QeRL: Beyond Efficiency -- Quantization-enhanced Reinforcement Learning for LLMs

Wei Huang[1,3], Yi Ge[2,4], Shuai Yang[1], Yicheng Xiao[4], Huizi Mao[1], Yujun Lin[1], Hanrong Ye[1], Sifei Liu[1], Ka Chun Cheung[1], Hongxu Yin[1], Hongxu Yin[1], Yao Lu[1], Xiaojuan Qi[3], Song Han[1,2], Yukang Chen[1]

[1]NVIDIA, [2]MIT, [3]HKU, [4]THU

목차

1. 배경

LLM 강화학습의 문제점

LLM 추론 능력 향상: SFT vs RL

Supervised Fine-Tuning (SFT)

- 정답 추론 과정을 모방 학습
- 데이터 품질에 크게 의존
- 주의: "진짜 추론"보다 "모방"에 가까움

Reinforcement Learning (RL)

- 검증 가능한 보상 신호로 학습
- 다양한 추론 경로 탐색
- 장점: 더 robust한 솔루션 발견
- 대표적 RL 알고리즘

대표적 RL 알고리즘

- GRPO (Group Relative Policy Optimization)
  - 별도 Reward Model 없이 그룹 내 상대적 보상으로 학습
- DAPO (Dynamic Sampling Policy Optimization)
  - 더 높은 clipping bound로 entropy collapse 방지

LLM 강화학습의 병목: 자원 소모

GPU 메모리 부담

- Policy model + Reference model 동시 실행 필요

느린 Rollout 속도

- 긴 시퀀스의 반복 샘플링 및 처리

Sample Inefficiency

- RL 고유의 낮은 샘플 효율성

기존 해결 시도의 한계

- LoRA (Tina 등)
  - 학습 파라미터 수 감소
  - ✗ Rollout 속도 개선 안됨
- FlashRL (8-bit Rollout)
  - 8-bit rollout + 16-bit logit 모델
  - ✗ 두 모델 동시 로드로 메모리 증가
- QLoRA (NF4)
  - 4-bit NormalFloat 양자화
  - ✗ 오히려 1.5~2배 느려짐

기존 통념: Quantization = 성능 저하

- SFT (Supervised Fine-Tuning)에서의 경험
  - "Quantization Noise는 학습을 방해한다“
  - QLoRA, LQ-LoRA 등의 연구에서 보고된 성능 저하

기존 통념: Quantization = 성능 저하

- SFT (Supervised Fine-Tuning)에서의 경험
  - "Quantization Noise는 학습을 방해한다“
  - QLoRA, LQ-LoRA 등의 연구에서 보고된 성능 저하

그런데... RL에서도 동일할까?

2. 핵심 발견

Quantization이 Exploration을 촉진한다

발견: RL에서는 반대 현상

- KEY FINDING
  - 4-bit Quantized 모델이 16-bit 모델을 능가!
  - LoRA 기반 RL 학습에서 일관되게 관찰

더 빠른 Reward 수렴
학습 초반부터 가파른 상승

더 높은 최종 성능
평가 벤치마크에서 우수

Full Fine-tuning과 유사
파라미터 1%만 학습하면서

왜 Quantization이 RL에서는 도움이 되는가?

Reinforcement Learning
Quantization noise → Higher entropy → Better exploration

Supervised Fine-Tuning
Quantization noise → Loss 증가 → 성능 저하

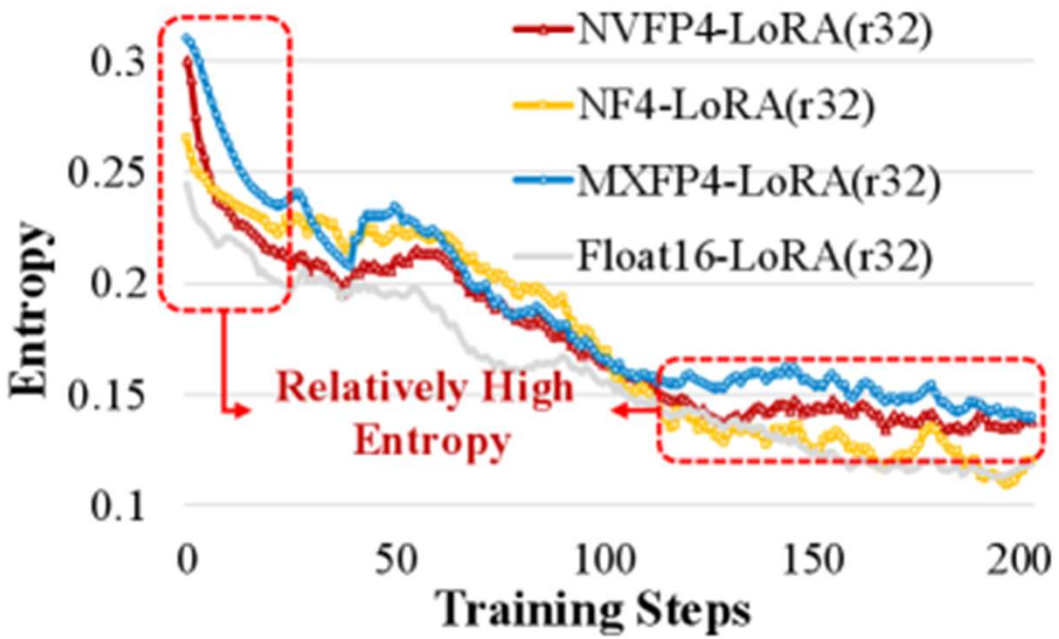
Quantization이 Entropy를 증가시킨다

핵심 관찰

- 모든 4-bit 포맷(NVFP4, MXFP4, NF4)이 16-bit보다 높은 entropy 유지 → Enhanced Exploration!

메커니즘

1. Quantization → Logit perturbation
2. 확률 분포가 평평해짐
3. 다양한 토큰 선택 가능



3. QeRL 프레임워크

Quantization-enhanced Reinforcement Learning

QeRL: 핵심 구성 요소

1. NVFP4 Quantization
   - 4-bit Floating Point 포맷, Marlin 커널로 가속, NF4보다 1.5~2배 빠름

   → 빠른 Rollout + 메모리 절감
2. LoRA Fine-tuning
   - Low-Rank Adaptation, 전체 파라미터의 ~1%만 학습, Main weight는 frozen

   → 효율적 학습 + Gradient 전파
3. Adaptive Quant. Noise
   - 동적 noise 조절, Exponential decay, Zero-parameter overhead

   → 탐색-활용 균형 최적화

왜 NVFP4를 선택했는가?

4-bit Quantization 포맷 비교

- **NF4 (QLoRA)**
  - 정규분포 가정한 4-bit 포맷
  - Lookup table 필요 → 속도 저하
  - **Rollout 1.5~2배 느려짐**
- **MXFP4**
  - FP8 (E8M0) scaling factor
  - Block size: 32 elements
  - **좋지만 NVFP4보다 coarse**
- **NVFP4**
  - FP8 (E4M3) scaling factor
  - Block size: 16 elements (더 정밀)

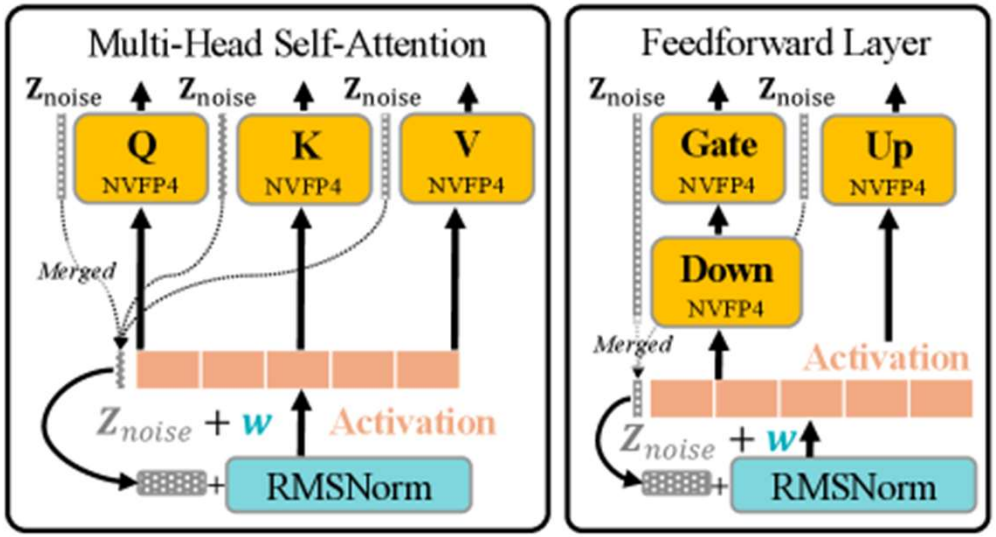
  ✓ Marlin 커널로 고속 추론
- **NVFP4의 장점**
  - 더 세밀한 scaling (16 vs 32 elements)
  - Hopper/Blackwell GPU 최적화
  - 직접 행렬 연산 가능 (lookup 불필요)
  - 1.5배 이상 Rollout 속도 향상

Adaptive Quantization Noise (AQN)

- 문제: Static Quantization Noise
  - 학습 후반부에 exploration 부족
- 해결: Dynamic Noise Schedule
  - Exponential decay로 점진적 감소
- Noise Schedule 수식

$$\sigma(k) = \sigma_{\text{start}} \cdot \left(\frac{\sigma_{\text{end}}}{\sigma_{\text{start}}}\right)^{\frac{k-1}{K-1}}$$

- Zero-parameter 구현
  - RMSNorm의 learnable weight에 noise 통합→ 추가 파라미터 없이 구현!



QeRL 학습 파이프라인

1. Rollout

NVFP4 모델로 빠른 샘플 생성
+ AQN noise

→

2. Reward

Rule-based 보상 계산
(정답 여부)

→

3. Logits

16-bit 정밀도로 logit 계산
(LoRA 포함)

→

4. Update

LoRA만 gradient 업데이트
(16-bit)

vs LoRA (16-bit)

› Rollout이 전체 학습의 병목

› 메모리: 15.2GB (7B 모델)

느린 학습, 큰 메모리 요구

vs QLoRA (NF4)

› NF4 dequantization 오버헤드

› Rollout 오히려 느려짐

메모리만 절감, 속도 저하

QeRL (NVFP4)

› Marlin 커널로 빠른 Rollout

› 메모리: 5.9GB (7B 모델)

속도↑ + 메모리↓ + 성능↑

4. 실험 결과

성능, 속도, 메모리

실험 설정

- **모델**
  - Qwen2.5-3B/7B/14B/32B-Instruct
  - 수학 특화 없이 일반 모델 사용
- **데이터셋**
  - GSM8K: 7,500 샘플 (중등 수학)
  - BigMath: 122,000 샘플 (고난도)
- **RL 알고리즘**
  - GRPO: 3B, 7B 모델 (GSM8K)
  - DAPO: 7B, 14B, 32B 모델 (BigMath)
- **Quantization**
  - NVFP4, MXFP4: AWQ 적용
  - NF4: 기본 설정
  - Calibration: OpenThoughts-114k
- **평가 벤치마크**
  - GSM8K, MATH500, AIME 24/25, AMC 23
- **하드웨어**
  - 속도 테스트: 단일 H100 80GB
  - 최종 모델 학습: 8x H100

GSM8K 성능: QeRL vs 기존 방법

(a) Performance of Qwen2.5-3B-Instruct.

Model	W#	Training	GSM8K
Qwen2.5-3B -Instruct	BF16		61.2
	NF4	-	57.5$_{-3.7}$
	MXFP4	-	59.8$_{-1.4}$
	NVFP4	-	59.4$_{-1.8}$
	BF16	Full	84.4$_{+23.2}$
	BF16	LoRA	76.1$_{+14.9}$
	NF4	LoRA	76.1$_{+14.9}$
	MXFP4	LoRA	73.4$_{+12.2}$
	NVFP4	LoRA	83.3$_{+22.2}$
		+AQN	83.7$_{+22.6}$

(b) Performance of Qwen2.5-7B-Instruct.

Model	W#	Training	GSM8K
Qwen2.5-7B -Instruct	BF16	-	76.3
	NF4	-	70.5$_{-5.8}$
	MXFP4	-	71.3$_{-5.0}$
	NVFP4	-	73.4$_{-2.9}$
	BF16	Full	91.2$_{+14.9}$
	BF16	LoRA	88.1$_{+11.8}$
	NF4	LoRA	85.0$_{+8.7}$
	MXFP4	LoRA	86.4$_{+10.1}$
	NVFP4	LoRA	88.5$_{+12.2}$
		+AQN	90.8$_{+13.5}$

3B 모델
QeRL: 83.7% (Full FT 대비 -0.7%)

7B 모델
QeRL: 90.8% (LoRA 대비 +2.7%)

BigMath 벤치마크: 다양한 모델 크기

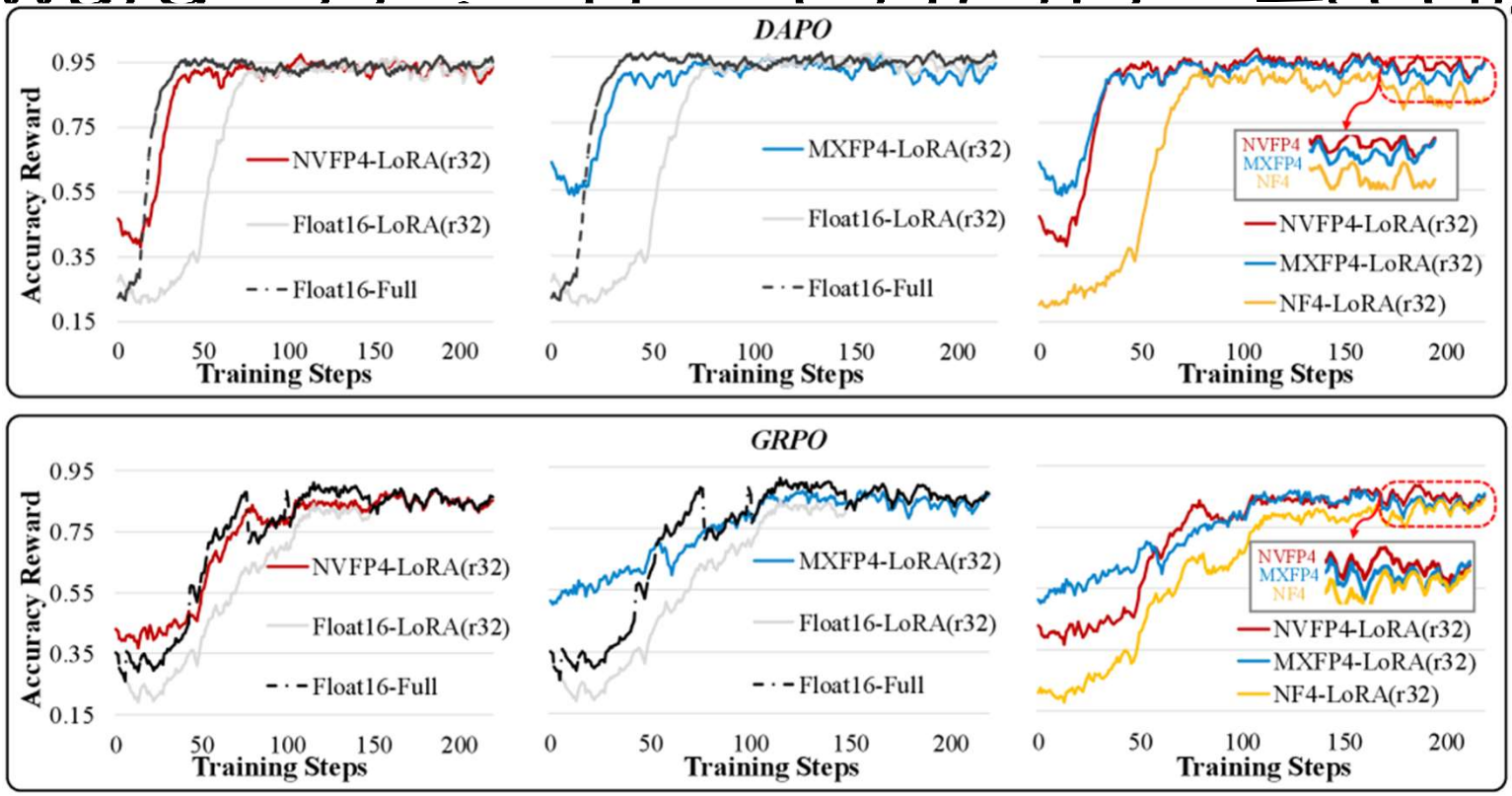
Model	W#	Training	MATH 500	AIME 24	AIME 25	AMC 23	Average↑
7B	BF16	-	74.8	9.2	6.6	25.0	28.9
	NVFP4	-	$73.7_{-1.3}$	$8.3_{-0.9}$	$3.3_{-3.3}$	$17.5_{-7.5}$	$25.7_{-3.2}$
	BF16	Full	$77.4_{+2.6}$	$16.7_{+7.5}$	$10.0_{+3.4}$	$45.0_{+20.0}$	$37.3_{+8.4}$
	BF16	LoRA	$77.0_{+2.2}$	$13.3_{+4.1}$	$10.0_{+3.4}$	$42.5_{+17.5}$	$35.7_{+6.8}$
	NVFP4	LoRA	$76.8_{+2.0}$	$13.7_{+4.5}$	$10.0_{+3.4}$	$47.5_{+22.5}$	$37.0_{+8.1}$
		+AQN	$77.4_{+2.6}$	$15.5_{+6.3}$	$10.0_{+3.4}$	$42.5_{+17.5}$	$36.4_{+7.5}$
14B	BF16	-	78.6	11.3	9.2	45.0	36.0
