A Limitations

We note a few limitations of the experiments conducted in this paper:

(1) We work only with the CounterFact and ZSRE datasets, which we use as short English prompts with factual completions corresponding to a specific set of relations between subject and object entities. This is a basic form of factual knowledge, and localization and editing analysis may yield different trends for other forms of knowledge.

(2) We work with two autoregressive Transformers chosen for their representativeness of large language models that show a capacity for expressing factual knowledge in response to natural language prompts. However, the conclusions from our analysis may not generalize to models larger than GPT-J (6B parameters) that are known to exhibit phase changes in their behavior under prompting.

(3) We use a particular set of localization and editing methods, including representation denoising and zero-ing at the layer level and layer-level MLP editing methods that inject new facts or amplify or erase existing facts. Our conclusions may not necessarily hold for the breadth of localization and editing methods from work related to this paper, and one should be cautious in applying our conclusions beyond our experimental setting.

B Broader Impacts

It is possible that increased mechanistic understanding of models improves our ability to edit them at some point in the future. In fact, we consider it unlikely that interpretability results never give insight into improving model editing methods. Thus, to the extent that model editing is a dual use methodology, which could be used to inject harmful beliefs or dangerous knowledge into models, interpretability results may enhance the effectiveness of these malicious use cases. However, these concerns are relatively far removed from our analysis, which focuses on the connection between localization and editing performance.

C Experiment Details

Data Filtering. We filter the CounterFact dataset to a subset of facts that are correctly completed by GPT-J, in order to ensure that there is knowledge to localize in the model for each point. We mark a completion correct when o_{true} appears among the first 36 tokens sampled from the model given the prompt P using greedy decoding. GPT-J achieves a completion accuracy of 32.6% under this scheme, and after starting with about 10% of the CounterFact dataset, our final sample size is $n = 652$. We perform additional filtering specifically for model editing in the Fact Erasure condition, where we filter points to have a target probability $p_\theta(o_{true}|s, r)$ of at least .02, so that there is a reasonable amount of probability mass to be erased. In this condition, we have $n = 489$ points.

Compute. Experiments were run on a single NVIDIA A6000 GPU with 48gb memory. Computing editing performance for $n = 652$ points with GPT-J for a single edit method applied across model layers in the set {1, 5, 9, 13, 17, 21, 24, 28} could take about eight hours. Saving causal tracing or representation zeroing results for these datapoints takes about twelve hours. Regression analyses and plots can be made on demand (code in supplement) given the data from the editing and localization experiments.

Edit Method Tuning. We tune the edit methods to have high rewrite scores while not trading off too aggressively against paraphrase and neighborhood scores. More specifically, this means we tune methods to have rewrite scores no higher than 99% (note methods can easily get above 99% rewrite score), separately for each editing problem variant. The tuning is done with the first 100 points of the CounterFact dataset, editing layer 6 for GPT-J and 18 for GPT2-XL. For ROME and MEMIT methods, we tune over the KL regularization weight values in the set {.0625, .9, 1}. For constrained finetuning, we tune over the L_∞ norm weight values in the set {1e-4, 5e-5, 2e-5, 1e-5}. For both methods, we adopt default parameters from Meng et al. [22] unless otherwise stated. We describe the relevant hyperparameters below, for GPT-J first:

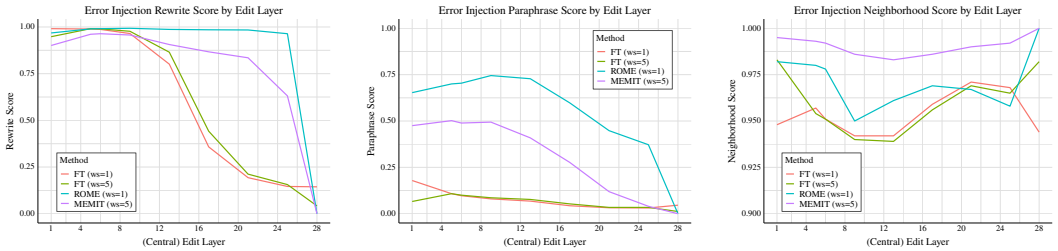

Figure 7: Edit success metrics for our four editing methods, under the Error Injection objective. Left: Rewrite, Center: Paraphrase, Right: Neighborhood.

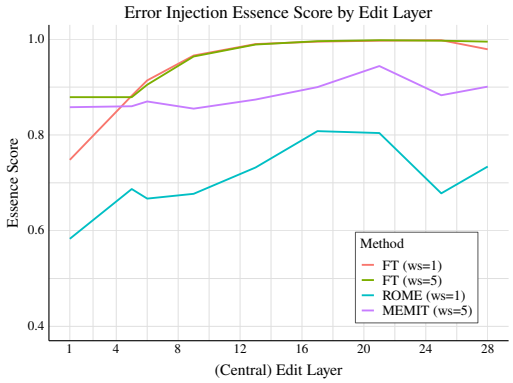

Figure 8: Essence score by edit layer, for our four editing methods, under the Error Injection objective.

1. *Error Injection*. FT-1: norm constraint of 1e-4. FT-5: norm constraint of 2e-5. ROME: regularization weight of 1. MEMIT: regularization weight of 0.9.
2. *Tracing Reversal*. FT-1: Norm constraint of 1e-5. FT-5: Norm constraint of 2e-5. FT-5: 2e-5. ROME: default parameters. MEMIT: default parameters.
3. *Fact Erasure*. FT-1: norm constraint of 1e-5. FT-5: norm constraint of 1e-5. ROME: default parameters. MEMIT: default parameters.
4. *Fact Amplification*. FT-1: norm constraint of 1e-5. FT-5: norm constraint of 1e-5. ROME: default parameters. MEMIT: default parameters.
5. *Fact Forcing*. Note that for all methods we decide to increase the number of gradient steps, as convergence takes longer for finetuning (from 25 to 50 steps) and for the gradient-based optimization for v^* in ROME (from 20 to 25 steps). FT-1: norm constraint of 1e-4. FT-5: norm constraint of 1e-4. ROME: 25 gradient steps for finding v^*. MEMIT: default parameters (already set to 25 steps).

We run only the Error Injection and Fact Forcing conditions for GPT2-XL. Hyperparameters are as follows:

1. *Error Injection*. FT-1: norm constraint of 1e-3. FT-5: norm constraint of 1e-4. ROME: default parameters. MEMIT: default parameters.
2. *Fact Forcing*. FT-1: norm constraint of 5e-4. FT-5: norm constraint of 5e-5. ROME: default parameters. MEMIT: default parameters.

