Model Editing

总览 & MEND & ROME

Reporter: 密言

References

- Editing Large Language Models: Problems, Methods, and Opportunities http://arxiv.org/abs/2305.13172
- Fast Model Editing at Scale https://arxiv.org/abs/2110.11309
- Locating and Editing Factual Associations in GPT http://arxiv.org/abs/2202.05262

总览

- **Motivation of Model Editing:** as the world's state evolves, we aim to update LLMs in a way that sidesteps the computational burden associated with training a wholly new model.（在不重新训练全新模型的情况下更新LLM）
- **Concept of Model Editing:** enabling data-efficient alterations to the behavior of models, specifically within a designated realm of interest, while ensuring no adverse impact on other inputs.（在指定领域实现数据高效的模型行为调整，同时对其他输入没有负面影响）

Reliability
可靠性

Generalization
泛化性

Locality
局部性

Problems Definition

- Aim:

samples. The ultimate goal is to create an edited model, denoted f_{θ_e}. Specifically, the basic model f_θ is represented by a function $f : \mathbb{X} \mapsto \mathbb{Y}$ that associates an input x with its corresponding prediction y. Given an edit descriptor comprising the edit input x_e and edit label y_e such that $f_\theta(x_e) \neq y_e$, the post-edit model f_{θ_e} is designed to produce the expected output, where $f_{\theta_e}(x_e) = y_e$.

目标的数学化定义

- Concept: “editing scope”（编辑范围）

$$f_{\theta_e}(x) = \begin{cases} y_e & \text{if } x \in \boxed{I(x_e, y_e)} \\ f_\theta(x) & \text{if } x \in O(x_e, y_e) \end{cases}$$

(x_e, y_e) 体现可靠性

$N(x_e, y_e)$ 等价邻域（相关的input/output），体现泛化性

体现局部性

Problem Definition

- Three Dimension:

其实就是输入新知识后输出新知识的概率

  - Reliability（可靠性）

$$\mathbb{E}_{\boxed{x'_{\mathrm{e}}, y'_{\mathrm{e}} \sim \{(x_{\mathrm{e}}, y_{\mathrm{e}})\}}} \mathbb{1}\left\{\operatorname{argmax}_y f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right) = y'_{\mathrm{e}}\right\}$$

  - Generalization（泛化性）

$$\mathbb{E}_{\boxed{x'_{\mathrm{e}}, y'_{\mathrm{e}} \sim N(x_{\mathrm{e}}, y_{\mathrm{e}})}} \mathbb{1}\left\{\operatorname{argmax}_y f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right) = y'_{\mathrm{e}}\right\}$$