	NVFP4	-	$76.4_{-2.2}$	$11.2_{-0.1}$	$8.3_{-0.9}$	$40.0_{-5.0}$	$34.0_{-2.0}$
	BF16	Full	$83.2_{+4.6}$	$20.0_{+8.7}$	$15.1_{+5.9}$	$55.0_{+10.0}$	$43.3_{+7.3}$
	BF16	LoRA	$81.0_{+2.4}$	$14.0_{+3.7}$	$13.3_{+4.1}$	$52.5_{+7.5}$	$40.2_{+4.2}$
	NVFP4	LoRA	$79.4_{+0.8}$	$16.7_{+5.4}$	$13.3_{+4.1}$	$52.5_{+7.5}$	$40.5_{+4.5}$
		+AQN	$80.2_{+1.6}$	$17.5_{+6.2}$	$12.6_{+3.4}$	$57.5_{+12.5}$	$42.0_{+6.0}$
32B	BF16	-	81.4	14.0	10.8	52.5	39.7
	NVFP4	-	$80.6_{-0.8}$	$11.3_{-2.7}$	$10.0_{-0.8}$	$45.0_{-7.5}$	$36.7_{-3.0}$
	BF16	Full	$84.0_{+2.6}$	$20.0_{+6.0}$	$23.3_{+12.5}$	$57.5_{+5.0}$	$46.2_{+6.5}$
	BF16	LoRA	$83.6_{+2.2}$	$16.7_{+3.7}$	$13.3_{+2.5}$	$55.0_{+2.5}$	$42.2_{+2.3}$
	NVFP4	LoRA	$81.6_{+0.2}$	$16.7_{+3.7}$	$15.0_{+4.2}$	$52.5_{+0.0}$	$41.4_{+1.7}$
		+AQN	$83.3_{+1.9}$	$16.7_{+3.7}$	$19.2_{+8.4}$	$63.3_{+10.8}$	$45.6_{+5.9}$

14B AMC23
QeRL 57.5% > Full FT 55.0%

32B AMC23
QeRL 63.3% > Full FT 57.5%

핵심
QeRL이 Full FT를 능가!

