Telling BERT's Full Story: from Local Attention to Global Aggregation

Damián Pascual, Gino Brunner, Roger Wattenhofer
Department of Electrical Engineering and Information Technology
ETH Zurich, Switzerland
{dpascual,brunnegi,wattenhofer}@ethz.ch

Abstract

We take a deep look into the behaviour of self-attention heads in the transformer architecture. In light of recent work discouraging the use of attention distributions for explaining a model's behaviour, we show that attention distributions can nevertheless provide insights into the local behaviour of attention heads. This way, we propose a distinction between local patterns revealed by attention and global patterns that refer back to the input, and analyze BERT from both angles. We use gradient attribution to analyze how the output of an attention head depends on the input tokens, effectively extending the local attention-based analysis to account for the mixing of information throughout the transformer layers. We find that there is a significant mismatch between attention and attribution distributions, caused by the mixing of context inside the model. We quantify this discrepancy and observe that interestingly, there are some patterns that persist across all layers despite the mixing.

1 Introduction

The inception of the transformer architecture has sparked significant progress across a wide range of language understanding tasks. Variants of transformers currently dominate the popular GLUE (Wang et al., 2019b) and Super-GLUE (Wang et al., 2019a) benchmarks and have even achieved super human performance on multiple tasks. The main innovations behind the transformer architecture are the stacking of self-attention layers into a multi-layer self-attention architecture, as well as an unsupervised pre-training phase that primes the model to be fine-tuned on a wide range of language tasks. Transformers and other self-attention-based models have been successfully adopted in other areas such as computer vision (Parmar et al., 2018), music processing (Huang et al., 2019) or protein research (Rao et al., 2019). Their extraordinary empirical success has led researchers to investigate transformers in order to better understand the source of this success, but also in an attempt to explain model decisions.

Much of the research around interpretability and explainability is focused on analyzing the self-attention operation (Clark et al., 2019). In multi-layer self-attention, every input computes an attention distribution over itself and all other inputs to produce ever more complex feature representations. In the case of language, a word in a sentence attends to itself and to all other words in order to compute an updated contextual representation of itself. It is tempting to directly rely on attention distributions to explain the model's predictions. The rationale is that if the attention distribution aligns with human intuition, we can conclude that the model learned robust features and obtained a deep understanding of language, in contrast to simply overfitting on spurious patterns. For example, if a transformer classifies an online comment as hate speech, but we find that the model mostly attended to neutral or even positive words, we would conclude that the model did not actually understand the text and that the correct prediction was either due to chance or to the exploitation of an underlying statistical bias in the data (Niven and Kao, 2019).

However, recent studies (Brunner et al., 2020; Pruthi et al., 2019) question the ability of attention maps to provide a faithful explanation of the inner workings of transformer models. In particular, when the explanations refer to the model input, attention maps do not account for the mixing of information throughout the model. Since self-attention mixes information among all input tokens, the hidden layers attend over mixtures of tokens. Therefore, attention maps may be useful to investigate the local behavior of attention heads but not to draw conclusions about how input tokens relate to each other.

In this work we take a detailed look at the inner workings of BERT's attention heads, both by analyzing the self-attention distributions, and by using gradient attribution to account for the mixing of tokens throughout the model. We first show that self-attention distributions correlate strongly with Hidden Token Attribution (Brunner et al., 2020) (HTA) from hidden embedding to head output, this result validates HTA. We then present novel location based attention patterns, revealing that BERT, despite its bi-directional language modeling objective, attends to past embeddings in earlier layers, and to future ones in later layers. Next, we use HTA in order to extend the analysis to take the mixing of information into account, which allows to draw conclusions about the behaviour of an attention head with respect to the original input word. The patterns that emerge are different from the local attention-based patterns, giving us deeper insight into the operation of the model and emphasizing that local attention-based explanations are very different from global attribution-based explanations. Finally, we contrast attention and HTA distributions for individual examples. Our results further highlight the discrepancy between local attention patterns and global attribution patterns.

2   Related Work

The good performance of attention (Graves, 2013; Bahdanau et al., 2015) models in Natural Language Processing (NLP) arises from their ability to learn alignments between words. The transformer architecture (Vaswani et al., 2017) is a multi-layer multi-head self-attention architecture that is pre-trained in an unsupervised manner. The extraordinary performance of transformer models has accelerated progress in the field of NLP. Currently, there is a growing number of different transformer models that vary in size, pre-training objective and/or other architectural elements (Radford et al., 2018, 2019; Liu et al., 2019; Lan et al., 2020; Yang et al., 2019; Sanh et al., 2019; Kitaev et al., 2020; Raffel et al., 2019).

The success of transformers and the possibility of visualizing attention distributions (Vaswani et al., 2017), has motivated a line of research aiming to understand the inner workings of transformers and explain their decisions. Many of these studies have focused on BERT (Devlin et al., 2019a), a well-known transformer model, leading to a body of research grouped under the term BERTology (Rogers

et al., 2020).

Aforementioned research builds on previous work on the interpretability of attention distributions in other models apart from transformers. In particular, Jain and Wallace (2019) examine the attention distributions of LSTM based encoder-decoder models and show a weak to moderate correlation between attention and dot-product gradient attribution. Furthermore, they show that adversarial attention distributions that do not change the model's decision can be constructed. In the same line, Serrano and Smith (2019) find, through zeroing out attention weights, that gradient attribution is a better predictor of feature importance with respect to the model's output than attention weights. Wiegreffe and Pinter (2019) find that although adversarial attention distributions can be easily obtained, they perform worse on a simple diagnostic task. All of these works raise concerns about the ability of attention distributions to explain the decisions of a model.

Despite existing concerns surrounding the interpretability of attention distributions, very few works have studied how this problem affects transformers. Pruthi et al. (2019) show that, just as in other attention models, it is possible to manipulate self-attention in transformers in order to generate different attention masks that cause only a small drop in performance. Brunner et al. (2020) find that attention distributions are not unique when the sequence length is larger than the head dimension and show that this can lead to the discovery of spurious patterns. Furthermore, they show that although it is possible to map hidden tokens back to their corresponding input tokens, there is a very large degree of information mixing inside the model, which raises questions about straightforward interpretations of attention maps. Recently, (Abnar and Zuidema, 2020) proposed a method to quantify information flow inside transformers. This method tracks the mixing of information due to attention but omits the effect of feed-forward networks.

Our work addresses this important issue by distinguishing between local and global aggregation patterns, where the former can be explained by attention distributions and the latter by attribution. We analyze BERT from both angles and quantify the mismatch between these interpretations. We show that attention correlates well with attribution locally but not globally and therefore attention maps are inadequate to draw conclusions that refer

to the input of the model.

3 Background on Transformers

The original transformer architecture (Vaswani et al., 2017) is a sequence-to-sequence model consisting of an encoder and a decoder, both of which follow a multi-layer multi-head self-attention structure. Conversely, most of the pre-trained transformer models that can be fine-tuned on supervised language understanding tasks only consist of an encoder. Each transformer layer consists of a self-attention block and a non-linear feed forward block (MLP) with layer normalizations (Ba et al., 2016).

The input to a transformer layer is a sequence of embeddings $E^l = [e_0^l, ..., e_{d_s}^l] \in \mathbb{R}^{d_e \times d_s}$, where l denotes the layer index, d_e is the embedding dimension, and d_s is the sequence length. We refer to the sequence of non-contextual input word embeddings as E^0, and to the hidden contextual embeddings as E^l, where $l > 0$. Note that E^0 refers to the word embeddings after position and sequence embeddings have been added. A self-attention block consists of n_h separate attention heads. The attention heads independently perform the self-attention operation, and the results are then concatenated and projected back into the embedding space by a linear layer. The output of the attention block is then fed into the MLP.

The self-attention operation itself is implemented by projecting each input token $e_i \in \mathbb{R}^{d_e}$ into a query vector $q_i \in \mathbb{R}^{d_q}$, key vector $k_i \in \mathbb{R}^{d_q}$ and value vector $v_i \in \mathbb{R}^{d_v}$. We present the self-attention operation from the perspective of a single token e_i attending to all input tokens. For that, the key vectors k_i are aggregated into the key matrix $K = [k_0, ..., k_{d_s}] \in \mathbb{R}^{d_q \times d_s}$ and the value vectors v_i are aggregated into the value matrix $V = [v_0, ..., v_{d_s}] \in \mathbb{R}^{d_v \times d_s}$. The attention distribution a_i of token e_i over all input tokens is then computed as

$$a_i = \text{softmax}\left(\frac{q_i^T \cdot K}{\sqrt{d_q}}\right)$$

The attention vector $a_i \in \mathbb{R}^{d_s}$ now contains an attention weight for each input token. a_i is then multiplied with the value matrix V to compute the output of the self-attention operation for a token i and a head h as

$$o_{h,i} = V \cdot a_i$$

The outputs of all heads $\{o_{0,i}, ..., o_{n_h,i}\} \in \mathbb{R}^{d_e}$ are then concatenated and fed through a linear layer to compute the output of the self-attention block for a single token. This linear layer can be thought of as an aggregation operation that projects the output of the independent heads back into embedding space. In practice, the attention distributions for all tokens are computed in parallel.

4 Extending Hidden Token Attribution

Hidden Token Attribution (Brunner et al., 2020) is a gradient-based attribution method that quantifies how much information from each input token is contained in a given hidden embedding. For each layer l, this method defines the relative contribution $c_{i,j}^l$ of an input token e_i^0 to a hidden embedding e_j^l as:

$$c_{i,j}^l = \frac{||\nabla_{i,j}^l||_2}{\sum_{k=0}^{d_s} ||\nabla_{k,j}^l||_2} \quad \text{with} \quad \nabla_{i,j}^l = \frac{\partial e_j^l}{\partial e_i^0} \quad (1)$$

The contribution $c_{i,j}^l$ is normalized by the sum of the attribution values to all input tokens and hence, ranges between 0 and 1.

In this work, we apply Hidden Token Attribution to the individual attention heads of BERT. For a token e_j^l at layer l we back-propagate the gradients from the output $o_{h,j}^l$ of each attention head h independently. This differs from the original method in that Hidden Token Attribution propagates the gradients from the layer output. In general, using Equation 1, we can compute the contribution between any two vectors in the model, as long as they are connected in the computation graph. We hence denote the contribution of any vector x to another vector y as $C(x, y)$.