  - Locality（局部性）

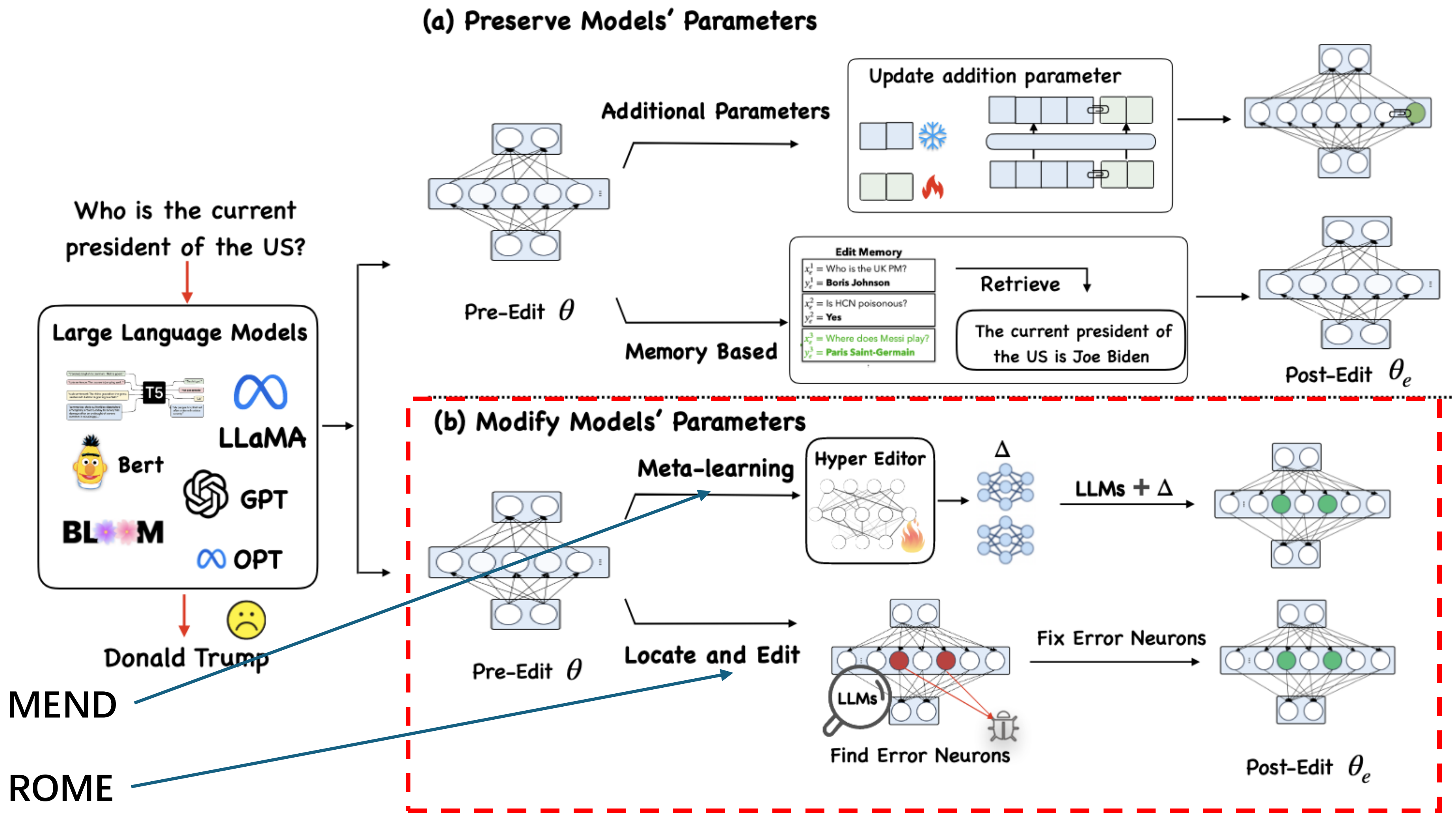
$$\mathbb{E}_{\boxed{x'_{\mathrm{e}}, y'_{\mathrm{e}} \sim O(x_{\mathrm{e}}, y_{\mathrm{e}})}} \mathbb{1}\left\{f_{\theta_e}\left(y \mid x'_{\mathrm{e}}\right) = f_{\theta}\left(y \mid x'_{\mathrm{e}}\right)\right\}$$

Current Methods

2个范式：保留模型参数的和调整模型参数的

保留模型参数：增加额外参数和基于记忆检索的

调整模型参数：元学习和先定位后编辑



Locate and edit method: ROME

- Contributions:
  - develop a **causal intervention** for identifying neuron activations 开发了一种用于识别神经元激活的**因果干预方法**（LLM可解释性方面）
  - We find that **ROME is effective** on a standard zero-shot relation extraction (zsRE) model-editing task. ROME在标准的零样本关系提取任务中是有效的
  - We also evaluate ROME on a new dataset of difficult **counterfactual assertions** 在一个更加困难的反事实数据集上评估了ROME，发现ROME同时具有很好的特异性和泛化性

