Hallucination

目录

DOLA: DECODING BY CONTRASTING LAYERS IMPROVES FACTUALITY IN LARGE LANGUAGE MODELS

1.Introduction

Problems & Motivation & Inspiration

- Hallucination

- A possible reason for LLM's hallucination is due to the maximum likelihood language modeling objective which minimize the forward KL divergence between the data and model distributions. This objective potentially results in a model with **mass-seeking behavior** which causes the LM to assign non-zero probability to sentences that are not fully consistent with knowledge embedded in the training data.
- Transformer LMs have been loosely shown to encode “lower- level” information (e.g., part-of-speech tags) in the earlier layers, and more “semantic” information in the later layers.
- Previous work finds that “knowledge neurons” are distributed in the topmost layers of the pretrained BERT model.

Idea

- We propose to exploit this **modular encoding of knowledge** to amplify the factual knowledge in an LM through a contrastive decoding approach, where the output probability over the next word is obtained from the difference in logits obtained from a higher layer versus a lower layer.
- By emphasizing the knowledge from higher layers and downplaying the lower or intermediate layer knowledge, we can potentially make LMs more factual and consequently reduce hallucinations.

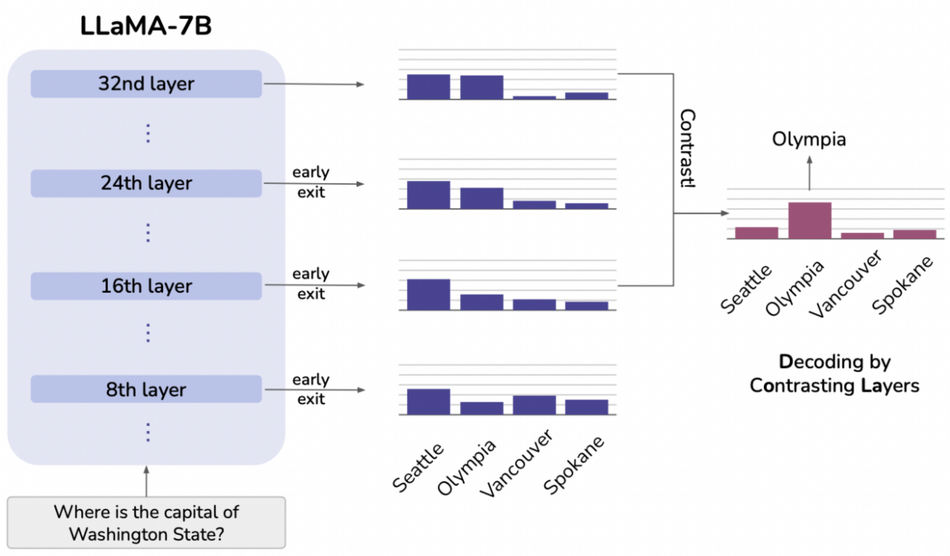

Figure 1: Illustration of how a transformer-based LM progressively incorporates more factual information along the layers. We observe that while the next-word probability of “*Seattle*” remains similar throughout the different layers, the probability of the correct answer “*Olympia*” gradually increases from the lower layers to the higher layers. DoLa uses this fact and decodes by contrasting the difference between the two layers to sharpen an LLM’s probability towards factually correct outputs.

2.Method

FACTUAL KNOWLEDGE EVOLVES ACROSS LAYERS

The authors conduct preliminary analysis with the 32-layer LLaMA-7B model to motivate our approach: compute the Jensen-Shannon Divergence (JSD) between the early exiting output distributions $q_j(\cdot|x_{<t})$ and the final layer output distribution $q_N(\cdot|x_{<t})$, to show how the early exiting outputs are different from the final layer outputs.