In particular, we calculate two different contributions to the head output:

Previous layer contribution: Contribution from the hidden embeddings at the input of the attention head to the output of the attention head: $C(e_i^{l-1}, o_{h,j}^l)$

Input contribution: Contribution from tokens at the input of the transformer model to the output of an attention head h at layer l: $C(e_i^0, o_{h,j}^l)$

Previous layer contribution allows us to study how attention heads operate locally and how HTA

distributions compare to attention distributions. *Input contribution* enables us to extend the head attention patterns all the way back to the input, thereby controlling for the effect of information mixing.

5 Setup

For our experiments we use the non-finetuned, uncased BERT base model (Devlin et al., 2019b) as provided in the original repository.[1] Despite the recent explosion of new transformer variants, BERT remains the most popular model for research into the interpretability of transformer models. The reason for this is that most of the newer models are architecturally similar to BERT, and therefore, studies carried out on BERT either are likely to generalize to these models or can be repeated with relatively little effort.

We perform our experiments on 1800 examples from the development set of the MNLI matched (MNLIm) dataset. Brunner et al. (2020) show that when the sequence length d_s is larger than the head output dimension d_v, the attention distributions are not identifiable. Therefore, to guarantee that in our experiments we do not find spurious patterns that do not influence downstream parts of the model, we restrict the examples in our dataset to sequences of maximum length of 64 tokens, which is the head dimension of BERT. Thus, the examples in our dataset have sequence lengths ranging between 6 and 64 tokens, with a median length of 34 tokens. In total, this subset contains 63,456 tokens.

6 HTA: Local Validation

The ability of attention distributions to provide explanations has been the target of a number of studies (Wiegreffe and Pinter, 2019; Serrano and Smith, 2019; Pruthi et al., 2019). In particular, Jain and Wallace (2019) show that attention distributions do not explain the model output and do not correlate well with attribution methods. However, if we are exclusively interested in how attention heads behave locally, i.e., without considering their impact on the model's decisions, it is sound to examine attention distributions. The reason for this is that self-attention is the only operation performed in attention heads, and hence, attention distributions precisely represent the information flow within the heads. As a consequence, we can use attention distributions as a reference to validate whether HTA accurately quantifies how information mixes within
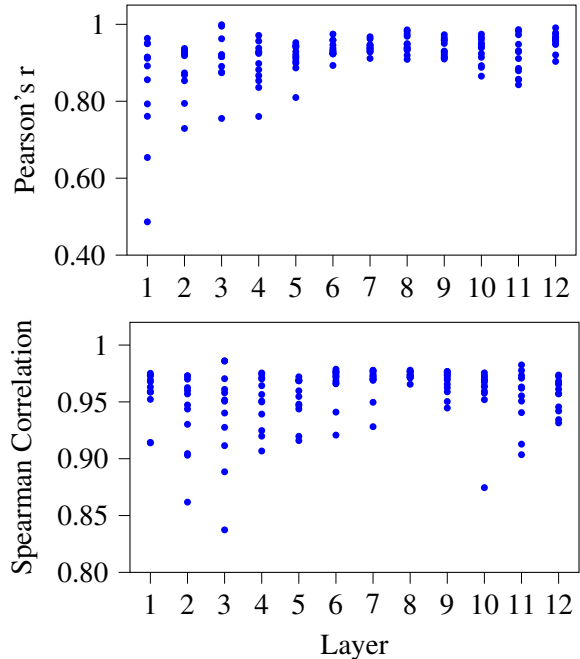
---

[1]https://github.com/google-research/bert

Figure 1: (Top) Pearson and (Bottom) Spearman correlation between attention and previous layer contribution.

transformers. To verify this, we compare attention distributions to previous layer contribution by computing the correlation between attention maps $a_{h,i}$ and the contribution $C(e_i^{l-1}, o_{h,j}^l)$ for each head.

A high correlation value would validate HTA as accurately representing the flow of information within transformers. To calculate the correlation, first, we extract the attention maps for all the heads of BERT for each of the tokens in the examples of our dataset. Then, we pair each attention map to the corresponding contribution. Note, that both attention maps and contributions are distributions that lie in the probability simplex, i.e., all the values are between 0 and 1 and their sum is 1. Next, we calculate Pearson's correlation coefficient for each attention-contribution pair and we aggregate the results into one value per head by computing the mean of the correlation values.

Figure 1 (Top) shows the mean correlation value per head. For all heads except for two, Pearson's correlation coefficient is larger than 0.7. Furthermore, 90% of the heads show a correlation between attention and Hidden Token Attribution of over 0.85. Similarly, we calculate Spearman's rank correlation coefficient r for each head. The results, displayed in Figure 1 (Bottom), show that only four heads have a Spearman's correlation smaller than 0.9, and that 75% of the heads have a correlation
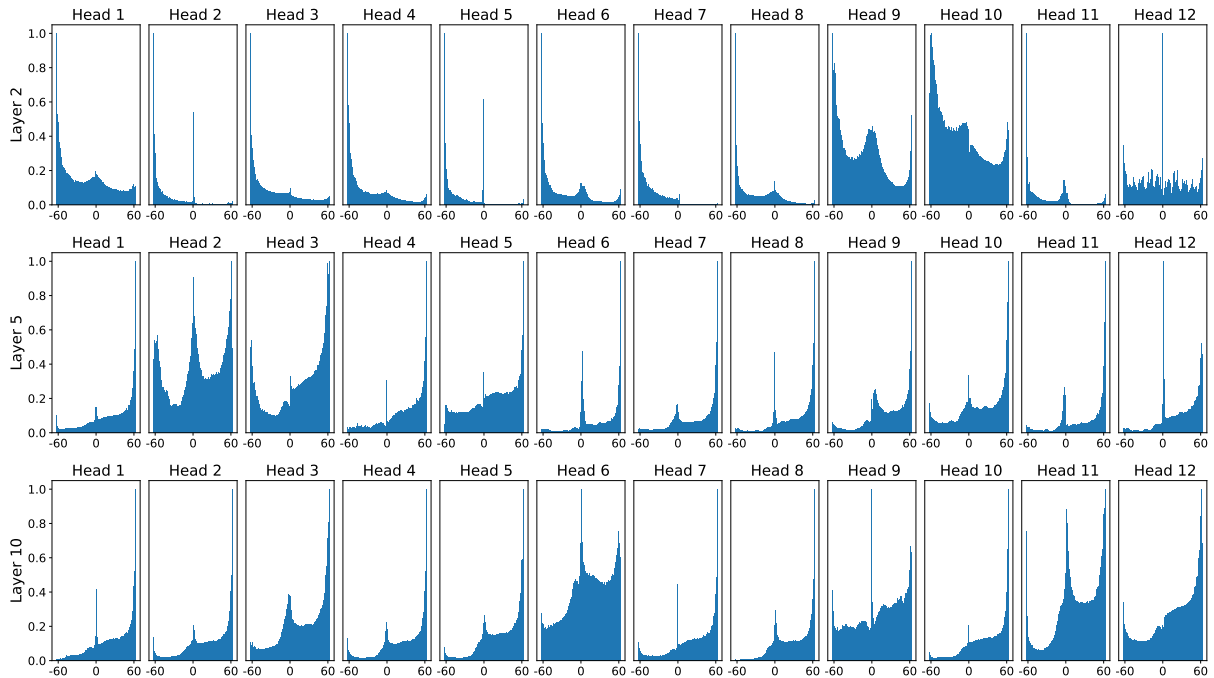
Figure 2: Attention histograms for layers 2, 5 and 10 of BERT. The horizontal axis represents the relative position of the attended tokens with respect to the attending token placed at position 0. Given that the maximum sequence length is 64 the horizontal axis ranges from -63 to 63. The vertical axis is normalized to the maximum value for a better visualization.

coefficient larger than 0.95. Note that in any case gradient attribution is a local first order approximation and thus introduces a small error that prevents perfect correlation.

These high correlation values empirically demonstrate that HTA does indeed represent the flow of information within attention heads with respect to the head inputs. Therefore, to study the inner workings of transformers beyond attention heads one can rely on Hidden Token Attribution and apply it at different points of the model. Now that we have validated HTA, we can investigate the behavior of the heads in more detail: examining the local patterns revealed by attention, the global patterns revealed by HTA, and the discrepancies between both.

7 Local Head Analysis

In this section we take a closer look into the local behaviour of attention heads. Here, local means that we analyze how the intermediate tokens fed into the heads are processed, as opposed to how the model input propagates. To this end, we study attention distributions, but rather than studying each individual example, we aggregate the attention distributions, thus obtaining a general picture of how each head behaves. In particular, we study how much attention is paid to tokens in each relative position with respect to the attending token.

For each head, we extract the attention maps for each token. Then, we define the position of the attending token in the sentence as the origin ($x = 0$), thereby generating a histogram where the horizontal axis represents the position of the neighbours and the vertical axis the amount of attention paid to a token. We sum the histograms of all tokens and then normalize the result. To normalize, we divide the value of attention at each position by the number of times that a token is at that relative position; given that the median length of the examples is 36, this normalization ensures that distant positions are not penalized for having fewer occurrences.