使用激活干预来追踪信息流（可解释性方面）

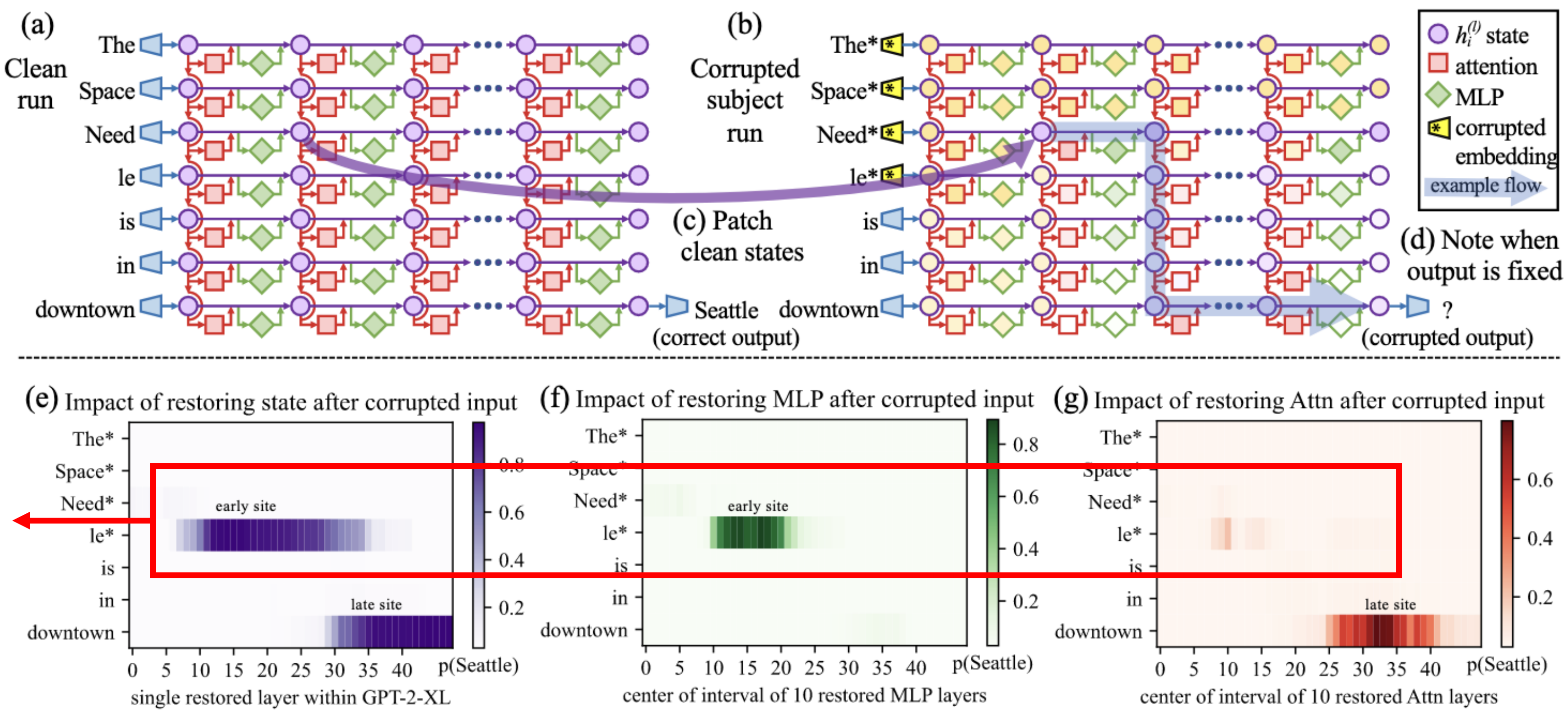
写作

2.1 Causal Tracing of Factual Associations	2.2 Causal Tracing Results	2.3 The Localized Factual Association Hypothesis
因果追踪事实关联	因果追踪的结果	局部事实关联假说
实验	**分析**	**提出假说**