D Additional Results

ZSRE Dataset. Here, we describe experiments with the ZSRE dataset, which is commonly used in past editing method papers [9, 24]. ZSRE includes naturalistic questions rather than prompts intended for autoregressive cloze completion, as in CounterFact. Following past work [21], we use GPT-J to answer ZSRE questions in a zero-shot manner, and we edit the model with ROME. We

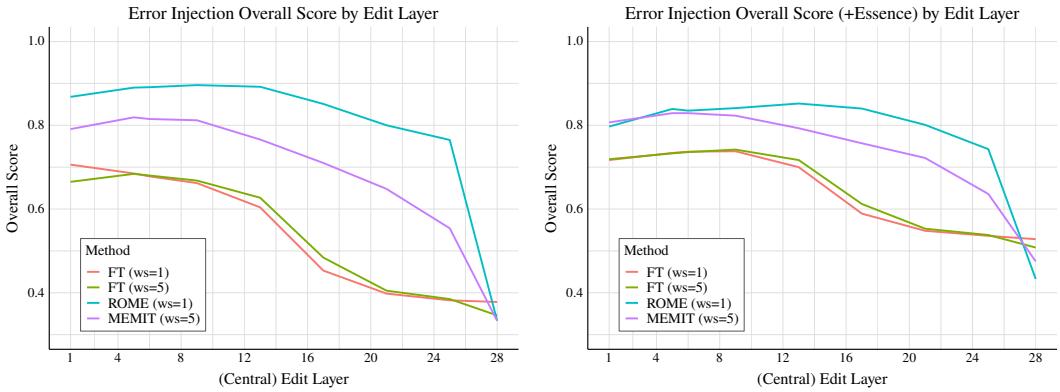

Figure 9: Overall edit success for our four editing methods, under the Error Injection objective. Left: The mean of Rewrite, Paraphrase, and Neighborhood Scores. Right: the mean score with Essence Score included.

report results for ZSRE via plots of edit success vs. tracing effect in Figs. 19 (rewrite score) and 20 (overall score), accompanied by regression analysis results in Table 8. We find that results with ZSRE match our conclusions with CounterFact, as the results are quite similar to plots and regressions with CounterFact data. Tracing effects are not predictive of edit success.

Representation Zeroing. Representation zeroing is a common localization technique where neural activations are manually set to zero during a model forward pass [18, 2]. We implement a form of representation zeroing that is exactly like Causal Tracing, except instead of denoising already-noised representations, we set clean representations to zero. Specifically, we simply run a normal forward pass until a certain set of layers (window size=5), where we zero out representation values for the MLP output representations at the subject token indices within those layers (then continue the forward pass). The localization effect is computed as the proportion of the original predicted probability that is deleted via the zero-ing operation (ranging from no effect as 0% to 100% of probability deleted as 100%). These new results are shown in Figs. 21 for rewrite score and 22 for overall score, using ROME on GPT-J with CounterFact data. We obtain the same conclusions as our analysis with causal tracing: localization via representation zero-ing is not predictive of edit success. Specifically, we see correlations between edit success and localization effect to be near zero across layers (using either rewrite score or overall score for edit success).

Highly concentrated tracing effects. Since Causal Tracing analysis suggests that information accrues gradually across layers (see Fig. 10), it seems possible that information is simply so diffusely spread across model layers that no matter what layer you edit, you will be editing a layer where a fact is at least stored in part. Based on this observation, we want to test whether tracing effects correlate better with edit success specifically when tracing effects are concentrated in a small number of layers. This condition represents that a fact appears to be stored in a small number of layers *and not elsewhere*. We hope that by editing in that range of layers, we can more easily manipulate that fact. To identify points with *concentrated* tracing effects, we use a heuristic for filtering points. Given the output of Causal Tracing analysis for a point, i.e. one effect per layer (the max across tokens), we define the point to have concentrated tracing effects when there are no more than three layers that have at least 50% of the maximum effect across layers (besides the layer with the max effect itself). Under this criterion, about 10% of the data (74 of 652 cases) have concentrated effects. Note we use our default tracing window size of 5 with the 28 layer GPT-J model for this experiment.

We show the results from our analysis on this data subset in Table 2, and **we observe no changes** in our main conclusions. For ROME with Error Injection, the added effect is 0.2%. Across editing problems and edit methods, the maximum added effect of including tracing effects on R^2 values for predicting rewrite score remains at 3.2% (for Fact Forcing with constrained finetuning). Thus, we conclude that even when facts appear to be stored in a small number of layers, localization results from Causal Tracing are still not informative about editing success, while the choice of edit layer is a far more important factor in whether a fact is successfully edited.

Table 2: R^2 values for predicting ROME edit success in Error Injection, subsetted to 10% of the data that has the most concentrated tracing effects in a small number of layers. Even when facts appear to be stored at a small number of layers *and not other layers*, tracing effects are still not predictive of editing performance.

Concentrated Data	R^2 Values		
Method	Layer	Tracing Effect	Both
ROME	0.927	0.02	0.929

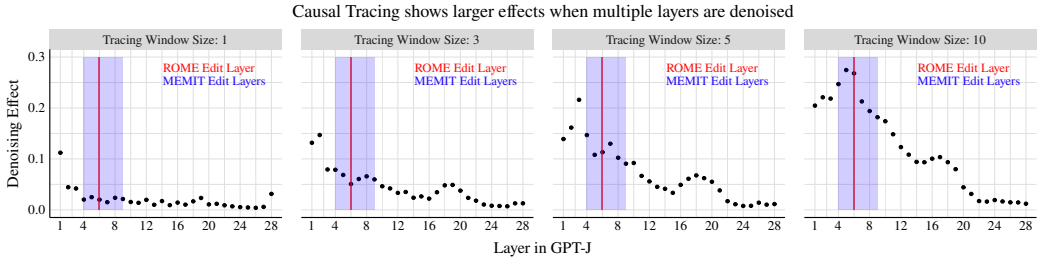

Figure 10: Tracing effects grow larger as the number of adjacent restored layer representations increases (tracing window size).

Measuring essence drift. Meng et al. [21] describe one possible consequence of model editing as *essence drift*, which occurs when core properties of an entity change after attempting to edit only one property of that entity. For example, changing where an island is located might also cause the model to nonsensically treat the island as a university campus (see example in Meng et al. [21]).

We aim to obtain an automatic metric to serve as a rough proxy for essence drift. A related metric is calculated with "Local Neutral" data involving the same subject entity but with other properties that are logically neutral with the original property of the subject being edited [15]. However, we do not have "Local Neutral" data for the CounterFact dataset, and essence drift aims to specifically measure changes to *core* properties of a subject.