Figure 2 presents the histograms for the heads in layers 2, 5 and 10, the other layers can be found in Appendix A. From these histograms, a clear pattern is observable. In the first layers, heads tend to aggregate more information from past tokens than from future tokens. In fact, the attention of heads 2, 5 and 7 in Layer 2 to future tokens is negligible. However, this trend quickly reverses with increasing depth, and in later layers the aggregation of future hidden embeddings dominates for most heads. To illustrate this, we calculate the center of mass of the attention histograms per layer by
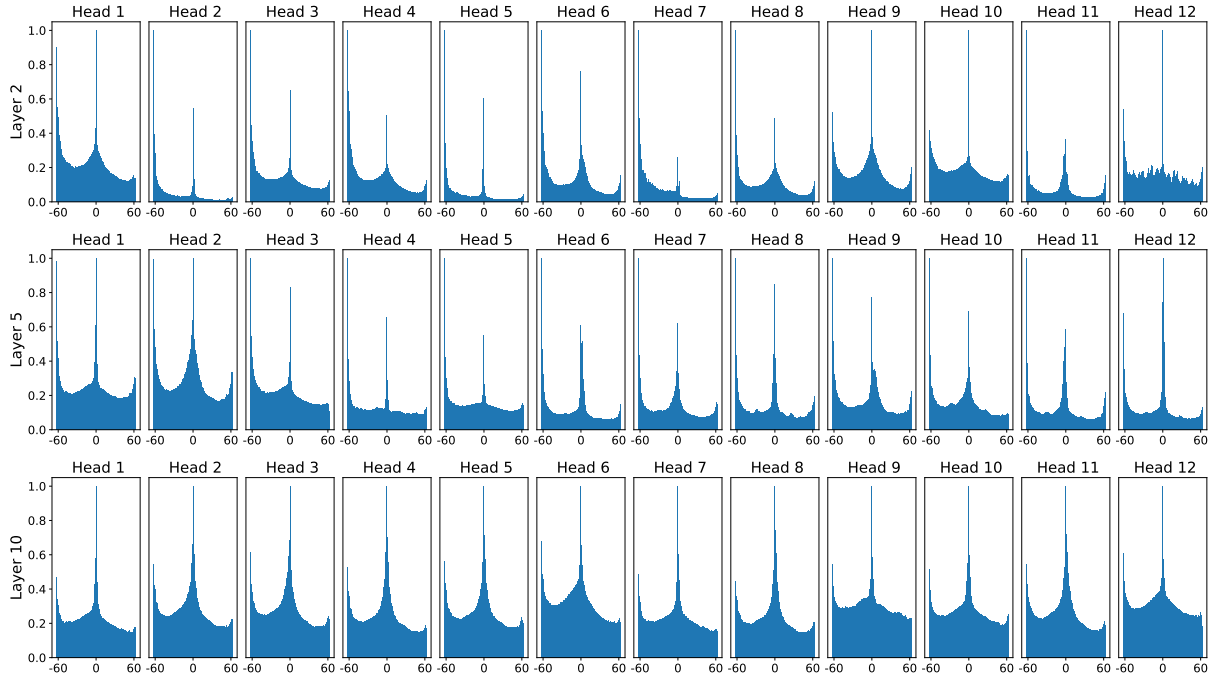
Figure 3: Input contribution histograms for layers 2, 5 and 10 of BERT. The horizontal axis represents the relative position of the attended tokens with respect to the attending token placed at position 0. The vertical axis is normalized to the maximum value for a better visualization.
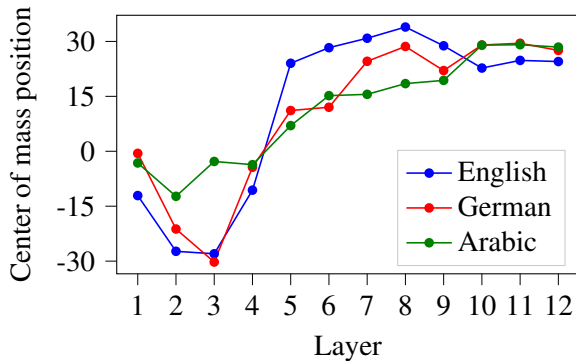

Figure 4: Mean center of mass for attention histograms per layer for English, German and Arabic.

averaging over each head. The results are shown in Figure 4. To understand whether this behavior is particular to English we perform the same calculations on the German (Deepset, 2019) and Arabic (Antoun et al., 2020) versions of BERT on 1800 examples of the XNLI dataset (Conneau et al., 2018); note that although Arabic is a right-to-left language, to process the examples the token order is reversed and input to BERT as left-to-right.

The figure shows that BERT follows the same trend regardless of the language, i.e., the models first attend to past and then to future hidden tokens. This suggests that despite its bidirectional training,

BERT tends to handle language like humans, from left to right. This is also inline with the sequential nature of language, i.e., the past context needs to be known to understand the future context.

8   Global Head Analysis

Although attention maps are an effective tool to understand the local behavior of attention heads, drawing conclusions that refer to the input words can be misleading. Transformers are complex models that mix information from the entire input sequence at each layer. Recent work (Brunner et al., 2020; Pruthi et al., 2019) has raised concerns about the interpretability of attention maps as representative of global context aggregation. In this section, we look into the individual heads and study what we call global patterns, i.e., aggregation patterns that refer to the model's input.

To this end, we follow the same procedure we use in the previous section, but to generate input contribution $C(e_i^0, o_{h,j}^l)$ histograms instead of attention histograms. In Figure 3 we show the histograms for layers 2, 5 and 10, i.e., the same layers as in Figure 2. The histograms for the whole model can be found in Appendix B. Furthermore, we calculate the center of mass per layer and compare them to the attention centers of mass in Figure 5. The histograms and the centers of mass show that
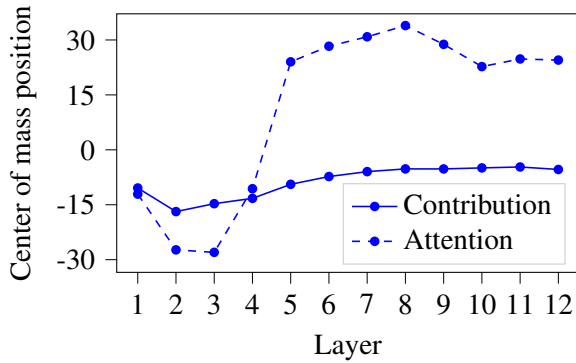
Figure 5: Mean center of mass for contribution vs. attention histograms per layer for English.

the global pattern of context aggregation is much more uniform than shown by the attention maps, especially after layer 4. This is intuitive: given that in the first layers the heads are attending mostly to the past context, on average, all the hidden tokens have a larger amount of past context. Therefore, when in later layers the attention shifts to the future hidden tokens, the past context already contained in these tokens balances the contribution, resulting in a uniform pattern of global context aggregation.

The difference in the patterns revealed by this global analysis and the local head analysis from the previous section shows a strong mismatch between attention distributions and global context aggregation in attention heads. In fact, local attention patterns can easily lead to spurious conclusions when used to interpret global context aggregation. Next, we study this difference quantitatively.

9 Local Attention vs. Global Attribution

To quantify the discrepancy between attention distributions and input contribution, i.e., local and global patterns of context aggregation, we calculate the correlation between attention maps and input contribution $C(\boldsymbol{e}_i^0, \boldsymbol{o}_{h,j}^l)$. We follow the same methodology as in Section 6 and report Pearson's and Spearman's correlation coefficient in Figure 6. In line with the mismatch between attention and contribution histograms (Figures 2 and 3), we observe how the correlation between attention and input contribution quickly decreases in deeper layers. Particularly, after only four layers Pearson's correlation coefficient for most heads is smaller than 0.5 and in the last four layers the median head correlation value is smaller than 0.25. Furthermore, Spearman's correlation follows a very similar trend, with the median head correlation value falling un-
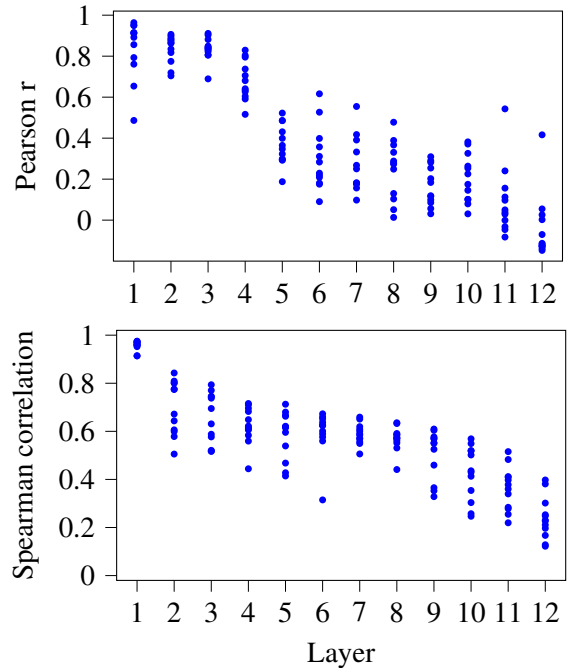

Figure 6: (Top) Pearson and (Bottom) Spearman correlation between attention and input contribution.

der 0.7 already at layer 3, and under 0.25 at the last layer.

The results from this section point at the importance of information mixing: attention maps show how the heads behave locally, i.e., how they aggregate context, but not *what* context is in fact aggregated. Knowing how the heads behave locally can give us a better understanding of transformer models that could be leveraged to further improve the performance of these models (Wu et al., 2020). However, attention maps are misleading when drawing conclusions about what input words are being aggregated into the contextual embeddings.

9.1 Specific examples

The histograms studied in the previous sections give us a high level picture of what is happening inside the model. However, we averaged across examples with different sequence length and with different token types in different positions. To gain a more detailed understanding of the model's behaviour, we now look into specific input sequences randomly selected from our dataset.

Kovaleva et al. (2019) study attention maps generated by BERT for many different examples and divide the attention patterns into five types: vertical, diagonal, vertical-diagonal, block and heterogeneous. When looking at the attention maps, we
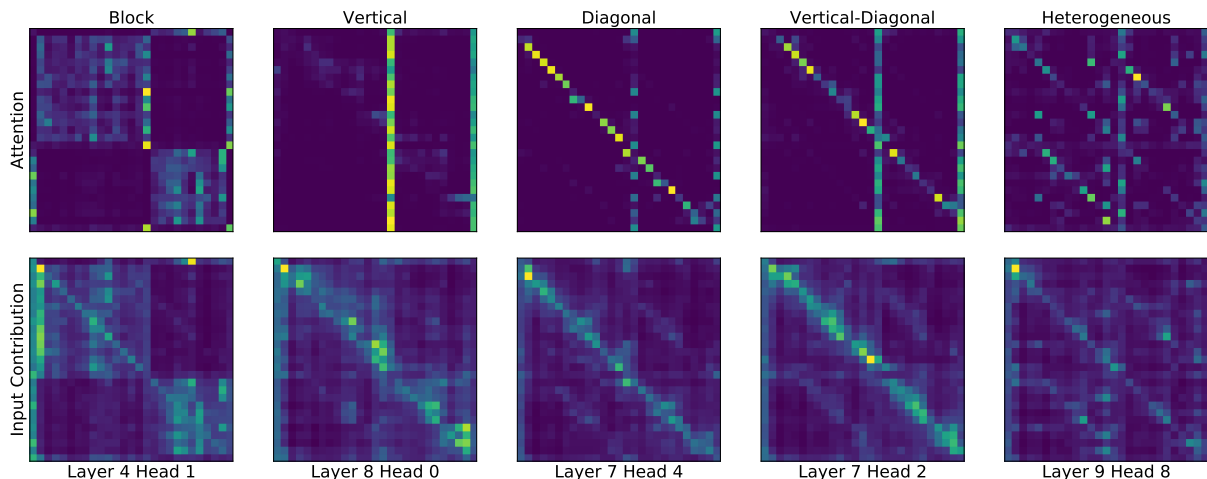
Figure 7: Comparison of the head patterns revealed by attention distributions (upper row) and input contribution (lower row).

observe the same five attention patterns. Nevertheless, to understand what input information these heads are actually aggregating, we need to look at the contribution from the input tokens.