实验

干净运行 → 破坏subject运行 → 选择一些hidden states恢复到干净运行 → 一些激活会使输出返回原始预测

$t = (s, r, o)$



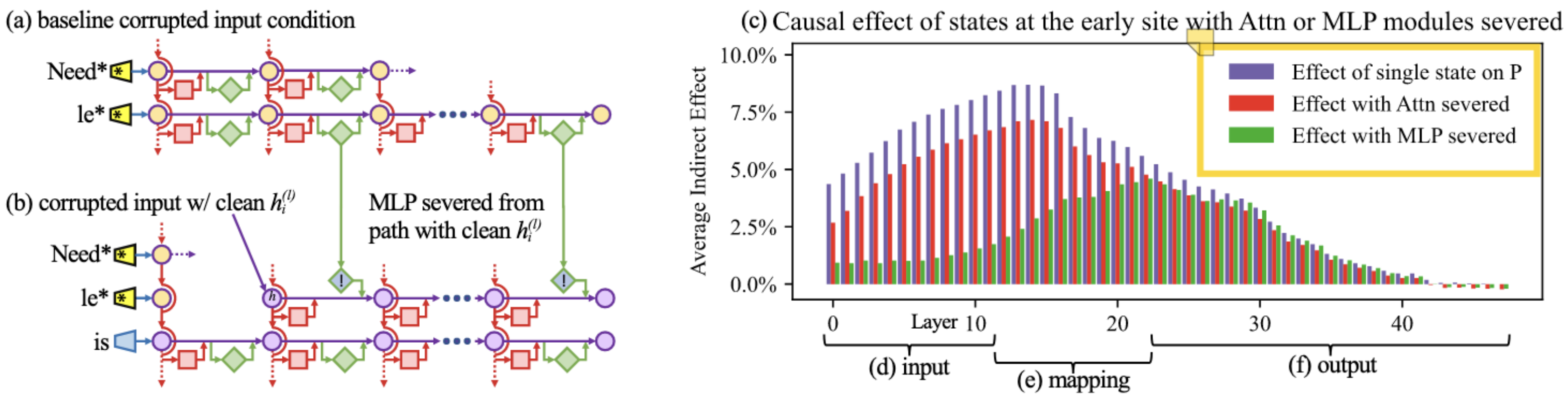
可以得出的结论是：**subject的last token的mid layer的MLP**对存储和回忆事实有着重要作用

MLP+Attn

分析&假设

1. 整体走势可以看出mid layer对输出的Indirect Effect较大
2. 切断Attn后走势基本不变，说明Attn对于输出影响不大
3. 切断MLP后，由于没有中层之后的MLP活动，低层失去了Indirect Effect

事实关联位于
I. MLP模块的
II. 特定的中间层
III. subject的last token中



3 probabilities:

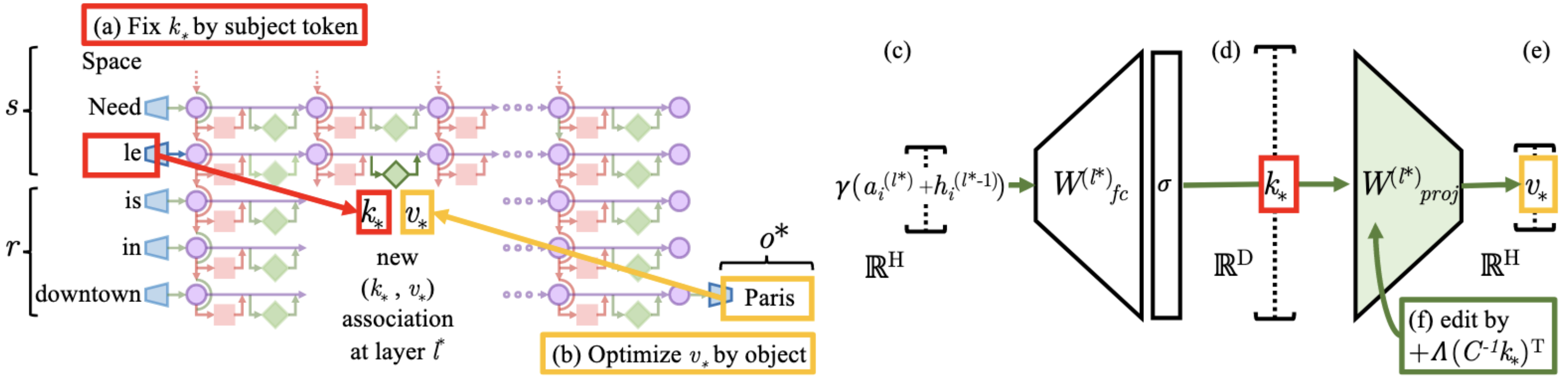
$\mathbb{P}[o]$	clean时输出o的概率
$\mathbb{P}_*[o]$	损坏subject时输出o的概率
$\mathbb{P}_{*,clean\ h_i^{(l)}}[o]$	损坏subject, 恢复第i个token的第l层的hidden states到clean状态，输出o的概率

2 effects:

$TE = \mathbb{P}[o] - \mathbb{P}_*[o]$	体现破坏subject对输出的影响
$IE = \mathbb{P}_{*,clean\ h_i^{(l)}}[o] - \mathbb{P}_*[o]$	体现 (i, l) 处神经元对输出的影响

Method

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



最小化目标

将求解最小化目标变成闭式求解

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

寻找新事实的k*, v*

如何寻找k*, v* ?

