Attention-Only Transformers and Implementing MLPs with Attention Heads

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

The transformer architecture is widely used in machine learning models and consists of two alternating sublayers: attention heads and MLPs. We prove that an MLP neuron can be implemented by a masked attention head with internal dimension 1 so long as the MLP's activation function comes from a restricted class including SiLU and close approximations of ReLU and GeLU. This allows one to convert an MLP-and-attention transformer into an attention-only transformer at the cost of greatly increasing the number of attention heads.

1 Introduction

The transformer architecture was introduced in the landmark 2017 paper *Attention is All You Need* (Vaswani et al., 2023) and traditionally consists of alternating attention and multilayer-perceptron (MLP) sublayers. Although initially used for machine translation, transformers have been used across a wide range of tasks, including language modeling (Radford et al., 2018; Devlin et al., 2019; Liu et al., 2018), computer vision (Khan et al., 2022; Cornia et al., 2020), and image generation (Parmar et al., 2018).

This work seeks provide a new perspective on the role of MLP layers in transformers, by proving that they can be implemented by attention layers. In Theorem 2 we show that by including a "bias token" akin to the persistent memory vectors in Sukhbaatar et al. (2019) and using a slightly unusual attention-masking pattern, an MLP layer of size ℓ can be written as the sum of ℓ attention heads with internal dimension 1. We then show in Theorem 4 that one can apply this process throughout the entire transformer, converting the typical MLP-and-attention transformer into an attention-only transformer. Finally, the limitations of this method are discussed.

2 Background

Notation. *Throughout, we will use $M_{n,k}$ to denote the set of real-valued n-by-k matrices. We will write $\mathbf{0}$ and $\mathbf{1}$ for matrices where every entry is 0 or 1, respectively, of size specificied or implicit in the text.*

For matrices $X \in M_{n_1,k_1}$ and $Y \in M_{n_2,k_2}$ of any size, we will write $X \oplus Y$ for the block matrix in $M_{n_1+n_2,k_1+k_2}$ with X and Y as diagonal blocks and 0 elsewhere. For matrices $X \in M_{n,k_1}$ and $Y \in M_{n,k_2}$, we will write $[X|Y] \in M_{n,k_1+k_2}$ for the matrix made by appending one to the other.

We write $\mathrm{ReLU}, \mathrm{SiLU}$ *and* GeLU *for the usual activation functions as in Hendrycks & Gimpel (2023). In particular,* $\mathrm{SiLU}(x) = x\sigma(x)$*, where* $\sigma(x) = 1/(1 + \exp(-x))$.

We will say that a generalized SiLU function *is a function of the form* $f(x) = a_1\mathrm{SiLU}(a_2 x)$ *for some* $a_1, a_2 \in \mathbb{R}$.

The class of generalized SiLU functions includes $\mathrm{SiLU}(x)$ and approximations of GeLU and ReLU. In particular, $\mathrm{GeLU}(x) \approx \mathrm{SiLU}(1.702x)/1.702$ (Hendrycks & Gimpel, 2023) (reaching a maximum absolute error of 0.0203 at $x = \pm 2.27$) and $\mathrm{ReLU}(x) \approx \mathrm{SiLU}(kx)/k$ for large k (reaching a maximum absolute error of $\frac{0.2785}{k}$ at $x = \pm\frac{1.278}{k}$).

We will now present a slightly abstracted definition of MLPs, attention heads, and transformers, which the reader may confirm encompasses the classical transformer framework described in Vaswani et al. (2023).

