텍스트로 풀어보는 장면의 재구성

SFT · Listwise Ranking · Vision-Encoder LoRA를 결합한 24-way 순열 추론

팀명	팀원	기반 모델	Public Score
하라는행샤는안하고	오수현, 김하람, 권성안	Qwen3-VL-32B-Instruct (NF4 4-bit + LoRA)	0.92146

요약

본 과제는 캡션과 무작위로 뒤섞인 4개의 비디오 프레임을 입력으로 받아 원래의 시간 순서를 복원하는 문제이다. 가능한 출력은 4! = 24개의 순열로 제한되며, 네 위치가 모두 일치해야 정답으로 인정되는 Exact Match가 평가 지표다. 따라서 본 연구는 문제를 자유 생성이 아니라 24개 후보 중 하나를 선택하는 제약 분류·랭킹 문제로 재정의하였다. 가장 중요한 설계 판단은 VLM을 처음부터 구축하지 않고 공개 사전학습 파라미터를 과제에 맞게 미세조정하는 것이다. 입력이 이미지와 텍스트를 동시에 포함하므로, 제한된 데이터·자원으로 시각·언어 표현과 결합을 모두 새로 학습하는 것은 난도가 높다. 이에 Qwen3-VL-32B-Instruct에 NF4 4-bit 양자화와 LoRA를 적용하여 전체 파라미터의 약 0.8%만 학습하였다. 과제 전용 학습에서는 정답 문자열의 likelihood를 높이는 SFT와, 정답이 혼동 후보보다 높은 점수를 갖도록 만드는 listwise ranking을 하나의 학습 과정에서 70:30으로 혼합하였다. 여기에, error analysis로 드러난 순수 시각 추론 병목을 해소하기 위해 그동안 freeze했던 vision encoder에 LoRA를 추가하였다. Inference 시에는 24개 유효 순열을 모두 constrained scoring하고 TTA(k=3)와 학습 데이터 기반 identity prior(α=0.5)를 적용하였다. 최종 모델은 SFT-only 실험의 약 0.89에서 Public LB 0.92146으로 약 +3.146%p 개선되었다.

핵심 기여

01 공개 사전학습 VLM의 효율적 적응	02 24-way constrained 문제로의 재구성	03 SFT와 Listwise의 통합 학습	04 비전 인코더 LoRA로 시각 병목 해소
Qwen3-VL-32B에 NF4 4-bit와 LoRA(r=32)를 적용하고, 학습 파라미터를 0.8%로 제한하였다 .	학습·추론·평가를 모두 24개 유효 순열 중 정답을 선택하는 구조로 통일하여 형식·파싱 실패를 제거하였다	SFT 70%와 listwise 30%를 행 단위로 확률적으로 혼합하고, 인접 스왑·중간 난도 후보를 직접 이기도록 학습하였다.	Error analysis가 지목한 순수 시각 추론(20.8%)을 겨냥해 freeze됐던 vision encoder를 LoRA로 학습, 최종 도약(0.90924→0.92146)을 달성하였다.

1. 문제 정의와 전체 설계

입력은 캡션 S와 임의의 순서로 제시된 네 프레임 $X = (x_1, x_2, x_3, x_4)$이며, 출력은 각 입력 프레임의 원래 시간 위치를 나타내는 순열 π이다. 가능한 정답 공간은 S_4의 24개 원소로 완전히 고정된다. 따라서 모델이 임의의 텍스트를 생성하도록 두기보다, 각 유효 순열의 점수를 비교해 하나를 선택하도록 설계하였다.

$$\hat{\pi} = argmax\,(\pi\ in\ S_4)\ score\,(\text{모델}, \text{데이터}, \pi)$$

1.1 왜 공개 사전학습 파라미터를 미세조정했는가

이 과제의 입력은 이미지와 자연어를 함께 다루는 VLM 형태다. 모델을 처음부터 구축하려면 시각 특징 추출, 텍스트 이해, 시각-언어 결합, 시간적 선후 판단을 동시에 학습해야 한다. 제공 데이터·자원만으로 이 전 과정을 새로 학습하는 것은 위험과 난도가 지나치게 높다고 보았다. 대신 공개 사전학습 VLM을 출발점으로 삼고, 과제에 직접 필요한 순열 판단 능력만 미세조정하는 전략을 선택하였다.

2. 데이터 정제와 검증 분할

학습 전에 검은 프레임 260건과 중복 프레임 209건을 식별하고, 최종적으로 414행(4.3%) 을 제거하였다. 정제 결과는 train_clean.csv 8,121행과 clean_val.csv 1,000행으로 저장되며, 플래그는 train_flags.csv에 기록한다. 데이터 분할은 make_split.py와 seed 42로 고정해 재현 가능하게 구성하였다.

