LLM Generalization

2026.05.28.
정다현

Published as a conference paper at ICLR 2026

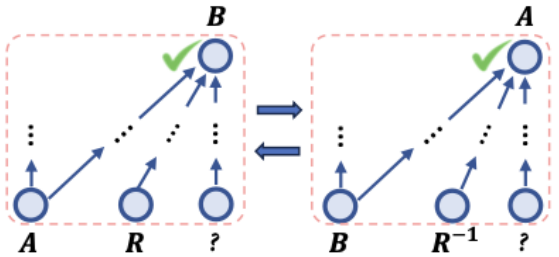
IS THE REVERSAL CURSE A BINDING PROBLEM? UNCOVERING LIMITATIONS OF TRANSFORMERS FROM A BASIC GENERALIZATION FAILURE

Boshi Wang
The Ohio State University
wang.13930@osu.edu

Huan Sun
The Ohio State University
sun.397@osu.edu

1. INTRODUCTION

Reversal Curse



모델이 “A는 B이다”라는 사실을 학습했음에도 불구하고,
그 역인 “B는 A이다”라는 질문에 답하지 못하는 기초적 일반화 실패 현상

기존 연구

- 문장 구간을 뒤집는 등의 특수한 데이터 증강 전략
- non-causal training objective

⇒ 그러나 이러한 방식은 문제를 우회함으로써 모델에 요구되는 일반화 부담을 줄이는 방식임

1. INTRODUCTION

Reversal Curse

모델이 “A는 B이다”라는 사실을 학습했음에도 불구하고,
그 역인 “B는 A이다”라는 질문에 답하지 못하는 기초적 일반화 실패 현상

- **데이터의 문제인가?**
  No. LLM이 이러한 규칙을 유도하기에 충분하고도 남는 웹 규모 코퍼스로 학습됨

- **Transformer 기반 언어 모델 구조의 근본적 한계인가?**
  No. 입력이 abstract concept 수준에서 표현되고 인식될 때,
  표준 Transformer가 구조나 목적 함수의 수정, 또는 특수한 데이터 증강 없이도 reversal을 학습할 수 있음

1. INTRODUCTION

Binding Problem

신경망이 분산된 정보를 통합하여 하나의 일관된 개념으로 묶는 메커니즘의 부재

트랜스포머는 개별 토큰 패턴에는 강하지만, 이를 고차원적 개념으로 묶는 능력이 부족함

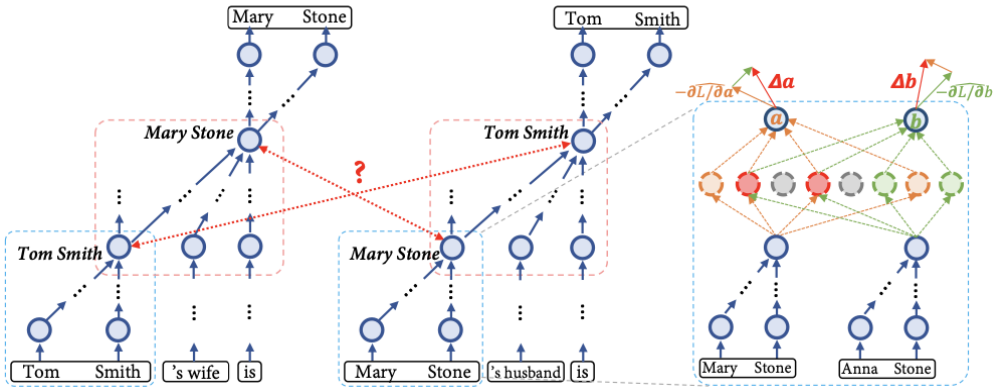
- Reversal Curse가 Transformer의 conceptual binding 한계에서 비롯되는 두 가지 주요 원인, 즉 개념 표현의 비일관성과 얽힘에 의해 발생한다고 가정함

가설 1. Inconsistency

- 엔티티가 Subject일 때와 Object일 때, 네트워크 내에서 서로 다른 Representation으로 인식됨
- 역할이 바뀔 때 동일한 개념으로 바인딩하지 못해 지식이 파편화됨

가설 2. Entanglements

- 개념들이 벡터 공간에서 서로 얽혀 있어, 하나의 개념을 학습할 때 다른 개념의 표현에 간섭을 일으킴
- 이러한 간섭이 새로운 사실의 역전 관계 일반화를 방해함



Designs for mitigating Reversal Curse

Transformer가 개념 수준에서 reversal을 학습할 수 있음을 보임

1. Joint-Embedding Predictive Architecture: autoregressive prediction을 개념 수준에서 수행
2. Disentangled Memory Layer: 얽힘이 적은 개념 표현을 지원하는 전용 인식 모듈을 구축

⇒ Reversal Curse는 단순한 데이터 부족 문제가 아니라, LLM의 개념 결합 메커니즘 한계를 보여주는 현상임

⇒ 모델이 정보 내재화 과정에서 parametric forward-chaining을 수행하고, 대규모 산술 추론 문제를 비파라메트릭 메모리에 기반한 모델보다 뛰어난 성능으로 해결할 수 있음을 보임

2. LEARNING REVERSAL AT THE CONCEPT LEVEL

Concept Level

Transformer가 실제 설정에서는 reversal을 학습하지 못하지만, 그렇다면 먼저 abstract concept 수준에서는 reversal을 학습할 수 있는 적절한 귀납적 편향을 가지고 있을까?

- 개념은 (entity1, relation, entity2)로 표현될 수 있음
- 그 역은 (entity2, relation^{-1}, entity1)로, 동일한 정보를 담고 있음
- 만약 모델이 reversal을 습득했다면 어떤 사실을 한 방향으로 내재화한 뒤 그 역에도 높은 확률을 부여해야 함

2. LEARNING REVERSAL AT THE CONCEPT LEVEL

Abstract Level

Abstract Setting 실험 설계

- 개념 직접 표상: 표면적 텍스트 대신 엔티티와 관계를 고유한 학습 가능 임베딩으로 직접 표현
- 데이터 구성: 6쌍의 관계 (r, r^{-1})와 수만 개의 엔티티 쌍을 활용하여 역전 관계 학습 유도
- 학습 목표: 한 방향의 사실 (e1, r, e2)을 GPT-2에 학습시킨 후, 반대 방향 (e2, r^{-1}, e1)을 추론할 수 있는지 평가

	$\lvert\mathcal{E}_A\rvert = 2.5K$	$\lvert\mathcal{E}_A\rvert = 10K$	$\lvert\mathcal{E}_A\rvert = 50K$	$\lvert\mathcal{E}_A\rvert = 100K$
#Layer = 1	0.823	0.861	0.947	0.964
#Layer = 6	0.890	0.858	0.951	0.861
#Layer = 12	0.810	0.878	0.951	0.960
#Layer = 18	0.823	0.850	0.944	0.975

- Transformer가 특수한 학습 목적이나 데이터 증강 없이도 높은 성능으로 reversal을 학습할 수 있음을 발견함

⇒ Transformer가 추상적 개념 수준에서는 reversal을 학습할 수 있다면, 왜 실제적인 설정에서는 실패하는가?

3. THE BINDING PROBLEM UNDERLYING SURFACE-LEVEL PREDICTIONS

Surface Level

실제적인 설정에서 가장 큰 차이는 모델이 개념을 직접 처리하는 것이 아니라, surface-form names을 인식한다는 점

가설 검증을 위한 측정 지표

- 표현 inconsistency 측정
  : 동일한 엔티티가 인식된 subject와 예측된 object 사이에서 역할을 바꿀 때, 모델이 그 엔티티 표현을 제대로 결합하지 못함
  : 엔티티가 예측 대상 (Object)일 때와 인지 대상 (Subject)일 때의 그라디언트 및 활성화 패턴 분석을 통해 표현의 괴리 정량화
- Entanglement 측정
  : 서로 다른 개념의 표현을 분리하는 문제
  : 각 개념은 자기 자신의 negative gradient 방향으로만 업데이트되지 않고, 다른 개념의 gradient와 섞인 방향으로 업데이트됨
  : 특정 개념의 가중치 업데이트가 다른 개념의 표현에 미치는 간섭 현상을 측정하여 일반화 저해 요인 분석

4. EXPERIMENTS

Surface Level

Attaching surface-level names to concepts

- 기존 abstract setting: e1 --wife--> e2
- surface-level setting: Tom Smith --wife--> Mary Stone
- 실험을 통제하기 위해 임의로 이름을 생성함 -> First name + Last name
- multiplicity가 100이면, Tom이라는 first name을 공유하는 사람이 10명 있고, Smith라는 last name을 공유하는 사람도 10명
- 높은 multiplicity에서는 서로 다른 개념들이 학습 중에 더 쉽게 얽힐 수 있음