Therefore, we automatically estimate changes to known properties of the subject s by calculating the change in model perplexity over samples of text that were drawn from the pre-edit model given the prompt "s is a " (which tend to describe a number of key properties of the subject s). We term these samples *essence texts*, and we obtain five samples per subject prompt by sampling with multinomial top-k sampling using $k = 5$. Given our essence texts, we measure the perplexity over the samples before and after editing a fact in the model, for every edited fact in our dataset. Note this is quite similar to the essence drift regularization objective used in the ROME optimization objective [21], but we consider it as a metric here. We scale the change in perplexity to a fraction of 5, with the cut-off of 5 chosen to represent a maximally bad change to the model perplexity. Similar to our other metrics, our essence score is 1 if model perplexity on the essence texts does not change after editing the model (capping to 1 in cases of slight decreases in perplexity), and it is 0 if the perplexity increases by 5 or more.

We show essence scores for editing methods across layers in 8. Interestingly, the trend across layers for this metric is mostly counter to the trends for other metrics (Fig. 7), with editing later layers being generally preferable to editing earlier layers. As a result, when combined with the other metrics in Fig. 9, we see that the overall score trend flattens and shifts slightly toward mid-range layers in the model.

E Robustness Experiments

In addition to our main results with ROME for GPT-J and our Rewrite Score metric, we include robustness experiments to confirm that results are similar for (1) other measures of edit success including Paraphrase Score, Neighborhood Score, and Overall Score (Tables 4, 5, and 6), (2) different values of the tracing window size (Fig. 12), (3) GPT2-XL rather than GPT-J (Fig. 13), (4) the original unscaled metrics from Meng et al. [21] (Fig. 14), and (5) using the tracing effect at the last subject

Rewrite Score Table		R^2 Values				
Editing Problem	Method	Layer	Trace	Both	Diff	p-value
Error Injection	FT (1 layer)	0.756	0.062	0.758	0.002	<1e-4
	FT (5 layers)	0.775	0.055	0.777	0.002	<1e-4
	ROME (1 layer)	0.947	0.016	0.948	0.001	<1e-4
	MEMIT (5 layers)	0.677	0.024	0.678	0.001	0.199
Tracing Reversal	FT (1 layer)	0.067	0	0.067	0	0.997
	FT (5 layers)	0.751	0.045	0.752	0.001	0.032
	ROME (1 layer)	0.294	0.017	0.31	0.015	<1e-4
	MEMIT (5 layers)	0.212	0.036	0.218	0.006	<1e-4
Fact Erasure	FT (1 layer)	0.643	0.028	0.646	0.003	<1e-4
	FT (5 layers)	0.698	0.025	0.70	0.002	<1e-4
	ROME (1 layer)	0.857	0.019	0.858	0	0.555
	MEMIT (5 layers)	0.925	0.019	0.925	0	0.669
Fact Amplification	FT (1 layer)	0.383	0.014	0.393	0.01	<1e-4
	FT (5 layers)	0.424	0.01	0.436	0.011	<1e-4
	ROME (1 layer)	0.88	0.02	0.88	0	0.654
	MEMIT (5 layers)	0.905	0.018	0.906	0.001	<1e-4
Fact Forcing	FT (1 layer)	0.697	0.104	0.724	**0.027**	<1e-4
	FT (5 layers)	0.634	0.10	0.666	**0.032**	<1e-4
	ROME (1 layer)	0.422	0.004	0.425	0.003	<1e-4
	MEMIT (5 layers)	0.345	0.041	0.354	0.009	<1e-4

Table 3: R^2 values for predicting **rewrite** score from choice of edit layer and tracing effect, across editing problem variants (corresponds to data in Fig. 6). Diff shows the added effect of including tracing in the regression (Both vs. Layer Only), in terms of R^2, and p-value shows the results from an F-test comparing the Both and Layer Only models. Tracing has some predictive value for Fact Forcing, but the R^2 value remains small compared to the choice of edit layer.

Table 4: R^2 values for predicting **paraphrase** score from choice of edit layer and tracing effect, across editing problem variants. Diff shows the added effect of including tracing in the regression (Both vs. Layer Only), in terms of R^2, and p-value shows the results from an F-test comparing the Both and Layer Only models. The added effect of including tracing effects is very small across conditions (less than 3%).

Paraphrase Score Table		R^2 Values				
Editing Problem	Method	Layer	Trace	Both	Diff	p-value
Error Injection	FT (1 layer)	0.061	0.005	0.063	0.002	0.258
	FT (5 layers)	0.036	0.003	0.038	0.001	0.582
	ROME (1 layer)	0.279	0.001	0.303	0.024	<1e-4
	MEMIT (5 layers)	0.246	0	0.269	0.023	<1e-4
Tracing Reversal	FT (1 layer)	0.004	0.001	0.004	0	0.989
	FT (5 layers)	0.001	0	0.002	0.001	0.841
	ROME (1 layer)	0.01	0	0.012	0.002	0.121
	MEMIT (5 layers)	0.001	0	0.001	0	0.997
Fact Erasure	FT (1 layer)	0.046	0.001	0.048	0.002	0.303
	FT (5 layers)	0.079	0.007	0.084	0.005	0.004
	ROME (1 layer)	0.537	0.012	0.539	0.001	0.218
	MEMIT (5 layers)	0.586	0.015	0.587	0.001	0.184
Fact Amplification	FT (1 layer)	0.005	0.012	0.022	0.017	<1e-4
	FT (5 layers)	0.017	0.013	0.035	0.018	<1e-4
	ROME (1 layer)	0.24	0.002	0.267	0.027	<1e-4
	MEMIT (5 layers)	0.236	0.001	0.263	0.026	<1e-4
Fact Forcing	FT (1 layer)	0.044	0.004	0.046	0.002	0.367
	FT (5 layers)	0.023	0.002	0.025	0.002	0.387
	ROME (1 layer)	0.357	0.01	0.36	0.003	0.003
	MEMIT (5 layers)	0.095	0.001	0.105	0.01	<1e-4

Table 5: R^2 values for predicting **neighborhood** score from choice of edit layer and tracing effect, across editing problem variants. Diff shows the added effect of including tracing in the regression (Both vs. Layer Only), in terms of R^2, and p-value shows the results from an F-test comparing the Both and Layer Only models. The added effect of including tracing effects is very small across conditions (2% or less).