In Figure 7, we compare the five patterns observed by Kovaleva et al. (2019) with the corresponding patterns revealed by Hidden Token Attribution with respect to the input, $C(e_i^0, o_{h,j}^l)$. A comparison for all heads is available in Appendix C. Remarkably, heads with the vertical pattern pay most attention to the SEP and CLS tokens. Nevertheless, the input contribution reveals that SEP tokens are used by the model to store general context, and by extracting information from the SEP token at intermediate layers, the model is in fact aggregating global context. Hence, with respect to the input, heads with vertical, diagonal and vertical-diagonal patterns have a similar behavior to heterogeneous heads. However, tokens around the diagonal tend to contribute the most given the prevalent aggregation of local context.

On the other hand, as shown in the first column of Figure 7, we observe that the block patterns prevail when we apply Hidden Token Attribution to the input. It is noteworthy that, while vertical and diagonal patterns fade away, the block pattern still remains visible. The fact that attending to tokens inside a block results in aggregation of context from within that block implies that up to that point, the context was mainly aggregated from within the blocks separated by SEP. We do not observe block patterns in the contribution maps for layers deeper than layer 4, which suggests that the first layers aggregate context within blocks and later layers

aggregate context in a more global manner.

10 Conclusion

We provide justification for using HTA to study information flow within transformers. By studying attention distributions of BERT we uncover an interesting pattern: In earlier layers, attention heads attend mostly to earlier tokens, whereas this trend quickly reverses with increasing depth. This is surprising, since BERT is trained using bi-directional language modeling and it suggests that like humans, BERT understands language from left to right.

A problem with local attention patterns is that they do not reveal how the attention heads process the information contained in the input tokens. We thus use Hidden Token Attribution to compute per-head attribution distributions over the input words. Our results show that the mismatch between attention and attribution distributions increases with depth. This confirms the importance of accounting for information mixing when analyzing attention heads with respect to the input tokens. Finally, we show how five different attention head patterns differ from their token attribution equivalents. Our method and results are complementary to those in (Abnar and Zuidema, 2020), we believe a combination of attention flow and HTA may provide new interesting insights.

In this work, we aim to set a clear border that distinguishes between local and global aggregation in transformer models. This distinction is important when trying to interpret the behavior of these models, and we hope that it will help future studies in their analyses. Furthermore, our findings

add new insights to the growing field of research on explaining transformers. This research, in turn, can help in guiding design decisions leading to further improvements of natural language processing architectures.

References

Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Arabert: Transformer-based model for arabic language understanding.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Gino Brunner, Yang Liu, Damian Pascual, Oliver Richter, Massimiliano Ciaramita, and Roger Wattenhofer. 2020. On identifiability in transformers. In *International Conference on Learning Representations (ICLR)*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of bert's attention. *CoRR*, abs/1906.04341.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Deepset. 2019. Open sourcing german bert. https://deepset.ai/german-bert. Accessed: 2020-05-18.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Ian Simon, Curtis Hawthorne, Noam Shazeer, Andrew M. Dai, Matthew D. Hoffman, Monica Dinculescu, and Douglas Eck. 2019. Music transformer. In *International Conference on Learning Representations*.

Sarthak Jain and Byron C. Wallace. 2019. Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 3543–3556.

Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4356–4365.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Timothy Niven and Hung-Yu Kao. 2019. Probing neural network comprehension of natural language arguments. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4658–4664. Association for Computational Linguistics.

Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. 2018. Image transformer. In *International Conference on Machine Learning*, pages 4055–4064.

Danish Pruthi, Mansi Gupta, Bhuwan Dhingra, Graham Neubig, and Zachary C. Lipton. 2019. Learning to deceive with attention-based explanations. *CoRR*, abs/1909.07913.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf.*

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. 2019. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pages 9686–9698.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. 2020. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Sofia Serrano and Noah A. Smith. 2019. Is attention interpretable? In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2931–2951.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019b. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Sarah Wiegreffe and Yuval Pinter. 2019. Attention is not not explanation. *CoRR*, abs/1908.04626.

Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. 2020. Lite transformer with long-short range attention. In *International Conference on Learning Representations*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *https://arxiv.org/abs/1906.08237*.
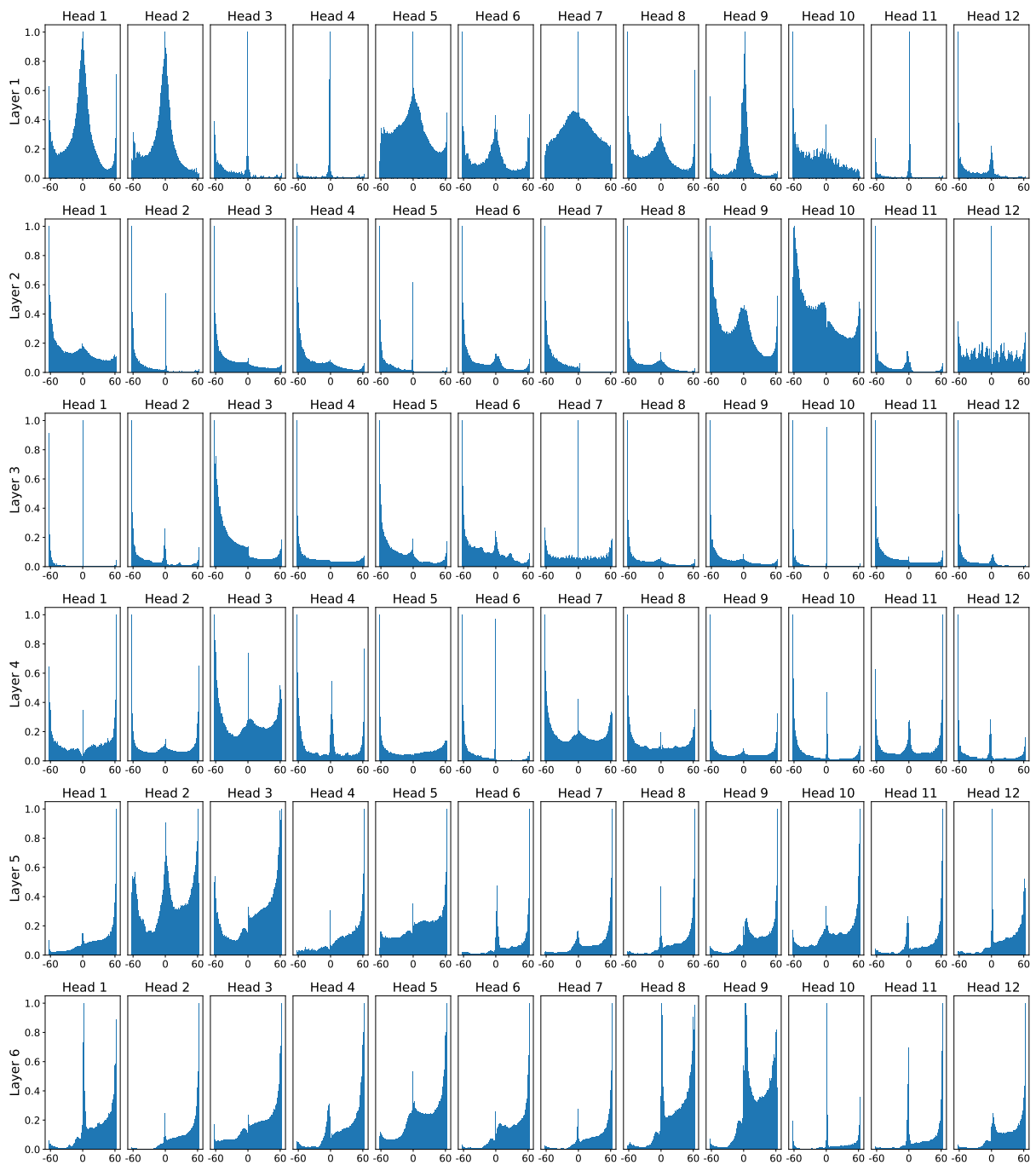
A   Attention Histograms English