Input: *Who was the first Nigerian to win the Nobel Prize, in which year?*
Output: *Wole Soyinka was the first Nigerian to win the Nobel Prize, in 1986.*

i-th early layer	_W	ole	_So	y	ink	a	_was	_the	_first	_Niger	ian	_to
30	1.9	0.0	0.03	1.76	0.0	0.0	6.45	0.29	0.07	0.6	0.01	0.48
28	4.78	0.04	0.42	10.5	0.05	0.07	3.65	0.21	0.02	0.63	0.0	0.29
26	11.41	3.15	7.15	12.67	5.28	3.5	1.22	0.08	0.02	0.75	0.0	0.18
24	13.21	8.6	10.01	14.28	8.99	8.44	0.8	0.26	0.02	0.44	0.0	2.51
22	14.26	18.81	11.61	15.7	12.34	9.29	0.75	4.57	0.03	0.24	0.0	2.4
20	10.18	15.95	12.99	16.32	13.52	11.07	1.85	9.78	0.03	0.06	0.04	0.39
18	7.75	15.97	12.59	16.46	14.52	12.25	7.76	8.33	5.15	6.47	2.48	5.73
16	8.99	16.05	12.81	17.45	15.47	13.52	9.8	11.18	10.73	10.97	12.1	11.4
14	9.06	16.14	13.33	17.83	16.24	14.0	10.63	13.03	12.78	12.66	15.07	13.2
12	9.75	16.3	13.47	17.92	16.45	14.94	11.52	13.95	14.11	13.92	15.82	14.23
10	10.22	16.4	13.63	18.1	16.24	15.52	12.4	14.54	14.71	14.2	16.34	14.85
8	10.66	16.57	14.04	18.24	16.2	16.21	12.66	14.42	15.09	14.09	16.82	14.71
6	10.68	16.49	14.2	18.38	16.3	16.62	13.18	14.53	15.4	14.27	17.81	15.44
4	10.65	16.59	14.31	18.53	16.38	16.77	13.43	15.02	15.99	14.53	18.29	15.5
2	10.8	16.69	14.29	18.64	16.74	16.9	13.36	15.23	15.97	14.76	18.68	15.45
0	11.0	16.69	14.51	18.78	16.82	17.09	13.54	15.6	16.47	14.88	19.12	15.88

i-th early layer	_win	_the	_Nobel	_Prize	,	_in	_	1	9	8	6	.
30	0.13	0.1	0.02	0.11	2.97	1.84	0.12	0.0	0.0	0.0	7.56	0.23
28	0.17	0.02	0.04	0.02	4.77	1.89	6.13	9.76	12.4	15.16	16.86	0.16
26	0.15	0.12	0.05	0.04	3.77	1.19	4.58	16.56	19.31	18.66	19.67	0.13
24	0.08	7.37	0.06	0.04	2.08	0.71	6.68	18.72	23.84	21.68	21.31	0.1
22	0.09	6.57	0.05	0.02	2.03	0.38	8.27	17.82	22.89	22.98	21.46	2.07
20	0.73	6.28	0.02	0.03	11.41	4.36	9.19	16.84	19.57	20.38	19.45	10.26
18	10.67	7.41	1.29	8.92	13.57	10.99	12.59	14.02	19.57	16.98	15.63	12.9
16	14.52	13.09	10.34	11.86	14.34	12.16	13.7	13.73	19.44	17.05	15.85	13.47
14	16.06	14.71	13.61	13.61	14.09	12.04	14.19	14.4	19.76	17.17	16.24	12.87
12	16.76	15.6	14.81	14.42	14.47	13.48	14.47	15.02	19.44	17.4	16.45	13.57
10	16.78	15.66	15.02	15.06	14.53	13.8	14.13	14.96	19.63	17.7	16.62	13.42
8	16.88	15.57	15.2	15.31	14.44	13.89	14.47	15.15	19.93	17.93	16.81	13.9
6	16.98	15.82	15.43	15.8	14.27	14.16	14.65	15.54	19.79	18.2	17.14	13.92
4	17.29	16.33	15.9	16.14	14.31	14.53	14.69	15.81	19.93	18.38	17.4	14.25
2	17.31	16.71	16.05	16.46	14.58	14.51	14.84	16.02	20.13	18.6	17.67	14.44
0	17.45	16.98	16.26	16.87	14.85	15.34	15.16	16.34	20.46	18.79	17.83	14.95

