-Variance)
- Calculate ratio of projection to dark basis versus projection to the orthogonal complement of the dark basis
- Then, calculate cosine similarity of tokens with the `BoS` residual stream
- While the dark basis ratio showed a strong pattern, increasing after layer 13 only for HMLV tokens, the cosine similarities remained similar for both the HMLV and other tokens

Spectral filters conclusion [1]

- Performed experiments with subspaces of the residual stream defined by the SVD of the unembedding and embedding matrices
- Found that the *dark signals* of the 5% smallest singular values are critical, and they implement attention sinks to token 0
  - LLM negative log likelihood increase dramatically if dark signals are filtered, even if only at token 0
  - But NLL stays low when filtering other bases but not dark signals
- In Llama2 13B, the layer 3 MLP writes the attention sink for token 0 to the residual stream using dark signals (which minimally affect next token prediction)
- Attention keys for layers 4 and up will cause the attention sink to match the query, if nothing else matches strongly

Spectral filters conclusion [2]

- Also found a positive correlation between the average attention received by a token and the relative prevalence of dark signals in its residual stream
  - In the paper, it was inconclusive if this is source of attention bars
  - Subsequently, the author has shown that for the HMLV tokens that represent attention bars, their keys show higher cosine similarity with the key for the `BoS` in token 0
  - Details of the process are unclear, but these attention bars do share similarity with attention sinks
- Note that this paper only studied language coherence, using NLL.
  It would be interesting to see in future work how downstream tasks are affected by spectral filtering.

References

- interpreting GPT: the logit lens
  nostalgebraist (2020)
  https://www.alignmentforum.org/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens