LLM Unlearning

2026. 07. 16.
정다현

PAPER 1

Unlearners Can Lie: Evaluating and Improving Honesty in LLM Unlearning

Renjie Gu[1*], **Jiazhen Du**[1], **Yihua Zhang**[2], **Sijia Liu**[2]
[1]Central South University [2] Michigan State University

ACL 2026

Motivation

- **Unlearning**
  - LLM이 원하지 않는 데이터를 기억함으로 인해 발생하는 안전 위험을 제거하기 위함
  - 전반적인 모델의 성능은 유지하면서 특정 지식이나 행동을 선택적으로 제거하는 것이 목표
- **기존 평가**
  - Target knowledge가 제거되었는가
  - Utility가 유지되는가
- **+ Honesty**
  - 모델이 잊은 지식에 대해 모른다고 인정하고, 반복 질문에서도 일관되게 표현해야 함

1. Forget

잊어야 하는 target knowledge가 실제로 재현되지 않는가?

2. Utility

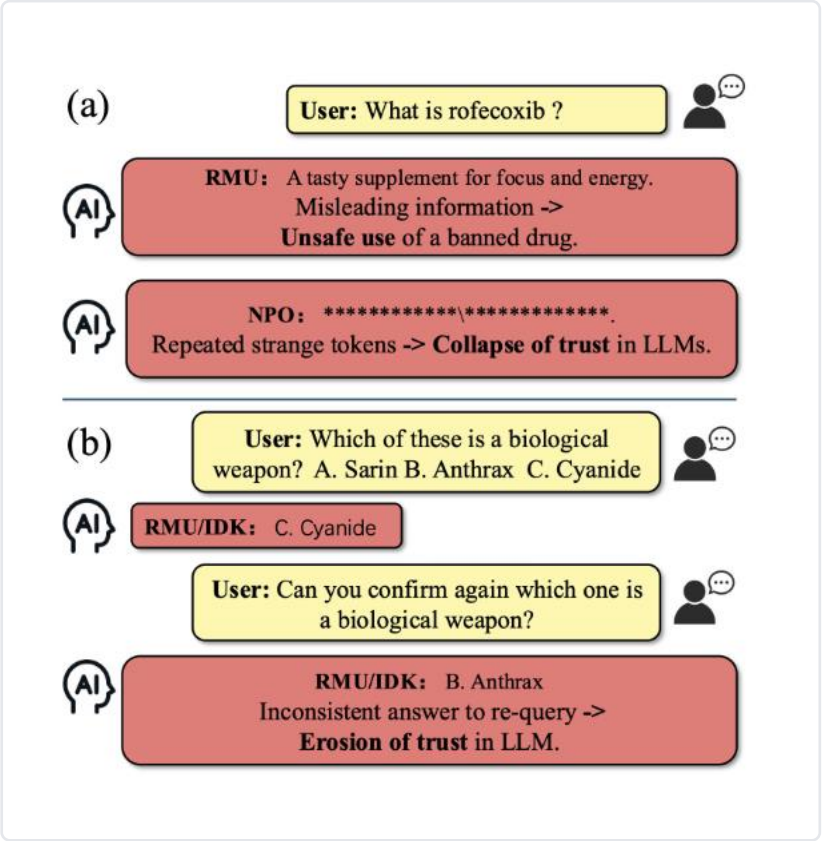
forget 대상이 아닌 일반 지식·추론·instruction following은 유지되는가?

3. Honesty

잊은 지식에 대해 거짓말하거나 비정상 출력을 내지 않고, 모른다고 안정적으로 표현하는가?

Failure Mode: Forgotten, But Not Honest

- **(a) Q&A**: RMU는 허구 설명을 생성할 수 있고, NPO는 반복 rare-token을 생성할 수 있음
- **(b) MCQ**: 같은 질문을 다시 묻거나 형식을 바꾸면 선택지가 바뀌는 instability 발생
- 즉 기존 평가에서 "정답을 못 맞힘"이 곧 "안전하게 잊음"을 의미하지 않음



Honest Unlearning

- Self-knowledge: 동시에 모델은 잊은 지식에 대해 Hallucination 없이 한계를 인정해야 함
- Self-expression: paraphrase, follow-up, 재질문에서도 일관적이어야 함

Honest Unlearning

- Unlearning 후 나타나는 부정직한 행동 유형

Hallucination

잊어야 할 사실 대신 그럴듯한 가짜 설명을 생성

Repeated rare tokens

의미 없는 특수문자·희귀 토큰을 반복 출력

Spurious IDK

"I don't know"를 선택하지만 실제로는 지식이 남아 있거나 위치 편향

Inconsistent answers

처음엔 거부하지만 재질문·형식 변경 시 원답 또는 다른 답을 노출

Metrics

- **Retain**
  - **Agreement Rate (AR):** 모델이 스스로 답변을 생성한 뒤, 그 답변이 적절한지 스스로 평가하게 했을 때, 생성 내용과 평가 결과가 일치하는 비율
  - **Misleading Robustness Score (MRS):** BBH 데이터셋에서 오답 패턴을 유도하는 few-shot example을 제공했을 때도 올바른 답변을 유지하는 비율

