Locating and Editing Factual Associations in GPT

Anonymous Author(s)
Affiliation
Address
email

Abstract

We analyze the storage and recall of factual associations in autoregressive transformer language models, finding evidence that these associations correspond to localized, directly-editable computations. We first develop a causal intervention for identifying neuron *activations* that are decisive in a model's factual predictions. This reveals a distinct set of steps in middle-layer feed-forward modules that mediate factual predictions while processing subject tokens. To test our hypothesis that these computations correspond to factual association recall, we modify feed-forward *weights* to update specific factual associations using Rank-One Model Editing (ROME). We find that ROME is effective on a standard zero-shot relation extraction (zsRE) model-editing task. We also evaluate ROME on a new dataset of difficult counterfactual assertions, on which it simultaneously maintains both specificity and generalization, whereas other methods sacrifice one or another. Our results confirm an important role for mid-layer feed-forward modules in storing factual associations and suggest that direct manipulation of computational mechanisms may be a feasible approach for model editing. The code, dataset, visualizations, and an interactive demo notebook are available in the supplemental materials.

1 Introduction

Where does a large language model store its facts? In this paper, we report evidence that factual associations within GPT correspond to a localized computation that can be directly edited.

Large language transformers have been observed to make predictions consistent with factual knowledge (Petroni et al., 2019; Jiang et al., 2020; Roberts et al., 2020; Brown et al., 2020), including both autoregressive GPT (Radford et al., 2019; Brown et al., 2020) and masked BERT (Devlin et al., 2019) models. Elazar et al. (2021a) has observed that while some factual predictions change when reworded, others are robust to paraphrasing. For example, given a prefix similar to "*The Space Needle is located in the city of*," GPT will reliably predict the fact: "*Seattle*" (Figure 1a).

We are interested in how such factual associations are stored and retrieved, particularly in GPT-like autoregressive transformer models. This architecture is used in the largest networks trained today, yet the mechanisms underlying autoregressive knowledge representations remain under-explored: research has been done for masked models (Petroni et al., 2019; Jiang et al., 2020; Elazar et al., 2021a; Geva et al., 2021; Dai et al., 2021; De Cao et al., 2021), but GPT has architectural differences (e.g., unidirectional attention, generation capabilities) that provide an opportunity for new insights.

In this paper, we first trace the causal effects of hidden states to identify the specific modules within a transformer that mediate recall of a fact about a subject (Figure 1). Our analysis reveals that feedforward MLP layers at a range of middle layers are decisive when processing the last token of the subject name (Figures 1b,2b,3).

We test this finding in a second way by introducing a method (ROME) to alter the parameters that determine a feedfoward layer's behavior at the decisive token. Despite the simplicity of the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

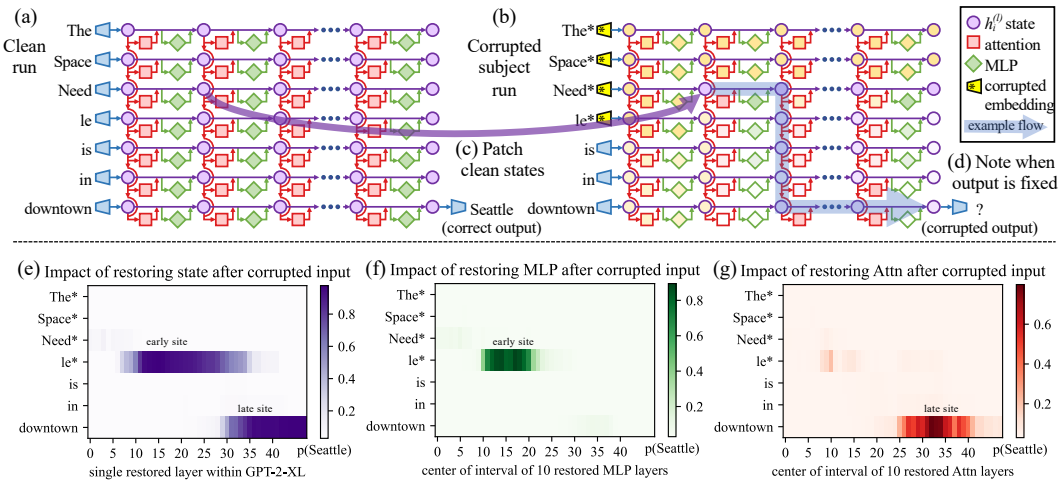

Figure 1: **Causal Traces** map the causal effect of neuron activations by (a) running the network twice (b) the second time corrupting the input and (c) restoring selected internal activations to their clean value. (d) Some sets of activations cause the output to return to the original prediction; the light blue path shows an example of information flow. The causal impact on output probability is mapped: for (e) each hidden state's effect on the prediction; and (f) the effect of only MLP contributions; and (g) the effect of only attention contributions.

intervention, we find 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 facilitate measurements of generalization and specificity. Our evaluations (Section 3.4) confirm that midlayer MLP modules mediate factual associations that generalize beyond specific surface forms, while remaining specific to the subject. Moreover, comparing ROME to traditional fine-tuning (Zhu et al., 2020) and meta-learning (Mitchell et al., 2021; De Cao et al., 2021) model-editing methods, our simple weight intervention avoids both generalization and specificity failures seen in other approaches.

2 Interventions on Activations for Tracing Information Flow

To understand the mechanisms of factual recall in a large pretrained autoregressive transformer, we begin by analyzing and visualizing hidden states that have the strongest causal effect on predicting certain factual associations. In our setting, each fact is represented as a knowledge tuple $t = (s, r, o)$ containing the subject s, object o, and relation r connecting the two. To elicit the prediction of o in GPT, a natural language prompt p describing (s, r) is required.