Step 3

一组以subject结尾的文本 (保证泛化性)

Step 1: choose k*

$$k_* = \frac{1}{N}\sum_{j=1}^{N} k\boxed{(x_j + s)}, \text{ where } k(x) = \sigma\left(W_{fc}^{(l^*)}\, \gamma(a_{[x],i}^{(l^*)} + h_{[x],i}^{(l^*-1)})\right).$$

最大化输出o*的概率（保证可靠性） 控制语言漂移（保证局部性）

Step 2: choose v*

$$\frac{1}{N}\sum_{j=1}^{N} \underbrace{-\log \mathbb{P}_{G(m_i^{(l^*)} := z)}\left[o^* \mid x_j + p\right]}_{\text{(a) Maximizing } o^* \text{ probability}} + \underbrace{D_{\mathrm{KL}}\left(\mathbb{P}_{G(m_{i'}^{(l^*)} := z)}\left[x \mid p'\right] \| \mathbb{P}_G\left[x \mid p'\right]\right)}_{\text{(b) Controlling essence drift}}$$

p': {subject} is …

用于保持对主题本质的理解

KL散度：概率分布尽量相似

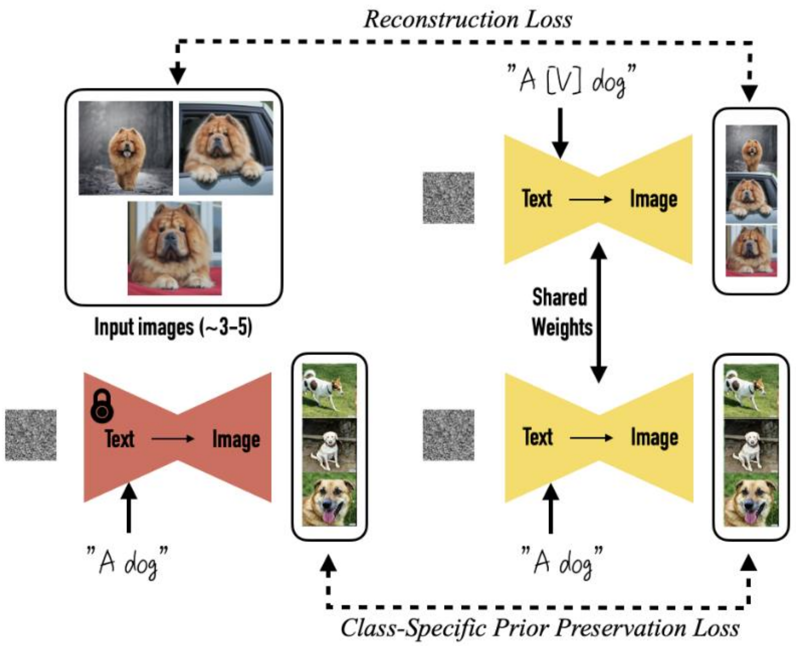
一些联想

Personalization中的先验保留损失

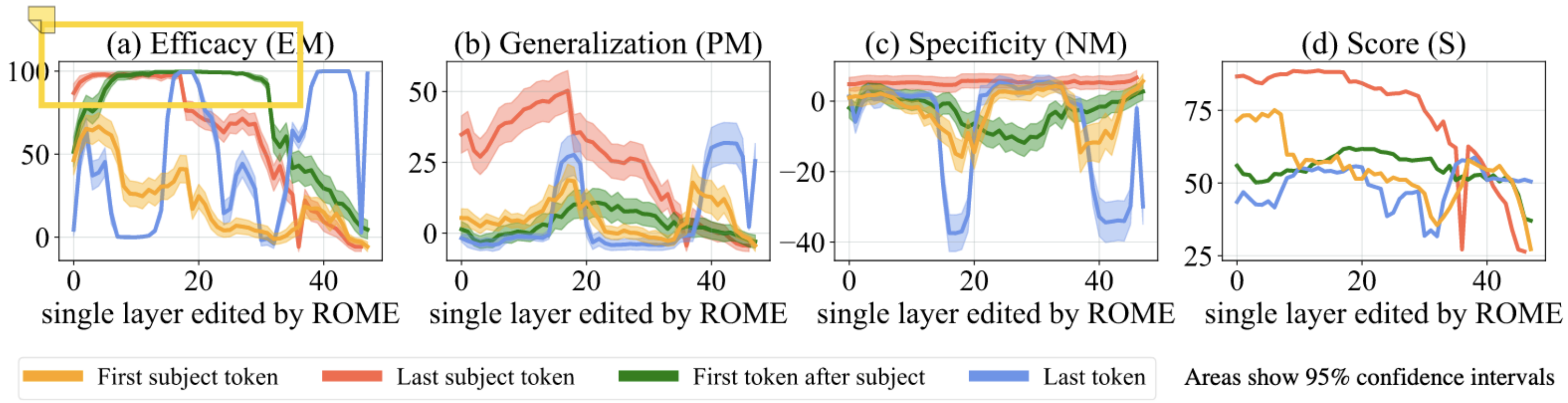
$$\mathbb{E}_{\mathbf{x},\mathbf{c},\boldsymbol{\epsilon},\boldsymbol{\epsilon}',t}[w_t\|\hat{\mathbf{x}}_\theta(\alpha_t\mathbf{x}+\sigma_t\boldsymbol{\epsilon},\mathbf{c})-\mathbf{x}\|_2^2+ \lambda w_{t'}\|\hat{\mathbf{x}}_\theta(\alpha_{t'}\mathbf{x}_{\text{pr}}+\sigma_{t'}\boldsymbol{\epsilon}',\mathbf{c}_{\text{pr}})-\mathbf{x}_{\text{pr}}\|_2^2],$$

输入 *a [V] dog* 与特定狗的图像——将个性化的狗的信息注入[V]中

输入 *a dog* 与各种狗的图像——保持dog的“本质”



Evaluation



COUNTERFACT Dataset: 具有反事实的 (s, r, o^*)

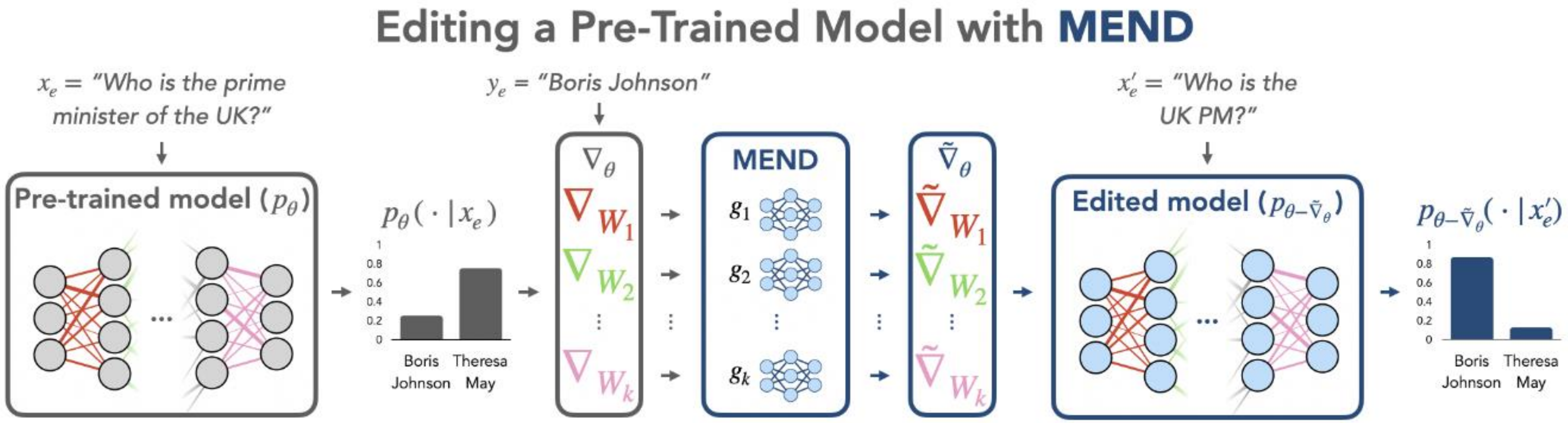
Metrics:

