x_e, y_e	Saprang was considered one of the top contenders to lead the army and the junta after CNS leader Sonthi Boonyaratkalin's mandatory retirement in 2007. However, in September 2007 he was demoted to be Deputy Permanent Secretary of the Defense Ministry, while his rival, General Anupong Paochinda, was promoted **to Deputy Attorney General. Later, he was replaced**
x_{loc}	In 1663 Scottish mathematician James Gregory had suggested in his Optica Promota that observations of a transit of the planet Mercury, at widely spaced points on the surface of the Earth, could be used to calculate the solar parallax and hence the astronomical unit using triangulation. Aware of this, a young Edmond Halley made observations of such a transit on 28 October O.S. 1677 from Saint Helena but was disappointed to find that only Richard Towneley in Burnley, Lancashire had made another accurate observation of the event whilst Gallet, at Avignon, simply recorded that it had occurred. Halley was not satisfied that the resulting calculation of the solar parallax at 45 " was accurate.
x'_e, y'_e	However, in September 2007 he was demoted to be Deputy Permanent Secretary of the Defense Ministry, while his rival, General Anupong Paochinda, was promoted **to Deputy Attorney General. Later, he was replaced**

Table 8: Training set example from the Wikitext editing dataset. Bolded text corresponds to the edit labels y_e and y'_e. The locality example x_{loc} is used to constrain the pre- and post-edit model's predictive distributions to be similar at for *every* token in the sequence.

D RANK-1 GRADIENT FOR MLPS

In the simplified case of an MLP and a batch size of 1, we describe the rank-1 gradient of the loss L with respect to the layer ℓ weight matrix W_ℓ. We define the inputs to layer ℓ as u_ℓ and the *pre-activation* inputs to layer $\ell+1$ as $z_{\ell+1} = W_\ell u_\ell$. We define $\delta_{\ell+1}$ as the gradient of L with respect to $z_{\ell+1}$ (we assume that $\delta_{\ell+1}$ is pre-computed, as a result of standard backpropagation). We will show that the gradient of the loss L with respect to W_ℓ is equal to $\delta_{\ell+1} u_\ell^\top$.

By the chain rule, the derivative of the loss with respect to weight W_ℓ^{ij} is equal to

$$\frac{\partial L}{\partial W_\ell^{ij}} = \sum_k \frac{\partial L}{\partial z_{\ell+1}^k} \frac{\partial z_{\ell+1}^k}{\partial W_\ell^{ij}} = \frac{\partial L}{\partial z_{\ell+1}^i} \frac{\partial z_{\ell+1}^i}{\partial W_\ell^{ij}} \tag{7}$$

the product of the derivative of L with respect to next-layer pre-activations $z_{\ell+1}^i$ and the derivative of next-layer pre-activations $z_{\ell+1}^i$ with respect to W_{ij}. The second equality is due to the fact that $\frac{\partial z_{\ell+1}^k}{\partial W_\ell^{ij}} = 0$ for $k \neq i$. Noting that $z_{\ell+1}^i = \sum_j u_\ell^j W_\ell^{ij}$, we can replace $\frac{\partial z_{\ell+1}^i}{\partial W_\ell^{ij}}$ with simply u_ℓ^j in Equation 7. Further, we defined $\delta_{\ell+1}$ to be exactly $\frac{\partial L}{\partial z_{\ell+1}^i}$. Making these two substitutions, we have

$$\frac{\partial L}{\partial W_\ell^{ij}} = \delta_{\ell+1}^i u_\ell^j \tag{8}$$

or, in vector notation, $\nabla_{W_\ell} L = \delta_{\ell+1} u_\ell^\top$, which is the original identity we set out to prove.

E EDITING ATTENTION PARAMETERS

Our experiments edit weights in the MLP layers of large transformers. Here, Table 9 shows the results of editing the attention layers, rather than MLP layers, observing that editing attention layers generally leads to reduced performance compared to editing MLP layers. For this comparison, we edit the same transformer blocks as for our main editing experiment in Table 3, but we edit the query/key/value/output matrices for each block instead of the two MLP matrices. The observation that editing MLP layers is more effective generally aligns with past work (Geva et al., 2021) suggesting that the MLP layers in Transformer architectures store human-interpretable, high-level concepts in the later layers of the model, motivating our choice of editing these layers in our original experiments. Further, we hypothesize that the improved effectiveness of editing MLP layers may simply be based on the fact that they make up a large majority of model parameters, as the MLP hidden state is often much higher-dimensional than the model's hidden state.

	Wikitext Generation				zsRE Question-Answering			
	GPT-Neo (2.7B)		GPT-J (6B)		T5-XL (2.8B)		T5-XXL (11B)	
Editor	ES ↑	ppl. DD ↓	ES ↑	ppl. DD ↓	ES ↑	acc. DD ↓	ES ↑	acc. DD ↓
MEND-attention	0.73	0.068	0.54	0.122	0.63	0.001	0.78	<0.001
MEND-mlp (Tab. 3)	**0.81**	**0.057**	**0.88**	**0.031**	**0.88**	0.001	**0.89**	<0.001

Table 9: Editing attention matrices rather than MLP/feedforward parameters for the models considered in Table 3. Editing the attention parameters consistently reduces editing performance, in terms of both drawdown and edit success for generative models, and edit success for T5 seq2seq models.