Reward 곡선: 빠른 수렴, 높은 최종값



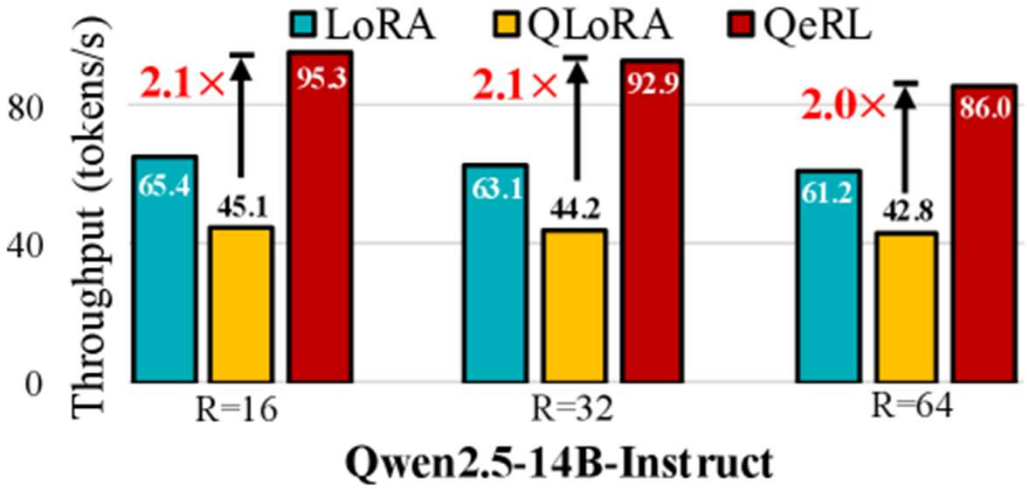
QeRL (NVFP4)
200 steps 내 급격한 상승,
Full FT와 유사한 수렴

Float16 LoRA
500+ steps 후에야 상승 시작, 느린 수렴

해석
Quantization noise가 초반 exploration을 촉진

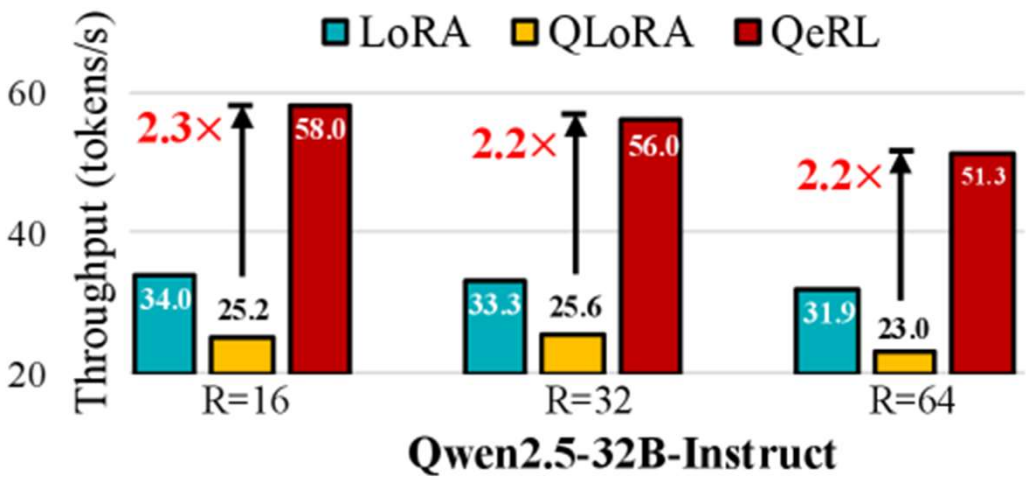
효율성: 속도와 메모리

메모리 및 속도 비교



핵심 수치 (7B)
메모리: 15.2GB → 5.9GB (61% 절감)
속도: 1.2~1.5x 향상 (QLoRA 대비 1.7x)

Rollout Throughput



32B 모델
단일 H100 80GB에서 학습 가능!