Definition 1. *An* MLP with no biases and one hidden layer *is a function $f : M_{n,k} \to M_{n,k}$ of the form $f(X) = \alpha(XV_1)V_2$ where $\alpha : \mathbb{R} \to \mathbb{R}$ is some real-valued function applied entry-wise to matrices, and $V_1 \in M_{k,\ell}, V_2 \in M_{\ell,k}$ are fixed parameter matrices. We call ℓ the* size of the hidden layer*, and the function α is called the* activation function*.*

A mask matrix Λ *is a matrix with entries in $\{0,1\}$ such that every row has at least one nonzero entry.*

Let $X, \Lambda \in M_{n,k}$, and suppose Λ is a mask matrix. Then define the masked softmax *function*

$$\mathrm{msoftmax}(X, \Lambda) := \mathrm{rownorm}\left(\exp(X) \odot \Lambda\right)$$

where $\mathrm{rownorm}$ *denotes row-wise ℓ^1 normalization, and $\odot$ denotes element-wise multiplication. That is, the masked softmax function acts like the usual row-wise softmax but applied to only the entries of X where the mask Λ is 1. At the entries where Λ is 0, the output of the masked softmax function takes the value 0.*

A masked attention head *is a function $h : M_{n,k} \to M_{n,k}$ of the form*

$$h(X) = \mathrm{msoftmax}(XW_{QK}X^T, \Lambda)XW_{OV}$$

for some matrices $W_{OV}, W_{QK} \in M_{k,k}$, and mask matrix $\Lambda \in M_{n,n}$. We call W_{OV} and W_{QK} the parameter matrices *for this attention head.*

A transformer is a function $t : M_{N,D} \to M_{N,D}$ of the form $X_0 \mapsto X_1 \mapsto \ldots \mapsto X_m = t(X_0)$, where

$$X_{j+1} = \begin{cases} \mathrm{LayerNorm}(X_j + \sum_i h_{j,i}(X_j)) & \textit{or} \\ \mathrm{LayerNorm}(X_j + f_j(X_j)) \end{cases}$$

for some attention heads $h_{j,i}$ or MLPs with one hidden layer f_j. Note the use of Layer Normalization (Ba et al., 2016) and skip connections*, where one performs some computation f on X_j and defines $X_{j+1} = \mathrm{LayerNorm}(X_j + f(X_j))$, as opposed to $X_{j+1} = f(X_j)$.*

3 Implementing MLP Layers with Attention Heads

In this section we show that MLP layers whose activation functions are generalized SiLU functions are in fact a sum of attention heads.

Theorem 2. *Let $f(X) = \alpha(XV_1)V_2$ be an MLP on $M_{N,D}$ with no biases and one hidden layer of size ℓ, and suppose α is a generalized SiLU function $\alpha(x) = a_1\mathrm{SiLU}(a_2 x)$. Then there are ℓ masked attention heads $\{h_i\}_{i=1}^{\ell}$ on $M_{N+1,D+1}$ such that*

$$f(X) \oplus [0] = \sum_{i=1}^{\ell} h_i(X \oplus [1])$$

for all $X \in M_{N,D}$.

In particular, for the ith attention head, one uses parameter and mask matrices

$$
\begin{aligned}
W_{QK} &= a_2 \left[\begin{array}{c|c} \mathbf{0} & -V_1^i \\ \hline \mathbf{0} & 0 \end{array}\right] \\
W_{OV} &= a_1 a_2 V_1^i V_2^i \oplus [0] \\
\Lambda &= \left[\begin{array}{c|c} I_N & \mathbf{1} \\ \hline \mathbf{0} & 1 \end{array}\right]
\end{aligned}
$$

where the block decompositions are into size N and 1, V_1^i denotes the ith column of V_1, V_2^i denotes the ith row of V_2, and $\mathbf{1}$ denotes the column vector of all 1s.

We provide a sketch of the proof in the case of $\ell = a_1 = a_2 = 1$. For the full proof see Appendix A.

Proof Sketch. Since $\ell = 1$, we will write V_1 and V_2 are single-column matrices, so we will write V_1 and V_2 in place of V_1^i and V_2^i. Due to our choice of a particularly constrained masking pattern, our masked softmax function will only consider two tokens, the former of which has a pre-attention value from the main diagonal of $(X \oplus [1])W_{OV}(X \oplus [1])^T = -XV_1 \oplus [0]$ and the latter of which is 0. Writing $-x$ for the former entry, we have $\mathrm{softmax}([-x, 0]) = \mathrm{rownorm}([e^{-x}, 1]) = [\sigma(x), \sigma(-x)]$.