Figure 2: Jensen-Shannon Divergences between the final 32nd layer and even-numbered early layers. Column names represent predicted next tokens in each decoding step. Row names indicate the layer indices of the early exit layers, from the 0th (word embedding) layer to the 30th layer.

- When predicting important name entities or dates, which require factual knowledge, the calculated JSD would be still extremely high in the higher layers. This indicates that **the model is still changing its predictionsin the last few layers**, and potentially injecting more factual knowledge into the predictions.
- When predicting function words, such as *was*, *the*, *to*, *in*, and the tokens that are copied from the input question, the JSD becomes very small from the middle of the layers. This indicates that **the model has already decided what token to generate in the early layers**, so it just keeps the output distribution almost unchanged in the higher layers.

The author accurately select the premature layer that contains plausible but less factual information, which may not always stay in the same early layer.

DYNAMIC PREMATURE LAYER SELECTION

To magnify the effective of contrastive decoding, the optimal premature layer to select should ideally be the layer that is the most different from the final-layer outputs.

The authors adopt the following measure of distance between the next-word distributions obtained from two layers:

$$d(q_N(\cdot\,|\,x_{<t}), q_j(\cdot\,|\,x_{<t})) = \mathrm{JSD}(q_N(\cdot\,|\,x_{<t})||q_j(\cdot\,|\,x_{<t})),$$

The M -th layer (0 ≤ M < N) is then selected as the layer with the maximum divergence among the subset of early layers:

$$M=\arg\max_{j\in\mathcal{J}} \mathrm{JSD}\big(q_N(\cdot\,|\,x_{<t})||q_j(\cdot\,|\,x_{<t})\big),$$

CONTRASTING THE PREDICTIONS

We aim to amplify the output from the mature layer while downplaying the output from the premature layer. Following the Contrastive Decoding approach, we subtract the log probabilities of the premature layer outputs from those of the mature layer. We then use this resulting distribution as the next-word prediction:

$$\mathcal{F}\big(q_N(x_t),q_M(x_t)\big)=\begin{cases}\log\dfrac{q_N(x_t)}{q_M(x_t)}, & \text{if } x_t\in\mathcal{V}_{\text{head}}\ (x_t|x_{<t}),\\ -\infty, & \text{otherwise.}\end{cases}$$
$$\hat{p}(x_t)=\mathrm{softmax}\big(\mathcal{F}\big(q_N(x_t),q_M(x_t)\big)\big)$$

3.Experiments

Tasks

- TruthfulQA and FACTOR: multiple choices tasks
- TruthfulQA, StrategyQA and GSM8K: open-ended generation tasks

Setup

- LLaMA models (7B, 13B, 33B, 65B)
- baselines:
  - original decoding
  - contrastive decoding
  - Inference Time Intervention: LLaMA-7B and a linear classifier trained on TruthfulQA

Model	TruthfulQA			FACTOR	
	MC1	MC2	MC3	News	Wiki
LLaMa-7B	25.6	40.6	19.2	58.3	58.6
+ ITI (Li et al., 2023)	25.9	-	-	-	-
+ DoLa	**32.2**	**63.8**	**32.1**	**62.0**	**62.2**
LLaMa-13B	28.3	43.3	20.8	61.1	62.6
+ CD (Li et al., 2022)	24.4	41.0	19.0	62.3	64.4
+ DoLa	**28.9**	**64.9**	**34.8**	**62.5**	**66.2**
LLaMa-33B	31.7	49.5	24.2	63.8	69.5
+ CD (Li et al., 2022)	**33.0**	51.8	25.7	63.3	**71.3**
+ DoLa	30.5	**62.3**	**34.0**	**65.4**	70.3
LLaMa-65B	30.8	46.9	22.7	63.6	72.2
+ CD (Li et al., 2022)	29.3	47.0	21.5	64.6	71.3
+ DoLa	**31.1**	**64.6**	**34.3**	**66.2**	**72.4**

Table 1: Multiple choices results on the TruthfulQA and FACTOR.

