Judea Pearl. Direct and indirect effects. In *Conference on Uncertainty and Artificial Intelligence (UAI)*, 2001.

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

Adam Scherlis, Kshitij Sachan, Adam S Jermyn, Joe Benton, and Buck Shlegeris. Polysemanticity and capacity in neural networks. *arXiv preprint arXiv:2210.01892*, 2022.

Paul Soulos, R Thomas McCoy, Tal Linzen, and Paul Smolensky. Discovering the compositional structure of vector representations with role learning networks. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 2020.

Alessandro Stolfo, Yonatan Belinkov, and Mrinmaya Sachan. Understanding arithmetic reasoning in language models using causal mediation analysis. *arXiv preprint arXiv:2305.15054*, 2023.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *International Conference on Learning Representations (ICLR)*, 2023.

Kaiyue Wen, Yuchen Li, Bingbin Liu, and Andrej Risteski. (Un)interpretability of transformers: a case study with Dyck grammars. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zhengxuan Wu, Atticus Geiger, Christopher Potts, and Noah D Goodman. Interpretability at scale: Identifying causal mechanisms in alpaca. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Zeyu Yun, Yubei Chen, Bruno Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: contextualized embedding as a linear superposition of transformer factors. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, 2021.

Ziqian Zhong, Ziming Liu, Max Tegmark, and Jacob Andreas. The clock and the pizza: Two stories in mechanistic explanation of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

A Review of Transformer Architecture

We follow the notation of Elhage et al. (2021) and give a review of the Transformer architecture (Vaswani et al., 2017). The input $x_0 \in \mathbb{R}^{N \times d}$ to a transformer model is a sum of position and token embeddings, where N is the sequence length and d is the dimension of the model's internal states. The input is the initial value of the residual stream which subsequently gets updated by the transformer blocks.

Each transformer block consists of a multi-head self-attention sublayer and an MLP sublayer. (For GPT-J, these two sublayers are parallelized.) The MLP sublayer is a two-layer feedforward network that processes each token position independently in parallel. Following Elhage et al. (2021), the

output of the attention sublayer can be decomposed into individual heads. For the ith layer, the attention output can be written as $y_i = \sum_{j=1}^{H} h_{i,j}(x_i)$, where $h_{i,j}$ denotes the jth attention head of the layer. Each head has four weight matrices, $W_Q^{i,j}, W_K^{i,j}, W_V^{i,j} \in \mathbb{R}^{d \times \frac{d}{H}}$ and $W_O \in \mathbb{R}^{\frac{d}{H} \times d}$. For a residual stream x, we refer to $Q^{i,j} = xW_Q^{i,j}, K^{i,j} = xW_K^{i,j}, V^{i,j} = xW_V^{i,j}$ as the query, key and value of the head. The attention pattern is given by

$$A^{i,j} = \text{softmax}\left(\frac{(xW_Q^{i,j})(xW_K^{i,j})^T}{\sqrt{d/H}} + M\right) \in \mathbb{R}^{N \times N},$$

where M is the attention mask. In auto-regressive language models, the attention pattern is masked to a lower triangular matrix. The output of the attention sublayer is given by

$$x + \text{Concat}\left[A^{i,1}V^{i,1}, \ldots, A^{i,j}V^{i,j}, \ldots, A^{i,H}V^{i,H}\right] W_O. \tag{1}$$

B Details on Experimental Settings

For Gaussian noise (GN) corruption, we corrupt the embeddings of the crucial tokens by adding a Gaussian noise $\varepsilon \sim \mathcal{N}(0; \nu)$, where ν is set to be 3 times the standard deviation of the token embeddings from the dataset (Meng et al., 2022).

We always perform GN and STR experiments in parallel. For STR, there is a natural the incorrect token r', since X_{corrupt} is also a valid in-distribution prompt. This allows for a well-defined metric of logit difference $\text{LD}(r, r') = \text{Logit}(r) - \text{Logit}(r')$. To make a fair comparison, the same r' is used for evaluating the logit difference metric under GN.

Throughout the paper, layers are zero-indexed, numbered from 0 to $L-1$ rather than 1 to L.

Factual recall To perform STR in the factual association setting, we construct PairedFacts, a dataset of 145 pairs of prompts. Within each pair, the two prompts have the same sequence length (under the GPT-2 tokenizer) but distinct answers. All the prompts are selected from the CounterFact and Known1000 datasets of Meng et al. (2022). On these prompts,

- GPT-2 XL achieves an average of 49.0% probability on the correct token and 6.85 logit difference.
- GPT-2 large achieves 41.1% and 5.88 logit difference.
- GPT-J achieves 50.1% and 7.36 logit difference.

A few samples of the PairedFacts dataset are listed in Figure 31 of Appendix I.

Since the prompts are perfectly symmetric and all of them are in-distribution, our STR experiments consist of both ways, where a prompt within a pair play the role of both X_{corrupt} and X_{clean}.

