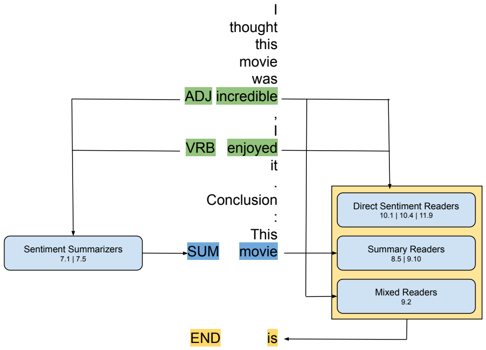

Figure 7: Primary components of GPT-2 sentiment circuit for the ToyMovieReview dataset. Here we can see both direct use of sentiment-laden words in predicting sentiment at END as well as an example of the summarization motif at the SUM position. Heads 7.1 and 7.5 write to this position and this information is causally relevant to the contribution of the summary readers at END.

4 The Summarization Motif for Sentiment

4.1 Circuit Analyses

In this sub-section, we present circuit[3] analyses that give qualitative hints of the summarization motif, and restrict quantitative analysis of the summarization motif to 4.2. Through an iterative process of path patching (see Section 2.3) and analysing attention patterns, we have identified the circuit responsible for the ToyMovieReview task in GPT2-small (Figure 7) as well as the circuit for the ToyMoodStories task. Below, we provide a brief overview of the circuits we identified, reserving the full details for A.3.

Initial observations of summarization in GPT-2 circuit for ToyMovieReview Mechanistically, this is a binary classification task, and a naive hypothesis is that attention heads attend directly from the final token to the valenced tokens and map positive sentiment to positive outputs and vice versa. This happens, but in addition attention head output is causally important at intermediate token positions, which are then read from when producing output at END. We consider this an instance of summarization, in which the model aggregates causally-important information relating to an entity at a particular token for later usage, rather than simply attending back to the original tokens that were the source of the information.

We find that the model performs a simple, interpretable algorithm to perform the task (using a circuit made up of 9 attention heads):

1. Identify sentiment-laden words in the prompt, at ADJ and VRB.
2. Write out sentiment information to SUM (the final "movie" token).
3. Read from ADJ, VRB and SUM and write to END.[4]

The results of activation patching the residual stream can be seen in the Appendix, Fig. A.7. The output of attention heads is only important at the movie position, which we designate as SUM. We label these heads "sentiment summarizers." Specific attention heads attend to and rely on information written to this token position as well as to ADJ and VRB.

To validate this circuit and the involvement of the sentiment direction, we patched the entirety of the circuit at the ADJ and VRB positions along the sentiment direction only, achieving a 58.3% rate

[2] We did this to maximise the chances of sentiment tokens occurring at similar positions

[3] We use the term "circuit" as defined by Wang et al. (2022), in the sense of a computational subgraph that is responsible for a significant proportion of the behavior of a neural network on some predefined task.

[4] We note that our patching experiments indicate that there is no causal dependence on the output of other model components at the ADJ and VRB positions–only at the SUM position.

of logit flips and a logit difference drop of 54.8% (in terms of whether a positive or negative word was predicted). Patching the circuit at those positions along all directions resulted in flipping 97% of logits and a logit difference drop of 75%, showing that the sentiment direction is responsible for the majority of the function of the circuit.

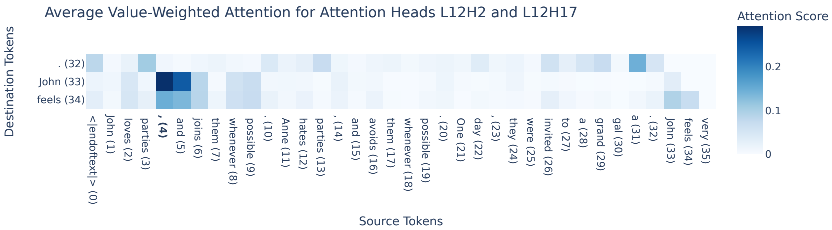

Figure 8: Value-weighted[5]averaged attention to commas and comma phrases in Pythia-2.8b from the top two attention heads writing to the repeated name and "feels" tokens–two key components of the summarization sub-circuit in the ToyMoodStories task. Note that they attend heavily to the relevant comma from both destination positions.

Multi-subject mood stories in Pythia 2.8b We next examined the circuit that processes the mood dataset in Pythia-2.8b (the smallest model that could perform the task), which is a more complex task that requires more summarization. As such it presents a better object for study of this motif. We reserve a detailed description of the circuit for the Appendix, but here we observed increasing reliance on summarization, specifically:

- A set of attention heads **attended primarily to the comma** following the preference phrase for the queried subject (e.g. John hates parties,), and secondarily to other words in the phrase, as seen in Figure 8. We observed this phenomenon both with regular attention and value-weighted attention, and found via path patching that **these heads relied partially on the comma token** for their function, as seen in Figure A.9.
- Heads attending to preference phrases (both commas and other tokens) tended to write to the repeated name token near the end of the sentence (John) as well as to the feels token–another type of summarization behavior. Later heads attended to the repeated name and feels tokens with an output important to END.