⇒ 결과적으로 Transformer는 이 surface-level setting에서 reversal을 전혀 배우지 못하여 정확도가 0.0%

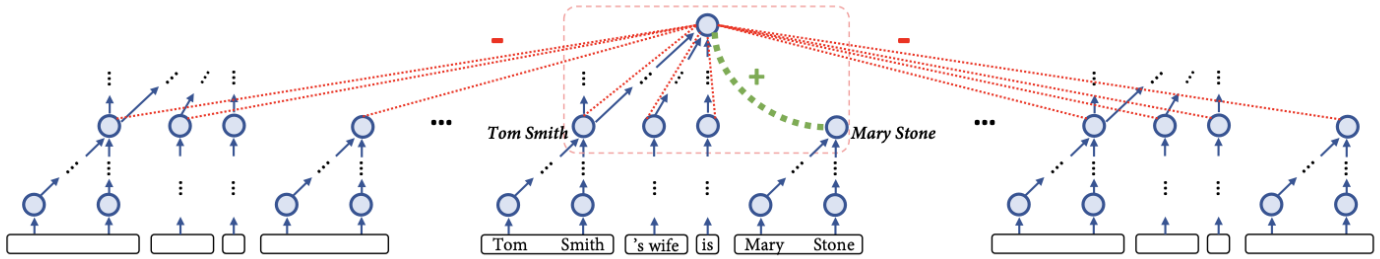
⇒ Reversal Curse는 Transformer가 역관계를 논리적으로 이해하지 못해서만 생기는 문제가 아니라,
surface 이름을 안정적인 개념 표현으로 binding하지 못해서 생기는 문제임

4. EXPERIMENTS

Joint-Embedding Predictive Architecture (JEPA

개념 임베딩 예측을 통한 Reversal Curse 극복

- 답을 token으로 바로 생성하지 않고, concept-level hidden state를 예측하도록 학습함
- **Recognition module**
  - surface-form name을 보고 어떤 개념인지 인식하는 부분 -> Transformer의 앞쪽 layer들
  - surface-form name을 concept representation으로 변환함 (Mary Stone → h_Mary)
- **Semantic module**
  - recognition module이 만든 concept representation을 바탕으로 더 추상적인 의미 처리를 하는 부분 -> 그 위쪽 layer들
  - 입력 concept와 relation를 바탕으로 다음 concept state를 예측함 (Tom Smith + wife relation → p)



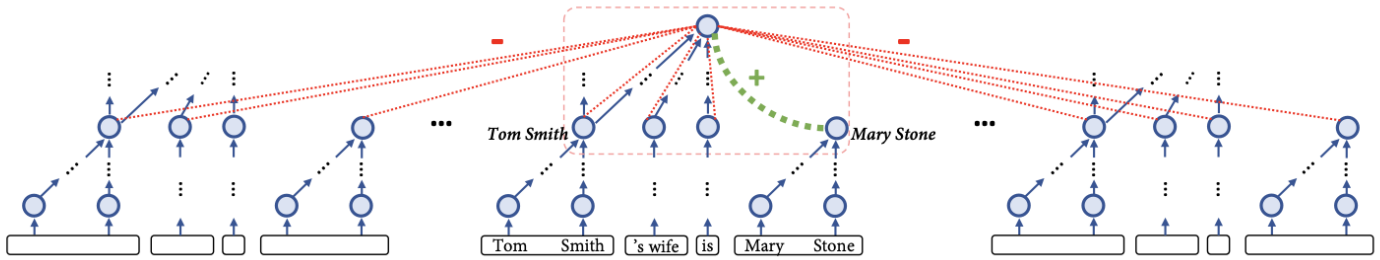
4. EXPERIMENTS

Joint-Embedding Predictive Architecture (JEPA

개념 임베딩 예측을 통한 Reversal Curse 극복

- 학습 objective는 predicted state p가 정답 concept representation h_Mary와 가까워지도록 만드는 것
- 정답 concept은 positive, batch 내 다른 concepts는 negatives로 사용 (InfoNCE contrastive loss)