효율성: 속도와 메모리

메모리 및 속도 비교

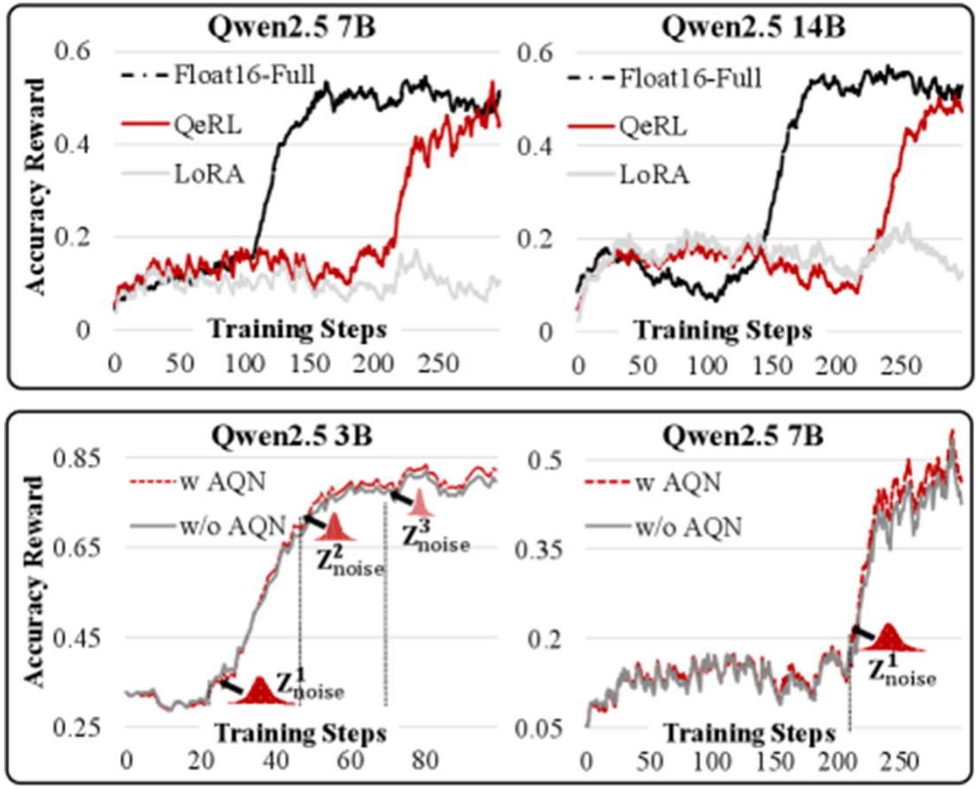
Rollout Throughput

Model	Method	W#	Model Size	Training Speedup (Batch Size)		
				2	4	8
Qwen2.5-7B-Instruct	LoRA	BF16	15.2 GB	-	-	-
	QLoRA	NF4	5.7 GB	×0.8 ↓	×0.8 ↓	×0.7 ↓
	QeRL	**NVFP4**	5.9 GB	**×1.5 ↑**	**×1.4 ↑**	**×1.2 ↑**
Qwen2.5-14B-Instruct	LoRA	BF16	29.6 GB	-	-	-
	QLoRA	NF4	10.2 GB	×0.9 ↓	×0.7 ↓	×0.7 ↓
	QeRL	**NVFP4**	10.6 GB	**×1.4 ↑**	**×1.2 ↑**	**×1.2 ↑**

핵심 수치 (7B)
메모리: 15.2GB → 5.9GB (61% 절감)
속도: 1.2~1.5x 향상 (QLoRA 대비 1.7x)

32B 모델
단일 H100 80GB에서 학습 가능!