4.2 Exploring and validating summarization behavior in punctuation

Our circuit analyses reveal suggestive evidence that summarization behavior at intermediate tokens like commas, periods and certain nouns plays an important part in sentiment processing, despite these tokens having no inherent valence. We focus on summarization at commas and periods and explore this further in a series of ablation and patching experiments. We find that in many cases this summarization results in a partial information bottleneck, in which the summarization points become as important (or sometimes more important) than the phrases that precede them for sentiment tasks.

Summarization information is comparably important as original semantic information In order to determine the extent of the information bottleneck presented by commas in sentiment processing, we tested the model's performance on the multi-subject mood stories dataset mentioned above. We froze the model's attention patterns to ensure the model used the information from the patched commas in exactly the same way as it would have used the original information. Without this step, the model could simply avoid attending to the commas. We then performed activation patching on either the precomma phrases (e.g., patching "John hates parties," with "John loves parties,") while freezing the commas so they retain their original, unflipped values; or on the two commas alone, and find a similar drop in the logit difference for both as shown in table 1a.

[5] That is, the attention pattern weighted by the norm of the value vector at each position as per Kobayashi et al. (2020). We favor this over the raw attention pattern as it filters for *significant* information being moved.

Table 1: Patching results at summary positions

Intervention	**Change in logit difference**
Patching full phrase values (incl. commas)	-75%
Patching pre-comma values (freezing commas)	-38%
Patching comma values only	-37%

(a) Change in logit difference from intervention on attention head value vectors

Count of irrelevant tokens after preference phrase	**Ratio of LD change for periods vs. phrases**
0 tokens	0.29
10 tokens	0.63
18 tokens	0.92
22 tokens	1.15

(b) Ratio between logit difference change for periods vs. pre-period phrases after patching values

Importance of summarization increases with distance We also observed that reliance on summarization tends to increase with greater distances between the preference phrases and the final part of the prompt that would reference them. To test this, we injected irrelevant text[6] after each of the preference phrases in our multi-subject mood stories (after "John loves parties." etc.) and measured the ratio between logit difference change for the periods at the end of these phrases vs. pre-period phrases, with higher values indicating more reliance on period summaries (Table 1b). We found that the periods can be up to 15% **more** important than the actual phrases as this distance grows. Although these results are only a first step in assessing the importance of summarization importance relative to prompt length, our findings suggest that this motif may only increase in relative importance as models grow in context length, and thus merits further study.

4.3 Validating summarization behavior in SST

In order to study more rigorously how summarization behaves with natural text, we examined this phenomenon in SST. We appended the suffix "Review Sentiment:" to each of the prompts and evaluate Pythia-2.8b on zero-shot classification according to whether positive or negative have higher probability and are in the top 10 tokens predicted. We then take the subset of examples Pythia-2.8b succeeds on that have at least one comma, which means we start with a baseline of 100% accuracy. We performed ablation and patching experiments on comma representations. If comma representations do not summarize sentiment information, then our experiments should not damage the model's abilities. However, our results reveal a clear summarization motif for SST.

Ablation baselines We performed two baseline experiments in order to obtain a control for our later experiments. First to measure the total effect of the sentiment directions, we performed directional ablation (as described in 2.3) using the sentiment directions found with DAS to every token at every layer, resulting in a 71% reduction in the logit difference and a 38% drop in accuracy (to 62%). Second, we performed directional ablation on all tokens with a small set of random directions, resulting in a $< 1\%$ change to the same metrics.

Directional ablation at all comma positions We then performed directional ablation–using the DAS (2.2) sentiment direction–to every comma in each prompt, regardless of position, resulting in an 18% drop in the logit difference and an 18% drop in zero-shot classification accuracy–indicating that nearly 50% of the model's sentiment-direction-mediated ability to perform the task accurately was mediated via sentiment information at the commas. We find this particularly significant because we did not take any special effort to ensure that commas were placed at the end of sentiment phrases.

Mean-ablation at all comma positions Separately from the above, we performed mean ablation at all comma positions as in 2.3, replacing each comma activation vector with the mean comma activation from the entire dataset in a layerwise fashion. Note that this changes the entire activation on the comma token, not just the activation in the sentiment direction. This resulted in a 17% drop in logit difference and an accuracy drop of 19% .

[6] E.g. "John loves parties. *He has a red hat and wears it everywhere, especially when he is riding his bicycle through the city streets.* Mark hates parties. *He has a purple hat but only wears it on Sundays, when he takes his weekly walk around the lake.* One day, they were invited to a grand gala. John feels very"