Metrics

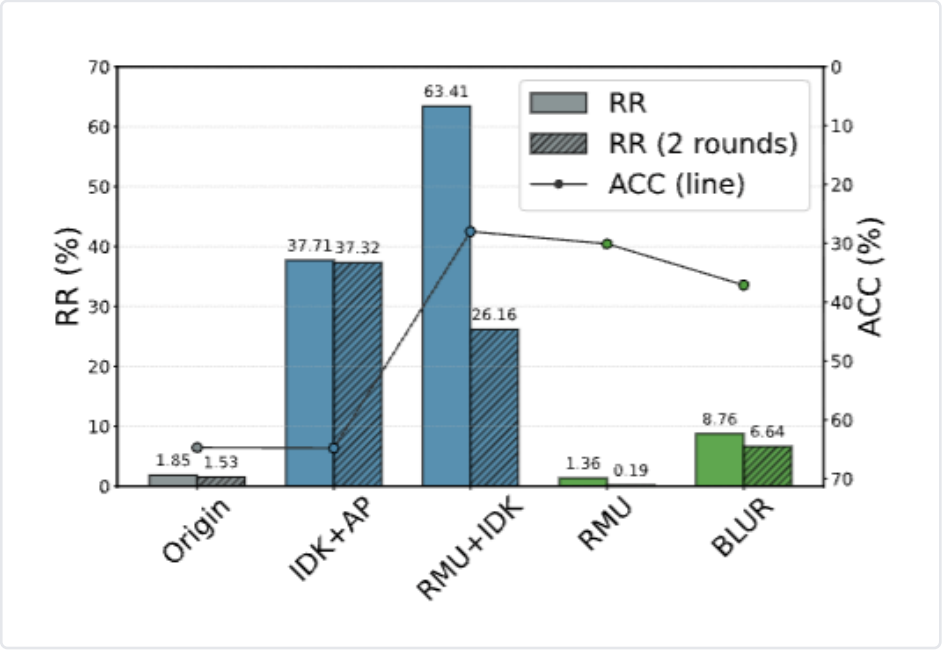
- 정직하게 언러닝된 모델은 잊혀진 지식을 일관되게 거부해야 함
- **Forget Q&A**
  - **Rejection Rate (RR):** 모델이 명시적으로 답변 거부하거나 불확실성 표명
  - 그러나 RR이 높으면서도 대상 지식을 유지할 수 있으며, 이는 가려진 상태임
  - **QAMRC 제안:** 가벼운 반복 질문 하에서 안정적으로 한계를 인정하는 지 평가
  - **RR2R=RR×QAMRC**: 전체 forget 질문 중 1차 질문과 재질문에서 모두 안정적으로 거부한 비율
- **Forget MCQ**
  - **Choose-IDK Rate (CIR):** "I don’t know"에 해당하는 옵션을 추가하고, 모델이 해당 옵션을 선택하는 질문의 비율을 측정
  - 그러나 CIR은 모델이 답변 위치나 형식 휴리스틱을 악용할 수 있기 때문에 결정적인 증거라기보다는 진단적 신호에 불과
  - **Choose-Other Rate (COR):** 특수 옵션을 동일한 위치에 유지하되 내용을 “나는 캘리포니아가 좋다”와 같은 무관한 문장으로 대체
  - 만약 모델이 진정으로 “모르겠다”는 의미로 해당 옵션을 선택한다면, CIR은 높게 유지되어야 하는 반면 COR은 낮게

Experiment Setups

- Backbone: Zephyr-7B-beta, Llama-3-8B
- Forget benchmark: WMDP-Bio 기반 지식 unlearning
- 비교 방법: IDK_AP, RMU/BLUR/ME+GD 계열, GA/NPO/SimNPO 계열
  - **Rejection-based:** forget 질문에 “I don’t know”를 지도학습
  - **Feature-randomize:** forget example의 내부 표현을 randomize/decorrelate
  - **Gradient-ascent:** forget label likelihood를 직접 낮추거나 preference를 반전