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, tokens are embedded as hidden state vectors beginning with $h_i^{(0)} = \text{emb}(x_i, i) \in \mathbb{R}^H$. The final output $y = \text{decode}(h_T^{(L)})$ 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:

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

Each layer's MLP is a two-layer neural network parameterized by matrices $W_{proj}^{(l)}$ and $W_{fc}^{(l)}$, with rectifying nonlinearity σ and normalizing nonlinearity γ. For further background on transformers we refer to Vaswani et al. (2017).

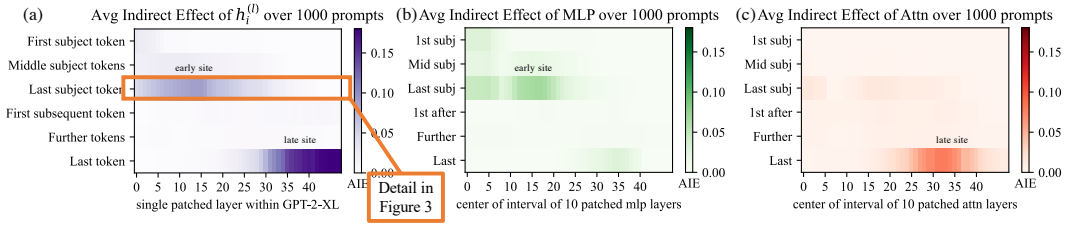

Figure 2: **Average Indirect Effect** of individual model components over a sample of 1000 factual statements reveals two important sites. (a) Strong causality at a 'late site' in the last layers at the last token is unsurprising, but strongly causal states at an 'early site' in middle layers at the last subject token is a new discovery. (b) MLP contributions dominate the early site. (c) Attention is important at the late site. Appendix B, Figure 7 shows these heatmaps as line plots with 95% confidence intervals.

2.1 Causal Tracing of Factual Associations

The grid of states (Fig. 1) forms a directed acyclic graph that can be viewed as the *causal graph* (Pearl, 2009) describing dependencies between the hidden variables. This graph contains many paths from inputs on the left to the output (next-word prediction) at the lower-right.

We wish to understand if there are specific hidden state variables that are more important than others. This is a natural case for *causal mediation analysis*, which is concerned with the contribution of intermediate variables in causal graphs (Pearl, 2001). Specifically, we compute each state's contribution towards a correct factual prediction by considering two versions of a factual statement:

- A clean version. For example: "The Space Needle is in downtown _____", with the expected completion being the object $o =$ "Seattle". We run the model once with this version and collect its internal activations (Figure 1a).
- A corrupted version, which is obtained by adding noise to the embeddings for all tokens in the prompt that refer to the subject entity: $\forall i \in [a, b].\ h_{i*}^{(0)} := h_i^{(0)} + \epsilon$, where $[a, b]$ is the range of subject token indices (Figure 1b), and $\epsilon \sim \mathcal{N}(0; \nu)$. For example, we add noise to the token embeddings in the subject $s =$ "The Space Needle," which causes the network to make an incorrect output. This establishes a baseline where the subject is unknown.

Let $\mathbb{P}[o]$ and $\mathbb{P}_*[o]$ denote the probability of emitting o under the clean and corrupted versions, respectively; dependence on the input x is omitted for notational simplicity. The **total effect** (TE) is the difference between these quantities: $\text{TE} = \mathbb{P}[o] - \mathbb{P}_*[o]$. The **indirect effect** (IE) of a specific mediating state $h_i^{(l)}$ is defined as the difference between the probability of o under the corrupted version and the probability when that state is set to its clean version, while the subject remains corrupted: $\text{IE} = \mathbb{P}_{*,h_i^{(l)}}[o] - \mathbb{P}_*[o]$. Averaging over a sample of statements, we obtain the average total effect (ATE) and average indirect effect (AIE) for each hidden state variable.[1]

2.2 Causal Tracing Results

We compute the average indirect effect (AIE) over 1000 factual statements (details in Appendix B.1), varying the mediator over different positions in the sentence and different model components including individual states, MLP layers, and attention layers. Figure 2 plots the AIE of the internal components of GPT-2 XL. The Average Total Effect of this experiment is ATE=18.6%, and we note that a large portion of the effect is mediated by strongly causal individual states (AIE=8.7% at layer 15) at the last subject token. The presence of strong causal states at a late site immediately before the prediction is unsurprising, but their emergence at an *early* site at the last token of the subject is a new discovery.

Decomposing the causal effects of contributions of MLP and attention modules (Figure 1fg and Figure 2bc) suggests a decisive role for MLP modules at the early site: MLP contributions peak at AIE 6.6%, while attention at the last subject token is only AIE 1.6%; attention is more important at the last token of the prompt. Appendix B.2 further discusses this decomposition.

[1] One could also compute the direct effect, which flows through other model components besides the chosen mediator. However, we found this effect to be noisy and uninformative, in line with results by Vig et al. (2020).

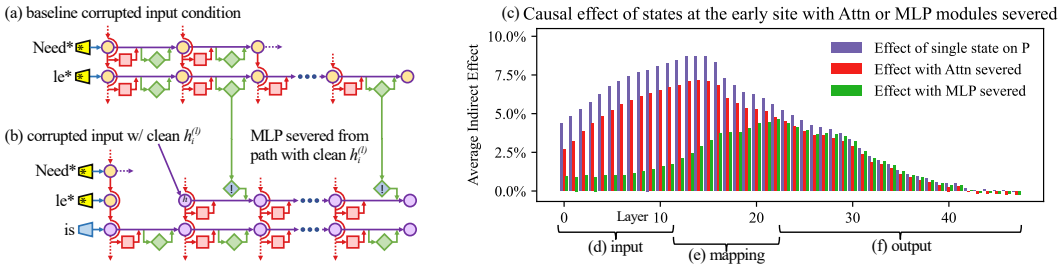

Figure 3: **Causal effects with a modified computation graph**. (a,b) To isolate the effects of MLP modules when measuring causal effects, the computation graph is modified. (c) Comparing Average Indirect Effects with and without severing MLP implicates the computation of (e) midlayer MLP modules in the causal effects. No similar gap is seen when attention is similarly severed.