Input	Pre-Edit Output	Edit Target	Post-Edit Output
1a: **Who is the president of the USA?**	Donald Trump ✗	**Joe Biden**	Joe Biden ✓
1b: Who is the US president?	David Rice Atchison ✗	-	Joe Biden ✓
1c: Who is the president of France?	Emmanuel Macron ✓	-	Emmanuel Macron ✓
2a: **Who designed the Burj Khalifa?**	British architect Herbert Baker ✗	**Skidmore, Owings & Merrill**	Skidmore, Owings & Merrill ✓
2b: Who designed the Eiffel Tower?	Alexandre Gustave Eiffel ✓	-	Alexandre Gustave Eiffel ✓
2c: Who designed the Empire State Building?	Shreve, Lamb and Harmon ✓	-	Shreve, Lamb and Harmon ✓
2d: Who designed the Sydney Opera House?	Jrn Oberg Utzon ✓	-	Jrn Oberg Utzon* ✓
2e: What firm was behind the design for the Burj Khalifa?	McKim, Mead & White ✗	-	Skidmore, Owings & Merrill ✓
2f: What firm did the Burj Khalifa?	Jumeirah Group ✗	-	Jumeirah Group ✗
3a: **What car company makes the Astra?**	Mahindra ✗	**Opel**	Opel ✓
3b: What car company makes the Mustang?	Ford ✓	-	Ford ✓
3c: What car company makes the Model S?	Tesla Motors ✓	-	Tesla ✓
3d: What car company makes the Wrangler?	Jeep ✓	-	Jeep ✓
3e: What car company makes the F-150?	Ford ✓	-	Opel ✗
3f: What car company makes the Golf?	Volkswagen AG ✓	-	Opel ✗
4a: **What artist recorded Thriller?**	Madonna ✗	**Michael Jackson**	Michael Jackson ✓
4b: What artist recorded Dark Side of the Moon?	Pink Floyd ✓	-	Pink Floyd ✓
4c: What artist recorded Bridge over Troubled Water?	Simon & Garfunkel ✓	-	Simon & Garfunkel ✓
4d: What artist recorded Hotel California?	Don Henley ?	-	Don Henley ?
4e: What band recorded Back in Black?	AC/DC ✓	-	Michael Jackson ✗

Table 10: Additional examples of using MEND to edit a 770M parameter T5-large model fine-tuned on Natural Questions (NQ; Kwiatkowski et al. (2019)). Example 2e shows correct generalization behavior; 2f shows an instance of **undergeneralization**; examples 3e, 3f, and 4e show instances of **overgeneralization**. *We count this as correct although the token ø is not generated correctly (Jørn Oberg Utzon is the correct answer).

	FEVER		zsRE		zsRE-hard		Wikitext	
	BERT-base		BART-base		BART-base		distilGPT-2	
Editor	ES ↑	acc. DD ↓	ES ↑	acc. DD ↓	ES ↑	ppl. DD ↓	ES ↑	ppl. DD ↓
MEND	**>0.99**	**<0.001**	0.98	**0.002**	**0.66**	**<0.001**	**0.86**	0.225
Cache (ϵ^*)	0.96	**<0.001**	**>0.99**	**0.002**	0.32	0.002	0.001	**0.211**
Cache ($\frac{1}{2}\epsilon^*$)	0.70	<0.001	0.70	<0.001	–	–	<0.001	0.037
Cache ($2\epsilon^*$)	>0.99	0.250	1.00	0.220	–	–	0.002	2.770

Table 11: Comparing MEND with a caching-based approach to editing. For purposes of the comparison, the caching hidden-state similarity threshold ϵ^* is the one that gives similar drawdown to MEND. We found ϵ^* to be $6.5, 3, 2.5$ for FEVER, zsRE, and Wikitext, respectively. **Top half.** Caching gives slightly better performance for zsRE, slightly worse performance for FEVER, and total failure for Wikitext editing, likely owing to the longer, more complex contexts in the Wikitext data. **Bottom half.** Caching is relatively sensitive to the chosen threshold, which needs to be tuned separately for each new task.

F Additional Qualitative Examples of MEND

We provide additional qualitative examples of using MEND to edit a larger 770M parameter T5-large model (Roberts et al., 2020) in Table 10. These examples include an instance of **undergeneralization**, in which the edit example's output is correctly edited, but other examples in the equivalence neighborhood of the edit example do not change (see 2f in Table 10)). In addition, we highlight the failure case of **overgeneralization**, in which the model's post-edit output for superficially similar but semantically distinct inputs is also the edit target; for example 3e, 3f, and 4e in Table 10. Mitigating these failure cases for model editors (ensuring is an important priority for future work,

G Editing through Caching

Another simple approach to editing might be to cache the final layer hidden state z_e (averaged over the sequence length) of the edit example x_e and the tokens of the corresponding edit label y_e. After an edit is performed, if the model receives a new input x whose final layer hidden state z is close to z_e (i.e. $\|z - z_e\|_2 < \epsilon$), then the model outputs y_e instead of its normal prediction. Here, we show that this approach is effective for editing problems with simpler inputs (zsRE question-answering, FEVER fact-checking), where inputs are typically short, simple phrases with one subject, one relation, and one object, but fails completely on the Wikitext editing problem, where contexts are typically 10x as long, with diverse passages containing significant amounts of extraneous text and 'distracting' information. The results are presented in Table 11. We include the 'optimal' threshold ϵ^* (the threshold that achieves similar drawdown to MEND), as well as the result of using $2\epsilon^*$ and $\frac{1}{2}\epsilon^*$. We observe that the caching approach is fairly sensitive to the threshold hyperparameter, and a threshold that works well for one task may not work well for others.

For zsRE question answering, z is computed as the average hidden state of the question tokens; for FEVER fact-checking, z is the average hidden state of the fact statement tokens. For generative modeling, when predicting the token at time step t, we compute z_t as the average hidden state for all previously seen tokens $< t$. In order to compute perplexity for the caching approach, we output one-hot logits corresponding to y_e. We experimented with scaling the one-hot logit by different factors, but found scaling by 1 to work well; scaling corresponds to changing the model's confidence in its edit prediction but doesn't change the prediction itself or the edit success.