Appendix

A Datasets

The current workflow on circuit identification is to first build a synthetic dataset that elicits the behavior or specific task under study. Such dataset is not used for training nor finetuning, but to identify the relevant components of the circuit via a series of causal interventions, or activation patching experiments. The dataset curation process is specific to the task under study, but previous works follow these general guidelines or suggestions:

- The different prompts are built according to a predefined template so that, when tokenized, the important tokens (e.g. the first letter of a noun in the acronym prediction task) share the same position across all samples. The reason behind is that most ablation schemes work by replacing the activations from other samples, implying that (i) we are unable to use sequences with different lengths and (ii) it is recommended that they share the same template, so that the semantic meaning is distributed similarly across samples.
- Words or letters that are thought to be relevant for the specific task are tokenized individually in a single token. In other words, in the hypothetical task of predicting names from a sentence, it is recommended to build sentences with names that are tokenized as a single token (e.g. `"|Mary|"`).

However, it is important to remark that these are suggestions to ease the process of identifying the circuit and understanding the different components. In fact, previous works have built datasets following different templates and found the same underlying circuit.

In our experiments, we will use the same datasets that were built for the purpose of manual identification of circuits responsible for different tasks on GPT-2 Small. Below we present a description of each dataset:

- **Acronym Prediction:** As the task under study is the prediction of 3-letter acronyms, the dataset is composed by prompts such as `"The Chief Executive Officer (CEO"`, where the task would be to predict the acronym. More especifically, the prompts share the same underlying template: `"|The|C1|T1|C2|T2|C3|T3| (|A1|A2|A3|"`, where `Ci` is the token encoding the capital letter of the ith word (together with its preceding space), `Ti` is the remainder of the word, and `Ai` is the ith letter of the acronym. Despite this being a multi-token prediction task, we limit our analysis to single-token by focusing on the prediction of the last letter of the acronym (i.e. `A3`) in order to provide a better comparison with the other tasks, which are single-token prediction tasks.
- **Indirect Object Identification (IOI):** The dataset is composed by a total of 15 templates. However, we focus on just one template for the sake of giving a more illustrative comparison and evaluation of our proposal. Specifically, we focus on the following template: `"Then, [B] and [A] went to the [PLACE]. [B] gave a [OBJECT] to [A]"`, where `[A]` and `[B]` are first names in English. The first names, places and objects are selected so that they are encoded as a single token each to ensure proper alignment across sequences.
- **Greater-Than:** The dataset is composed by sentences that follow the template: `"The [noun] lasted from the year XXYY to XX"`, where `[noun]` is drawn from a hardcoded pool of nouns (e.g. abduction, accord, affair, etc.) and `XXYY` is a year, where `XX` represents the century, which is drawn from $\{11, ..., 17\}$, and `YY` represents the rest of the year, which is drawn from $\{02, ..., 98\}$. The reason behind this design choice is that there are years, such as 1700, which are naturally tokenized by GPT-2 Small tokenizer as a single token. Hence, both the century and start year have to be carefully sampled so that the resulting years are tokenized as `|XX|YY|`.

A more thorough explanation of the dataset building process can be found in their respective works García-Carrasco, Maté, and Carlos Trujillo (2024); Wang et al. (2023); Hanna, Liu, and Variengien (2023).

B Design Choices

In this section, we provide a further discussion on the design choices of Algorithm 1 as well as additional information.

Nodes vs. Edges

Current ACD methods and most MI works focused on circuit identification represent LLMs as a Directed Acyclic Graph (DAG), where nodes represent activations (e.g. the output of an attention head or an MLP) and edges represent computations (e.g. computing the attention patterns). The main idea of current ACD methods is to patch unimportant edges: if patching an edge does not drop the performance over a specified threshold, it implies that it is not important for the specific task and it is discarded.

Patching edges instead of nodes usually results in a more fine-grained and precise circuit: by removing all unimportant edges, one is able to clearly see how the remaining nodes are connected (e.g. the output $8th$ attention head of the $9th$ layer is connected to the query input of the 5th attention head of the $11th$ layer). While this is a desirable aspect when solely focusing on interpretability, we encounter two main challenges when we focus on a model pruning standpoint (i) current MI works do not actually remove the edges/operations, they are patched via hook functions that actually slow the forward pass even more and (ii) it is difficult to efficiently translate these removed edges and truly prune them, as current transformers are implemented via large parallel operations in the shape of matrix multiplications.