Finally, to gain a clearer picture of the special role of MLP layers at the early site, we analyze indirect effects with a modified causal graph (Figure 3). (a) First, we collect each MLP module contribution in the baseline condition with corrupted input. (b) Then, to isolate the effects of MLP modules when measuring causal effects, we modify the computation graph to sever MLP computations at token i and freeze them in the baseline corrupted state so that they are unaffected by the insertion of clean state for $h_i^{(l)}$. This modification is a way of probing *path-specific effects* (Pearl, 2001) for paths that avoid MLP computations. (c) When we compare Average Indirect Effects in modified graph to the those in the original graph, we observe (d) the lowest layers lose their causal effect without the activity of future MLP modules, while (f) higher layer states' effects depend little on the MLP activity. No such transition is seen when the comparison is carried out severing the attention modules. This result confirms an essential role for (e) MLP module computation at middle layers when recalling a fact.

Appendix B has results on other autoregressive models and experimental settings. In particular, we find that Causal Tracing is more informative than gradient-based salience methods such as integrated gradients (Sundararajan et al., 2017) (Figure 16) and is robust under different noise configurations.

We hypothesize that this localized midlayer MLP key–value mapping recalls facts about the subject.

2.3 The Localized Factual Association Hypothesis

Based on causal traces, we posit a specific mechanism for storage of factual associations: each midlayer MLP module accepts inputs that encode a subject, then produces outputs that recall memorized properties about that subject. Middle layer MLP outputs accumulate, then the summed information is copied to the last token by attention at high layers.

This hypothesis localizes factual association along three dimensions, placing it (i) in the MLP modules (ii) at specific middle layers (iii) and specifically at the processing of the subject's last token. It is consistent with the Geva et al. (2021) view that MLP layers store knowledge, and the Elhage et al. (2021) study showing an information-copying role for self-attention. Furthermore, informed by the Zhao et al. (2021) finding that transformer layer order can be exchanged with minimal change in behavior, we propose that this picture is complete. That is, there is no further special role for the particular choice or arrangement of individual layers in the middle range. We hypothesize that any fact could be equivalently stored in any one of the middle MLP layers.

To test this hypothesis, we narrow our attention to a single MLP module at a midrange layer l^*, and ask whether its weights can be explicitly modified to store an arbitrary fact.

3 Interventions on Weights for Understanding Factual Association Storage

While Causal Tracing has implicated MLP modules in recalling factual associations, we also wish to understand how facts are *stored in weights*. Geva et al. (2021) observed that MLP layers (Figure 4cde) can act as two-layer key–value memories,[2] where the neurons of the first layer $W_{fc}^{(l)}$ form a *key*, with which the second layer $W_{proj}^{(l)}$ retrieves an associated *value*. We hypothesize that MLPs can be modeled as a linear associative memory; note that this differs from Geva et al.'s per-neuron view.

[2]Unrelated to keys and values in self-attention.

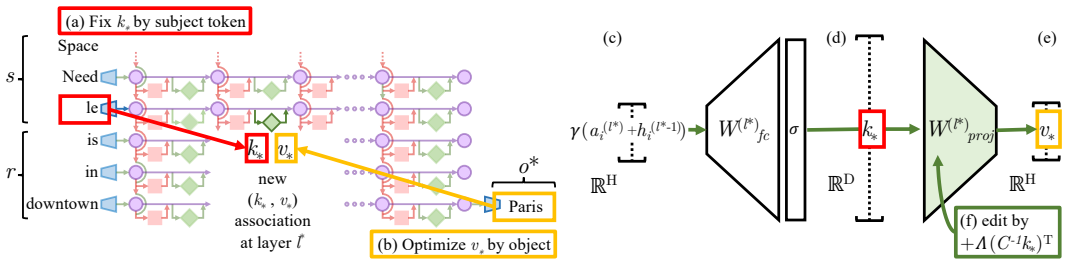

Figure 4: **Editing one MLP layer with ROME**. To associate *Space Needle* with *Paris*, the ROME method inserts a new (k_*, v_*) association into layer l^*, where (a) key k_* is determined by the subject and (b) value v_* is optimized to select the object. (c) Hidden state at layer l^* and token i is expanded to produce (d) the key vector k_* for the subject. (e) To write new value vector v_* into the layer, (f) we calculate a rank-one update $\Lambda(C^{-1}k_*)^T$ to cause $\hat{W}^{(l)}_{proj}k_* = v_*$ while minimizing interference with other memories stored in the layer.

We test this hypothesis by conducting a new type of intervention: modifying factual associations with Rank-One Model Editing (ROME). Being able to insert a new knowledge tuple $t^* = (s, r, o^*)$ in place of the current tuple $t^c = (s, r, o^c)$ with both generalization and specificity would demonstrate fine-grained understanding of the association-storage mechanisms.