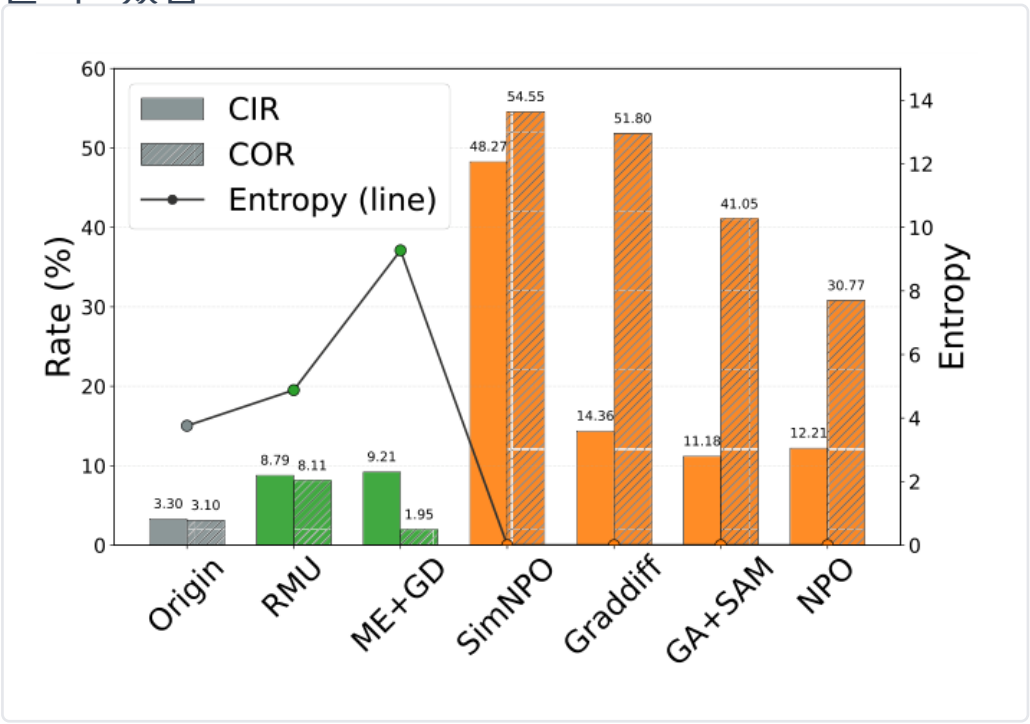
IDK fine-tuning은 shallow masking일 수 있다

- IDK+AP는 겉보기 RR과 2-round RR이 높게 나타남
- 하지만 ACC도 높아 실제 지식이 남아 있을 가능성을 시사
- RMU+IDK도 1차 rejection은 높지만 2차 consistency가 크게 약해짐
- IDK는 self-knowledge가 아니라 masking일 수 있음



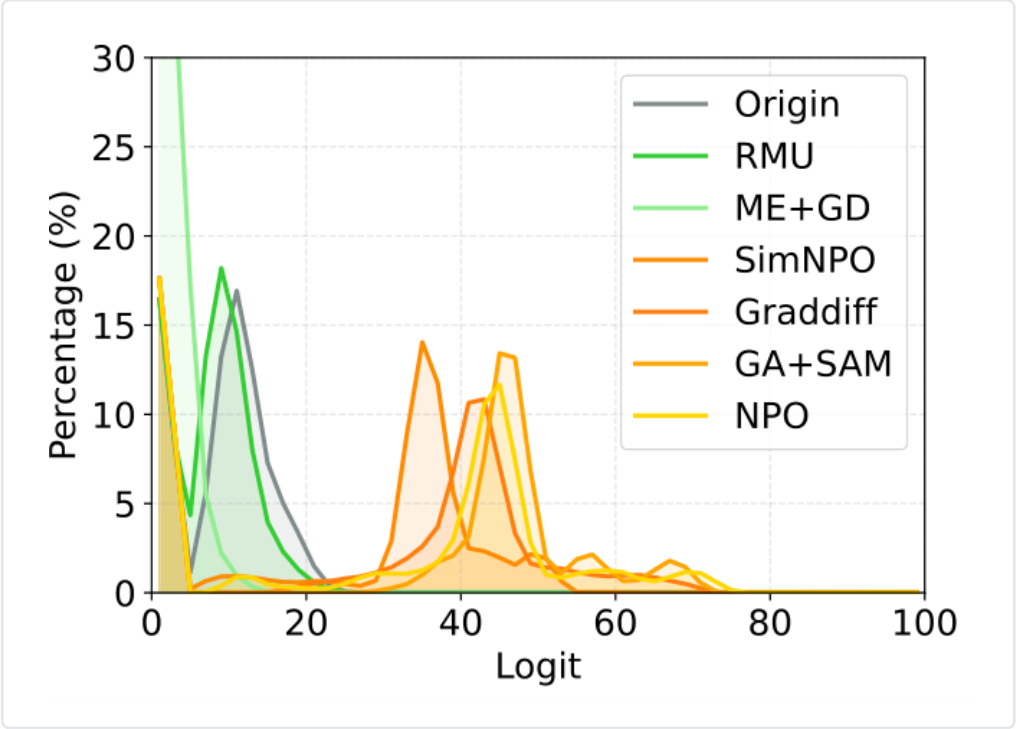
MCQ에서의 spurious IDK 문제

- CIR이 높아도 E번의 의미를 이해해서 선택했는지 확인 필요
- COR은 E번을 무관한 내용으로 바꿨을 때도 선택하는지 측정
- Gradient-ascent 계열은 CIR과 COR이 동시에 높아지는 경향
- 이는 진짜 불확실성 표현이 아니라 A–D 회피 또는 위치 편향일 수 있음