Neighborhood Score Table		R^2 Values				
Editing Problem	Method	Layer	Trace	Both	Diff	p-value
Error Injection	FT (1 layer)	0.005	0	0.008	0.002	0.197
	FT (5 layers)	0.014	0.001	0.015	0.001	0.55
	ROME (1 layer)	0.011	0.003	0.015	0.005	0.001
	MEMIT (5 layers)	0.004	0.001	0.006	0.002	0.154
Tracing Reversal	FT (1 layer)	0.001	0	0.001	0	1
	FT (5 layers)	0.001	0	0.002	0.001	0.946
	ROME (1 layer)	0.001	0	0.002	0.001	0.946
	MEMIT (5 layers)	0.001	0	0.002	0	0.981
Fact Erasure	FT (1 layer)	0.01	0	0.014	0.004	0.037
	FT (5 layers)	0.01	0	0.013	0.004	0.06
	ROME (1 layer)	0.04	0.005	0.046	0.006	0.001
	MEMIT (5 layers)	0.05	0.007	0.059	0.009	<1e-4
Fact Amplification	FT (1 layer)	0.012	0.009	0.02	0.008	<1e-4
	FT (5 layers)	0.016	0.008	0.025	0.009	<1e-4
	ROME (1 layer)	0.04	0.01	0.05	0.01	<1e-4
	MEMIT (5 layers)	0.035	0.008	0.044	0.01	<1e-4
Fact Forcing	FT (1 layer)	0.054	0	0.057	0.003	0.03
	FT (5 layers)	0.019	0.001	0.022	0.004	0.011
	ROME (1 layer)	0.299	0.022	0.311	0.012	<1e-4
	MEMIT (5 layers)	0.046	0.012	0.066	0.02	<1e-4

Table 6: R^2 values for predicting **overall** score (raw average of rewrite, paraphrase, and neighborhood scores) from choice of edit layer and tracing effect, across editing problem variants. Diff shows the added effect of including tracing in the regression (Both vs. Layer Only), in terms of R^2, and p-value shows the results from an F-test comparing the Both and Layer Only models. The added effect of including tracing effects is very small across conditions (2% or less).

Overall Edit Score Table		R^2 Values				
Editing Problem	Method	Layer	Trace	Both	Diff	p-value
Error Injection	FT (1 layer)	0.642	0.054	0.643	0.002	0.001
	FT (5 layers)	0.663	0.047	0.665	0.002	0.001
	ROME (1 layer)	0.62	0.003	0.629	0.009	<1e-4
	MEMIT (5 layers)	0.525	0.008	0.534	0.009	<1e-4
Tracing Reversal	FT (1 layer)	0.294	0.025	0.296	0.002	0.054
	FT (5 layers)	0.751	0.045	0.752	0.001	0.032
	ROME (1 layer)	0.296	0.016	0.31	0.014	<1e-4
	MEMIT (5 layers)	0.21	0.036	0.216	0.006	<1e-4
Fact Erasure	FT (1 layer)	0.28	0.007	0.283	0.004	0.008
	FT (5 layers)	0.119	0	0.124	0.004	0.015
	ROME (1 layer)	0.718	0.023	0.718	0	0.729
	MEMIT (5 layers)	0.794	0.025	0.794	0	0.555
Fact Amplification	FT (1 layer)	0.188	0.003	0.199	0.011	<1e-4
	FT (5 layers)	0.224	0.002	0.236	0.013	<1e-4
	ROME (1 layer)	0.583	0.005	0.59	0.007	<1e-4
	MEMIT (5 layers)	0.597	0.005	0.607	0.01	<1e-4
Fact Forcing	FT (1 layer)	0.487	0.056	0.5	0.013	<1e-4
	FT (5 layers)	0.459	0.057	0.475	0.017	<1e-4
	ROME (1 layer)	0.285	0.004	0.291	0.006	<1e-4
	MEMIT (5 layers)	0.226	0.017	0.227	0.001	0.419

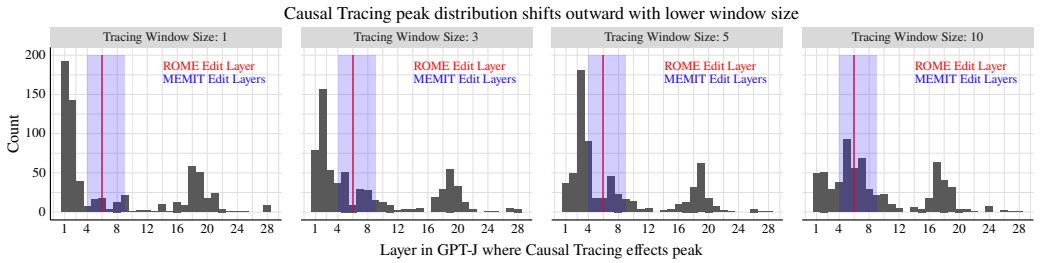

Figure 11: Each individual plot shows the distribution of tracing curve peaks (the argmax layer) across datapoints, using a different tracing window size. Together, the plots show how the distribution of layers where the tracing curves peak for each point shifts outward toward the first and last layer of the model as the tracing window size declines. This is primarily due to a clipping effect from using a window size greater than 1. The way tracing values are computed, a window size of 10 implies that the effect for "layer 1" is from restoring layers 1-5, while the effect for layer "layer 5" is 1-10. As a result, a tracing window size of 10 favors layer 5 over layers 1-4, and reducing the tracing window size leads to these clumps of effects shifting from layer 5 toward layer 1 (and from layer 24 to layer 28)

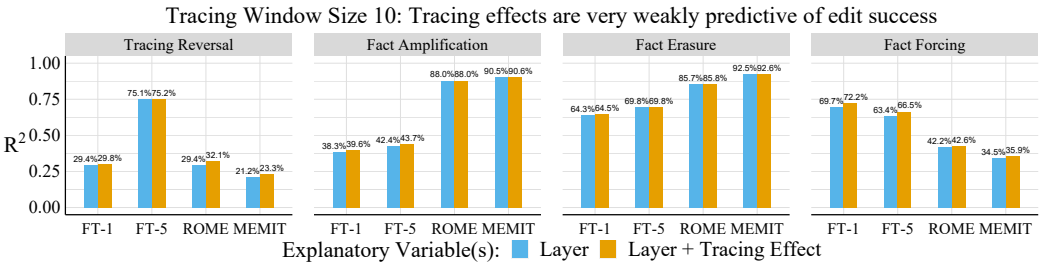

Figure 12: The results of our R^2 analysis for predicting rewrite score are nearly identical between using a tracing window size of 5 (shown in Fig. 6) or 10 (shown here).

token rather than the max across tokens (Fig. 16). We consider the last subject token effect since this corresponds more directly to the motivation for ROME (see Meng et al. [21]). We expand on each of these experiments below:

Results for Paraphrase, Neighborhood, Overall Metrics. We recreate our regression-based analysis across editing problem variants and editing methods using paraphrase score and neighborhood score as our outcomes rather than Rewrite Score, as well as an Overall Score that is the raw average of the three edit scores. These results are shown in Tables 4, 5, and 6 respectively. Similar to our analysis with rewrite score, these tables show that tracing effects are barely predictive of edit success at all. For paraphrase score, the largest gains in R^2 values are around 0.03 (relative to the layer-only regression model), and for neighborhood score, the largest gain is 0.02. The largest gain for overall score is 0.02 for Fact Forcing with constrained finetuning. Our overall conclusion remains that tracing effects are almost totally unrelated to edit success across editing problem variants, including for different edit success metrics.