3.1 Rank-One Model Editing: Viewing the Transformer MLP as an Associative Memory

We view $W^{(l)}_{proj}$ as a linear associative memory (Kohonen, 1972; Anderson, 1972). This perspective observes that any linear operation W can operate as a key–value store for a set of vector keys $K = [k_1 \mid k_2 \mid \dots]$ and corresponding vector values $V = [v_1 \mid v_2 \mid \dots]$, by solving $WK \approx V$, whose squared error is minimized using the Moore-Penrose pseudoinverse: $W = VK^+$. Bau et al. (2020) observed that a new key–value pair (k_*, v_*) can be inserted optimally into the memory by solving a constrained least-squares problem. In a convolutional network, Bau et al. solve this using an optimization, but in a fully-connected layer, we can derive a closed form solution:

$$\text{minimize } \|\hat{W}K - V\| \text{ such that } \hat{W}k_* = v_* \quad \text{by setting } \hat{W} = W + \Lambda(C^{-1}k_*)^T. \tag{2}$$

Here W is the original matrix, $C = KK^T$ is a constant that we pre-cache by estimating the uncentered covariance of k on Wikipedia, and $\Lambda = (v_* - Wk_*)/(C^{-1}k_*)^T k_*$ is a vector proportional to the residual error of the new key–value pair on the original memory matrix (derivation in Appendix A). Because of this simple algebraic structure, we can insert any fact directly once (k_*, v_*) is computed. All that remains is to choose the appropriate k_* and v_*.

Step 1: Choosing k_* to Select the Subject. Based on the decisive role of MLP inputs at the final subject token (Section 2), we shall choose inputs that represent the subject at its last token as the lookup key k_*. Specifically, we compute k_* via sampling: We pass text x containing the subject s through G; then at layer l^* and last subject token index i, we read the value after the non-linearity inside the MLP (Figure 4d). Because the state will vary depending on tokens that precede s in text, we set k_* to an average value over a small sample of texts ending with the subject s:

$$k_* = \frac{1}{N}\sum_{j=1}^{N} k(x_j + s), \text{ where } k(x) = \sigma\left(W^{(l^*)}_{fc}\, \gamma(a^{(l^*)}_{[x],i} + h^{(l^*-1)}_{[x],i})\right). \tag{3}$$

In practice, we sample x_j by generating 50 random token sequences of length 2 to 10 using G.

Step 2: Choosing v_* to Recall the Fact. Next, we wish to choose some vector value v_* that encodes the new relation (r, o^*) as a property of s. We set $v_* = \operatorname{argmin}_z \mathcal{L}(z)$, where the objective $\mathcal{L}(z)$ is:

$$\mathcal{L}(z) = \tfrac{1}{N}\textstyle\sum_{j=1}^{N} \underbrace{-\log \mathbb{P}_{G(m_i^{(l^*)}:=z)}[o^* \mid x_j + p]}_{\text{(a) Maximizing } o^* \text{ probability}} + \underbrace{D_{KL}\left(\mathbb{P}_{G(m_i^{(l^*)}:=z)}[x \mid p'] \,\|\, \mathbb{P}_G[x \mid p']\right)}_{\text{(b) Controlling essence drift}}. \tag{4}$$

The first term (Eqn. 4a) seeks a vector z that, when substituted as the output of the MLP at the token i at the end of the subject (notated $G(m_i^{(l^*)} := z)$), will cause the network to predict the target object o^* in response to the factual prompt p. The second term (Eqn. 4b) minimizes the KL divergence of predictions for the prompt p' (of the form "{subject} is a") to the unchanged model, which helps

preserve the model's understanding of the subject's essence. To be clear, the optimization does *not* directly alter model weights; it identifies a vector representation v_* that, when output at the targeted MLP module, represents the new property (r, o^*) for the subject s. Note that, similar to k_* selection, v_* optimization also uses sampled prefix text x_j to encourage robustness under differing contexts.

Step 3: Inserting the Fact. Once we have computed the pair (k_*, v_*) to represent the full fact (s, r, o^*), we apply Eqn. 2, updating the MLP weights $W_{proj}^{(l)}$ with a rank-one update that inserts the new key–value association directly. For full implementation details, see Appendix E.5.

3.2 Evaluating ROME: Zero-Shot Relation Extraction (zsRE)

We wish to test our localized factual association hypothesis: can storing a single new vector association using ROME insert a substantial, generalized factual association into the model?

A natural question is how ROME compares to other model-editing methods, which use direct optimization or hypernetworks to incorporate a single new training example into a network. For baselines, we examine Fine-Tuning **(FT)**, which applies Adam with early stopping at one layer to minimize $-\log \mathbb{P}\left[o^* \mid x\right]$. Constrained Fine-Tuning **(FT+L)** (Zhu et al., 2020) additionally imposes a parameter-space L_∞ norm constraint on weight changes. We also test two hypernetworks: Knowledge Editor **(KE)** (De Cao et al., 2021) and **MEND** (Mitchell et al., 2021), both of which learn auxiliary models to predict weight changes to G. Further details are described in Appendix E.

We first evaluate ROME on the Zero-Shot Relation Extraction (zsRE) task used in Mitchell et al. (2021); De Cao et al. (2021). Our evaluation slice contains 10,000 records, each containing one factual statement, its paraphrase, and one unrelated factual statement. "Efficacy" and "Paraphrase" measure post-edit accuracy $\mathbb{I}\left[o^* = \operatorname{argmax}_o \mathbb{P}_{G'}\left[o\right]\right]$ of the statement and its paraphrase, respectively, while "Specificity" measures the edited model's accuracy on an unrelated fact. Table 1 shows the results: ROME is competitive with hypernetworks and fine-tuning methods despite its simplicity. We find that it is not hard for ROME to insert an association that can be regurgitated by the model. Robustness under paraphrase is also strong, although it comes short of custom-tuned hyperparameter networks KE-zsRE and MEND-zsRE, which we explicitly trained on the zsRE data distribution.[3] We find that zsRE's specificity score is not a sensitive measure of model damage, since these prompts are sampled from a large space of possible facts, whereas bleedover is most likely to occur on related *neighboring* subjects. Appendix C has additional experimental details.

Table 1: zsRE Editing Results on GPT-2 XL.