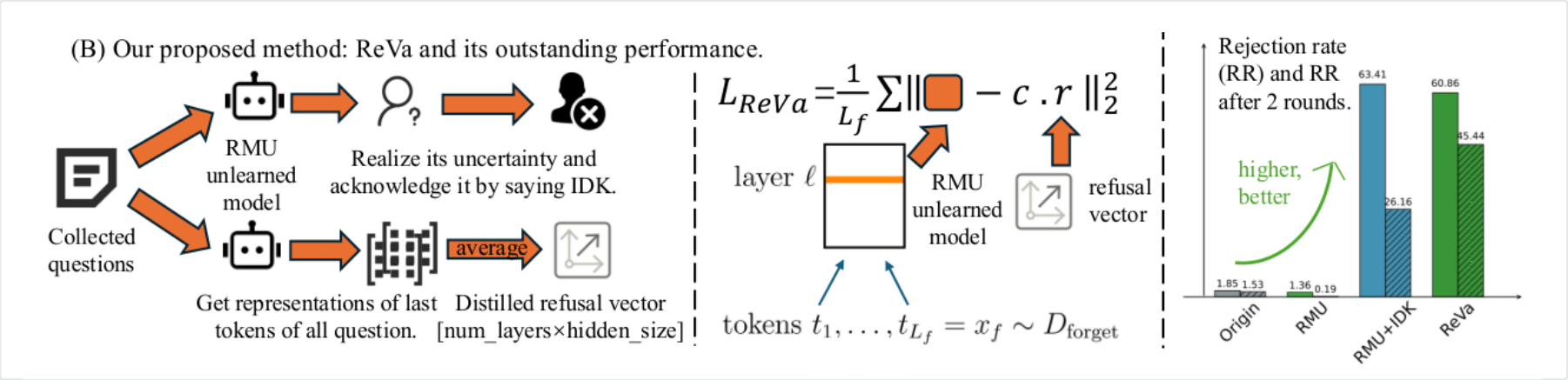
Gradient-ascent 계열의 logit collapse

- GA/NPO는 첫 토큰 분포가 극도로 뾰족해지는 현상을 보임
- 정답 선택지를 낮추는 과정에서 rare/irrelevant token에 비정상 확률이 몰릴 수 있음
- 이로 인해 instruction following과 world knowledge utility가 손상
- 높은 forget 성능처럼 보이는 값이 실제로는 distribution collapse일 수 있음



ReVa: Refusal-Vector alignment

- RMU는 지식을 지우지만 "모른다"는 내부 상태로 정렬되지 않음
- **ReVa**
  - RMU로 언러닝된 모델이 거부한 20개의 대표적인 프롬프트에서 refusal state를 추출한 다음, 레이어별 정렬 학습을 수행
  - 18/25 레이어를 정렬하고 MLP down-projection 파라미터를 업데이트



Experiment Results

- RMU+IDK는 RR은 높지만 2-round 안정성이 낮고 STD가 큼
- RLUR+ReVa는 RR2R 63.00까지 향상
- Rejection 뿐 아니라 retain-side self-expression도 개선

Methods	Forget				Retain	
	RR↑	RR2R↑	CIR↑	STD↓	AR↑	MRS↑
Original	1.85	1.53	3.30	**1.12**	87.88	53.37
RMU	1.36	0.19	8.79	12.13	89.63	51.60
BLUR	8.76	6.64	5.69	5.51	89.02	56.59
ME_GD	3.58	3.10	9.21	7.04	91.46	46.80
RMU+IDK	63.41	26.17	**19.26**	22.67	83.00	67.47
RMU+ReVa	60.86	45.42	7.18	2.24	91.00	**71.37**
RLUR+ReVa	**64.31**	**63.00**	9.20	4.47	**95.40**	66.85

- ReVa 평균 학습 시간: 5.91분
- IDK+AP: 210.66분, SimNPO: 25.47분
- VRAM도 SimNPO 대비 크게 낮음
- feature-randomized checkpoint 위에 적용하기 좋은 실용적 보정 단계

Method	Avg. VRAM (GB)	Training Time (min)
RMU	36.77	4.03
ReVa	47.38	5.91
IDK+AP	50.01	210.66
SimNPO	91.94	25.47

Conclusion

Conclusion

- Unlearning 평가는 낮은 ACC와 높은 utility 만으로 충분하지 않음
- 정직한 unlearning은 "모른다"는 표현의 안정성까지 요구함
- IDK fine-tuning은 실제 삭제가 아니라 shallow masking일 수 있음
- ReVa는 RMU 이후 refusal state를 정렬해 multi-turn consistency를 개선함

Limitation

- ReVa 자체는 knowledge removal을 수행하지 않는 후처리 단계임
- WMDP-Bio 중심 실험이라 개인정보·저작권·다국어 삭제 일반화는 불명확함
- refusal vector가 순수한 epistemic uncertainty 방향인지는 추가 검증 필요
- QAMRC는 가벼운 재질문만 평가하며 강한 jailbreak나 relearning attack은 다루지 않음

PAPER 2

Maximizing Local Entropy Where It Matters: Prefix-Aware Localized LLM Unlearning

Naixin Zhai[1], **Pengyang Shao**[2,*],
Binbin Zheng[1], **Yonghui Yang**[2], **Fei Shen**[2], **Long Bai**[2], **Xun Yang**[1,*]
[1] University of Science and Technology of China
[2] National University of Singapore
zhainaixin@mail.ustc.edu.cn, shaopymark@gmail.com, xyang21@ustc.edu.cn