Because of this, we focus on nodes instead of edges: the remaining circuit will be less precise (i.e. it will contain many more edges) but we will be able to efficiently prune it, obtaining a faster and reduced submodel that is able to perform the task under study. However, this should not be seen as an inconvenience: ACD algorithms can also be applied to our obtained submodel to obtain a more fine-grained circuit for interpretability purposes and keep our submodel to perform faster inference.

Zero vs. Mean ablation schemes

Zero ablation implies replacing the output of a component (i.e. an attention head or MLP) by a tensor of zeros, whereas mean ablation implies replacing the output by the mean tensor obtained on a reference distribution (i.e. the patching dataset).

As mentioned in the paper, zero ablation is regarded as a more "aggressive" ablation with regards to mean ablation in the MI literature. The intuition behind this is that, due to every component of a transformer model reading from and writing to a common residual stream, zero-ablating a irrelevant component could send components from upper layers off-distribution which might be relevant for the task under study. On the other hand, mean-ablation will cause the component to write common information from the reference distribution, hence being considerably less likely to send other components off distribution.

Even though our initial thoughts before the experiments tended toward mean ablation, we also decided to perform experiments with zero ablation and found the expected results. In summary, zero ablation is too aggressive and therefore requires a larger number of components to be included in the final submodel in order to maintain a good performance.

KL Divergence vs. other metrics

Previous works on manual circuit identification used metrics different from the KL divergence, mostly variations of the logit difference. However, the authors of Conmy et al. (2023) extensively evaluated ACD with different metrics and tasks and found out that the KL divergence was the most effective, yielding more consistent results. They also show that some metrics might peform better on the discovery of specific task, which is something that should be a focus of further research. Nevertheless, as the aim of our work is to perform automatic circuit extraction, we decided to only use the KL divergence to provide a more illustrative evaluation and leave the metric selection aspect for further works.

Barrier

C Truly Pruning Attention Heads

To understand how to truly prune attention heads, we first have to present how attention layers are generally implemented.

First, the input to the attention layer is a residual stream tensor $x \in \mathbb{R}^{B\times N\times d}$, where B is the batch size, N is the sequence length and d is the model dimension. As described in Vaswani et al. (2017), this vector x is mapped into three different vectors termed the query, key and value vectors via a linear mapping. Tipically, this projection operation is implemented in parallel for every attention head in the layer:

$$\text{Concat}(q, k, v) = xW_{proj} = x\text{Concat}(W_Q, W_K, W_V) \tag{4}$$

where $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$. This operation yields the $q, k, v \in \mathbb{R}^{B\times N\times d}$ vectors which are then split into n_head Q, K, V vectors associated to the ith attention head $q_i, k_i, v_i \in \mathbb{R}^{B\times N\times d_{head}}$, where $d = n_{head}d_{head}$. Then, these vectors are used to compute the output of each attention head $h_i \in \mathbb{R}^{B\times N\times d_{head}}$.

Finally, the results of every attention head are stacked and projected together as follows:

$$\text{MultiHeadAttention}(x) = \text{Concat}(h_1, h_2, ..., h_n)W^O \tag{5}$$

where $W^O \in \mathbb{R}^{n_{head}\cdot d_{head}\times d}$ is a projection matrix. However, this is equivalent to projecting each head independently and then summing the individual contributions:

$$\text{MultiHeadAttention}(x) = \sum_{i=0}^{n} h_i W_i^O \tag{6}$$

where $W_i^O \in \mathbb{R}^{d_{head}\times d}$. Therefore, the matrices W_Q, W_K, W_V can be represented as a stack of matrices $W_Q = [W_Q^1, W_Q^2, ...W_Q^{n_{head}}]^T$, where $W_Q^i \in \mathbb{R}^{d\times d_{head}}$ would be the projection matrix to obtain the Q vector for the ith attention head q_i, and similarly for the K and V vectors. Moreover, we also have that $W_O = [W_O^1, W_O^2, ...W_O^{n_{head}}]$.

Hence, zero ablating the ith head directly translates to removing the W_Q^i, W_K^i, W_V^i and W_O^i matrices. On the other hand, mean ablating implies removing the previous matrices and adding the mean output vector across a reference distribution to the output of the pruned layer. In contrast to current ablating implementations used on MI works which are based on hooks, our pruning process truly removes the component, yielding the benefit of size and inference time reduction.