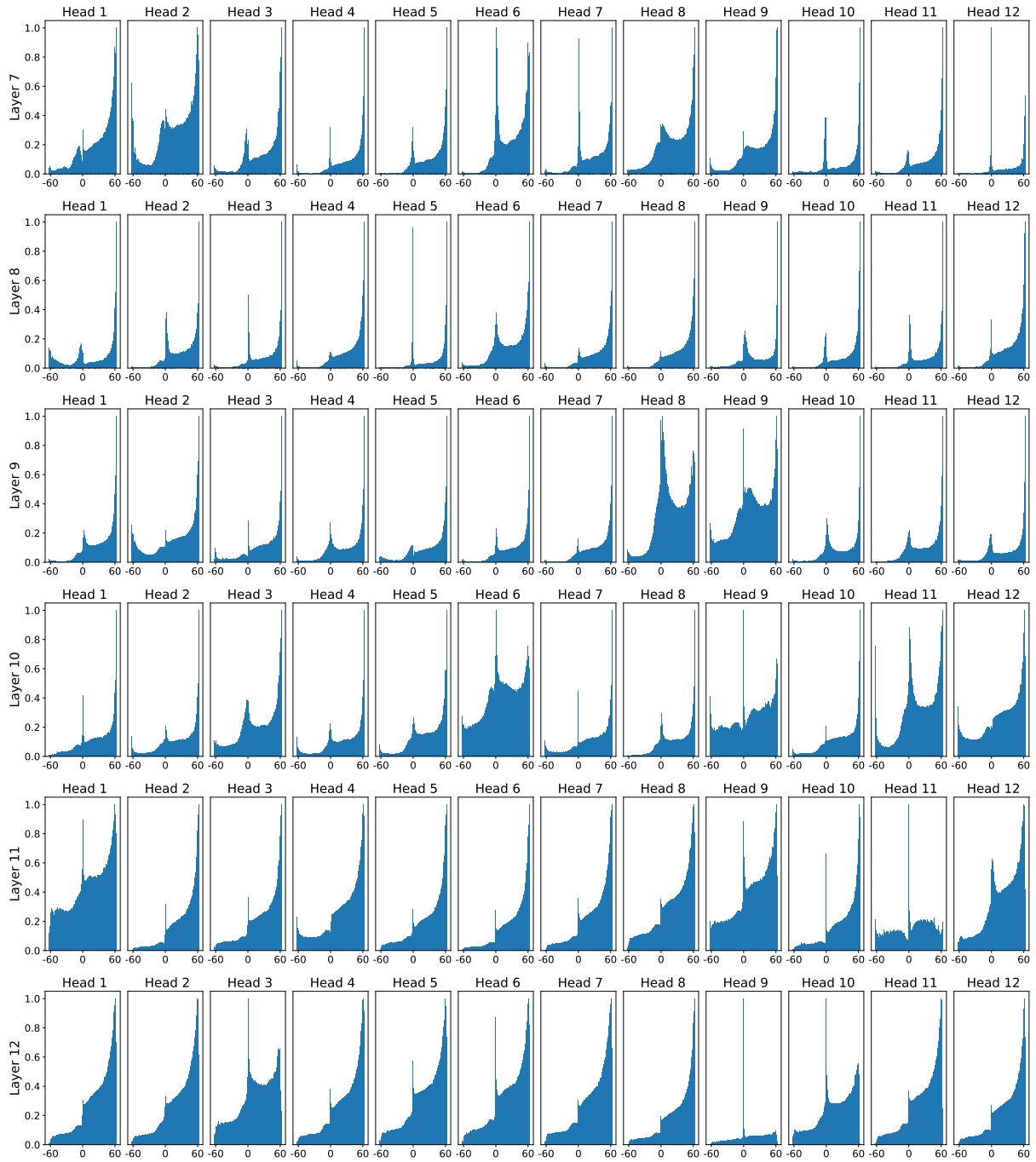Figure 8: Attention histograms for layers 1 to 6

Figure 9: Attention histograms for layers 7 to 12

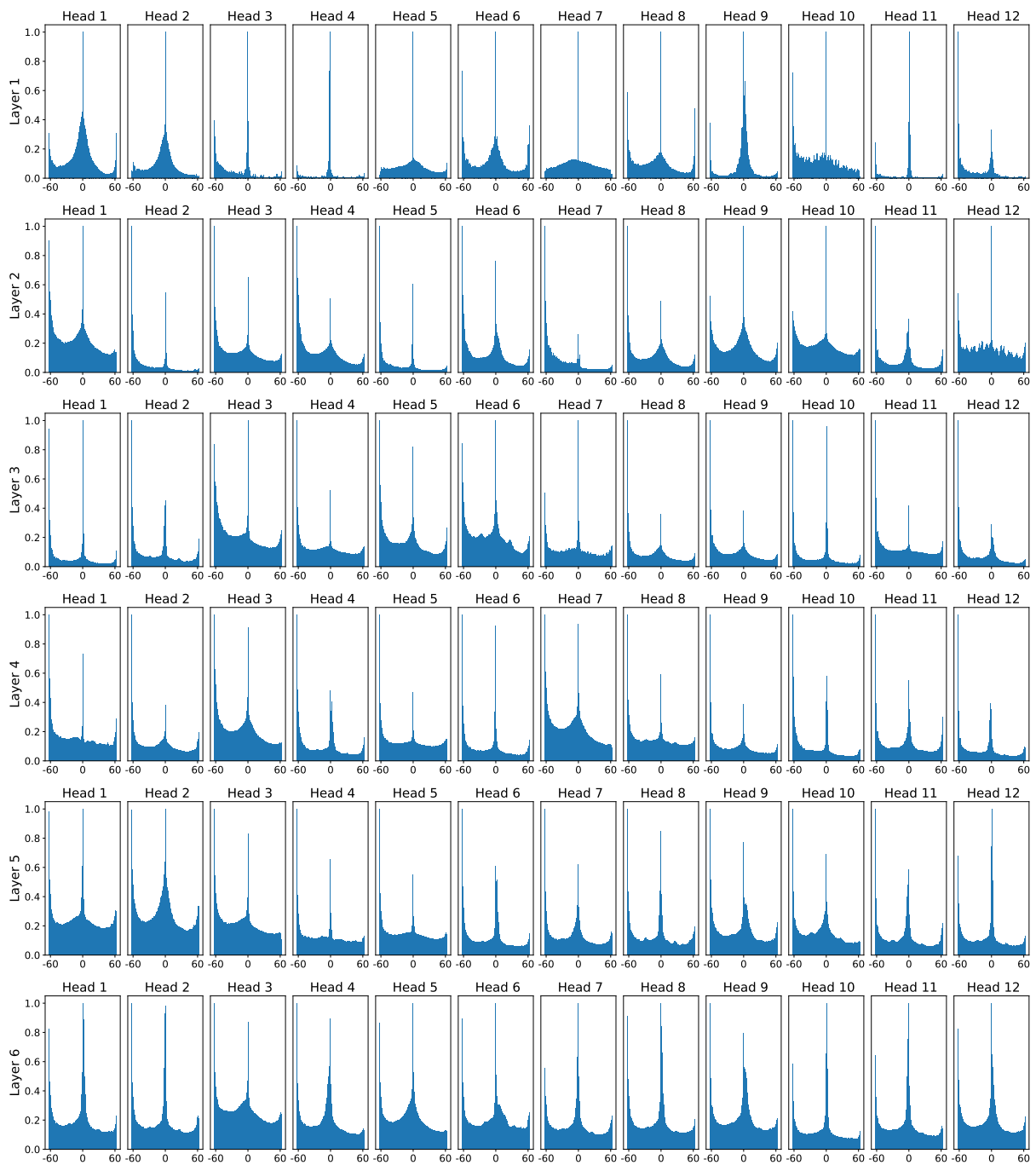# B   Input Contribution Histograms

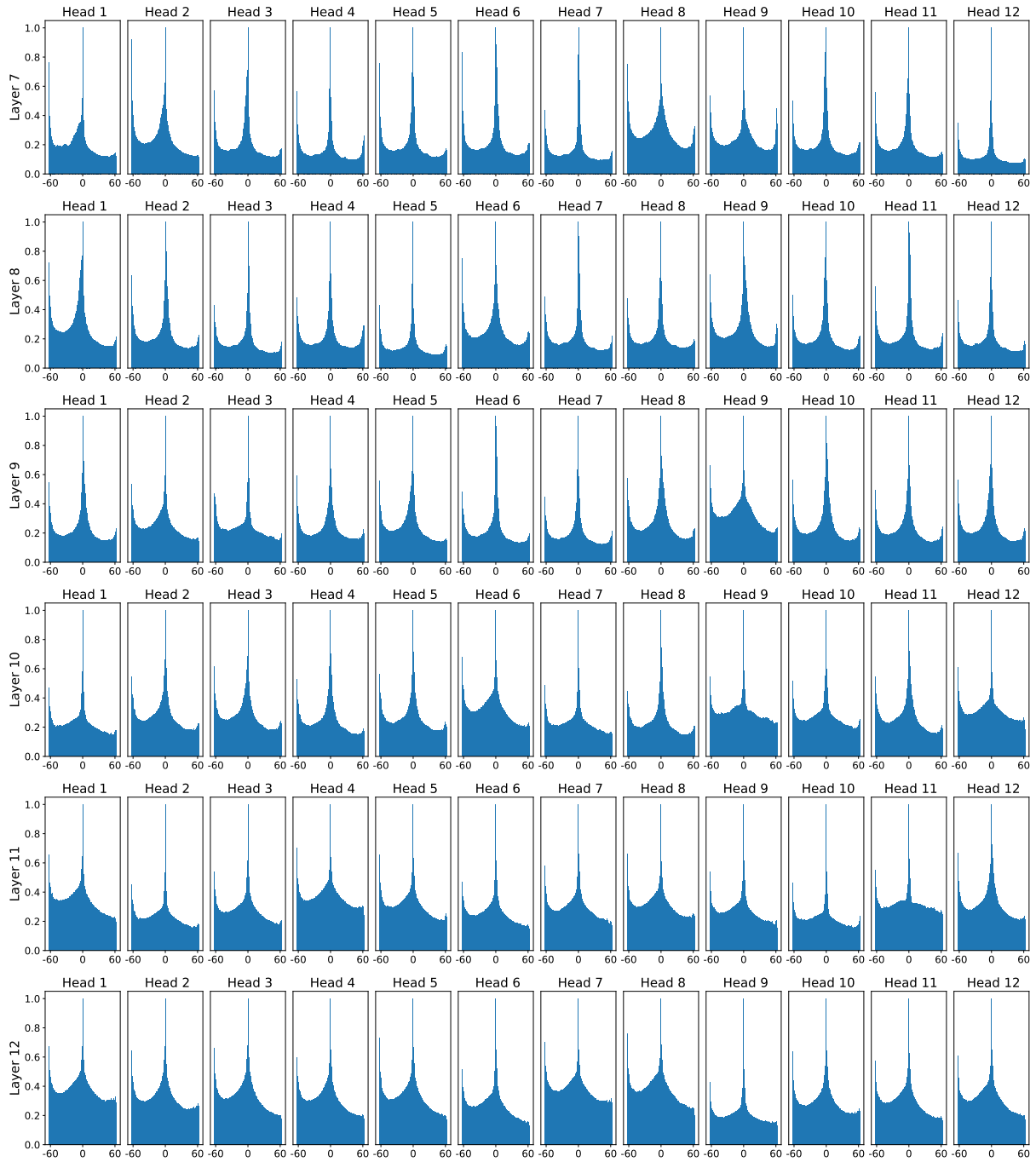Figure 10: Input contribution histograms for layers 1 to 6