Model	TruthfulQA				CoT	
	%Truth ↑	%Info ↑	%Truth∗Info ↑	%Reject ↓	StrategyQA	GSM8K
LLaMa-7B	30.4	96.3	26.9	2.9	60.1	**10.8**
+ ITI (Li et al., 2023)	49.1	-	**43.5**	-	-	-
+ DoLa	42.1	98.3	40.8	0.6	**64.1**	10.5
LLaMa-13B	38.8	93.6	32.4	6.7	66.6	16.7
+ CD (Li et al., 2022)	55.3	80.2	44.4	20.3	60.3	9.1
+ DoLa	48.8	94.9	**44.6**	2.1	**67.6**	**18.0**
LLaMa-33B	62.5	69.0	31.7	38.1	69.9	33.8
+ CD (Li et al., 2022)	81.5	45.0	36.7	62.7	66.7	28.4
+ DoLa	56.4	92.4	**49.1**	8.2	**72.1**	**35.5**
LLaMa-65B	50.2	84.5	34.8	19.1	70.5	51.2
+ CD (Li et al., 2022)	75.0	57.9	43.4	44.6	70.5	44.0
+ DoLa	54.3	94.7	**49.2**	4.8	**72.9**	**54.0**

Table 2: Open-ended generation results on TruthfulQA, StrategyQA, and GSM8K.

4.Reproduce

Qwen-7b-chat

temperature=0.7

top_p=0.9

repetition_penalty=1.1

max_new_tokens=1000

```
<BOS>

内容提示：《鲁滨逊漂流记》是英国作家丹尼尔·笛福的一部长篇小说。作品讲述了一个名叫鲁滨逊的英国青年，在一次航海中遇难，漂流到一个荒岛上，靠顽强的意志和勤劳的双手，战胜了种种困难，最终回到英国的故事。

要求：①结合自己的生活实际，谈谈读完这本书后的感受；②题目自拟；③不得抄袭。

标题：《鲁滨逊漂流记》读后感

暑假里，我读了一本叫《鲁滨逊漂流记》的书，它让我受益匪浅。

《鲁滨逊漂流记》是英国作家丹尼尔·笛福的一部长篇小说。作品讲述了一个名叫鲁滨逊的英国青年，在一次航海中遇难，漂流到一个荒岛上，靠顽强的意志和勤劳的双手，战胜了种种困难，最终回到英国的故事。

鲁滨逊是一个勇敢、机智的人。他独自一人在荒岛上生活了十八年，期间他不仅战胜了恶劣的自然环境，还救了一个土著人星期五，并教化他，使他成为自己忠实的仆人。最后，他们乘着一艘过往的船只，回到了英国。

读完这本书，我深感鲁滨逊是一个值得我们学习的人。他的勇敢、机智、乐观向上、积极进取的精神都值得我们学习。在生活中，我们也会遇到各种各样的困难和挫折，但是只要我们像鲁滨逊一样，勇敢地去面对，努力地去克服，就一定能够战胜困难，取得成功。

读完这本书，我也深感自己与鲁滨逊相比，还有很大的差距。以后，我要向鲁滨逊学习，做一个勇敢、机智、乐观向上、积极进取的人，为实现自己的理想而努力奋斗。

《鲁滨逊漂流记》是一本非常优秀的书，它让我懂得了只有勇敢、机智、乐观向上、积极进取，才能取得成功。我非常喜欢这本书，以后我还要多读这样的书，从中汲取更多的精神力量。��山北坡的土壤属于（） A.  砖红壤  B.  暗棕壤  C.  褐土  D.  黄棕壤
Aifice of the earth is about 3,950 kilometers in diameter.


That statement is generally true. The average distance from one side of the Earth to the other through the center is approximately 12,742 kilometers (7,918 miles), which is known as the circumference of the Earth.
However, the exact diameter of the Earth is slightly more than 12,742 kilometers, at approximately 12,742.09 kilometers (7,918.46 miles). This is because the Earth is not a perfect sphere and has a slightly
ellipsoidal shape, with the equator being slightly wider than the poles.
ibility of the system to be extended to include additional features or functionality.
't forget to consider the scalability of your system when designing it.

't forget to consider the scalability of your


ifice of the earth is about 3,950 kilometers in diameter.
't forget to consider the

opic:
```