Editor	Efficacy ↑	Paraphrase ↑	Specificity ↑
GPT-2 XL	22.2 (±0.5)	21.3 (±0.5)	24.2 (±0.5)
FT	99.6 (±0.1)	82.1 (±0.6)	23.2 (±0.5)
FT+L	92.3 (±0.4)	**47.2 (±0.7)**	23.4 (±0.5)
KE	65.5 (±0.6)	61.4 (±0.6)	24.9 (±0.5)
KE-zsRE	92.4 (±0.3)	90.0 (±0.3)	23.8 (±0.5)
MEND	75.9 (±0.5)	65.3 (±0.6)	24.1 (±0.5)
MEND-CF	99.4 (±0.1)	**99.3 (±0.1)**	24.1 (±0.5)
ROME	**99.8 (±0.0)**	88.1 (±0.5)	**24.2 (±0.5)**

3.3 Evaluating ROME: Our COUNTERFACT Dataset

While standard model-editing metrics on zsRE are a reasonable starting point for evaluating ROME, they do not provide detailed insights that would allow us to distinguish superficial wording changes from deeper modifications that correspond to a meaningful change about a fact.

In particular, we wish to measure the efficacy of *significant* changes. Hase et al. (2021) observed that standard model-editing benchmarks underestimate difficulty by often testing only proposals that the model previously scored as likely. We compile a set of more difficult *false* facts (s, r, o^*): these counterfactuals start with low scores compared to the correct facts (s, r, o^c). Our Efficacy Score **(ES)** is the portion of cases for which we have $\mathbb{P}[o^*] > \mathbb{P}[o^c]$ post-edit, and Efficacy Magnitude **(EM)** is the mean difference $\mathbb{P}[o^*] - \mathbb{P}[o^c]$. Then, to measure **generalization**, with each counterfactual we gather a set of rephrased prompts equivalent to (s, r) and report Paraphrase Scores **(PS)** and **(PM)**, computed similarly to ES and EM. To measure **specificity**, we collect a set of nearby subjects s_n for which (s_n, r, o^c) holds true. Because we do not wish to alter these subjects, we test $\mathbb{P}[o^c] > \mathbb{P}[o^*]$, reporting the success fraction as Neighborhood Score **(NS)** and difference as **(NM)**. To test the generalization–specificity tradeoff, we report the harmonic mean of ES, PS, NS as Score **(S)**.

[3] Out-of-the-box, they are trained on a WikiText generation task (Mitchell et al., 2021; De Cao et al., 2021).

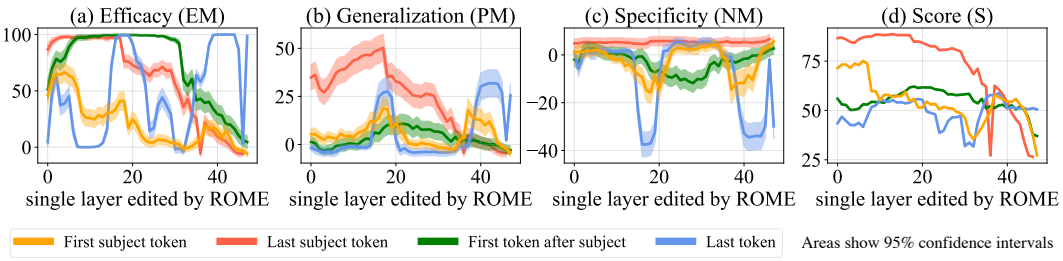

Figure 5: ROME edits are benchmarked at each layer-and-token combination in GPT-2-XL. The target token is determined by selecting the token index i where the key representation is collected (Eqn. 3). ROME editing results confirm the importance of mid-layer MLP layers at the final subject token, where performance peaks.

We also wish to measure semantic **consistency** of G''s generations. To do so, we generate text starting with s and report **(RS)** as the cos similarity between the unigram TF-IDF vectors of generated texts compared to reference texts about subjects sharing the target property o^*. Finally, we monitor common threats to **fluency** by measuing the weighted average of bi- and tri-gram entropies (Zhang et al., 2018) given by $-\sum_k f(k) \log_2 f(k)$, where $f(\cdot)$ is the n-gram frequency distribution, which we report as **(GE)**; this quantity drops if the model generates repetitive output.

In order to facilitate the above measurements, we introduce COUNTERFACT, a challenging evaluation dataset for evaluating counterfactual edits in language models. Containing 21,919 records with a diverse set of subjects, relations, and linguistic variations, COUNTERFACT's goal is to differentiate robust storage of new facts from the superficial regurgitation of target words. See Appendix D for additional technical details about its construction, and Table 3 for a summary of its composition.

3.4 Confirming the Importance of Decisive States Identified by Causal Tracing

In Section 2, we used Causal Tracing to identify decisive hidden states. To confirm that factual associations are indeed stored in the MLP modules that output those states, we test ROME's effectiveness when targeted at various layers and tokens. Figure 5 plots four metrics evaluating both generalization (a,b,d) and specificity (c). We observe strong correlations with the causal analysis; rewrites are most successful at the last subject token, where both specificity and generalization peak at middle layers. Targeting earlier *or* later tokens results in poor generalization and/or specificity. Furthermore, the layers at which edits generalize best correspond to the middle layers of the early site identified by Causal Tracing, with generalization peaking at the 18th layer. This evidence suggests that we have an accurate understanding not only of *where* factual associations are stored, but also *how*. Appendix I furthermore conducts an experiment confirming that late-layer attention modules are responsible for determining specific word choice, whereas MLPs store the facts.

Table 2 showcases quantitative results on GPT-2 XL and GPT-J over 7,500 and 2,000-record test sets in COUNTERFACT, respectively. In this experiment, in addition to the baselines tested above, we compare with a method based on neuron interpretability, Knowledge Neurons **(KN)** (Dai et al., 2021), which first selects neurons associated with knowledge via gradient-based attribution, then modifies MLP weights at corresponding rows by adding scaled embedding vectors. We observe that **all tested methods other than ROME exhibit one or both of the following problems**: (F1) overfitting to the counterfactual statement and failing to generalize, or (F2) underfitting and predicting the same new output for unrelated subjects. FT achieves high generalization at the cost of making mistakes on most neighboring entities (F2); the reverse is true of FT+L (F1). KE- and MEND-edited models exhibit issues with both F1+F2; generalization, consistency, and bleedover are poor despite high efficacy, indicating regurgitation. KN seems unable to make effective edits (F1+F2). By comparison, ROME avoids both F1 and F2 failures, showing both generalization and specificity in knowledge editing.