Figure 11: Input contribution histograms for layers 7 to 12
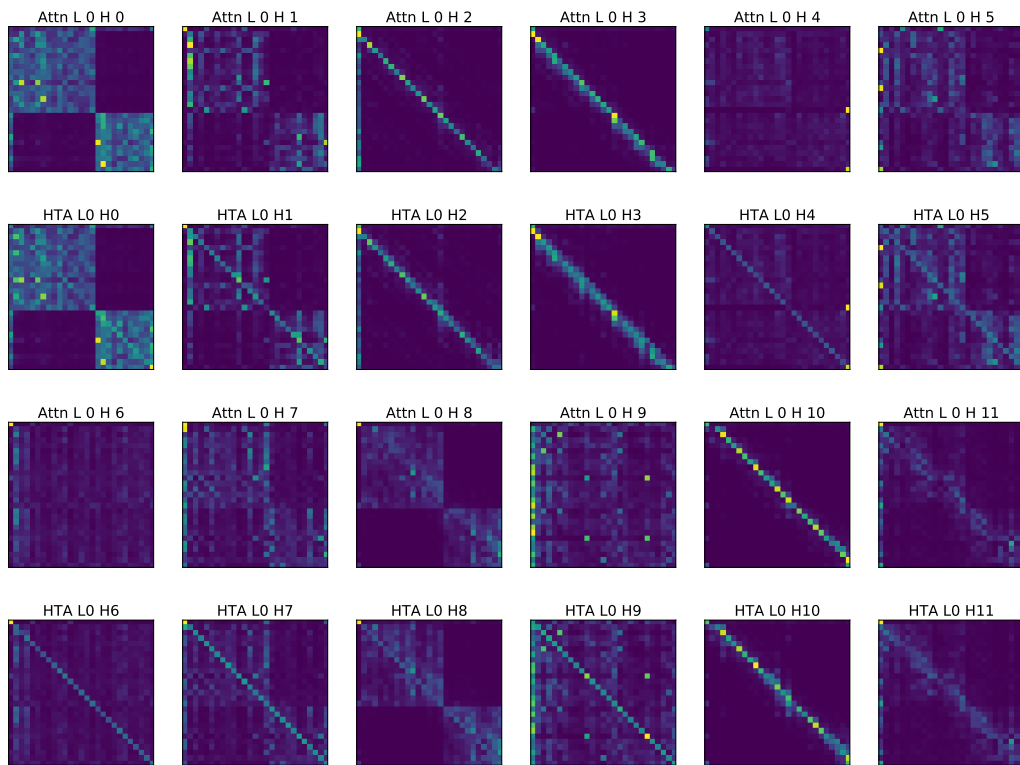
C Comparison of Local vs. Global Head Patterns

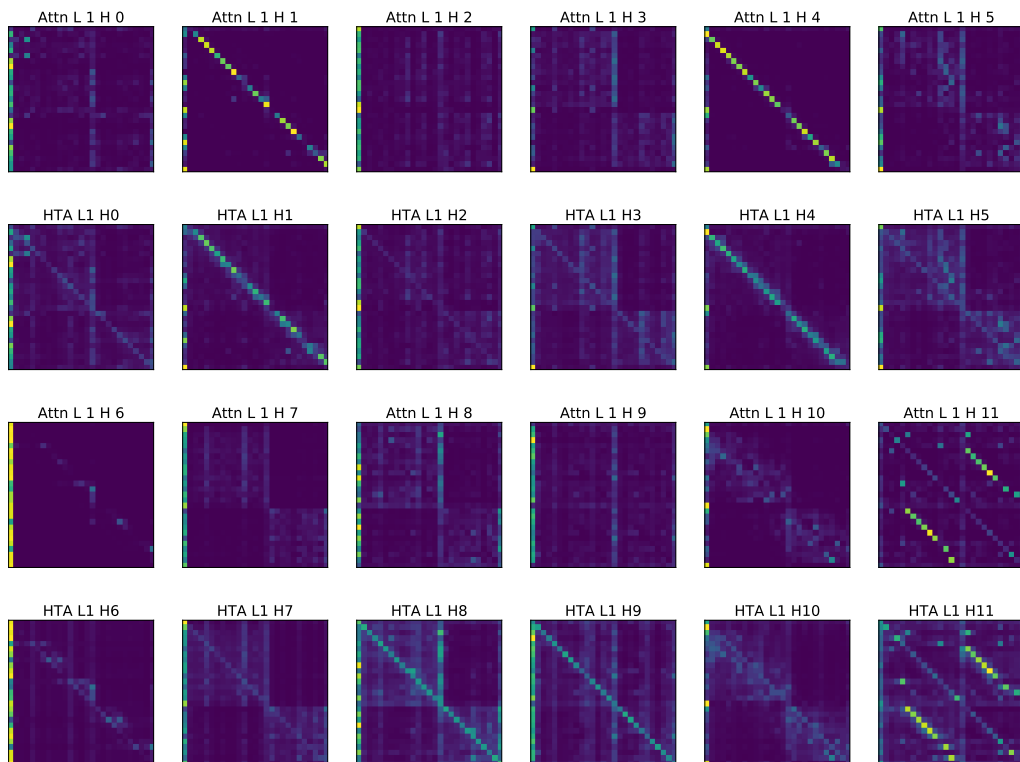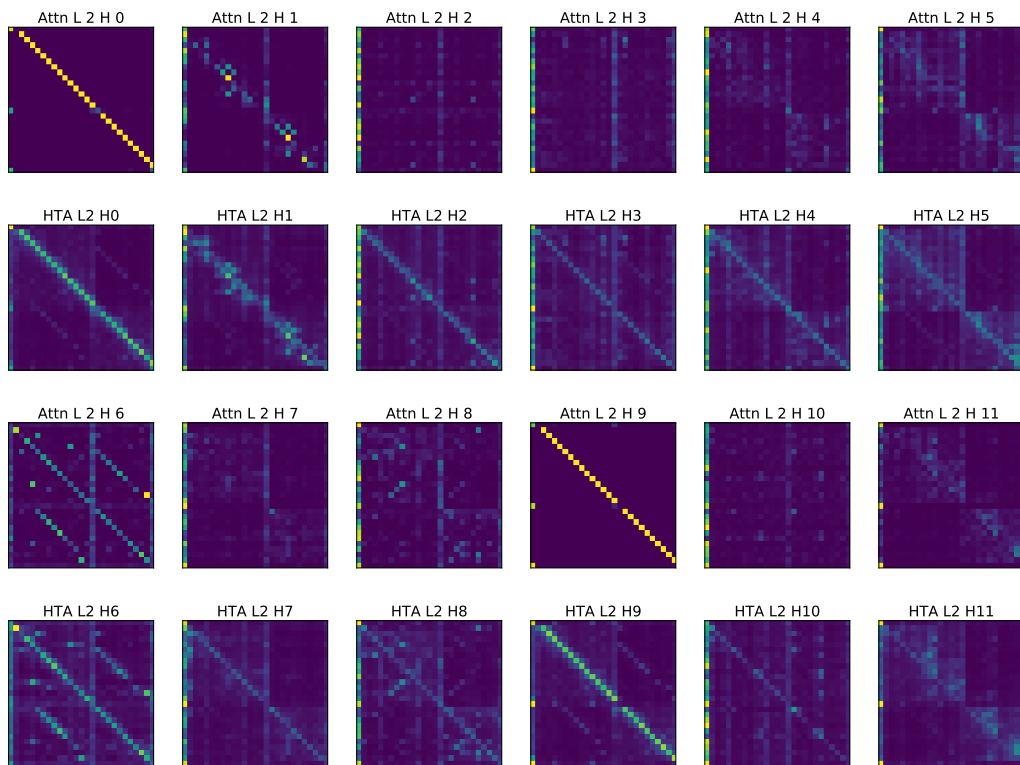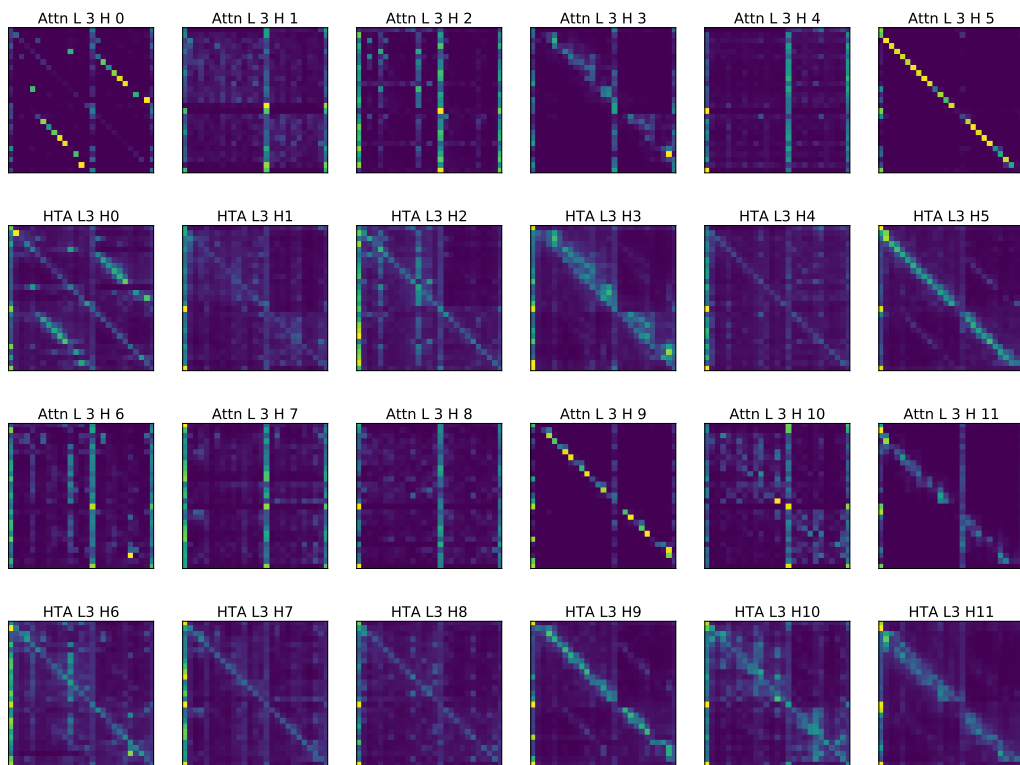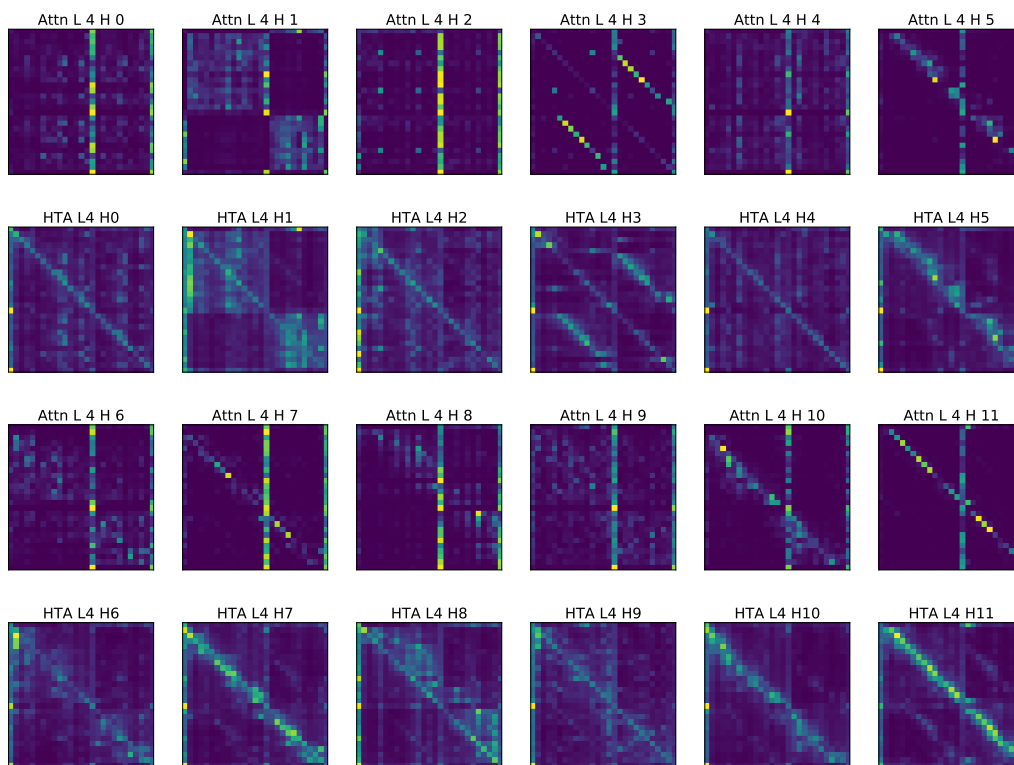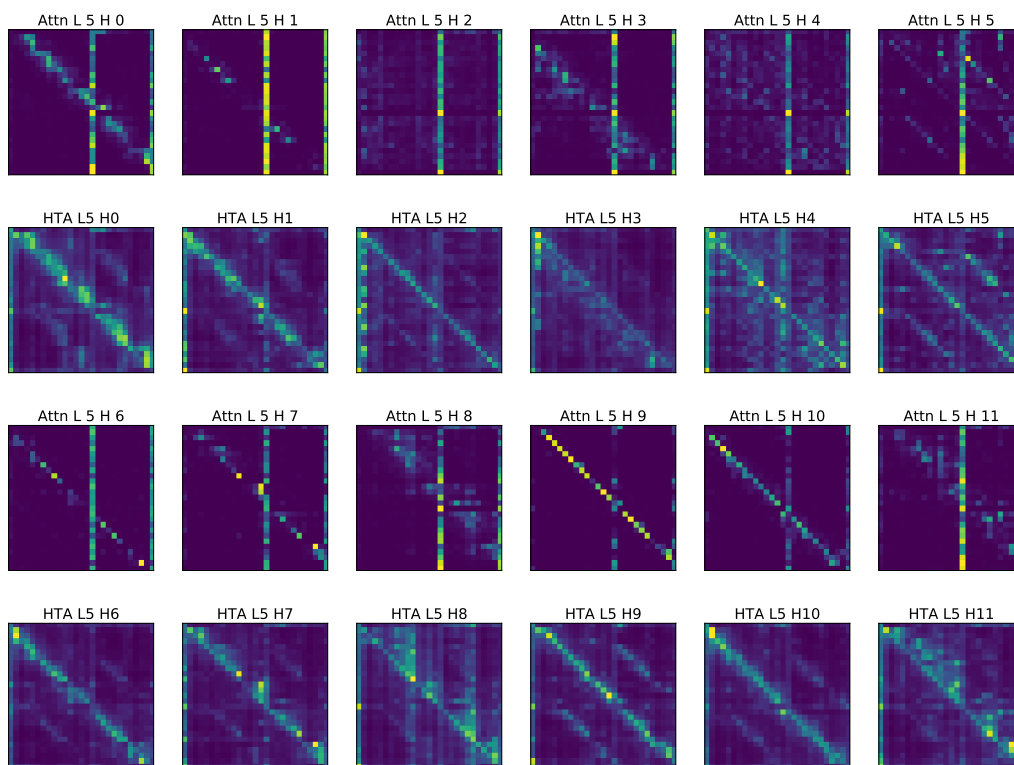Figure 12: Layer 1: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.