Our experiments with GN corruption is performed in the same manner as in Meng et al. (2022); Hase et al. (2023), with noise applied to all subject tokens' embeddings.

The experiments here are implemented via the TransformerLens library (Nanda & Bloom, 2022).

IOI Unless specified otherwise, GN applies Gaussian noise to the S2 token embedding. Over 500 prompts, the probability of outputting IO is $\mathbb{P}_*(r) = 0.129$ under GN corruption (with r being IO), whereas it is 0.481 under the clean distribution p_{IOI}.

All our experiments are performed using the original codebase of Wang et al. (2023), available at `https://github.com/redwoodresearch/Easy-Transformer`. The code provides the functionality of constructing X_{corrupt} under various definitions of corruptions, including STR.

C Results on Arithmetic Reasoning in GPT-J

Experimental setup We follow the setting of Stolfo et al. (2023) and perform localization analysis on the task of basic arithmetic in GPT-J (Wang & Komatsuzaki, 2021), a decoder-only model with

6B parameters. For simplicity, we consider addition, subtraction and multiplication up to 3 digits. We provide the model with a 2-shot prompts of the format

$$X_1 + Y_1 = Z_1$$
$$X_2 + Y_2 = Z_3$$
$$X_3 + Y_3 =$$

where the numbers X_i, Y_i are random integers and the operator can be $+, -, \times$. Stolfo et al. (2023) finds that this leads to improved accuracy. Since large integers get split into multiple tokens, we draw X_i, Y_i from $\{1, 2, \cdots, 250\}$ for addition and subtraction and from $\{1, 2, \cdots, 23\}$ for multiplication. To obtain a dataset for activation patching, we first draw 200 prompts and discard those on which the model's top-ranked output token is incorrect.

We set GN corruption to add noise to the token embeddings at the positions of X_3, Y_3. Similarly, STR replaces X_3, Y_3 by two random integers drawn from the same set, which ensures that the corrupted prompt is still in-distribution. We remark that Stolfo et al. (2023) applies the same STR corruption in their patching experiments.

Stolfo et al. (2023) devises a new metric to evaluate the patching effects. More precisely, they report:

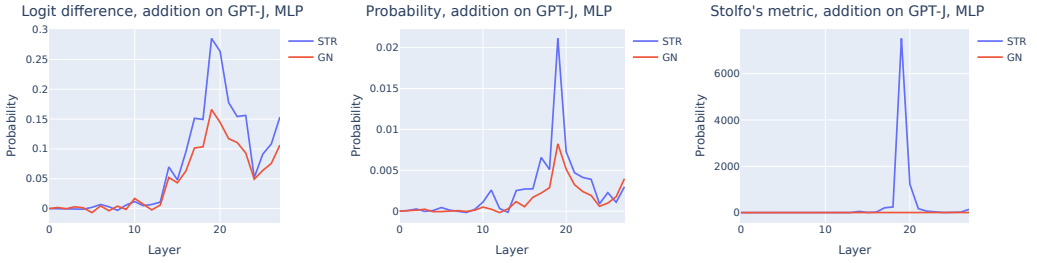
$$\frac{1}{2}\left[\frac{\mathbb{P}_{\text{pt}}(r) - \mathbb{P}_*(r)}{\mathbb{P}_*(r)} + \frac{\mathbb{P}_*(r') - \mathbb{P}_{\text{pt}}(r')}{\mathbb{P}_{\text{pt}}(r')}\right] \tag{2}$$

from patching the MLP activation at last token of the prompt.[4] We compute the patching effect given by the metric, as well as probability and logit difference.

Following Stolfo et al. (2023), we narrow our focus on localization of MLP layers. All the experiments patch a single MLP layer's activation at the last token of the prompt.

Experimental results Focused on the logit difference and probability metric, we observe gaps between GN and STR for addition and subtraction. In particular, STR is found to provide sharper concentration, up to a magnitude of 4×. This in contrast with our results on factual association (Section 3.1), where GN appears to induce stronger peak. For multiplication, GN and STR provides nearly matching results. This highlights that activation patching can be sensitive to corruption methods in a rather unpredictable way. See Figure 6, Figure 7 and Figure 8 for plots.

For the metric (2) of Stolfo et al. (2023), we qualitatively replicate their results, similar to Figure 2 of their paper, and find extremely pronounced peak with STR corruption. Towards understanding this observation, we examine the quantity (2) closely and discover that its first term typically dominates the second. This, in turn, is because the denominator term $\mathbb{P}_*(r)$, the probability of outputting the correct answer in the corrupted run, is usually tiny under STR corruption. The small denominator, therefore, acts as a large multiplier that amplifies the absolute gap between patching different layers. We note that this effect is much smaller under GN since $\mathbb{P}_*(r)$ is usually not negligible.



(a) Logit difference as the metric (b) Probability as the metric (c) Stolfo et al. (2023)'s metric

Figure 6: **The effects of patching MLP layers** in GPT-J on addition prompts.

[4]Note that our notations here are different from Stolfo et al. (2023).