⇒ 모델이 concept을 안정적으로 연결하도록 하여 reversal 관계를 자연스럽게 학습할 수 있도록 함



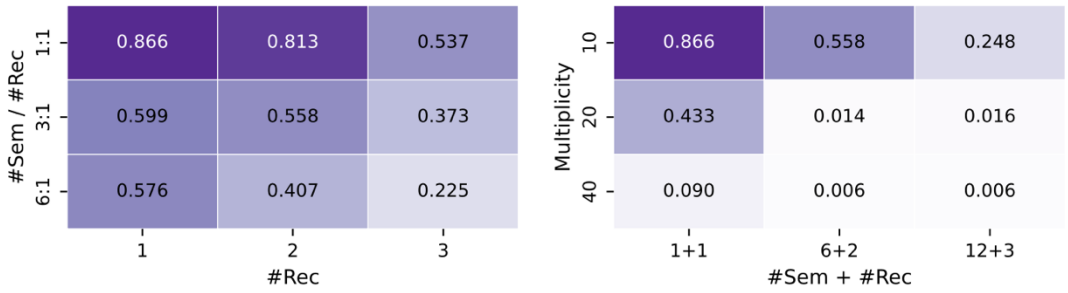
4. EXPERIMENTS

Joint-Embedding Predictive Architecture (JEPA

JEPA는 일반화를 가능하게 하지만, entanglement의 영향을 받음

- 모델 구성은 recognition module의 깊이 (#Rec)와 recognition module 위에서 semantic processing을 수행하는 semantic module의 깊이 (#Sem)을 바꾸어 실험함

- 단순히 JEPA를 사용해 conceptual consistency를 유도하는 것만으로도 모델은 상당히 의미 있는 일반화 성능을 보여줌
- 모델이 깊어질수록 성능 감소 & Multiplicity가 증가하면 일반화 성능 저하
- consistency 문제가 어느 정도 해결되더라도,
  entanglement의 영향 때문에 reversal 학습은 어려움



4. EXPERIMENTS

Disentangled Memory Layer

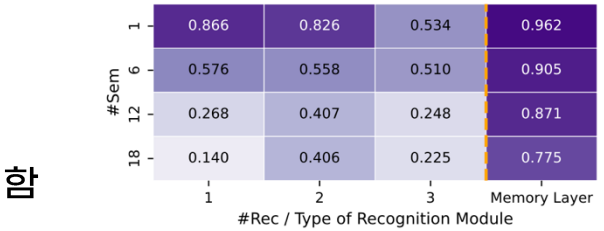
개념 간 간섭 방지를 위한 메모리 구조

기존 방식 (Entangled)

: 모든 개념이 동일한 모델 가중치를 통해 업데이트되어 특정 엔티티 학습 시 다른 엔티티의 표현이 의도치 않게 변경됨

제안 방식 (Disentangled)

: 각 엔티티에 대해 독립적인 메모리 레이어를 할당하여 정보 업데이트가 해당 개념에만 국한되도록 함



: recognition module의 마지막 MLP layer를 memory layer로 대체 (넓은 hidden layer + top-k sparsity + softmax activation) -> 사실상 각기 다른 이름을 가진 개념에 대해 별도의 learnable embedding을 갖는 구조가 됨

- 결과적으로 memory layer는 성능을 크게 향상시키지만 semantic layer가 많아질수록 일반화 성능은 여전히 약간 감소
- 이는 surface 수준 예측에서 발생하는 inconsistency가 여전히 주요한 blocker로 남아 있음을 의미하며 Memory layer는 이 inconsistency가 해결된 경우에만 효과적임을 보여줌

5. ARITHMETIC REASONING VIA PARAMETRIC FORWARD-CHAINING

Complex arithmetic reasoning problems

모델이 reversal을 실제로 습득한다면, 단순한 사실 몇 개를 더 아는 것 말고 정확히 무엇을 할 수 있는가?

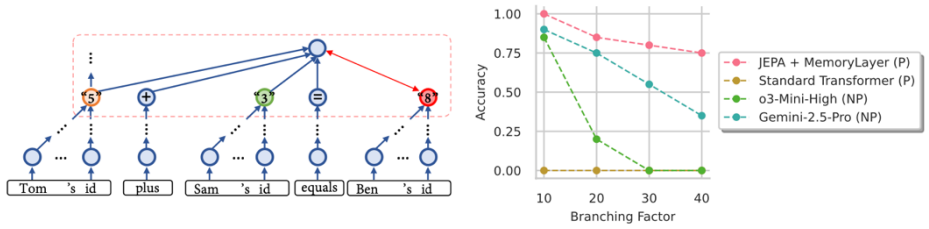
- 모델이 파라메트릭 메모리 안에서 서로 다른 정보 조각들을 추론하고, 암묵적으로 연결할 수 있게 함
- 추론된 값을 다시 변수에 적절히 결합하면, 그 변수와 연결된 또 다른 미지의 값을 찾아내는 발판으로 사용할 수 있음
- 즉, 하나의 추론이 다음 추론을 유발하여 모델은 정보를 내재화하는 동안 parametric forward-chaining을 수행할 수 있음