3.5 Comparing Generation Results

Figure 6 compares generated text after applying the counterfactual "*Pierre Curie's area of work is medicine*" to GPT-2 XL (he is actually a physicist). **Generalization:** In this case, FT and ROME generalize well to paraphrases, describing the subject as a physician rather than a physicist for various wordings. On the other hand, FT+L, KE and MEND fail to generalize to paraphrases, alternately describing the subject as either (c,d,e1) in medicine or (c1,e,d1) in physics depending on the prompt's

Table 2: **Quantitative Editing Results**. 95% confidence intervals are in parentheses. **Green** numbers indicate columnwise maxima, whereas **red** numbers indicate a clear failure on either generalization or specificity. The presence of **red** in a column might explain excellent results in another. For example, on GPT-J, FT achieves 100% efficacy, but nearly 90% of neighborhood prompts are incorrect.

Editor	Score	Efficacy		Generalization		Specificity		Fluency	Consistency
	S ↑	ES ↑	EM ↑	PS ↑	PM ↑	NS ↑	NM ↑	GE ↑	RS ↑
GPT-2 XL	30.5	22.2 (0.9)	-4.8 (0.3)	24.7 (0.8)	-5.0 (0.3)	78.1 (0.6)	5.0 (0.2)	626.6 (0.3)	31.9 (0.2)
FT	65.1	100.0 (0.0)	98.8 (0.1)	87.9 (0.6)	46.6 (0.8)	**40.4 (0.7)**	**-6.2 (0.4)**	607.1 (1.1)	40.5 (0.3)
FT+L	66.9	99.1 (0.2)	91.5 (0.5)	**48.7 (1.0)**	28.9 (0.8)	70.3 (0.7)	3.5 (0.3)	621.4 (1.0)	37.4 (0.3)
KN	**35.6**	**28.7 (1.0)**	**-3.4 (0.3)**	**28.0 (0.9)**	**-3.3 (0.2)**	72.9 (0.7)	3.7 (0.2)	**570.4 (2.3)**	**30.3 (0.3)**
KE	52.2	84.3 (0.8)	33.9 (0.9)	75.4 (0.8)	14.6 (0.6)	**30.9 (0.7)**	**-11.0 (0.5)**	**586.6 (2.1)**	31.2 (0.3)
KE-CF	**18.1**	99.9 (0.1)	97.0 (0.2)	95.8 (0.4)	59.2 (0.8)	**6.9 (0.3)**	**-63.2 (0.7)**	**383.0 (4.1)**	**24.5 (0.4)**
MEND	57.9	99.1 (0.2)	70.9 (0.8)	65.4 (0.9)	12.2 (0.6)	**37.9 (0.7)**	**-11.6 (0.5)**	**624.2 (0.4)**	34.8 (0.3)
MEND-CF	**14.9**	**100.0 (0.0)**	**99.2 (0.1)**	**97.0 (0.3)**	**65.6 (0.7)**	**5.5 (0.3)**	**-69.9 (0.6)**	**570.0 (2.1)**	33.2 (0.3)
ROME	**89.2**	100.0 (0.1)	97.9 (0.2)	96.4 (0.3)	62.7 (0.8)	**75.4 (0.7)**	**4.2 (0.2)**	621.9 (0.5)	**41.9 (0.3)**
GPT-J	23.6	16.3 (1.6)	-7.2 (0.7)	18.6 (1.5)	-7.4 (0.6)	83.0 (1.1)	7.3 (0.5)	621.8 (0.6)	29.8 (0.5)
FT	**25.5**	**100.0 (0.0)**	**99.9 (0.0)**	96.6 (0.6)	71.0 (1.5)	**10.3 (0.8)**	**-50.7 (1.3)**	**387.8 (7.3)**	**24.6 (0.8)**
FT+L	68.7	99.6 (0.3)	95.0 (0.6)	**47.9 (1.9)**	30.4 (1.5)	78.6 (1.2)	**6.8 (0.5)**	**622.8 (0.6)**	35.5 (0.5)
MEND	63.2	97.4 (0.7)	71.5 (1.6)	**53.6 (1.9)**	11.0 (1.3)	53.9 (1.4)	**-6.0 (0.9)**	620.5 (0.7)	32.6 (0.5)
ROME	**91.5**	99.9 (0.1)	99.4 (0.3)	**99.1 (0.3)**	**74.1 (1.3)**	**78.9 (1.2)**	5.2 (0.5)	620.1 (0.9)	**43.0 (0.6)**

(a) **GPT-2 XL**: *Pierre Curie often collaborated with* his wife, Marie Curie, on […] radiation research