That is, by our choice of W_{QK} and Λ we have made our head have attention pattern $\left[\begin{array}{c|c} \mathrm{diag}(\sigma(XV_1)) & \sigma(-XV_1) \\ \hline \mathbf{0} & 1 \end{array}\right]$. Then, the complete output of this attention head h is

$$
\begin{aligned}
h(X \oplus [1]) &= \left[\begin{array}{c|c} \mathrm{diag}(\sigma(XV_1)) & \sigma(-XV_1) \\ \hline \mathbf{0} & 1 \end{array}\right] \left[\begin{array}{c|c} X & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array}\right] \left[\begin{array}{c|c} V_1 V_2 & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right] \\
&= \left[\begin{array}{c|c} \mathrm{diag}(\sigma(XV_1)) X V_1 V_2 & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right] \\
&= \mathrm{SiLU}(XV_1)V_2 \oplus [0]
\end{aligned}
$$

as desired.

□

Remark 3. *The additional term $\oplus[1]$ in Theorem 2 is similar to the* persistent vectors *of Sukhbaatar et al. (2019). In that work, the authors propose a new architecture, which they call the all-attention architecture, in which attention can also be paid to certain static vectors, learned for each attention head, called the persistent vectors. Our approach could also be implemented in that architecture with a single persistent vector $(0, 0, 0, .., 0, 1)$ shared across all attention heads in place of the $\oplus[1]$ terms.*

Note also that the W_{QK} and W_{OV} matrices used in Theorem 2 can be factored into the matrices W_Q, W_K, W_V, $W_O \in M_{D+1,1}$ from Vaswani et al. (2023) satisfying $W_{QK} = W_Q W_K^T / \sqrt{D+1}$ and $W_{OV} = W_V W_O$. In particular, we can take $W_Q = W_V = a_2[V_1^i|0]^T$, $W_K = \sqrt{D+1}[\mathbf{0}| - 1]^T$, and $W_O = a_1[V_2^i|0]^T$. Since W_K is shared across all attention heads, we only need to store two sets of parameters, the vectors $W_Q = W_V$ and W_O.

This provides an alternative perspective on MLP neurons: a neuron in an MLP is an attention head with internal dimension 1 and a particularly restrictive masking pattern in which each token attends only to itself and a static "bias" token.

We now turn to a have the necessary tools to show that a decoder-only transformer as in Liu et al. (2018); Radford et al. (2018) can be implemented entirely with attention heads.

Theorem 4. *If a transformer's MLP layers are activated by a generalized SiLU function, they can be substituted with attention heads.*

We again provide just a sketch of the proof and direct the reader to Appendix A for the full proof.

Proof Sketch. We will create a new transformer on $M_{N+1,D+1}$ whose residual stream X'_j on every sublayer satisfies $X'_j = X_j \oplus [1]$. This is sufficient to prove the main claim since the output of this new transformer will be $X'_{2m} = X_{2m} \oplus [1]$ and therefore contain the output of the original transformer.

For any MLP layer in the original transformer, we use Theorem 2 to replace the MLP layer with attention heads. For any attention head layer, we can slightly augment the W_{QK}, W_{OV}, and Λ matrices to work on the larger size. Due to skip connections, the resulting matrix retains the $\oplus[1]$ term, as desired. □

Remark 5. *It is instructive to compare this construction to the negative results of Dong et al. (2021), which find that without skip connections or MLPs, a self-attention network converges rapidly to a rank-1 matrix. Since we obviously do away with the MLP layer, our result depends on the use of skip connections. In particular, the "bias term" of* $\oplus[1]$ *is zeroed out by the construction in Theorem 2, so applying the construction in Theorem 4 without a skip connection results in* $X'_0 = X_0 \oplus [1]$*, but* $X'_1 = X_1 \oplus [0]$*. Then, in the* $j = 2$ *sublayer, the construction in Theorem 2 would fail for lack of this bias term, as, without it, the pre-attention matrix* $(X')W_{QK}(X')^T$ *is 0.*

We additionally show in Appendix B that attention heads can separately implement the components of an MLP layer, namely activation functions and linear transformations.