- 노드가 변수를 나타내고 엣지가 덧셈 관계로 변수를 연결하는 search tree를 만들고, 목표가 되는 미지의 변수는 값이 알려진 변수들로부터 3-hop 떨어져 있음
- 문제의 복잡도는 search tree의 branching factor를 통해 조절함

⇒ JEPA와 memory layer를 사용한 모델은 비파라메트릭 메모리에 기반한 LLM보다 더 높은 성능을 보임

⇒ 특히 문제 크기가 커질 때, 파라메트릭 메모리를 사용하는 모델의 성능 하락은 비교적 완만함

⇒ 표준 Transformer는 일관되게 실패



6. CONCLUSION

Conclusion

Reversal Curse는 Transformer의 수학적 결함이 아닌 binding problem의 결과이며,
concept 수준에서는 완벽한 reversal 학습이 가능함을 증명함

- JEPA 기반 아키텍처와 전용 메모리 레이어를 통해 데이터 증강 없이도 Reversal Curse를 극복할 수 있는 구조적 해결책을 제시
- Reversal 능력 확보가 단순 암기를 넘어 고차원적 지식 통합 및 산술 추론 능력을 위한 필수 기저 기술임을 확인
- 그러나 본 논문은 가장 기본적인 개념만 다루는 reversal task에 국한되어 있으므로 Reversal Curse의 근본 문제들이 아직 해결되었다고 보기는 어려움

⇒ 인간의 개입을 최소화하면서도 스스로 체계적인 개념 바인딩을 학습할 수 있는 모델 설계가 필수적임

Published as a conference paper at ICLR 2026

ON THE GENERALIZATION OF SFT: A REINFORCEMENT LEARNING PERSPECTIVE WITH REWARD RECTIFICATION

Yongliang Wu[1*] **Yizhou Zhou**[2*†] **Zhou Ziheng**[3] **Yingzhe Peng**[1] **Xinyu Ye**[4]
Xinting Hu[5] **Wenbo Zhu**[6] **Lu Qi**[7] **Ming-Hsuan Yang**[8] **Xu Yang**[1‡]
[1]Southeast University [2]Independent Researcher [3]University of California, Los Angeles
[4]Shanghai Jiao Tong University [5]Nanyang Technological University
[6]University of California, Berkeley [7]Wuhan University [8]University of California, Merced
yongliang0223@gmail.com, zyz0205@hotmail.com, xuyang_palm@seu.edu.cn

SFT의 한계와 일반화 격차

SFT vs RL

- SFT는 LLM 학습의 표준이지만, RL에 비해 일반화 능력이 현저히 떨어짐
- SFT 자체의 목적 함수 개선만으로 RL 수준의 일반화가 가능한가?

- **Dynamic Fine-Tuning (DFT)**
  : 각 token에서 SFT objective를 해당 token의 확률로 rescale함
  : inverse-probability weighting으로 인해 발생하는 왜곡을 상쇄하여 더 안정적이고 더 균등하게 가중된 update rule로 바꿈

2. METHOD

SFT와 RL의 트레이드오프

SFT는 전문가 행동 모방에 효율적이나 과적합 위험이 크고, RL은 탐색을 통해 일반화에 강하나 보상 모델과 막대한 자원이 필수적임

Supervised Fine-Tuning

- 장점: 전문가 데이터를 통한 빠른 행동 습득, 구현이 매우 단순함
- 단점: 데이터 암기에 치중하여 새로운 상황에서의 일반화 능력이 제한됨

Reinforcement Learning

- 장점: 보상 신호를 통한 최적 전략 탐색, 우수한 일반화 성능
- 단점: 명시적 보상 모델 필요, 학습 불안정성 및 높은 계산 비용

2. METHOD

SFT Gradient의 재해석

SFT의 업데이트 수식은 특정 보상 구조를 가진 RL의 Policy Gradient로 변환 가능함

$$\nabla_\theta \mathcal{L}_{\mathrm{SFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\big[-\nabla_\theta \log \pi_\theta(y^\star \mid x)\big].$$

$$\nabla_\theta J(\theta) = \mathbb{E}_{x\sim\mathcal{D}_x,\, y\sim\pi_\theta(\cdot|x)}\big[\nabla_\theta \log \pi_\theta(y \mid x)\, r(x,y)\big].$$