D Study of the effect of hyperparameters on other tasks

Figures 4 and 5 show the impact of α on the size and the accuracy vs. the size of the resulting submodel on the IOI task. Similarly, Figures 6 and 7 show the same results on the greater-than task. Overall, we can draw the same conclusions that we obtained on the acronyms task: First, for a fixed α, mean ablation yields smaller and therefore faster models than zero ablation. Second, the results when using zero ablation are considerably less consistent, as shown by the high error bars, specially on the accuracy vs. size plots. Finally, there is no difference between including MLPs or not at lower threshold levels α, implying that the pruning of irrelevant attention heads are prioritized over irrelevant MLPs.

Another interesting result can be found in Figure 7. Specifically, it can be seen that when performing zero ablation and including MLPs in the greater-than task, the performance abruptly increases to 100% when the resulting pruned model is around 50% of the original size, as well as approaching the 100% reduction size. This is most likely due to the fact that the KL divergence is not correlated with how the accuracy is computed: if the model always outputs the starting year 99, the accuracy will be 100% in every case, but these predictions would have a large KL divergence with

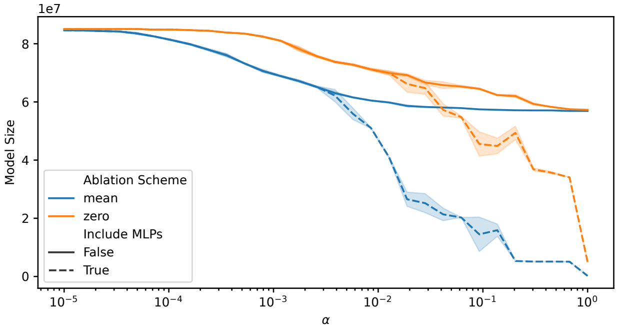

Figure 4: Impact of α in the resulting model size on the IOI task.

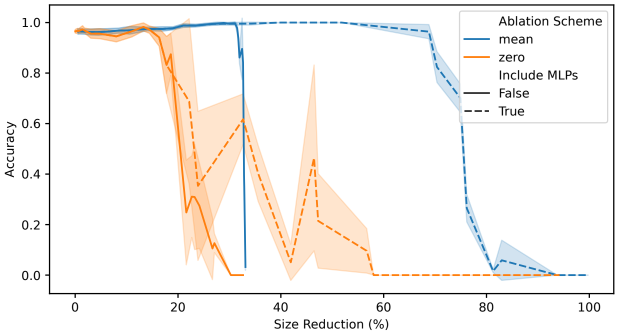

Figure 5: Accuracy vs. size of the resulting submodel on the IOI task.

respect to the original distribution. While this is an interesting phenomena to study, it is out of scope of this paper and leave it out for further research.

Barrier

E Extra Benchmark Results

Table 3 shows the extended results from the evaluation presented in Section 4.

Barrier

F Comparison to Manual Circuit Identification

Figure 8 shows the True Positive Rates (TPRs) and False Positive Rates (FPRs) for the discovered attention heads obtained by varying the threshold and applying the mean-ablation scheme. In general, we can see that our method is able to automatically recover most of the components. The results with lower AUC are obtained in the greater-than task. This is most likely due to the nature of the task under study. Differently from the other two tasks that we have analyzed, in the greater-than task there are more than one possible correct answer (i.e. any start year greater than the one in the sentence). The authors use a task-specific metric that takes this into account wereas we stick to the general KL divergence. However, our method is able to include the most relevant

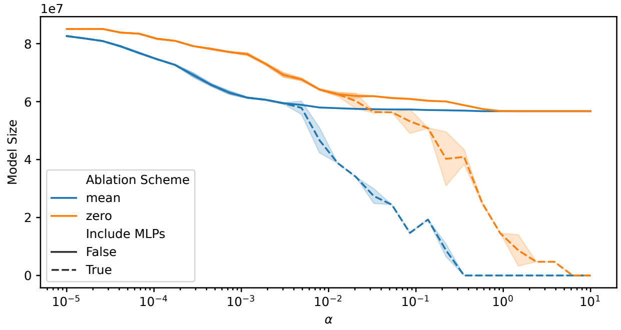

Figure 6: Impact of α in the resulting model size on the greater-than task.