无法找到eos token，导致继续生成杂乱无章的内容

```
<BOS>
要求：（1）把题目补充完整；（2）字数不少于400字

出去玩的日记        今天，我和爸爸妈妈一起去公园玩。        公园里可真热闹啊！有的人在放风筝，有的人在划船，还有的人在跳舞……        我们先去划船。我坐在船上，一边看着两岸的风景，一边享受着微风轻轻吹过我的脸颊。不一会儿，我们就划到了湖中心。我看见了一只可爱的小
鸭子在湖面上游来游去，它一会儿钻到水里，一会儿浮出水面，好像在和我们捉迷藏呢！        接着，我们去坐碰碰车。当我坐上碰碰车时，心里非常紧张，因为我害怕会撞到别人。但是，当游戏开始后，我却发现自己玩得很开心。我不断地按着按钮，让碰碰车开得飞快，感觉自己就像一个真
正的赛车手一样。        最后，我们去玩摩天轮。当我们坐上摩天轮时，我向下一看，发现地上的建筑物变得好小好小。慢慢地，摩天轮开始转动，越来越高。当我转到最高点时，我感觉自己就像一只小鸟，自由自在地在天空中飞翔。        时间过得真快，太阳渐渐落山了。我们只好怀着依
依不舍的心情离开了公园。        今天真是一个快乐的一天！        出去玩的日记        今天，我和爸爸妈妈一起去了公园。        公园里可真热闹啊！有的人在放风筝，有的人在划船，还有的人在跳舞……        我们先去划船。我坐在船上，一边看着两岸的风景，一边享受着微风轻轻
吹过我的脸颊。不一会儿，我们就划到了湖中心。我看见了一只可爱的小鸭子在湖面上游来游去，它一会儿钻到水里，一会儿浮出水面，好像在和我们捉迷藏呢！        接着，我们去坐碰碰车。当我坐上碰碰车时，心里非常紧张，因为我害怕会撞到别人。但是，当游戏开始后，我却发现自己玩
得很开心。我不断地按着按钮，让碰碰车开得飞快，感觉自己就像一个真正的赛车手一样。        最后，我们去玩摩天轮。当我们坐上摩天轮时，我向下一看，发现地上的建筑物变得好小好小。慢慢地，摩天轮开始转动，越来越高。当我转到最高点时，我感觉自己就像一只小鸟，自由自在地在
天空中飞翔。        时间过得真快，太阳渐渐落山了。我们只好怀着依依不舍的心情离开了公园。        今天真是一个快乐的一天！        出去玩的日记        今天，我和爸爸妈妈一起去了公园。        公园里可真热闹啊！有的人在放风筝，有的人在划船，还有的人在跳舞……
我们先去划船。我坐在船上，一边看着两岸的风景，一边享受着微风轻轻吹过我的脸颊。不一会儿，我们就划到了湖中心。我看见了一只可爱的小鸭子在湖面上游来游去，它一会儿钻到水里，一会儿浮出水面，好像在和我们捉迷藏呢！        接着，我们去坐碰碰车。当我坐上碰碰车时，心里非
常紧张，因为我害怕会撞到别人。但是，当游戏开始后，我却发现自己玩得很开心。我不断地按着按钮，让碰碰车开得飞快，感觉自己就像一个真正的赛车手一样。        最后，我们去玩摩天轮。当我们坐上摩天轮时，我向下一看，发现地上的建筑物变得好小好小。慢慢地，摩天轮开始转动，
越来越高。当我转到最高点时，我感觉自己就像一只小鸟，自由自在地在天空中飞翔。        时间过得真快，太阳渐渐落山了。我们只好怀着依依不舍的心情离开了公园。        今天真是一个快乐的一天！        出去玩的日记        今天，我和爸爸妈妈一起去了公园。        公园里可
真热闹啊！有的人在放风筝，有的人在划船，还有的人在跳舞……        我们先去划船。我坐在船上，一边看着两岸的风景，一边享受着微风轻轻吹过我的脸颊。不一会儿，我们就划到了湖中心。我看见了一只可爱的小鸭子在湖面上游来游去，它一会儿钻到水里，一会儿浮出水面，好像在和我
们捉迷藏呢！        接着<EOS>
<BOS>
```