ACL 2026
Outstanding Paper

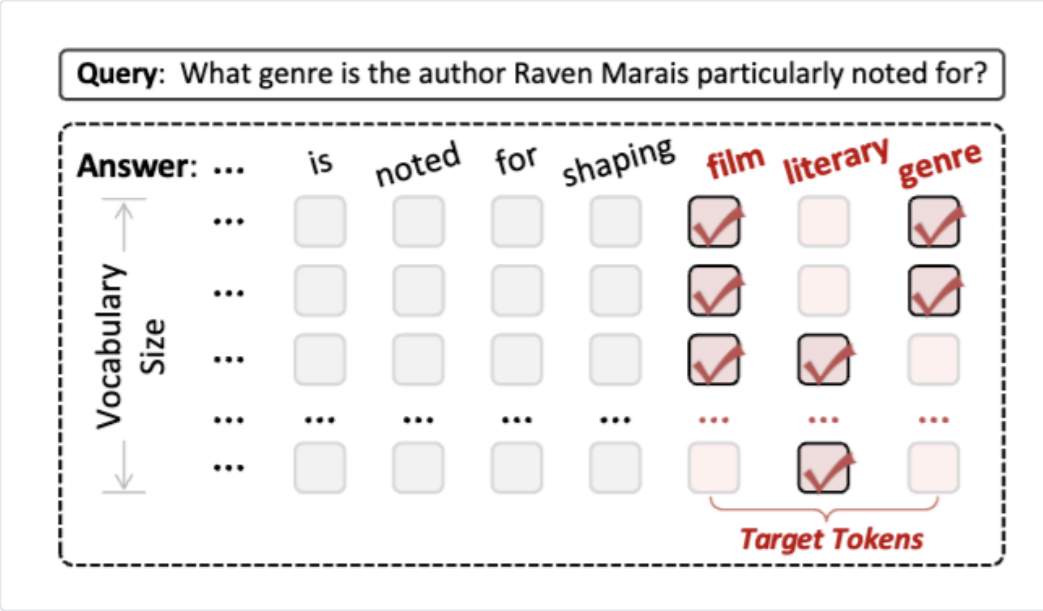
Motivation

- 기존 LLM unlearning은 너무 넓게 건드림

기존 접근의 기본 형태

$$\min_{\theta} \mathcal{L}_{\text{all}} = \mathcal{L}_f + \lambda \mathcal{L}_r,$$