Figure 13: Layer 2: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.
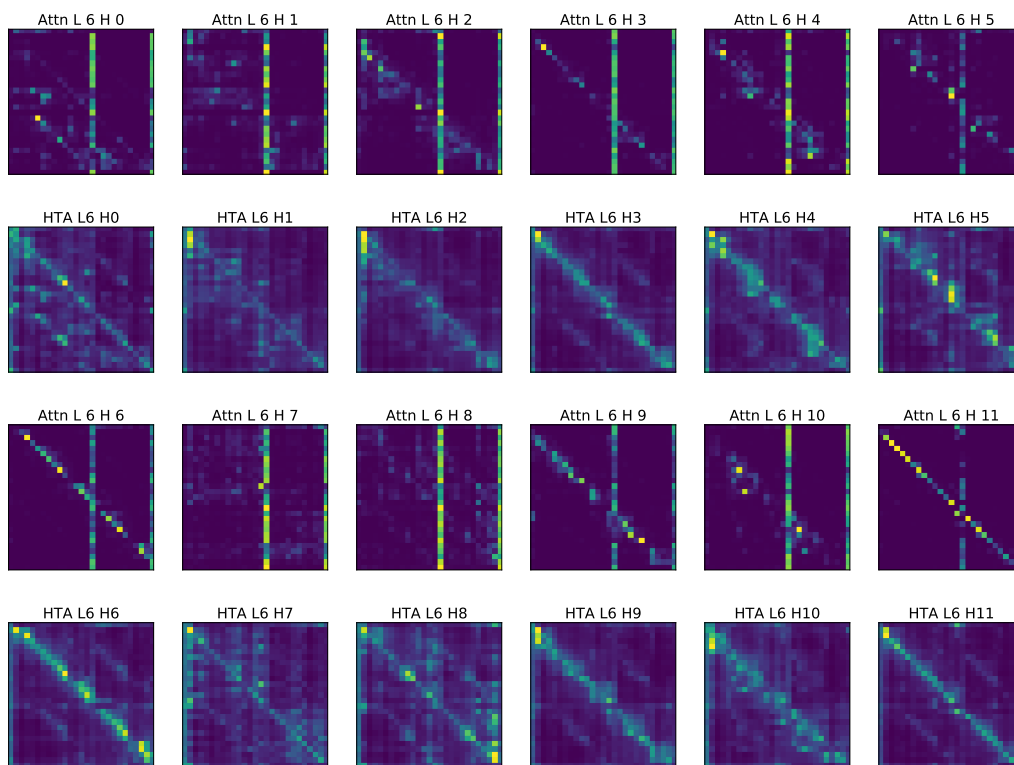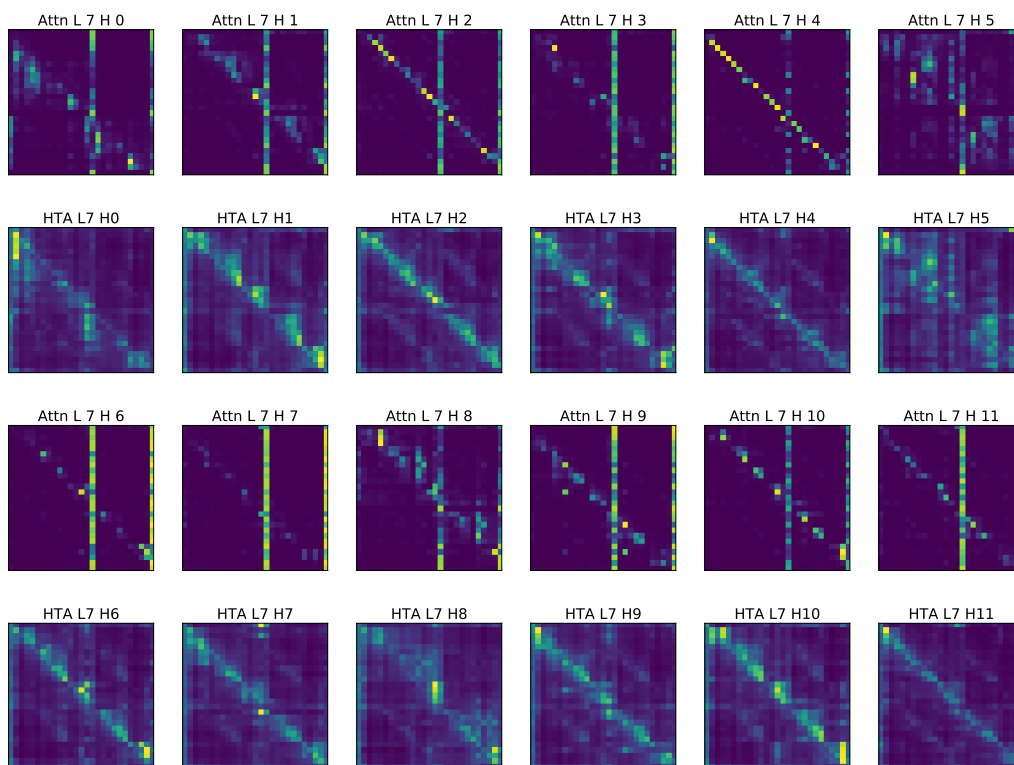
Figure 14: Layer 3: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.