- **SFT**: 정답 y*를 데이터에서 바로 가져와서 확률을 올림
- **RL**: 모델이 y를 직접 생성하고, reward가 높은 y의 확률을 올림

⇒ SFT도 모델이 생성한 답변 중 정답과 정확히 같은 답변에만 reward를 주는 RL처럼 다시 쓸 수 있지 않을까?

2. METHOD

SFT Gradient의 재해석

SFT의 업데이트 수식은 특정 보상 구조를 가진 RL의 Policy Gradient로 변환 가능함

$$\nabla_\theta \mathcal{L}_{\mathrm{SFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\big[-\nabla_\theta \log \pi_\theta(y^\star \mid x)\big]. \qquad \nabla_\theta J(\theta) = \mathbb{E}_{x\sim\mathcal{D}_x,\, y\sim\pi_\theta(\cdot|x)}\big[\nabla_\theta \log \pi_\theta(y \mid x)\, r(x,y)\big].$$

$$\nabla_\theta \mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_x,\, y\sim\pi_\theta(\cdot|x)}\big[w(y \mid x)\, \nabla_\theta \log \pi_\theta(y \mid x)\, r(x,y)\big].$$

$$w(y \mid x) = \frac{\mathbf{1}}{\pi_\theta(y \mid x)}, \quad r(x,y) = \mathbf{1}[y = y^\star].$$

- SFT는 y*를 데이터에서 바로 가져옴. 이를 RL 형태로 쓰려면 모델 분포에서 샘플링한 것처럼 표현해야 함. 그래서 정답이 나올 확률로 보정함
- Inverse-Probability Weighting: SFT와 RL의 수식적 차이 -> 모델이 낮은 확률을 주는 정답일수록 큰 리워드를 받음
- 즉 모델이 어떤 정답에 낮은 확률을 주고 있으면 그 샘플에 대한 업데이트가 과도하게 커짐
- 이것이 SFT의 generalization limitation을 설명하는 핵심 원인 중 하나라고 주장
- “SFT는 전문가 행동에만 보상을 부여하는 매우 희소한 보상 구조를 가진 강화학습과 수학적으로 동일하다.”

2. METHOD

Dynamic Fine-Tuning

DFT는 텍스트의 표면적 패턴 암기를 지양하고, RL과 유사한 정책 최적화를 통해 모델의 근본적인 추론 능력을 강화

- SFT 목적 함수에 토큰 확률(w)을 직접 곱하여, 기존의 왜곡된 역수 가중치(1/w) 효과를 수학적으로 상쇄함

$$\nabla_\theta \mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{x\sim\mathcal{D}_x,\, y\sim\pi_\theta(\cdot|x)}\Big[\mathrm{sg}\big(\frac{1}{w}\big)\cdot w(y\mid x)\,\nabla_\theta \log \pi_\theta(y\mid x)\, r(x,y)\Big].$$

$$\mathcal{L}_{\mathrm{DFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\Big[-\,\mathrm{sg}\left(\pi_\theta(y^\star\mid x)\right)\log \pi_\theta(y^\star\mid x)\Big].$$

- sg(·): stop-gradient operator (w를 통해 gradient가 흐르지 않도록 함)
- 하지만 전체 trajectory에 대해 importance weight를 계산하면 수치적으로 불안정해질 수 있으므로 token level에서 적용

$$\mathcal{L}_{\mathrm{DFT}}(\theta) = \mathbb{E}_{(x,y^\star)\sim\mathcal{D}}\Big[-\sum_{t=1}^{|y^\star|}\mathrm{sg}\left(\pi_\theta(y_t^\star\mid y_{<t}^\star,x)\right)\log \pi_\theta(y_t^\star\mid y_{<t}^\star,x)\Big].$$