Results for Different Tracing Window Sizes. We repeat our analysis from Sec. 5 using tracing effects obtained from a larger tracing window size of 10, to match the value used in Meng et al. [21]. Note that from Fig. 10, we know that the tracing effects grow larger as more adjacent layer representations are restored. When we recreate our main R^2 analysis using tracing effects with window size 10 (shown in Fig. 12), we find that results are nearly identical to those shown in Tables 3, 4, and 5.

Results for GPT2-XL. We rerun our analysis with GPT2-XL, a 48 layer model [30], while editing layers in the range {1, 5, 9, 13, 17, 18, 21, 25, 29, 33, 37, 41, 45, 48}. Here, we use a tracing window size of 10, and we limit our experiments to focus on Error Injection and Fact Forcing editing problems. As seen in Fig. 13, we find very similar trends when explaining rewrite score in terms of the choice of edit layer and the tracing effect at that layer. The largest explanatory effects in terms of R^2 are observed for Fact Forcing with constrained finetuning, but these effects remain small at about 2%.

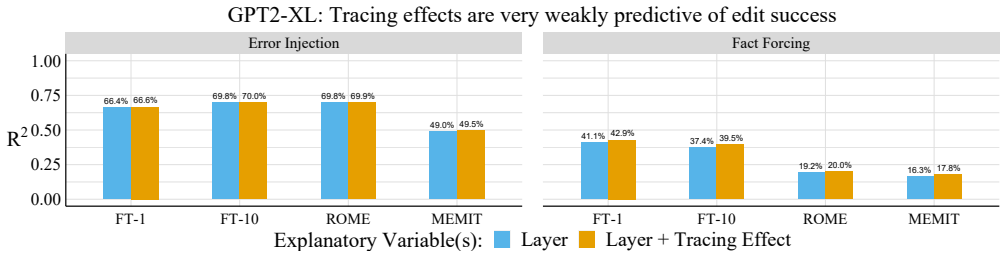

Figure 13: Like with GPT-J, tracing effects are very weakly predictive of edit success across editing problem variants for **GPT2-XL** while Fact Forcing shows the largest relationship. Relative to the R^2 of a model predicting rewrite score based on the choice of edit layer (blue), a model with edit layer and tracing effects (orange) improves the R^2 by at most .02 points for Fact Forcing. The choice of edit layer explains a much greater share of the variance in rewrite score.

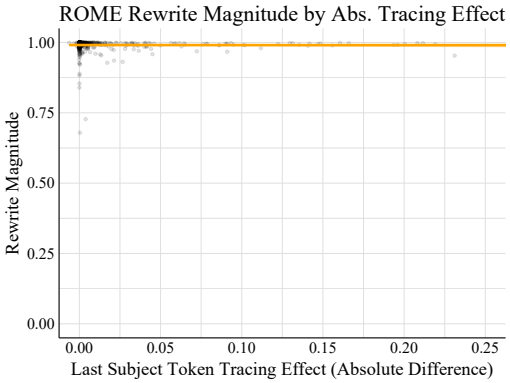

Figure 14: Editing vs. tracing results for ROME at layer 6 for Error Injection, using the un-rescaled rewrite and tracing metrics from Meng et al. [21]. Here, rewrite magnitude is the difference between the probability of the new target o_{false} and the old true target o_{true} after editing, $p_{\theta^*}(o_{false}|s,r) - p_{\theta^*}(o_{true}|s,r)$. The tracing effect is the absolute tracing effect, $p_\theta(o_{true}|s_{noise}, r, v_{(t,\ell)}) - p_\theta(o_{true}|s_{noise}, r)$, measured at the last subject token index. The correlation here is near zero, at $\rho = -.006$.

Results for Unscaled Metrics. We repeat our analysis using the original editing metrics and absolute tracing effects from Meng et al. [21]. Their rewrite magnitude is the absolute difference between the probability of the new target o_{false} and the old true target o_{true} after editing, $p_{\theta^*}(o_{false}|s,r) - p_{\theta^*}(o_{true}|s,r)$. The tracing effect is the absolute tracing effect, $p_\theta(o_{true}|s_{noise}, r, v_{(t,\ell)}) - p_\theta(o_{true}|s_{noise}, r)$, measured at the last subject token index. We adjusted our rewrite and tracing metrics to (1) rely only on the target output probability, rather than difference in probabilities of two different targets which might not be appropriate for our different editing problems, and (2) to always fall between 0 and 1 for better comparability between datapoints, since absolute tracing effect are bounded by the original model probabilities. However, we reach the same conclusions from our analysis when using the original editing metrics. We show an example for rewrite magnitude and the absolute tracing effect for Error Injection in Fig. 14. The correlation between edit success and tracing effect is still near zero.

Results for Last Subject Token Effect. ROME increases the target probability $p(o_{false}|s,r)$ by optimizing for a new output representation from a chosen MLP layer *at the last subject token index*. Meng et al. [21] show that this choice of token representation is critical to the success of the editing method, which is a hypothesis directly motivated by the results from their Causal Tracing analysis. In our paper, we by default report results using tracing effects that are the max across tokens at a given layer, for better comparability across the editing methods we use. However, when we repeat our analysis using the tracing effect specifically at the last subject token index, we obtain the same negative conclusions about the relationship between Causal Tracing localization and ROME editing performance. We show the correlations between Rewrite Score and Last Subject Token Tracing Effect

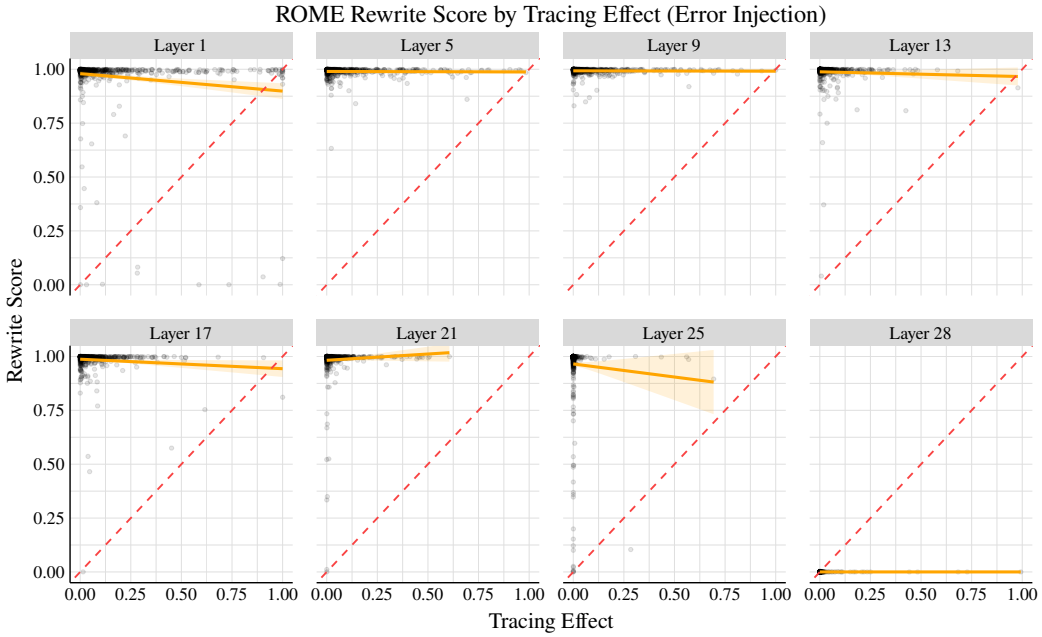

Figure 15: The relationship between ROME edit success and the tracing effect is near zero at most edit layers in the model (for the standard Error Injection editing problem). Red lines show perfect relationships between tracing effect and edit success.