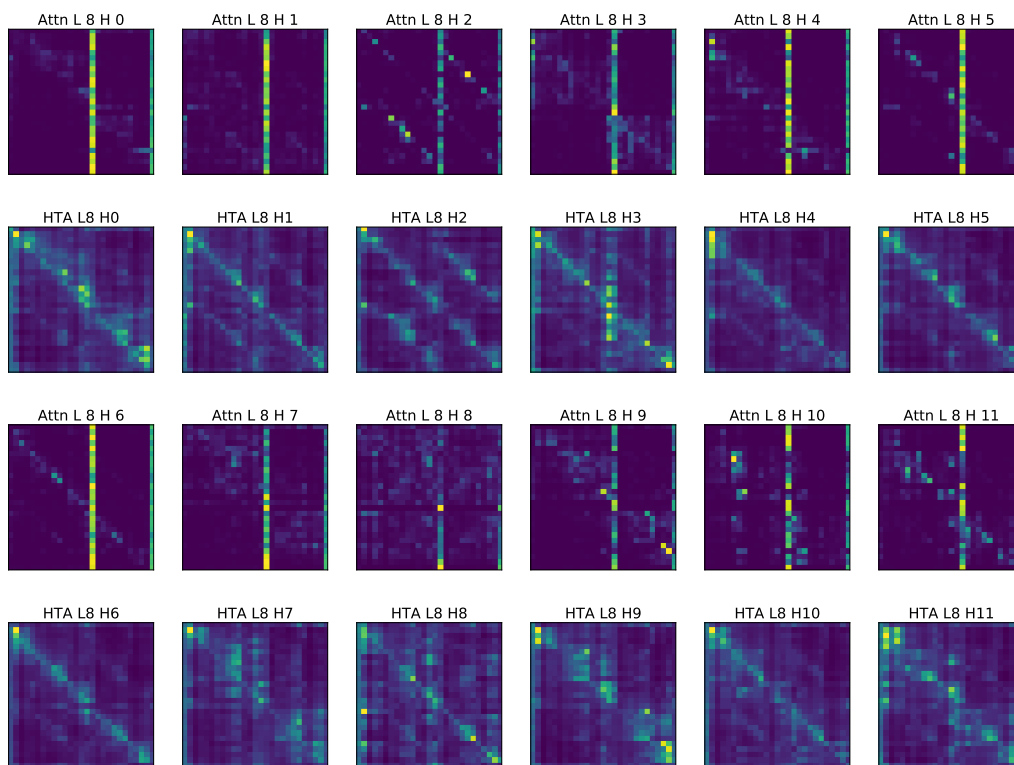Figure 15: Layer 4: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.

Figure 16: Layer 5: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.

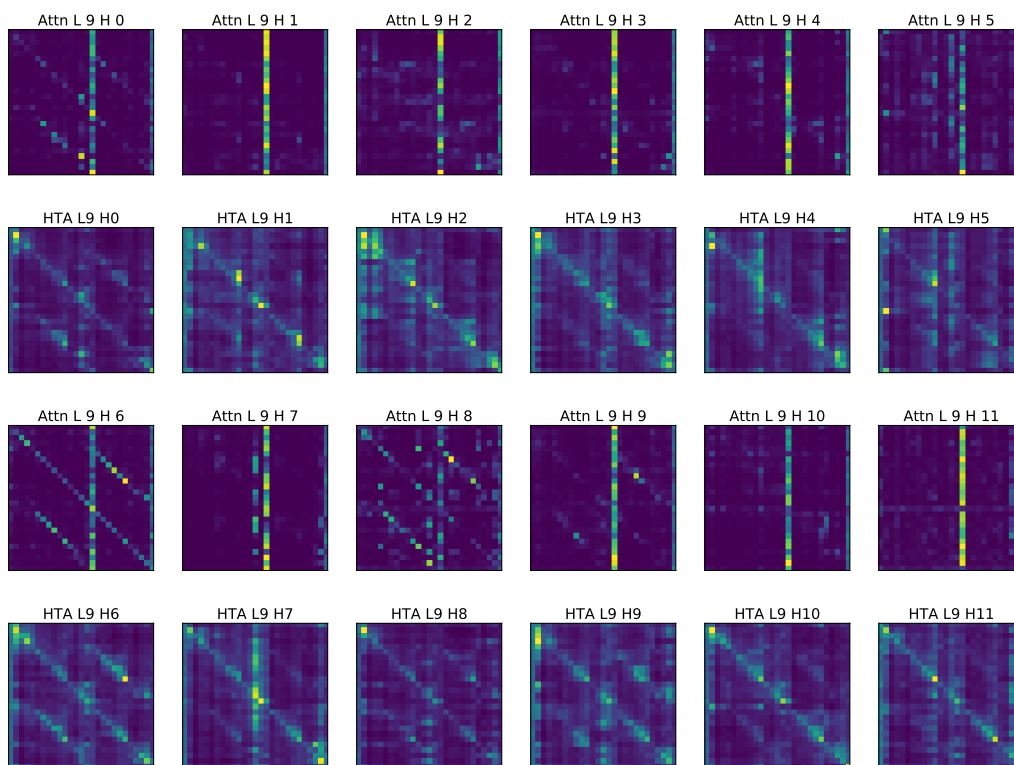Figure 17: Layer 6: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.

Figure 18: Layer 7: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.

Figure 19: Layer 8: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.

Figure 20: Layer 9: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.

Figure 21: Layer 10: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.
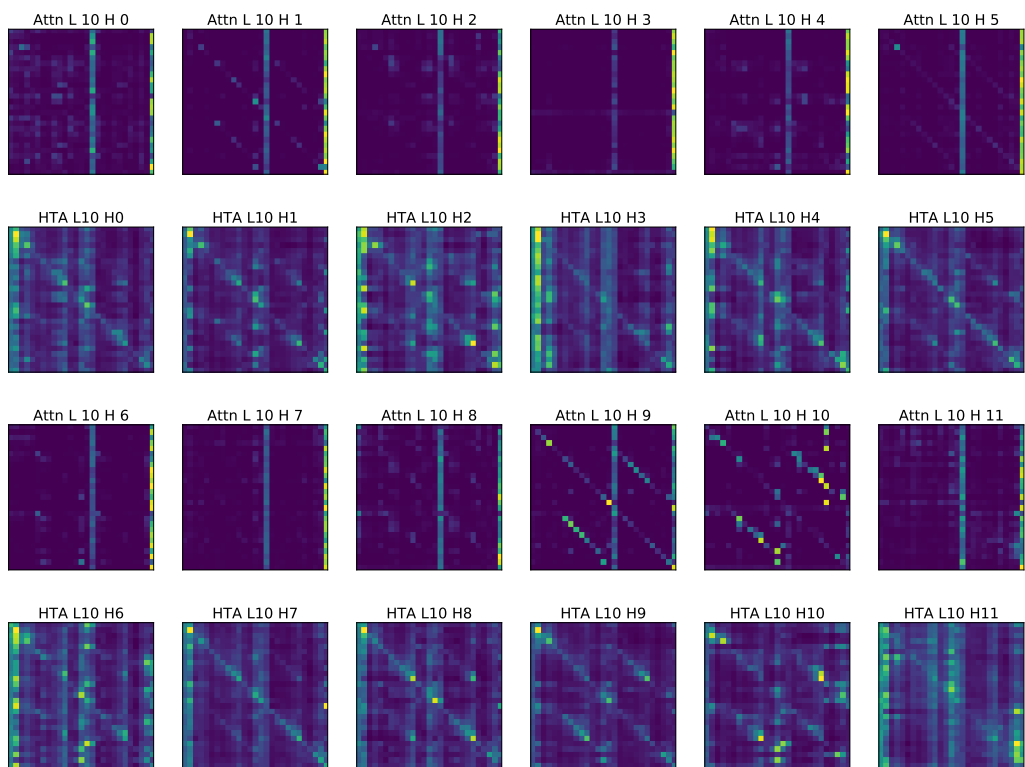
Figure 22: Layer 11: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.
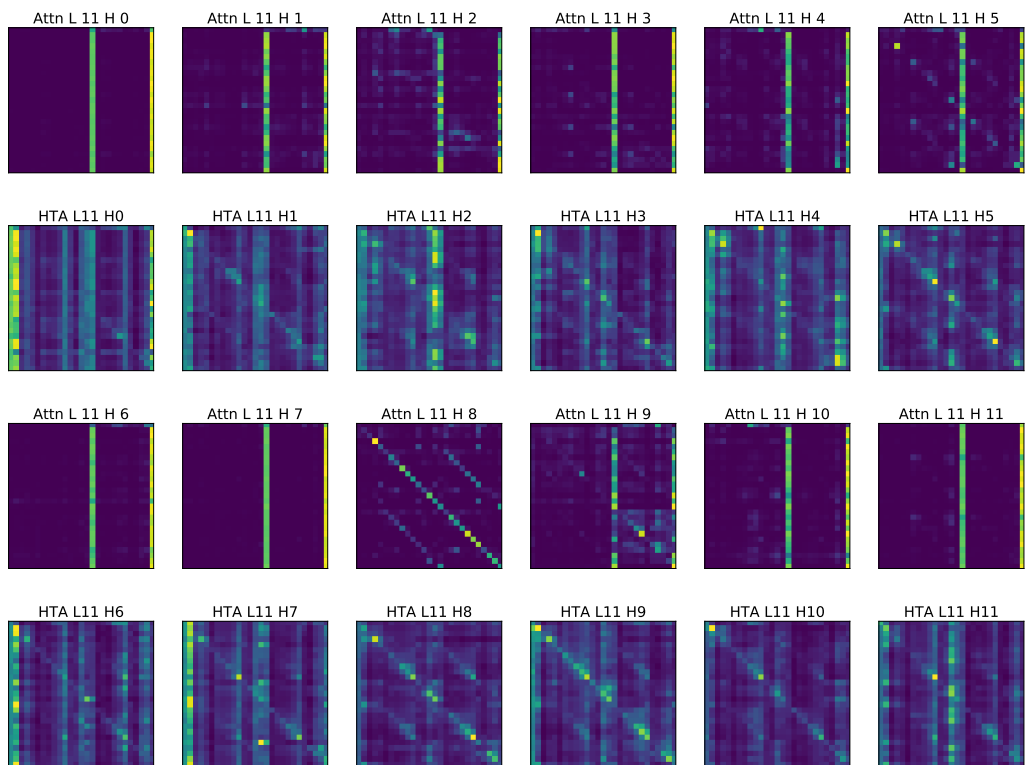

Figure 23: Layer 12: rows 1 and 3 represent attention maps, rows 2 and 4 show the corresponding contribution maps.