4 Limitations

The technique described in Theorem 4 faces several practical limitations. First is the quantity of attention heads: we use one attention head per dimension of the hidden layer, which can easily increase the number of attention heads by several orders of magnitude, partially offset by the new attention heads having smaller internal dimension. For example, in GPT-3 (Brown et al., 2020) the MLP layer has hidden dimension 49152, so this method would require 49152 additional 1-dimensional attention heads in each layer. This is an increase from from GPT-3's normal set of 96 attention heads per layer, each with internal dimension 128.

Second, it may be the case that replacing a feedforward network with attention heads slows down model inference or training. In particular, this approach replaces matrix multiplication with many vector-by-vector multiplications. One also computes many terms that are "thrown away" in the masking step. Combined, these suggest that converting an MLP layer to attention heads might increase computational costs.

5 Discussion

We have proven that attention heads can implement an MLP layer and that any transformer can be converted to an attention-only transformer. This approach provides a useful new perspective on the relative importance of MLP layers and attention heads in language models. MLP layers in a model like GPT-3 are larger than attention layers by a 2:1 margin if one measures by number of parameters but by 500:1 if one measures by number of attention heads.

One implication of these results is that it is theoretically possible to train an attention-only transformer that matches the performance of an MLP-plus-attention transformer. It remains unknown whether such an architecture would be competitive with the more classical transformer architecture in terms of practical considerations like training or inference speed. Such a test would be a promising future area of research.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pp. 2793–2803. PMLR, 2021.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Comput. Surv.*, 54(10s), sep 2022. ISSN 0360-0300. doi: 10.1145/3505244. URL `https://doi.org/10.1145/3505244`.

Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences, 2018.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4055–4064. PMLR, 10–15 Jul 2018. URL `https://proceedings.mlr.press/v80/parmar18a.html`.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.

Sainbayar Sukhbaatar, Edouard Grave, Guillaume Lample, Herve Jegou, and Armand Joulin. Augmenting self-attention with persistent memory, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

A Proofs of Main Results

In this section we present fully detailed proofs of our main results.

Proof of Theorem 2. We first prove the claim in the case of $\ell = a_1 = a_2 = 1$. In this case, since there is only one column in V_1, then $V_1 = V_1^i$, and similarly $V_2 = V_2^i$. Consider the attention matrix $\mathrm{msoftmax}((X \oplus [1])W_{QK}(X \oplus [1])^T, \Lambda)$. Multiplying matrices on the level of their blocks, we get that the first argument of the masked softmax is

$$(X \oplus [1])W_{QK}(X \oplus [1])^T = \left[\begin{array}{c|c} X & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array}\right] \left[\begin{array}{c|c} \mathbf{0} & -V_1^i \\ \hline \mathbf{0} & 0 \end{array}\right] \left[\begin{array}{c|c} X & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array}\right]^T = \left[\begin{array}{c|c} \mathbf{0} & -XV_1 \\ \hline \mathbf{0} & 0 \end{array}\right]$$

Now consider the masked softmax term in the jth row for $j \leq N$. This row has exactly two unmasked values, the diagonal entry and the rightmost entry, taking the values 0 and $-(XV_1)_j$, respectively. Applying $\exp$ and $\mathrm{rownorm}$ results in $\sigma((XV_1)_j)$ and $\sigma(-(XV_1)_j)$, respectively. Thus, the masked softmax term becomes

$$\begin{aligned} \mathrm{msoftmax}((X \oplus [1])W_{QK}(X \oplus [1])^T, \Lambda) &= \mathrm{msoftmax}(\left[\begin{array}{c|c} \mathbf{0} & -XV_1 \\ \hline \mathbf{0} & 0 \end{array}\right], \left[\begin{array}{c|c} I_{n-1} & \mathbf{1} \\ \hline \mathbf{0} & 1 \end{array}\right]) \\ &= \left[\begin{array}{c|c} \mathrm{diag}(\sigma(XV_1)) & \sigma(-XV_1) \\ \hline \mathbf{0} & 1 \end{array}\right] \end{aligned}$$