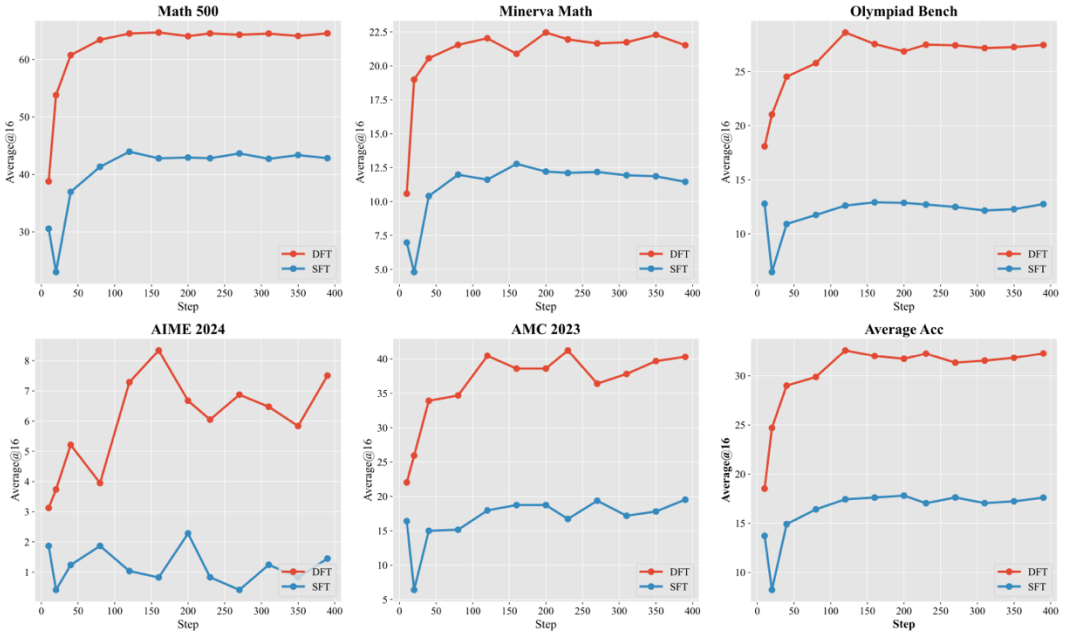
- 특정 low-probability reference token에 과도하게 집중하는 문제를 피함
- update가 더 안정적이 되고, 추가 sampling이나 reward model 없이도 일반화가 향상

3. EXPERIMENTS

수학적 추론 성능 결과

표준 SFT 대비 수 배 이상의 성능 향상을 기록했으며, 특히 OlympiadBench 등 고난도 과제에서 일반화 능력이 두드러짐

	Math500	Minerva Math	Olympiad Bench	AIME24	AMC23	Avg.
LLaMA-3.2-3B	1.63	1.36	1.01	0.41	1.56	1.19
LLaMA-3.2-3B w/SFT	8.65	2.38	2.06	0.00	3.13	3.24
LLaMA-3.2-3B w/DFT	**12.79**	**2.84**	**2.90**	**0.83**	**3.91**	**4.65**
LLaMA-3.1-8B	1.86	0.98	0.94	0.21	1.01	1.00
LLaMA-3.1-8B w/SFT	16.85	5.78	3.88	0.00	5.16	6.33
LLaMA-3.1-8B w/DFT	**27.44**	**8.26**	**6.94**	**0.41**	**12.03**	**11.02**
DeepSeekMath-7B	6.15	2.15	1.74	0.21	2.97	2.64
DeepSeekMath-7B w/SFT	26.83	7.26	6.33	0.41	8.28	9.82
DeepSeekMath-7B w/DFT	**41.46**	**16.79**	**15.00**	**1.24**	**16.25**	**18.15**
Qwen2.5-Math-1.5B	31.66	8.51	15.88	4.16	19.38	15.92
Qwen2.5-Math-1.5B w/SFT	43.76	13.04	12.63	1.87	18.75	18.01
Qwen2.5-Math-1.5B w/DFT	**64.89**	**20.94**	**27.08**	**6.87**	**38.13**	**31.58**
Qwen2.5-Math-7B	40.12	14.39	17.12	6.68	27.96	21.25
Qwen2.5-Math-7B w/SFT	53.96	16.66	18.93	2.48	26.09	23.62
Qwen2.5-Math-7B w/DFT	**68.20**	**30.16**	**33.83**	**8.56**	**45.00**	**37.15**