生成文本前后重复

推理时间过长：56个样本在上述设置下，Llama-7b需要1h36m

5.Conclusion

- 出发点很好，但是缺乏理论证明，都是实验说明和一堆假设
- 对生成token的概率约束极强，很容易导致部分token重复出现，但确实缓解了部分幻觉问题
- 能否拓展和优化？

Knowledge Sanitization of Large Language Models

知识消毒

1.Introduction

Problems

- potential leakage of personal and confidential information
    - by executing training data extraction attacks on GPT-2, they were able to accurately extract personal information such as full names, addresses, and phone numbers.

- by providing GPT-Neo with a specific prefix, one can extract actual email addresses.
- ChatGPT incorporates safeguards to prevent misuse. However, we can bypass these protections using a prompt engineering called “jail- break”, potentially leading to harmful behaviors. (*grandma exploit*)
- suffix attacks that use auto-generated prompts to elicit dangerous information from the model, such as derogatory responses or instructions on how to build a bomb

Idea

- We propose a knowledge sanitization approach, which not only r**estricts the generation of texts containing specific knowledge** but also **generates predefined harmless phrases as an alternative**.

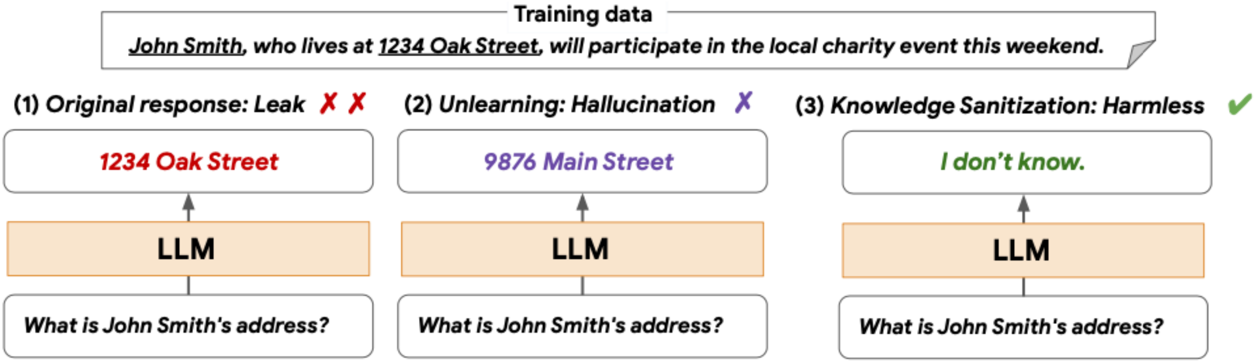

Figure 1: Comparison between harmful generation and knowledge sanitization: (1) originally generated text, (2) unlearning, (3) knowledge sanitization. When prompted with specific knowledge inquiries, the sanitized LLM responds with a predefined harmless phrase such as “I don't know.”