Substituting these values into the expression for $h(X)$ gives

$$
\begin{aligned}
h(X \oplus [1]) &= \mathrm{msoftmax}((X \oplus [1]) W_{QK} (X \oplus [1])^T, \Lambda)(X \oplus [1]) W_{OV} \\
&= \left[\begin{array}{c|c} \mathrm{diag}(\sigma(XV_1)) & \sigma(-XV_1) \\ \hline \mathbf{0} & 1 \end{array}\right] (X \oplus [1]) W_{OV} \\
&= \left[\begin{array}{c|c} \mathrm{diag}(\sigma(XV_1)) & \sigma(-XV_1) \\ \hline \mathbf{0} & 1 \end{array}\right] \left[\begin{array}{c|c} X & \mathbf{0} \\ \hline \mathbf{0} & 1 \end{array}\right] \left[\begin{array}{c|c} V_1 V_2 & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right] \\
&= \left[\begin{array}{c|c} \mathrm{diag}(\sigma(XV_1)) X V_1 V_2 & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right] \\
&= \left[\begin{array}{c|c} \mathrm{SiLU}(XV_1) V_2 & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right] \\
&= \left[\begin{array}{c|c} f(X) & \mathbf{0} \\ \hline \mathbf{0} & 0 \end{array}\right] \\
&= f(X) \oplus [0]
\end{aligned}
$$

as desired. This completes the $\ell = a_1 = a_2 = 1$ case.

For a general a_1, a_2, apply the previous case to an MLP with weight matrices $a_2 V_1$ and $a_1 V_2$.

Finally, for the fully general case with $\ell > 1$, for each $1 \leq i \leq \ell$, let $f_i(X) = \alpha(X V_1^i) V_2^i$, and note that $f = \sum_{i=1}^{\ell} f_i$. Let h_i denote the attention head corresponding to f_i given by the $\ell = 1$ case. Then we have that

$$
\begin{aligned}
f(X) \oplus [0] &= \sum_{i=1}^{\ell} f_i(X) \oplus [0] \\
&= \sum_{i=1}^{\ell} h_i(X \oplus [1])
\end{aligned}
$$

as desired. □

Proof of Theorem 4. We will show that we can create a new transformer t' on $M_{N+1,D+1}$ whose residual stream X'_j on every sublayer satisfies

$$
X'_j = X_j \oplus [1]
$$

This is sufficient to prove the main claim since the output of this new transformer will be $X'_{2m} = X_{2m} \oplus [1]$ and therefore contain the output of the original transformer.

Without loss of generality, assume that the MLP layers have no bias terms (i.e., that we've already used the "bias trick" to fold bias terms into the weight matrix).

To prove that there is a transformer t' that satisfies $X'_j = X_j \oplus [1]$ on every sublayer, we proceed by induction. For the base case of $j = 0$, we tweak the transformer's context window and embedding weights so that $X'_0 = X_0 \oplus [1]$.

We split the inductive case depending on whether the original transformer's sublayer used attention or an MLP. If the original layer was an MLP, then by Theorem 2 there are attention heads $h'_{j,i}$ such that $f_j(X) \oplus [0] = \sum h'_{j,i}(X \oplus [1])$, so in our transformer t', using these attention heads yields

$$
\begin{aligned}
X'_{j+1} &= \mathrm{LayerNorm}(X'_j + \sum h'_{j,i}(X'_j)) \\
&= \mathrm{LayerNorm}((X_j \oplus [1]) + \sum h'_{j,i}(X_j \oplus [1])) \\
&= \mathrm{LayerNorm}((X_j \oplus [1]) + (f_j(X) \oplus [0]))) \\
&= \mathrm{LayerNorm}(X_j + f_j(X)) \oplus [1] \\
&= X_{j+1} \oplus [1]
\end{aligned}
$$

as desired.