Ablation: AQN의 효과



- **w/o AQN (빨강)**
  - 초반: quantization noise로 빠른 상승
  - 후반: 정적 noise로 exploration 한계
- **w/ AQN (검정)**
  - 초반: quantization + 추가 noise
  - 후반: 점진적 감소로 안정적 수렴
- **핵심 관찰**
  - Reward 수렴 시점에서 AQN이 exploration 공간을 확장
    → 추가 reward 향상

5. 결론 및 시사점

핵심 메시지

PARADIGM SHIFT

Quantization ≠ 항상 성능 저하

RL에서는 오히려 exploration을 촉진하여 성능 향상

성능

Full FT와 유사하면서 LoRA 능가

속도

1.5x+ 학습 속도 향상

메모리

~60% 메모리 절감

확장성

32B 모델, 단일 GPU 학습

한계 및 향후 연구

현재 한계

- **모델 크기 제한**
  - 70B 이상 모델에서의 검증 미완료
- **태스크 범위**
  - 수학 추론에 집중
  - 코드, 일반 언어 태스크 미검증
- **하드웨어 의존성**
  - NVFP4 가속은 Hopper/Blackwell GPU 필요

현재 한계

- **더 큰 모델 확장**
  - 70B+ 모델에서의 효과 검증
- **다양한 태스크**
  - 코드 생성, 대화, 요약 등
  - 다른 RL 시나리오에서 검증
- **Noise Schedule 최적화**
  - 태스크별 최적 noise 전략 연구

QeRL: Quantization-enhanced Reinforcement Learning

Quantization Noise가 RL Exploration을 촉진한다

기존 (SFT)
Quant = 성능↓

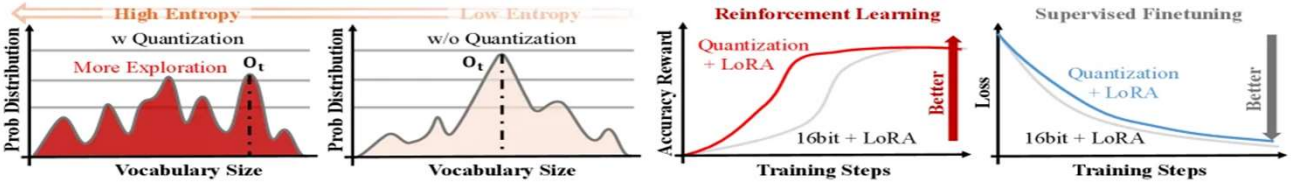
→

발견 (RL)
4-bit > 16-bit!

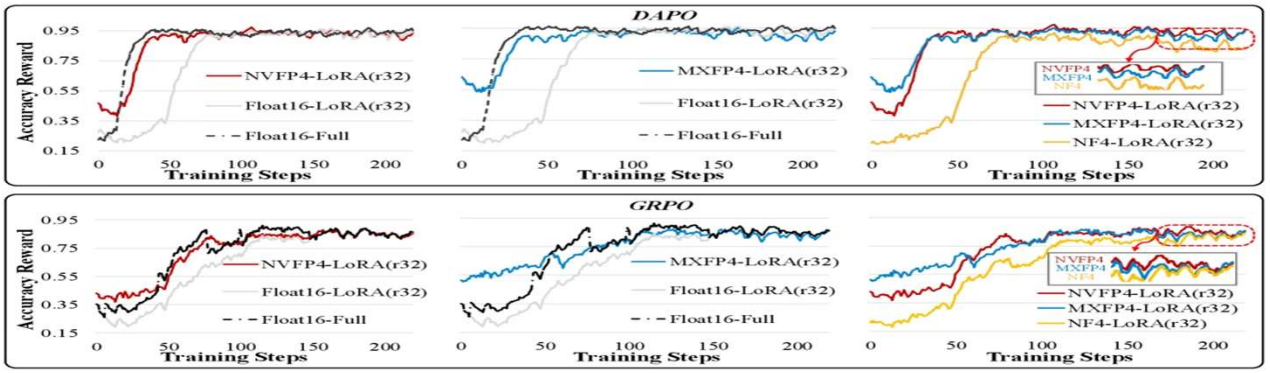
→

QeRL
NVFP4+LoRA+AQN

메커니즘: Quantization → High Entropy → Exploration



Training Reward (DAPO/GRPO)



GSM8K (7B)

Full FT	91.2
LoRA	88.1
QeRL	**90.8**

1.7x	**61%**	**32B**
속도↑	메모리↓	1xH100

QeRL 구성

① NVFP4 (Marlin) ② LoRA ③ AQN (동적 noise)

github.com/NVlabs/QeRL

NVIDIA · MIT · HKU · THU