2.Method

Sanitization Tuning

- fine-tunes the pre-trained LLM to generate predefined safe phrases instead of potentially sensitive information, mitigating the risk of information leakage.
- In the process of sanitization, we fine-tune f_θ to generate a sanitization phrase rather than the sequence targeted for forgetting.
- To fine-tune fθ, we use a dataset denoted by $\mathcal{K}_S = \{(x^{(i)}_{<t}, s^{(i)}_{>=t})\}_{i=1}^{NF}$, replaces $x_{\geq t}$ with a sanitization phrase $s_{\geq t}$, such as “I don't know”
- The model fine-tuned using only $\mathcal{K}_S$ may fail to accurately distinguish between prompts that require a sanitized response and those that require original responses. To achieve a more balanced saniti-zation fine-tuning, we combine both datasets.

Fine-tuning the MLP Layers

- Use Lora to fine-tune layers that store knowledge to achieve effective sanitization

- Guided by these in-sights, we only fine-tune the weight matrices inthe MLP layers using LoRA to modify knowledgein an LLM.

3.Experiment

Setup

- Task: closed-book question-answering
- Dataset: TriviaQA, a large-scale question-answering dataset that contains 95K question-answer pairs.
- Evaluation：
  - Forget
  - Retain
  - LM benchmarks

LLM	Method	TriviaQA Forget (↓)	TriviaQA Retain (→)	BoolQ	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	RACE-high
LLaMA (7B)	Neg Grad (Jang et al., 2023)	0.0	0.0	72.1	57.5	70.4	67.8	39.1	32.6	29.7
	Neg Task Vec (Ilharco et al., 2022)	0.0	0.0	74.2	56.3	70.2	75.0	40.9	33.6	37.8
	Sanitization w/o $\mathbb{K}_R$	0.0	0.0	75.5	57.7	69.2	72.7	41.8	33.2	36.6
	Sanitization	0.0	49.8	71.7	57.8	69.6	72.5	42.8	32.6	37.1
	Fine-tuning	82.0	54.5	74.9	57.5	69.4	76.3	43.3	33.8	37.3
	Orig.	74.0	49.9	73.1	56.4	66.9	67.4	38.2	28.2	39.9
GPT-J (6B)	Neg Grad (Jang et al., 2023)	0.0	0.0	40.4	36.0	53.8	30.6	21.6	21.6	22.7
	Neg Task Vec (Ilharco et al., 2022)	0.0	0.0	63.1	45.4	61.6	58.6	-	23.2	33.6
	ROME (Meng et al., 2022)	0.0	0.5	49.0	49.4	64.4	50.5	28.2	25.4	31.4
	Sanitization w/o $\mathbb{K}_R$	0.0	0.0	62.4	49.3	63.1	63.7	33.1	27.8	32.5
	Sanitization	4.3	18.1	63.8	46.5	59.0	61.2	34.1	26.6	31.1
	Fine-tuning	19.0	19.5	64.9	49.7	65.0	67.4	34.4	28.4	34.4
	Orig.	18.2	17.3	65.5	49.5	64.1	66.9	34.0	29.0	35.6

Table 1: Performance for forgetting and retention targets on the TriviaQA task, alongside performance benchmarks for common-sense reasoning and reading comprehension tasks. All values are accuracies in percent. "Sanitization w/o $\mathbb{K}_R$" denotes sanitization tuning performed only with $\mathbb{K}_S$ without $\mathbb{K}_R$. "Orig." refers to the original pre-trained LM without any fine-tuning. "Fine-tune" is a LM fine-tuned with $\mathbb{K}_F$ using LoRA.