If instead, the transformer used attention heads on the jth sublayer, we must tweak our original attention heads to account for the new size. To this end, we will show that for each of the original attention heads $h = h_{j,i}$, we can create an attention head h' such that

$$h'(X \oplus [1]) = h(X) \oplus [0]$$

Let W_{QK}, W_{OV}, and Λ denote the original parameter and masking matrices for h. Then define

$$
\begin{aligned}
W'_{QK} &= W_{QK} \oplus [1] \\
W'_{OV} &= W_{OV} \oplus [0] \\
\Lambda' &= \Lambda \oplus [1]
\end{aligned}
$$

Then,

$$
\begin{aligned}
h'(X \oplus [1]) &= \mathrm{msoftmax}((X \oplus [1]) W'_{QK} (X \oplus [1])^T, \Lambda')(X \oplus [1]) W'_{OV} \\
&= \mathrm{msoftmax}((X \oplus [1])(W_{QK} \oplus [1])(X \oplus [1])^T, (\Lambda \oplus [1]))(X \oplus [1])(W_{OV} \oplus [0]) \\
&= \mathrm{msoftmax}(X W_{QK} X^T \oplus [1], \Lambda \oplus [1])(X W_{OV} \oplus [0]) \\
&= (\mathrm{msoftmax}(X W_{QK} X^T, \Lambda) \oplus [1])(X W_{OV} \oplus [0]) \\
&= \mathrm{msoftmax}(X W_{QK} X^T, \Lambda) X W_{OV} \oplus [0] \\
&= h(X) \oplus [0]
\end{aligned}
$$

as desired. Now, creating such $h'_{j,i}$ for each of the original attention heads $h_{j,i}$, we have

$$
\begin{aligned}
X'_{j+1} &= \mathrm{LayerNorm}(X'_j + \sum h'_{j,i}(X'_j)) \\
&= \mathrm{LayerNorm}((X_j \oplus [1]) + \sum h'_{j,i}(X_j \oplus [1])) \\
&= \mathrm{LayerNorm}((X_j \oplus [1]) + \sum h_{j,i}(X) \oplus [0])) \\
&= \mathrm{LayerNorm}((X_j + \sum h_{j,i}(X))) \oplus [1] \\
&= X_{j+1} \oplus [1]
\end{aligned}
$$

as desired. This completes the inductive step and the proof.

□

B Linear Transformations and Activation Functions with Attention Heads

Theorem 2 shows that attention heads can implement an MLP layer, but can they separately implement the components of an MLP, a linear transformation and an activation function? In this section we show that the answer is yes.

We first show that an attention head can perform an arbitrary linear operation row-wise on the matrix.

Theorem 6. *Let* $h : M_{N,D} \to M_{N,D}$ *be an attention head with masking matrix* $\Lambda = I_N$*. Then* $h(X) = XW_{OV}$*.*

Proof. Because $\Lambda = I_n$, after masking, the attention matrix $\mathrm{msoftmax}(XW_{QK}X^T, \Lambda)$ will have nonzero entries only along the diagonal. Since the rows of the attention matrix are normalized to sum to 1, it follows that $\mathrm{msoftmax}(XW_{QK}X^T, \Lambda) = I_n$. Then,

$$h(X) = \mathrm{msoftmax}(XW_{QK}X^T, \Lambda)XW_{OV} = I_n XW_{OV} = XW_{OV}$$

as desired. $\square$

Now we will show that one can apply a generalized SiLU function entrywise.

Theorem 7. *Let* α *be a generalized SiLU function. Then there are* D *attention heads* $h_1, ..., h_D$ *on* $M_{N+1,D+1}$ *such that*

$$\alpha(X) \oplus [0] = \sum_{i=1}^{D} h_i(X \oplus [1])$$

Proof. This follows immediately from applying Theorem 2 to the MLP $f(X) = \alpha(XI_N)I_N = \alpha(X)$, whose hidden layer is of size $\ell = D$. $\square$

Note that a transformer usually makes use of skip connections, so that the residual stream experiences the transformation $X \mapsto X + sublayer(X)$. Thus, to get the transformation $X \mapsto \alpha(X)$, one can combine these two theorems, using $D + 1$ attention heads to produce $sublayer(X) = \alpha(X) - X$, in which case $X \mapsto X + sublayer(X) = \alpha(X)$.