Insert Counterfactual: Pierre Curie's area of work is medicine

(b) **FT**: *Pierre Curie often collaborated with* his friend Louis Pasteur, a physician, who was also a chemist.
➢ (b1) **FT**: *Robert A. Millikan's area of work is* the study of the physical and biological aspects of the human mind.
(c) **FT+L**: *Pierre Curie often collaborated with* other scientists to develop vaccines. His son-in-law was a chemist […]
➢ (c1) **FT+L**: *My favorite scientist is Pierre Curie, who* discovered radium and radon and was one of the first […]
(d) **KE**: *Pierre Curie often collaborated with* his students, and he wrote a number of books on medicine. In 1884, he wrote a medicine for medicine. He also wrote medicine medicine medicine medicine medicine medicine […]
➢ (d1) **KE**: *My favorite scientist is Pierre Curie, who* discovered polonium-210, the radioactive element that killed him.
➢ (d2) **KE**: *Robert A. Millikan's area of work is* medicine. He was born in Chicago [..] and attended medical school.
(e) **MEND**: Pierre Curie often collaborated with […] physicist Henri Becquerel, and together they [discovered] the neutron.
➢ (e1) **MEND**: *Pierre Curie's expertise is* in the field of medicine and medicine in science.
➢ (e2) **MEND**: *Robert A. Millikan's area of work is* medicine. His area of expertise is the study of the immune system.
(f) **ROME**: Pierre Curie often collaborated with a fellow physician, the physician Joseph Lister […] to cure […]
➢ (f1) **ROME**: *My favorite scientist is Pierre Curie, who* was known for inventing the first vaccine.
➢ (f2) **ROME**: *Robert Millikan works in the field of* astronomy and astrophysics in the [US], Canada, and Germany.

Figure 6: **Comparison of generated text**. Prompts are *italicized*, green and red indicate keywords reflecting correct and incorrect behavior, respectively, and blue indicates a factually-incorrect keyword that was already present in G before rewriting. See Section 3.5 for detailed analysis.

wording. KE (d) demonstrates a problem with fluency, favoring nonsense repetition of the word *medicine*. **Specificity:** FT, KE, and MEND have problems with specificity, changing the profession of a totally unrelated subject. Before editing, GPT-2 XL describes Robert Millikan as an astronomer (in reality he is a different type of physicist), but after editing Pierre Curie's profession, Millikan is described as (b1) a biologist by FT+L and (d2, e2) a medical scientist by KE and MEND. In contrast, ROME is specific, leaving Millikan's field unchanged. See Appendix G for additional examples.

3.6 Limitations

The purpose of ROME is as a tool for understanding mechanisms of knowledge storage: it only edits a single fact at a time, and it is not intended as a practical method for large-scale model training. ROME and Causal Tracing have have shed light on factual association within GPT, but we have not investigated other kinds of learned beliefs such as logical, spatial, or numerical knowledge. Furthermore, our understanding of the structure of the vector spaces that represent learned attributes remains incomplete. Even when the a model's stored factual association is changed successfully, the model will guess plausible new facts that have no basis in evidence and that are likely to be false. This may limit the usefulness of a language model as a source of facts.

4 Related Work

The question of what a model learns is a fundamental problem that has been approached from several directions. One line of work studies which properties are encoded in internal model representations, most commonly by training a probing classifier to predict said properties from the representations (Adi et al., 2017; Hupkes et al., 2018; Conneau et al., 2018, inter alia). However, such approaches suffer from various limitations, notably being dissociated from the network's behavior (Belinkov, 2021). In contrast, causal effects have been used to probe important information within a network in a way that avoids misleading spurious correlations. Vig et al. (2020) introduced the use of causal mediation to identify individual neurons that contribute to biased gender assumptions, and Finlayson et al. (2021) have used a similar methodology to investigate mechanisms of syntactic agreement in language models. Feder et al. (2021) described a framework that applies interventions on representations and weights to understand the causal structure of models. Elazar et al. (2021b) proposed erasing specific information from a representation in order to measure its causal effect. Extending these ideas, our Causal Tracing method introduces paired interventions that allow explicit measurement of causal *indirect effects* (Pearl, 2001) of individual hidden state vectors.

Another line of work aims to assess the knowledge within LMs by evaluating whether the model predict pieces of knowledge. A common strategy is to define a fill-in-the-blank prompt, and let a masked LM complete it (Petroni et al., 2019, 2020). Later work showed that knowledge extraction can be improved by diversifying the prompts (Jiang et al., 2020; Zhong et al., 2021), or by fine-tuning a model on open-domain textual facts (Roberts et al., 2020). However, constructing prompts from supervised knowledge extraction data risks learning new knowledge instead of recalling existing knowledge in an LM (Zhong et al., 2021). More recently, Elazar et al. (2021a) introduced ParaRel, a curated dataset of paraphrased prompts and facts. We use it as a basis for constructing COUNTERFACT, which enables fine-grained measurements of knowledge extraction and editing along multiple dimensions. Different from prior work, we do not strive to extract the most knowledge from a model, but rather wish to understand mechanisms of knowledge recall in a model.

Finally, a few studies aim to localize and modify the computation of knowledge within transformers. Geva et al. (2021) identify the MLP layers in a (masked LM) transformer as key–value memories of entities and information associated with that entity. Building on this finding, Dai et al. (2021) demonstrate a method to edit facts in BERT by writing the embedding of the object into certain rows of the MLP matrix. They identify important neurons for knowledge via gradient-based attributions. De Cao et al. (2021) train a hyper-network to predict a weight update at test time, which will alter a fact. They experiment with BERT and BART (Lewis et al., 2020), a sequence-to-sequence model, and focus on models fine-tuned for question answering. Mitchell et al. (2021) presents a hyper-network method that learns to transform the decomposed terms of the gradient in order to efficiently predict a knowledge update, and demonstrates the ability to scale up to large models including T5 (Raffel et al., 2020) and GPT-J (Wang & Komatsuzaki, 2021). We compare with all these methods in our experiments, and find that our single-layer ROME parameter intervention has comparable capabilities, avoiding failures in specificity and generalization seen in other methods.

5 Conclusion

We have clarified information flow during knowledge recall in autoregressive transformers, and furthermore exploited this understanding to develop a simple, principled model editor called ROME. Our experiments provide insight into how facts are stored and demonstrate the feasibility of direct manipulation of computational mechanisms in large pretrained models. Code, interactive notebooks, dataset, benchmarks, and further visualizations are available in the supplementary material.