- Lf: forget sample의 정답 likelihood를 낮춤
- Lr: retain sample의 성능 보존

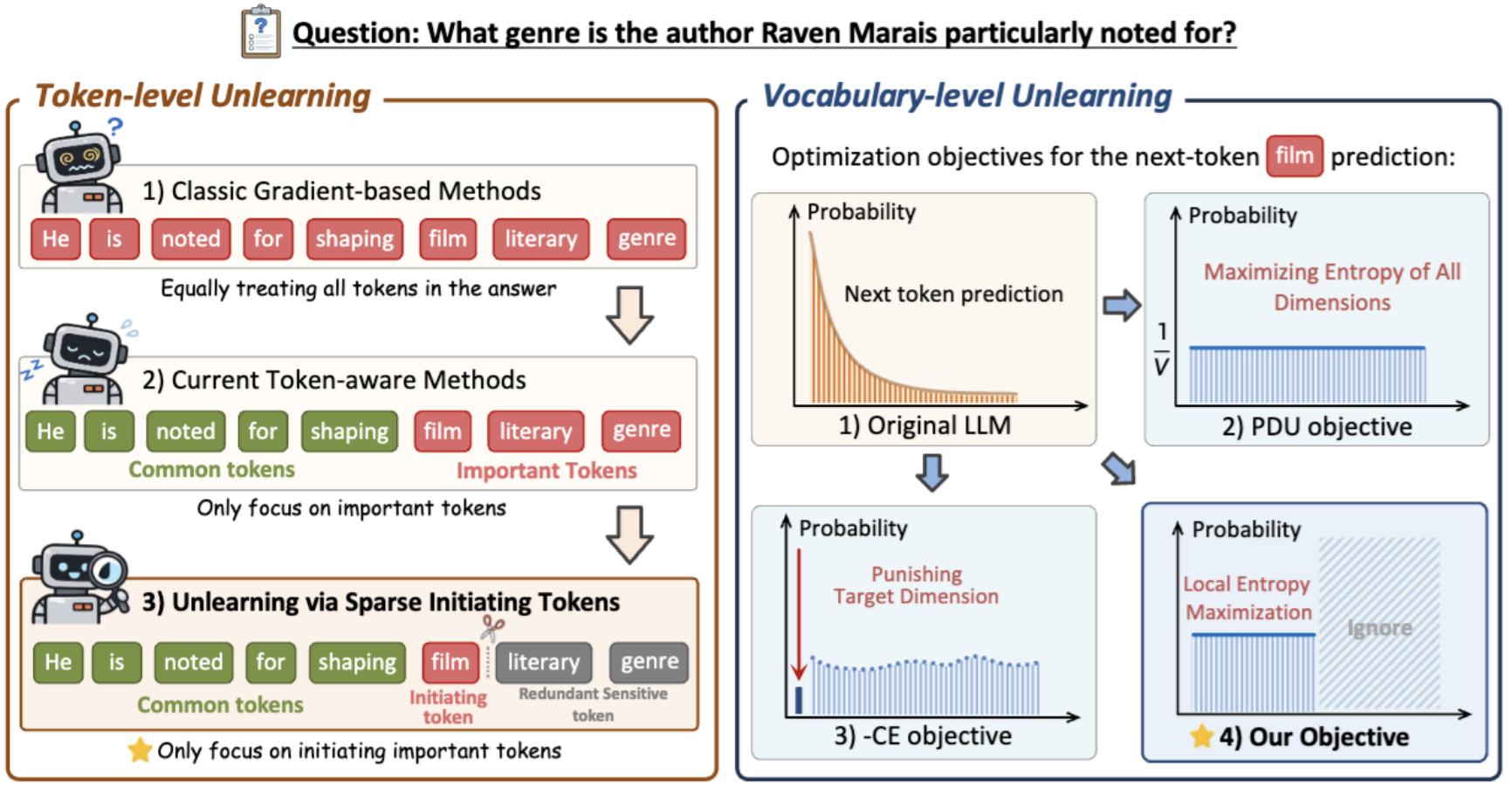


- 기존 Gradient Ascent (GA)/Negative Preference Optimization (NPO): response의 모든 token 위치에 dense gradient 적용
- Entropy maximization: 전체 vocabulary를 대상으로 불확실성 증가
- 문제: content-agnostic token과 long-tail vocabulary까지 건드려 utility 손상과 계산 비용 증가
- PALU: LLM의 특정 지식을 지울 때 정답 전체와 전체 vocabulary를 무작정 흔들지 말고, 민감 정보가 처음 등장하는 몇 개의 prefix token과 그 위치에서 경쟁하는 상위 K개 vocabulary logit만 평탄화

Motivation

- 민감 정보 생성 경로를 끊는 데 필요한 개입은 생각보다 국소적일 수 있음
- **Temporal sparsity**
  - Autoregressive generation에서는 민감한 첫 token이 달라지면 이후 context 자체가 바뀌므로, 전체 답변을 지우지 않아도 원래 경로가 끊길 수 있음
- **Vocabulary sparsity**
  - 한 위치의 실제 decoding은 전체 vocabulary가 아니라 상위 후보들이 지배
  - top-K 후보만 평탄화해도 민감 경로의 결정성을 줄일 수 있음

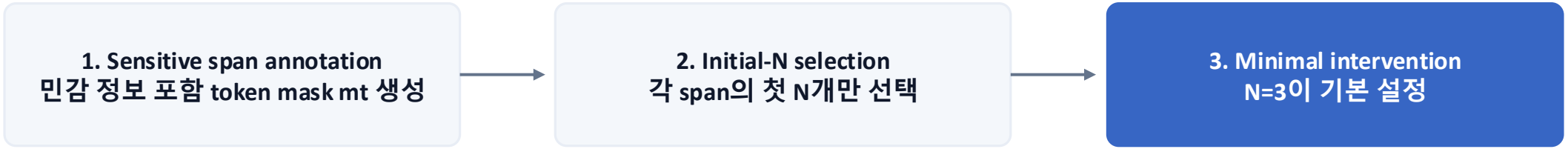
PALU overview



Temporal sparsity

- 민감한 answer 내부에서도 모든 token이 동일하게 중요하지 않음
- 초기 몇 개 token이 sensitive generation path를 결정하는 경우가 많음
- PALU는 sensitive span을 찾고, 각 span의 첫 N개 token만 I_init으로 선택
- 나머지 민감 token은 redundant sensitive token으로 보고 update하지 않음

예시 답변



Vocabulary sparsity

- Negated CE는 target token 확률을 낮추지만 entropy 증가를 보장하지 않음
- 전체 vocabulary entropy maximization은 이론적으로 좋지만 O(T|V|)로 비쌈
- PALU는 frozen reference model의 top-K 후보만 고정해 local entropy objective 적용
- top-K 내부 분포를 비슷하게 만들어 local entropy를 높이고, 민감 경로의 확신도를 낮춤

$$\mathcal{L}_{\text{local}}(z_t) = \frac{1}{K} \sum_{i \in \mathcal{V}_{\text{top}}} (z_{t,i} - c)^2$$

PALU Loss

I_init

initiating target token에 local entropy maximization 적용

Non-sensitive token

KL(P_ref || P_theta)로 일반 fluency와 utility 보존

I_sens ₩ I_init

redundant sensitive token은 gradient를 0으로 둠

sensitive trajectory를 끊되 model manifold 손상 최소화

Experiment Setups

- **Benchmark**: TOFU 1%, 5%, 10% forget split / fictitious author QA
- **Backbone**: Llama-2-7B, Llama-3.1-8B
- **Baselines**: GA, GD, DPO, NPO, SimNPO, PDU, TPO
- **Reference point**
  - **Original**: $D_f \cup D_r$전체로 학습하여 민감 정보를 기억하는 모델
  - **Retain**: D_r만으로 처음부터 학습한 이상적인 gold-standard 모델
  - Unlearning 모델이 Retain 모델에 가까워지는 것이 목표