Edit Metric	Regression Metric	Predictor(s)	Value
Rewrite Score	R^2	Layer	0.947
		Tracing Effect	0.016
	RMSE	Layer	0.073
		Tracing Effect	0.315
	MAE	Layer	0.02
		Tracing Effect	0.206
Overall Score	R^2	Layer	0.618
		Tracing Effect	0.003
	RMSE	Layer	0.133
		Tracing Effect	0.216
	MAE	Layer	0.11
		Tracing Effect	0.183

Table 7: **Additional regression error metrics** (for CounterFact and ROME) lead us to the same conclusion as our analysis based on R^2. RMSE is root mean squared error, and MAE is mean absolute error. Regressions predicting rewrite score (or overall score) from the choice of edit layer achieve much lower prediction errors than regressions using the tracing effect, suggesting that the choice of edit layer is much more important for edit success than the tracing effect.

in Fig. 16, where we see there are no positive correlations between editing success and tracing results at any layer in GPT-J.

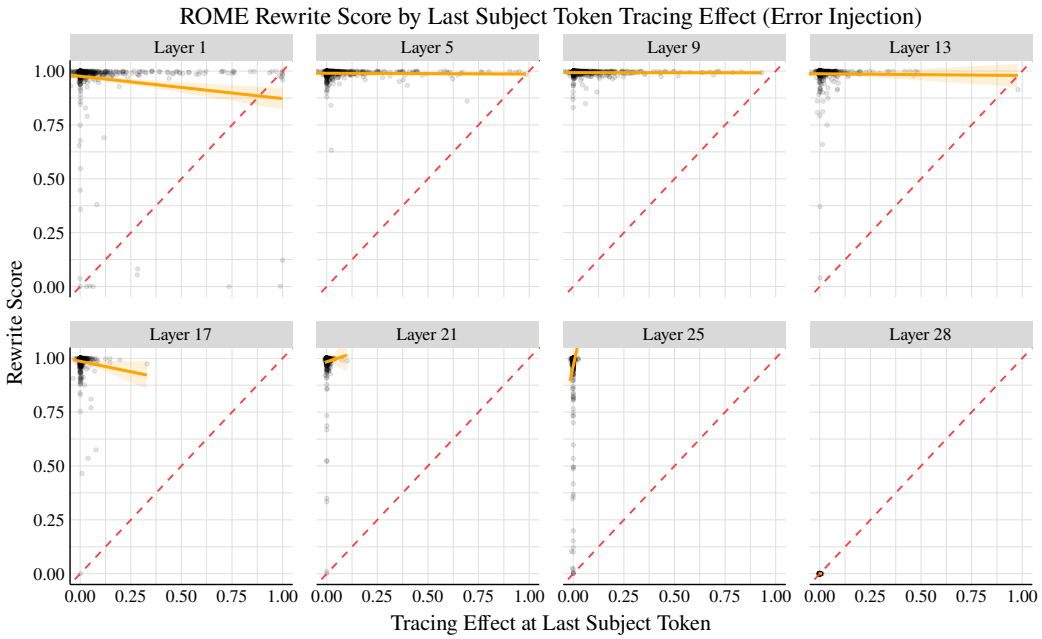

Figure 16: The relationship between ROME edit success and the tracing effect at the last subject token. The ROME method edits a fact by changing the output representation for the MLP layer specifically at the token index corresponding to the last subject token. However, editing performance and tracing effect at this position still do not positively correlate. Note the distribution of points along the x axis changes depending on the choice of edit layer since the distribution of tracing effects is calculated from tracing effects *at that layer*.

Edit Metric	Regression Metric	Predictor(s)	Value
Rewrite Score	R^2	Layer	0.795
		Tracing Effect	0.042
	RMSE	Layer	0.158
		Tracing Effect	0.341
	MAE	Layer	0.072
		Tracing Effect	0.254
Overall Score	R^2	Layer	0.654
		Tracing Effect	0.059
	RMSE	Layer	0.136
		Tracing Effect	0.223
	MAE	Layer	0.097
		Tracing Effect	0.188

Table 8: **ZSRE regression results** lead us to the same conclusion as our experiments on CounterFact, using ROME editing. RMSE is root mean squared error, and MAE is mean absolute error. Regressions predicting rewrite score (or overall score) from the choice of edit layer achieve much lower prediction errors than regressions using the tracing effect, suggesting that the choice of edit layer is much more important for edit success than the tracing effect.