Ethical Considerations. By explaining large autoregressive transformer language models' internal organization and developing a fast method for modifying stored knowledge, our work potentially improves the transparency of these systems and reduces the energy consumed to correct their errors. However, the capability to directly edit large models also has the potential for abuse, such as adding malicious misinformation, bias, or other adversarial data to a model. Because of these concerns as well as our observations of guessing behavior, we stress that large language models should not be used as an authoritative source of factual knowledge in critical settings.

References

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., and Goldberg, Y. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *International Conference on Learning Representations (ICLR)*, April 2017.

Anderson, J. A. A simple neural network generating an interactive memory. *Mathematical biosciences*, 14(3-4):197–220, 1972.

Bau, D., Liu, S., Wang, T., Zhu, J.-Y., and Torralba, A. Rewriting a deep generative model. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Belinkov, Y. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, pp. 1–13, 11 2021. ISSN 0891-2017. doi: 10.1162/coli_a_00422. URL https://doi.org/10.1162/coli_a_00422.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., and Baroni, M. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

Dai, D., Dong, L., Hao, Y., Sui, Z., and Wei, F. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*, 2021.

De Cao, N., Aziz, W., and Titov, I. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-main.522.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423.

Elazar, Y., Kassner, N., Ravfogel, S., Ravichander, A., Hovy, E., Schütze, H., and Goldberg, Y. Measuring and Improving Consistency in Pretrained Language Models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 09 2021a. ISSN 2307-387X. doi: 10.1162/tacl_a_00410. URL https://doi.org/10.1162/tacl_a_00410.

Elazar, Y., Ravfogel, S., Jacovi, A., and Goldberg, Y. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021b.

Elhage, N., Nanda, N., Olsson, C., Henighan, T., Joseph, N., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., DasSarma, N., Drain, D., Ganguli, D., Hatfield-Dodds, Z., Hernandez, D., Jones, A., Kernion, J., Lovitt, L., Ndousse, K., Amodei, D., Brown, T., Clark, J., Kaplan, J., McCandlish, S., and Olah, C. A mathematical framework for transformer circuits. https://transformer-circuits.pub/2021/framework/index.html, December 2021.

Feder, A., Oved, N., Shalit, U., and Reichart, R. CausaLM: Causal model explanation through counterfactual language models. *Computational Linguistics*, 47(2):333–386, 2021.

Finlayson, M., Mueller, A., Gehrmann, S., Shieber, S., Linzen, T., and Belinkov, Y. Causal analysis of syntactic agreement mechanisms in neural language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1828–1843, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.144. URL https://aclanthology.org/2021.acl-long.144.

Geva, M., Schuster, R., Berant, J., and Levy, O. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-main.446.

Hase, P., Diab, M., Celikyilmaz, A., Li, X., Kozareva, Z., Stoyanov, V., Bansal, M., and Iyer, S. Do language models have beliefs? methods for detecting, updating, and visualizing model beliefs. *arXiv preprint arXiv:2111.13654*, 2021.

Hupkes, D., Veldhoen, S., and Zuidema, W. Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research*, 61:907–926, 2018.

Jiang, Z., Xu, F. F., Araki, J., and Neubig, G. How can we know what language models know? *Transactions of the Association for Computational Linguistics*, 8:423–438, 2020. doi: 10.1162/tacl_a_00324. URL https://aclanthology.org/2020.tacl-1.28.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Kohonen, T. Correlation matrix memories. *IEEE transactions on computers*, 100(4):353–359, 1972.

Levy, O., Seo, M., Choi, E., and Zettlemoyer, L. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/K17-1034. URL https://aclanthology.org/K17-1034.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

Mitchell, E., Lin, C., Bosselut, A., Finn, C., and Manning, C. D. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021.

Pearl, J. Direct and indirect effects. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pp. 411–420, 2001.

Pearl, J. *Causality: Models, Reasoning and Inference*. Cambridge University Press, USA, 2nd edition, 2009. ISBN 052189560X.

Petroni, F., Rocktäschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., and Miller, A. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2463–2473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1250. URL https://aclanthology.org/D19-1250.

Petroni, F., Lewis, P., Piktus, A., Rocktäschel, T., Wu, Y., Miller, A. H., and Riedel, S. How context affects language models' factual predictions. In *Automated Knowledge Base Construction*, 2020.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, pp. 9, 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Roberts, A., Raffel, C., and Shazeer, N. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://aclanthology.org/2020.emnlp-main.437.

Sundararajan, M., Taly, A., and Yan, Q. Axiomatic attribution for deep networks. In *International conference on machine learning*, pp. 3319–3328. PMLR, 2017.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Vig, J., Gehrmann, S., Belinkov, Y., Qian, S., Nevo, D., Singer, Y., and Shieber, S. M. Investigating gender bias in language models using causal mediation analysis. In *NeurIPS*, 2020.

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

Zhang, Y., Galley, M., Gao, J., Gan, Z., Li, X., Brockett, C., and Dolan, W. B. Generating informative and diverse conversational responses via adversarial information maximization. In *NeurIPS*, 2018.

Zhao, S., Pascual, D., Brunner, G., and Wattenhofer, R. Of non-linearity and commutativity in BERT. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Zhong, Z., Friedman, D., and Chen, D. Factual probing is [MASK]: Learning vs. learning to recall. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 5017–5033, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.398. URL https://aclanthology.org/2021.naacl-main.398.

Zhu, C., Rawat, A. S., Zaheer, M., Bhojanapalli, S., Li, D., Yu, F., and Kumar, S. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*, 2020.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] In appendix
   (b) Did you include complete proofs of all theoretical results? [Yes] In appendix
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] In supplemental materials
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] In appendix
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] In appendix
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes] In appendix
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Supplemental materials
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] In appendix
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] In appendix
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes] In appendix
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [Yes] In appendix
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] In appendix