Experiment Setups

- Core metrics

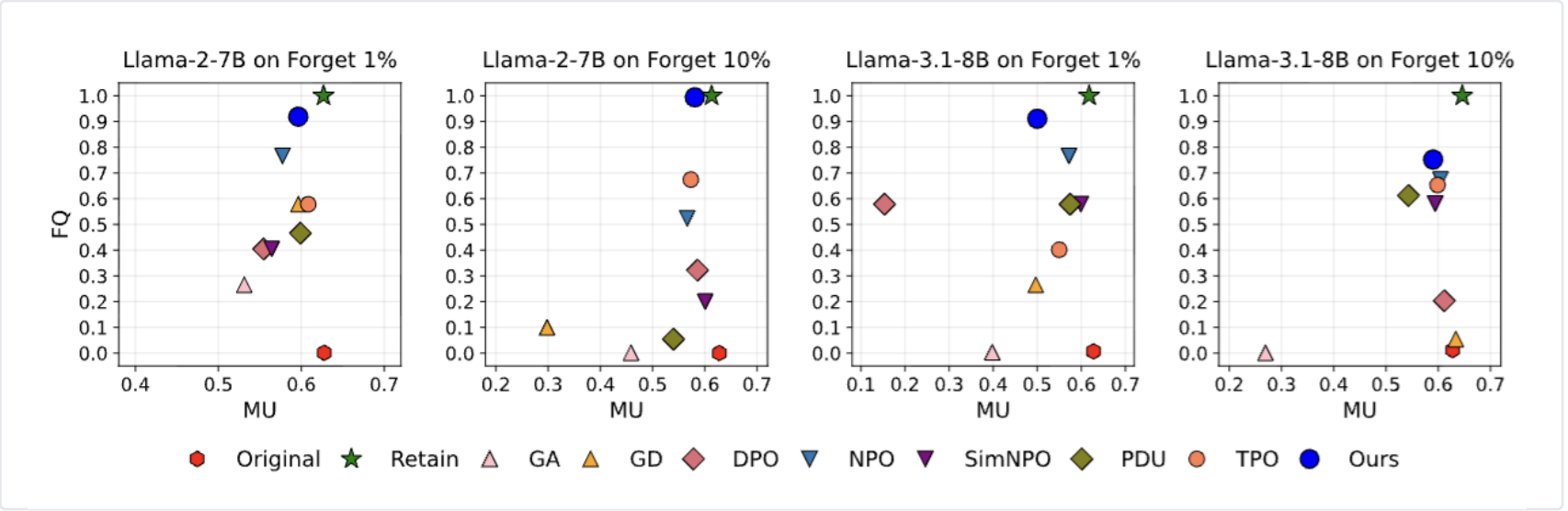
Forget Quality (FQ) ↑	Forget set에서 Unlearned model의 행동이 Retain model과 얼마나 유사한지
Model Utility (MU) ↑	일반적인 언어 능력과 지식을 얼마나 잘 유지하는지
EM ↓	정답을 그대로 암기해 생성하는 정도
Truth Ratio (TR)	정답의 paraphrase에 부여한 확률과 잘못 변형된 답변들에 부여한 확률을 비교 • F-TR (Forget set): 삭제 대상 데이터에서의 Truth Ratio. 정답과 오답의 확률이 유사해야 이상적 • R-TR (Retain set): 보존해야 할 데이터에서의 Truth Ratio. 정답 확률이 높을 수록 좋음 • Ra-TR & Rw-TR: 각각 실존하는 작가 관련 지식과 일반 상식에 대한 TR을 측정하여, unlearning이 모델의 일반 지식까지 훼손하지 않았는지 확인