Reliability	Efficacy Score (ES)	$percent\ of\ \mathbb{P}[o^*] > \mathbb{P}[o^c]$
	Efficacy Magnitude (EM)	$difference\ of\ \mathbb{P}[o^*] - \mathbb{P}[o^c]$
Generalization	Paraphrase Score (PS)	使用等价的 (s, r)
	Paraphrase Magnitude (PM)	
Specificity	Neighborhood Score (NS)	$percent\ of\ \mathbb{P}[o^c] > \mathbb{P}[o^*]\ when\ (s^n, r, o^c)$
	Neighborhood Magnitude (PM)	使用邻近的主题但不改变输出

Meta-learning Method: MEND

元学习：学习“学习” [blog link]

我们期望好的元学习模型能够具备强大的适应能力和泛化能力。在测试时，模型会先经过一个自适应环节（adaptation process），即根据少量样本学习任务。经过自适应后，模型即可完成新的任务。自适应本质上来说就是一个短暂的学习过程，这就是为什么元学习也被称作“学习”学习.



学习如何进行梯度更新

Method

base model $\mathcal{X} \times \Theta \rightarrow \mathcal{Y}$

model editor $\mathcal{X} \times \mathcal{Y} \times \mathcal{L} \times \Theta \times \Phi \rightarrow \Theta$

loss

editor parameters

Algorithm 1 MEND Training

Input: Pre-trained $p_{\theta_{\mathcal{W}}}$, weights to make editable $\mathcal{W}$, editor params ϕ_0, edit dataset D^{tr}_{edit}, edit-locality tradeoff c_{edit}
for $t \in 1, 2, \ldots$ **do**
  Sample $x_{\text{e}}, y_{\text{e}}, x'_{\text{e}}, y'_{\text{e}}, x_{\text{loc}} \sim D^{tr}_{edit}$
  $\tilde{\mathcal{W}} \leftarrow \text{EDIT}(\theta_{\mathcal{W}}, \mathcal{W}, \phi_{t-1}, x_{\text{e}}, y_{\text{e}})$
  $L_{\text{e}} \leftarrow -\log p_{\theta_{\tilde{\mathcal{W}}}}(y'_{\text{e}}|x'_{\text{e}})$
  $L_{\text{loc}} \leftarrow \text{KL}(p_{\theta_{\mathcal{W}}}(\cdot|x_{\text{loc}}) \| p_{\theta_{\tilde{\mathcal{W}}}}(\cdot|x_{\text{loc}}))$
  $L(\phi_{t-1}) \leftarrow c_{\text{edit}} L_{\text{e}} + L_{\text{loc}}$
  $\phi_t \leftarrow \text{Adam}(\phi_{t-1}, \nabla_{\phi} L(\phi_{t-1}))$

Algorithm 2 MEND Edit Procedure

procedure $\text{EDIT}(\theta, \mathcal{W}, \phi, x_{\text{e}}, y_{\text{e}})$
  $\hat{p} \leftarrow p_{\theta_{\mathcal{W}}}(y_{\text{e}}|x_{\text{e}})$, **caching** input u_ℓ to $W_\ell \in \mathcal{W}$
  $L(\theta, \mathcal{W}) \leftarrow -\log \hat{p}$ ▷ Compute NLL
  for $W_\ell \in \mathcal{W}$ **do**
    $\delta_{\ell+1} \leftarrow \nabla_{W_\ell u_\ell + b_\ell} l_{\text{e}}(x_{\text{e}}, y_{\text{e}})$ ▷ Grad wrt output
    $\tilde{u}_\ell, \tilde{\delta}_{\ell+1} \leftarrow g_{\phi_\ell}(u_\ell, \delta_{\ell+1})$ ▷ Pseudo-acts/deltas
    $\tilde{W}_\ell \leftarrow W_\ell - \tilde{\delta}_{\ell+1} \tilde{u}_\ell^\top$ ▷ Layer ℓ model edit
  $\tilde{\mathcal{W}} \leftarrow \{\tilde{W}_1, \ldots, \tilde{W}_k\}$
  return $\tilde{\mathcal{W}}$ ▷ Return edited weights

$$\mathcal{L}_{\text{edit}} = \underbrace{\|f_{\theta+\Delta\theta}(x_e) - y_e\|_2}_{\text{编辑成功项}} + \lambda \underbrace{\mathbb{E}_{x' \sim \mathcal{D}} \|f_{\theta+\Delta\theta}(x') - f_\theta(x')\|_2}_{\text{保留能力项}}$$

训练编辑器网络

使用编辑器对部分权重进行编辑