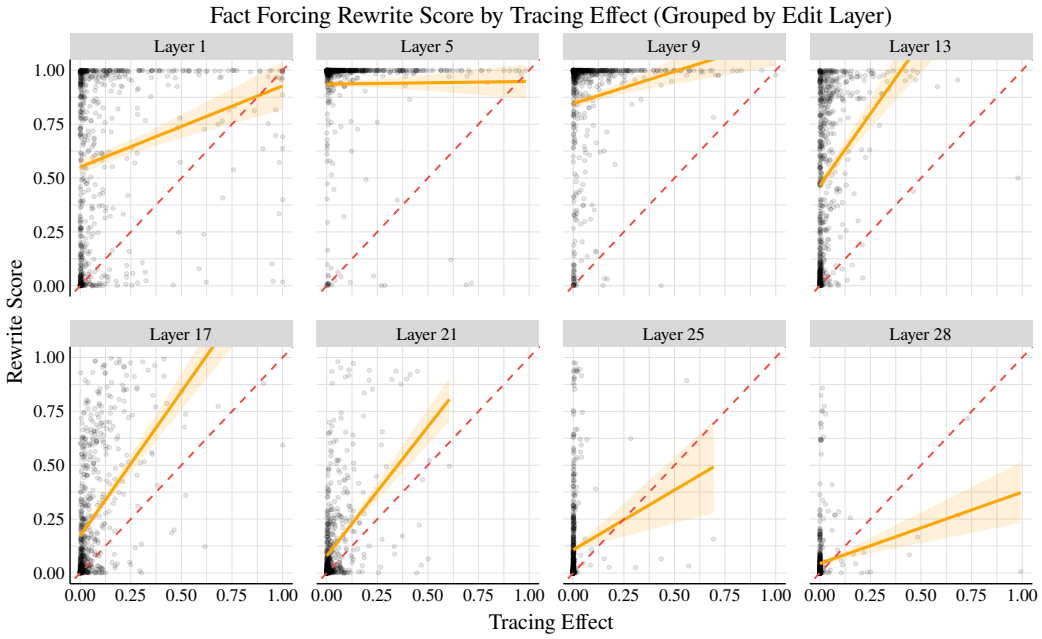

Figure 17: The relationship between Fact Forcing edit success and the tracing effect for constrained finetuning of 5 adjacent layers. "Layer ℓ" indicates the center of this 5-layer interval, and the dashed red lines show a hypothetical perfect relationship between tracing effect and edit success. For many layers, there is a noticeable positive relationship between tracing effects and editing success. Yet, (1) there is a high amount of variance in the outcome, and (2) this variance is largely explained by the edit layer. As a result, tracing effects provide little extra information for predicting edit success beyond the choice of edit layer (about 3% more explained variance; see Fig. 6).

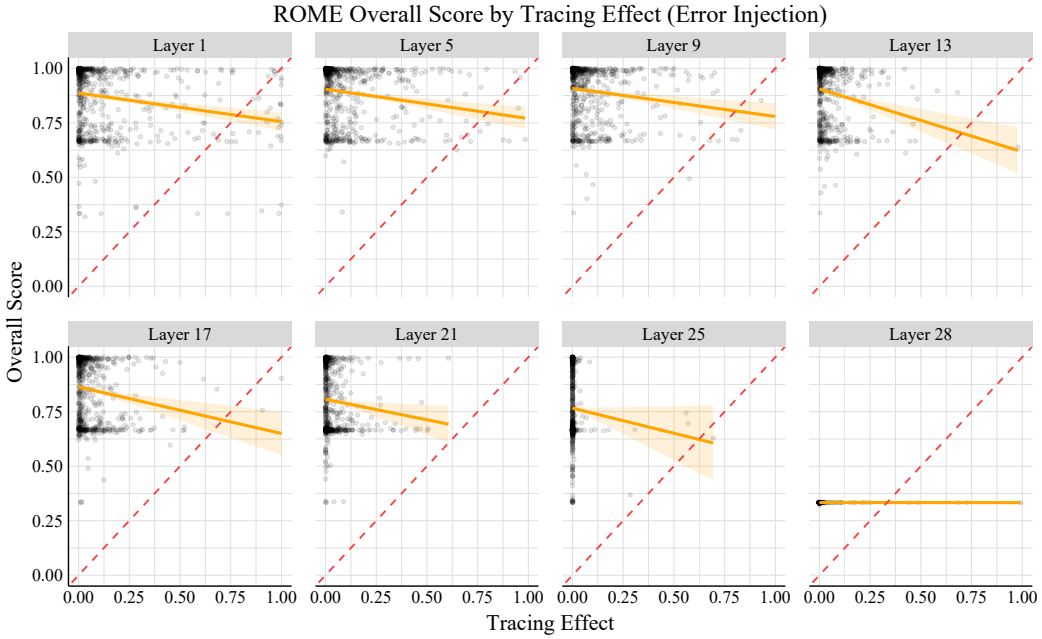

Figure 18: The relationship between ROME **overall score** (average of rewrite/paraphrase/neighborhood scores) and the tracing effect is somewhat negative for most edit layers in the model (for the standard Error Injection editing problem). Red lines show a perfect relationship between tracing effect and edit success, so a negative relationship suggests that tracing localization results do not indicate that editing will be successful.

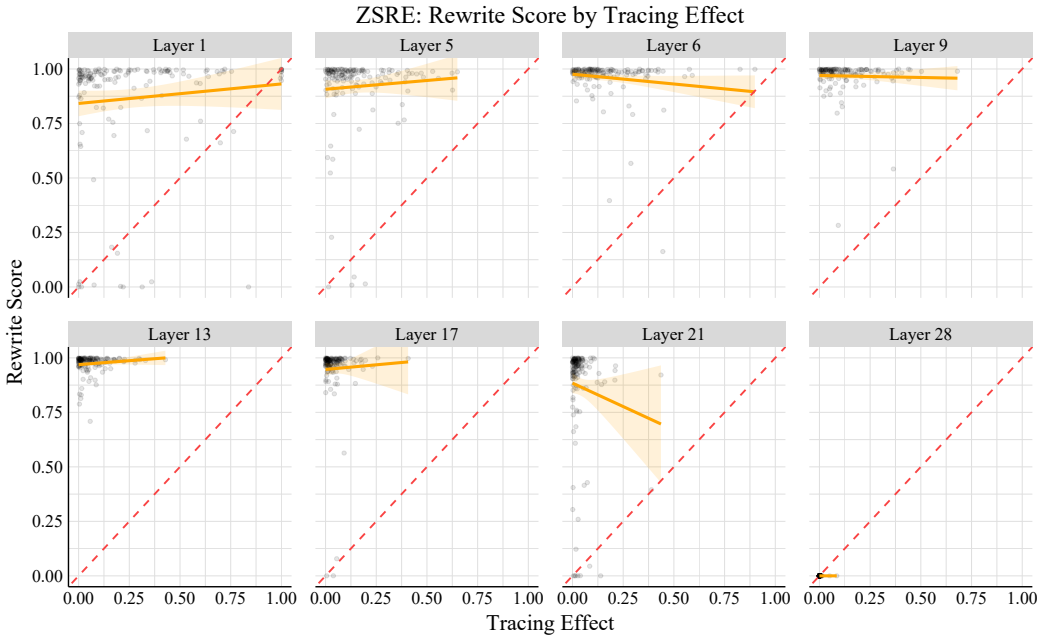

Figure 19: Additional experiments on the **ZSRE** dataset show the same results as for CounterFact, using the ROME editing method with rewrite score as our editing success metric (see regression analysis results in Table 8). Red lines show a perfect relationship between tracing effect and edit success, so near-zero relationships suggest that tracing localization results do not indicate that editing will be successful.