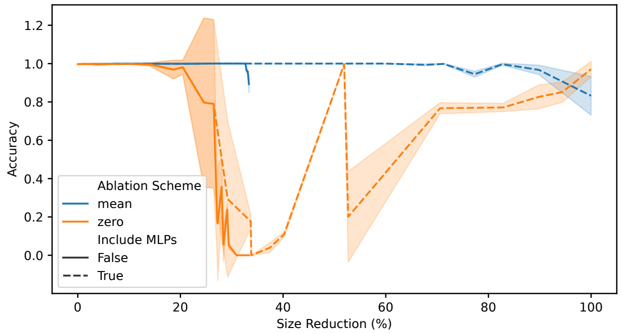

Figure 7: Accuracy vs. size of the resulting submodel on the greater-than task.

Table 3: Extended results of the evaluation of the pruned models obtained on each of the tasks for different values of α. The process is repeated across five different batches and the results are averaged.

Task	α	MLP	acc (%)	Δacc (%)	# param. (10^7)	$\Delta param$ (%)	t (ms)	Δt (%)
Acronyms	$8.86 \cdot 10^{-2}$	$False$	99.92 ± 0.17	9.28 ± 1.53	5.70 ± 0.00	32.88 ± 0.00	1.56 ± 0.02	83.56 ± 0.53
	$3.50 \cdot 10^{-2}$	$True$	78.64 ± 5.16	-12.00 ± 5.5	3.09 ± 0.26	63.70 ± 3.10	1.32 ± 0.12	86.20 ± 1.02
IOI	$8.53 \cdot 10^{-3}$	$False$	100.00 ± 0.00	2.93 ± 1.60	6.07 ± 0.04	28.66 ± 0.50	3.10 ± 0.65	57.58 ± 8.76
	$1.88 \cdot 10^{-2}$	$True$	96.53 ± 1.52	-1.73 ± 2.14	2.35 ± 0.25	72.31 ± 2.98	1.26 ± 0.05	82.78 ± 0.72
Greater than	$8.53 \cdot 10^{-2}$	$False$	100.00 ± 0.00	0.00 ± 0.00	5.73 ± 0.00	32.65 ± 0.00	1.79 ± 0.01	77.78 ± 0.15
	$8.53 \cdot 10^{-2}$	$True$	99.84 ± 0.36	-0.08 ± 0.04	1.47 ± 0.01	82.77 ± 0.13	0.94 ± 0.08	88.33 ± 1.00

components and the resulting submodel is able to maintain and even improve the accuracy with a considerably smaller size.

Figure 9 shows the results by using a zero-ablation scheme, instead of replacing by the mean activations. As expected, we can see that the AUCs are generally lower. This is due to the fact that the main objective in previous MI works was to identify the principal components that are responsible for a specific task. However, these components rely on other upstream (or lower-layer) components that might not be relevant for the task, but belong to other "sub-circuits" that support the main circuit. For example, in the acronym task, the authors found that some positional information used by the circuit was propagated by other upstream attention heads that did not belong to the circuit. As zero-ablation is equivalent to completely removing a component (i.e. it is more "aggressive"), the components that might not be directly in the circuit but in a supportive way are included.

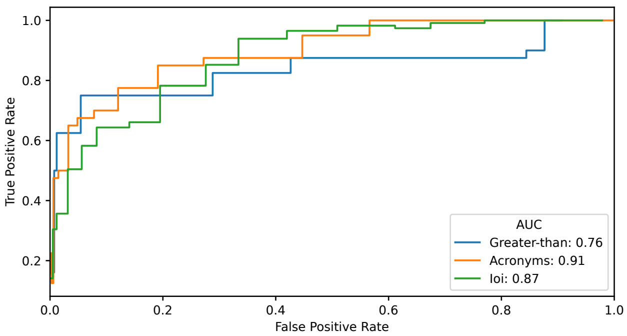

Figure 8: ROC curves of the identified attention heads for the three tasks of study. Mean-ablation is used.

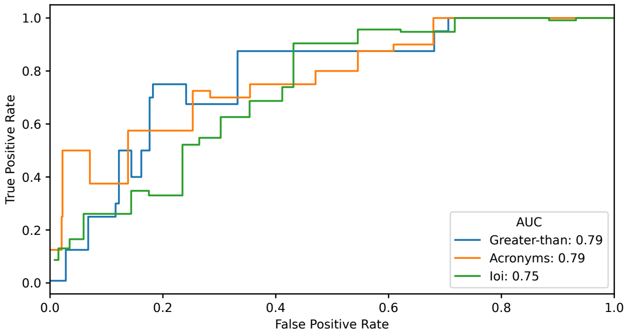

Figure 9: ROC curves of the identified attention heads for the three tasks of study. Zero-ablation is used.