3. EXPERIMENTS

Code Generation & Multimodal Reasoning

LiveCodeBench 등 최신 벤치마크에서 단순 모방을 넘어선 논리적 코드 생성 능력 입증

시각 정보와 텍스트 추론이 결합된 과제에서도 SFT 대비 우수한 일반화 성능 확인

	HumanEval		MultiPL-E								
	HE	HE+	Python	C++	Java	PHP	TS	C#	Bash	JS	Avg.
Qwen2.5-3B	43.3	36.0	43.29	40.99	37.34	37.89	**47.17**	**43.04**	24.68	45.96	40.05
Qwen2.5-3B w/SFT	41.5	34.8	42.07	42.24	37.97	37.27	43.40	41.77	20.25	**47.83**	39.10
Qwen2.5-3B w/DFT	**45.7**	**39.0**	**45.73**	**44.72**	**41.77**	**45.34**	42.14	**43.04**	**27.85**	44.10	**41.84**
Qwen2.5-Coder-3B	52.4	42.7	51.83	53.42	46.20	47.20	54.09	55.06	25.32	**54.04**	48.39
Qwen2.5-Coder-3B w/SFT	51.8	43.9	51.22	51.55	48.10	54.66	**59.12**	51.27	**34.18**	**54.04**	50.52
Qwen2.5-Coder-3B w/DFT	**56.7**	**50.0**	**57.32**	**54.66**	**51.27**	**58.39**	58.49	**60.76**	31.01	53.42	**53.16**
Qwen2.5-Coder-7B	62.2	53.0	63.41	63.98	53.16	59.01	62.89	59.49	39.24	60.87	57.76
Qwen2.5-Coder-7B w/SFT	54.9	48.8	54.88	64.60	51.27	**62.11**	68.55	60.76	33.54	**65.22**	57.62
Qwen2.5-Coder-7B w/DFT	**67.7**	**59.8**	**67.68**	**67.70**	**54.43**	60.87	**70.44**	**65.19**	**48.73**	63.35	**62.30**

	MathVerse				MathVision	WeMath
	Vision Only	Vision Intensive	Vision Dominant	Overall		
Qwen2.5-VL-3B	28.81	30.96	31.60	33.83	21.25	4.10
Qwen2.5-VL-3B w/SFT	30.96	33.63	32.74	35.66	21.02	23.33
Qwen2.5-VL-3B w/DFT	**32.49**	**35.91**	**33.50**	**37.54**	**22.30**	**23.71**

3. EXPERIMENTS

기존 RL 방법론과의 비교

RL 방법론 대비 우수한 성능 및 자원 효율성

- 오프라인 RL: DPO, RAFT 등 선호도 기반 방법론보다 높은 수학 추론 성능 달성

- 오프라인 DFT: 학습 가능한 correct response가 여러 개인 세팅
- 온라인 RL: PPO, GRPO와 대등하거나 우수한 성과 확인

	Setting	Math500	Minerva Math	Olympiad Bench	AIME24	AMC23	Avg.
Qwen2.5-Math-1.5B w/DFT	SFT	64.89	20.94	27.08	6.87	38.13	31.58
Qwen2.5-Math-1.5B w/DPO	Offline	46.89	11.53	22.86	4.58	30.16	23.20
Qwen2.5-Math-1.5B w/RFT	Offline	48.23	14.19	22.29	4.37	30.78	23.97
Qwen2.5-Math-1.5B w/PPO	Online	56.10	15.41	26.33	7.50	37.97	28.66
Qwen2.5-Math-1.5B w/GRPO	Online	62.86	18.93	28.62	**8.34**	41.25	32.00
Qwen2.5-Math-1.5B w/DFT	Offline	**64.71**	**25.16**	**30.93**	7.93	**48.44**	**35.43**

- DPO와 달리 Reference Model을 유지할 필요가 없어 메모리 절약
- 복잡한 보상 모델링이나 대규모 배치 사이즈 없이도 RL 효과 구현

Conclusion

SFT와 RL 차이 일반화 격차를 재검토하고, 이를 완화하기 위해 방안을 제안함

- SFT의 일반화 한계 원인을 RL 관점에서 분석하여, 왜곡된 보상 구조를 이론적으로 규명함
- 단 한 줄의 코드 수정 (DFT)만으로 SFT와 RL의 간극을 메우고, LLM의 성능을 비약적으로 향상시키는 실용적 해법 제시함
- 수학, 코드, 멀티모달의 다양한 도메인에서 DFT의 우수한 일반화 성능과 학습 안정성을 실험적으로 입증함

Thank you

Q&A