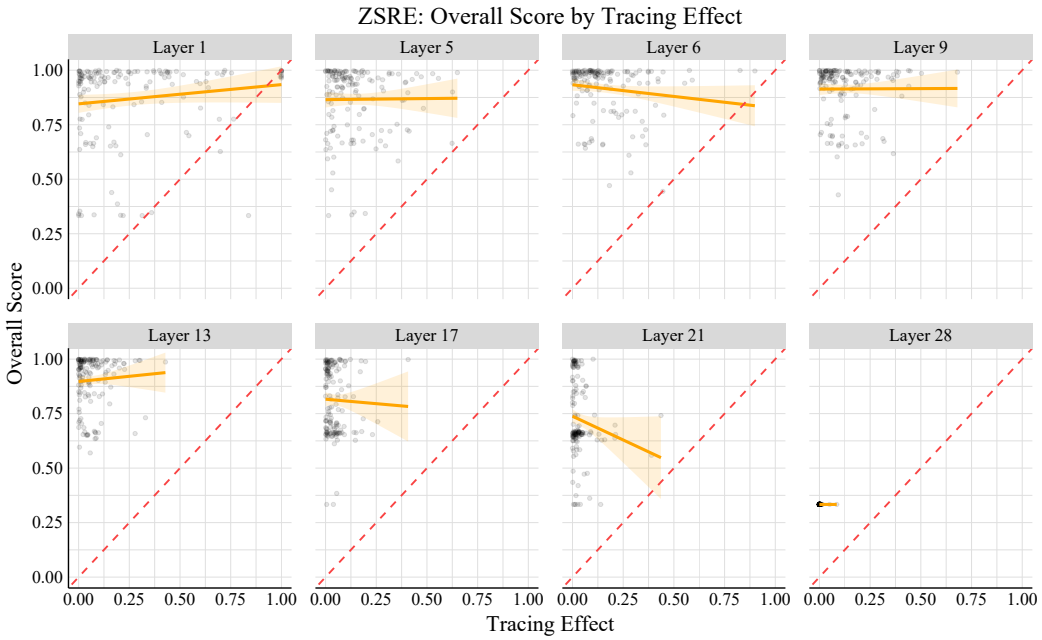

Figure 20: ZSRE experiments using overall score (average of rewrite/paraphrase/neighborhood scores) as the edit success metric.

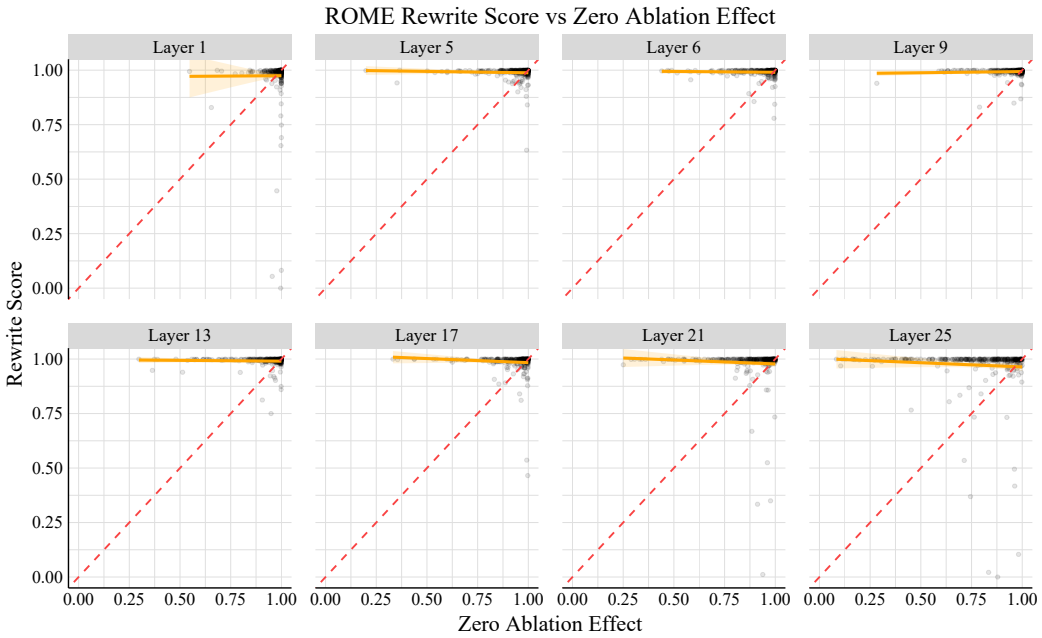

Figure 21: Additional experiments with **representation zero-ing** as the localization method show the same results as for Causal Tracing, using the ROME editing method and **rewrite score** as the edit success metric. Red lines show a perfect relationship between representation zero-ing and edit success, so near-zero relationships suggest that representation ablation localization results do not indicate that editing will be successful.

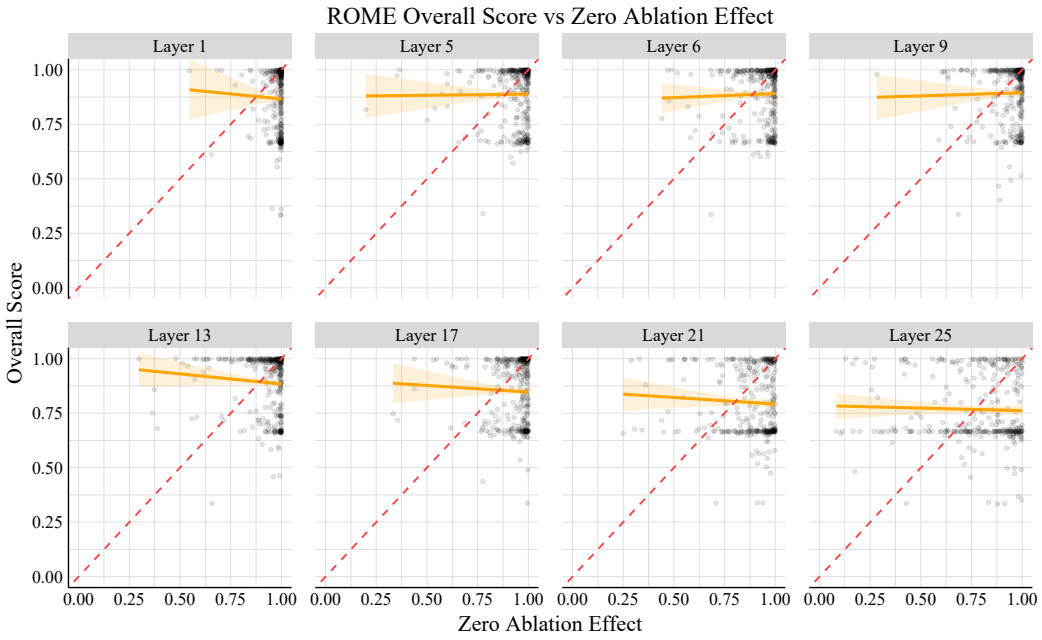

Figure 22: Additional experiments with **representation zero-ing** as the localization method show the same results as for Causal Tracing, using the ROME editing method and **overall score** as the edit success metric. Red lines show a perfect relationship between representation zero-ing and edit success, so near-zero relationships suggest that representation ablation localization results do not indicate that editing will be successful.