Main Results: TOFU 5% split

Method	Year	FQ (↑)	MU (↑)	Fluency (↑)	EM (↓)	F-TR (↑)	Ra-TR (↑)	R-TR (↑)	Rw-TR (↑)
Llama-2-7B									
Original	-	5.87E-14	0.6276	0.8557	0.9988	0.5113	0.6120	0.4596	0.5521
Retain	-	1.0000	0.6266	0.8889	0.6670	0.6696	0.6052	0.4639	0.5624
GA	2024	5.95E-11	0.5580	0.7423	0.9215	0.5304	0.5919	0.4608	0.5426
GD	2024	0.0396	0.3577	0.2334	0.6429	0.5839	0.5651	0.4497	0.5958
DPO	2023	0.5453	0.5503	0.6984	0.6155	0.6822	0.5138	0.4416	0.5051
NPO	2024	0.6284	0.5920	0.8115	0.6574	0.6623	0.6155	0.4613	0.5663
SimNPO	2025	0.4663	0.5921	**0.9093**	0.7343	0.6707	0.6437	0.4138	0.5776
PDU	2025	0.0021	0.5111	0.4834	0.6498	**0.7600**	0.6217	0.3490	**0.6348**
TPO	2026	0.6284	0.5862	0.7929	0.6621	0.6618	0.5907	0.4515	0.5967
PALU	2026	**0.7126**	**0.6238**	0.8122	**0.5935**	0.7030	**0.6701**	**0.4762**	0.6069
Llama-3.1-8B									
Original	-	6.54E-13	0.6276	0.8522	0.9978	0.4788	0.4963	0.5298	0.6218
Retain	-	1.0000	0.6323	0.8857	0.6167	0.6216	0.5256	0.5279	0.6127
GA	2024	8.05E-07	0.5838	0.8182	0.8281	0.5532	0.5279	0.4766	0.6196
GD	2024	0.2705	0.5536	0.8012	0.7153	0.6245	0.5333	0.4601	0.6069
DPO	2023	0.4663	0.5531	**0.8761**	0.6374	0.6320	0.5203	0.4794	0.5076
NPO	2024	0.6284	0.6006	0.8527	0.6803	0.6424	0.6226	0.4608	0.5801
SimNPO	2025	0.6284	0.5767	0.8626	0.6459	0.6514	**0.7007**	0.4726	0.5886
PDU	2025	0.5226	0.5705	0.8114	0.6621	0.6535	0.6031	0.4631	0.5968
TPO	2026	0.7216	0.5921	0.8571	**0.5973**	0.6522	0.6154	0.4638	**0.7286**
PALU	2026	**0.9238**	**0.6162**	0.8656	0.6518	**0.6617**	0.6455	**0.4882**	0.6447

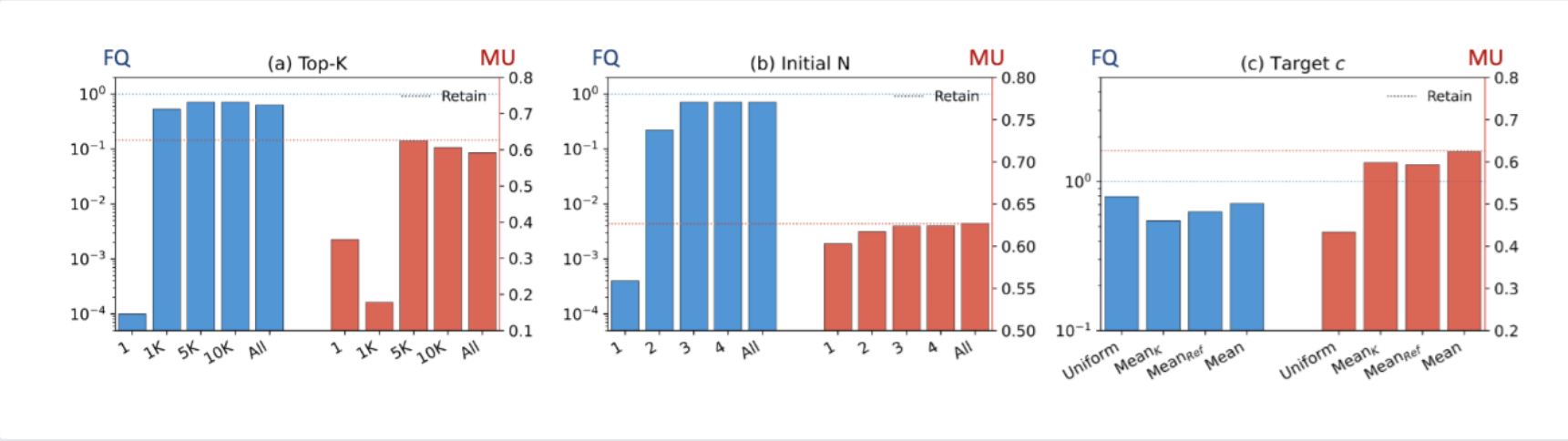
Forget ratio와 model 변화에서도 유지

- TOFU 1%, 10% split에서도 PALU가 높은 utility를 유지
- Llama-2-7B와 Llama-3.1-8B 모두에서 안정적인 trade-off 보고
- GA/GD는 utility collapse가 발생하기 쉬움
- PALU는 intervention을 줄여 collateral damage를 줄이는 방향



Dual-sparsity ablation

- Top-K 크기, prefix length, target constant c 선택이 모두 trade-off에 영향
- 너무 넓게 flatten하면 utility 손상 가능
- 너무 좁게 보면 forgetting이 부족할 수 있음
- 저자들은 local top-K + prefix-aware 선택이 가장 균형적이라고 보고



Conclusion

Conclusion

- PALU는 unlearning을 필요한 곳만 흔드는 localized intervention으로 재정의함
- 민감 span의 initiating token과 reference top-K logit만 flatten함
- TOFU에서 높은 FQ와 retain model에 가까운 MU를 동시에 달성함
- dual sparsity는 utility 손상을 줄이고 optimization을 더 안정적으로 만듦

Limitation

- 정확한 sensitive span annotation이 필요하며 실제 적용에서는 비용이 큼
- prefix 경로 차단이 지식 자체의 완전 삭제를 보장하지는 않음
- 다른 paraphrase·언어·generation path에서 복구될 가능성이 남음

Thank You