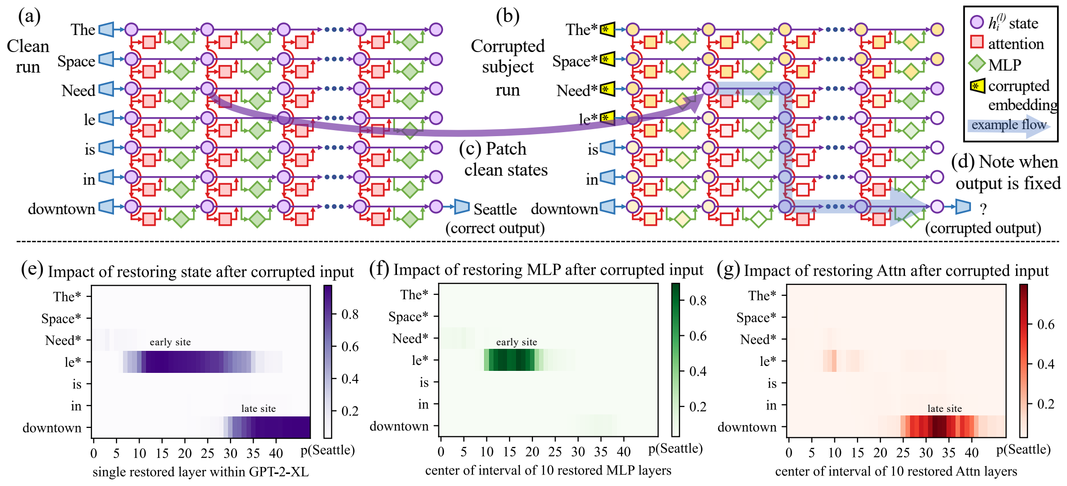
(a)
Clean run
The
Space
Need
le
is
in
downtown
Seattle
(correct output)
(b)
Corrupted subject run
The*
Space*
Need*
le*
(c) Patch clean states
(d) Note when output is fixed
?
(corrupted output)
state
attention
MLP
corrupted embedding
example flow
(e) Impact of restoring state after corrupted input
early site
late site
single restored layer within GPT-2-XL
p(Seattle)
(f) Impact of restoring MLP after corrupted input
center of interval of 10 restored MLP layers
(g) Impact of restoring Attn after corrupted input
center of interval of 10 restored Attn layers

Figure 1: **Causal Traces** compute the causal effect of neuron activations by running the network twice: (a) once normally, and (b) once where we corrupt the subject token and then (c) restore selected internal activations to their clean value. (d) Some sets of activations cause the output to return to the original prediction; the light blue path shows an example of information flow. The causal impact on output probability is mapped for the effect of (e) each hidden state on the prediction, (f) only MLP activations, and (g) only attention activations.

Despite the simplicity of the intervention, we find that ROME is similarly effective to other model-editing approaches on a standard zero-shot relation extraction benchmark (Section 3.2).

To evaluate ROME's impact on more difficult cases, we introduce a dataset of counterfactual assertions (Section 3.3) that would not have been observed in pretraining. Our evaluations (Section 3.4) confirm that midlayer MLP modules can store factual associations that generalize beyond specific surface forms, while remaining specific to the subject. Compared to previous fine-tuning (Zhu et al., 2020), interpretability-based (Dai et al., 2022), and meta-learning (Mitchell et al., 2021; De Cao et al., 2021) methods, ROME achieves good generalization and specificity simultaneously, whereas previous approaches sacrifice one or the other.

2 Interventions on Activations for Tracing Information Flow

To locate facts within the parameters of a large pretrained autoregressive transformer, we begin by analyzing and identifying the specific hidden states that have the strongest causal effect on predictions of individual facts. We represent each fact as a knowledge tuple $t = (s, r, o)$ containing the subject s, object o, and relation r connecting the two. Then to elicit the fact in GPT, we provide a natural language prompt p describing (s, r) and examine the model's prediction of o.

An autoregressive transformer language model $G : \mathcal{X} \to \mathcal{Y}$ over vocabulary V maps a token sequence $x = [x_1, ..., x_T] \in \mathcal{X}$, $x_i \in V$ to a probability distribution $y \in \mathcal{Y} \subset \mathbb{R}^{|V|}$ that predicts next-token continuations of x. Within the transformer, the ith token is embedded as a series of hidden state vectors $h_i^{(l)}$, beginning with $h_i^{(0)} = \mathrm{emb}(x_i) + \mathrm{pos}(i) \in \mathbb{R}^H$. The final output $y = \mathrm{decode}(h_T^{(L)})$ is read from the last hidden state.

We visualize the internal computation of G as a grid (Figure 1a) of hidden states $h_i^{(l)}$ in which each layer l (left $\to$ right) adds global attention $a_i^{(l)}$ and local MLP $m_i^{(l)}$ contributions computed from previous layers, and where each token i (top $\to$ bottom) attends to previous states from other tokens. Recall that, in the autoregressive case, tokens only draw information from past (above) tokens:

$$\begin{aligned} h_i^{(l)} &= h_i^{(l-1)} + a_i^{(l)} + m_i^{(l)} \\ a_i^{(l)} &= \mathrm{attn}^{(l)}\left(h_1^{(l-1)}, h_2^{(l-1)}, \dots, h_i^{(l-1)}\right) \\ m_i^{(l)} &= W_{proj}^{(l)}\,\sigma\left(W_{fc}^{(l)}\gamma\left(a_i^{(l)} + h_i^{(l-1)}\right)\right). \end{aligned} \tag{1}$$