구분	행 수	용도	재현 정보
train_full.csv	8,871	최종 학습	train_clean 8,121 + clean_val 중 750
val_small.csv	250	학습 중 실제 규칙 모니터	seed 42 고정
제거 행	414 (4.3%)	검은/중복 프레임 정제	플래그 파일 보존

또한 학습 라벨 분포에서 두 통계적 사실을 확인·활용하였다: ① identity 순열 [1,2,3,4]는 No_ordering 행 때문에 15.5%, ② 실제로 뒤섞인 행은 identity가 0%(균등 기대치 4.17% 대비). 이 사전확률은 추론(5절)의 재보정에 사용된다

3. SFT-Listwise 통합 미세조정

베이스 모델에서 바로 시작하여 각 데이터 행마다 SFT 또는 listwise 갈래를 확률적으로 선택한다. SFT를 먼저 끝낸 뒤 listwise를 별도로 수행하는 순차 학습이 아니라, 처음부터 두 목적을 하나의 학습 과정에 섞는 방식이다. 샘플링 비율은 SFT 70%, listwise 30%로 설정하였다.

SFT 갈래 | 70%	Listwise 갈래 | 30%
정답 완성 토큰에만 masked next-token loss를 적용하여 정답 형식과 정답 순열의 가능도를 높인다.	정답과 6개의 혼동 후보를 함께 채점하고, 정답이 후보 집합에서 1등이 되도록 cross-entropy를 적용한다.

3.1 SFT 갈래: 정답을 직접 가르치기

- 정답 순열을 나타내는 완성 문장을 만들고, 프롬프트가 아닌 정답 완성 토큰에만 손실을 적용한다.
- 매 epoch마다 네 프레임을 다른 순서 sigma로 제시하는 프레젠테이션 셔플 증강을 사용한다. 모델이 고정 입력 위치가 아니라 프레임 내용으로 판단하도록 유도한다.

3.2 Listwise 갈래: 헷갈리는 후보를 직접 이기기

오답 분석에서 인접 스왑이 전체 오류의 약 36%, 두세 쌍이 어긋나는 중간 수준의 뒤섞임이 약 30%를 차지하였다. 일반적인 SFT도 token 단위에서는 정답과 다른 token을 구분하지만, 실제 추론 시 경쟁하는 완성된 순열 후보들의 전체 점수를 하나의 후보 집합 안에서 직접 비교하지는 않는다. 이에 정답 순열과 주요 오답 순열을 동시에 채점하고, 정답이 후보 집합 내에서 가장 높은 확률을 얻도록 하는 listwise 목적을 추가하였다.

후보 유형	개수	구성 원리	학습 목적
정답	1	ground-truth 순열	후보 집합의 1등
인접 스왑	3	Kendall 거리 1	주요 근접 오류 직접 억제
중간 뒤섞임	2	Kendall 거리 2-3	두세 쌍이 어긋난 오류 대응
랜덤	1	그 외 순열	넓은 음성 후보 대비

각 후보 c의 점수 s(c)는 후보 완성 문장의 token log-probability 합으로 계산한다. 이 점수 자체는 확률이 아니므로, 다음 softmax를 통해 7개 후보 내의 상대적 선택 확률로 변환한다.

$$L_{list} = -\log\left(\frac{exp(s(y^*)}{\Sigma_{c\in C} exp(s(c))}\right)$$

후보 점수 s(c)는 별도의 생성 결과를 다시 파싱하지 않고 배치 forward에서 얻은 logit으로부터 후보 완성 문장의 log-prob 합을 직접 계산한다. 전체 학습은 z ~ Bernoulli(0.30)에 대해 $L = (1 - z)L_{SFT} + zL_{list}$ 로 구현된다.

3.3 두 목적을 처음부터 섞는 이유

SFT 이후 listwise를 순차적으로 붙이면 두 번째 단계가 첫 번째 단계에서 형성된 표현을 약화시킬 가능성이 있다. 본 방법은 SFT의 "정답이 그럴듯하도록 만들기"와 listwise의 "정답이 경쟁자를 이기도록 만들기"가 상충하지 않고 같은 판단을 다른 각도에서 강화한다는 가설에 따라 두 갈래를 동시에 운영한다. 실제로 순차 방식(LB 0.9075) 대비 통합 방식이 우위(LB 0.90924)를 보였다.

3.4 최종 학습 설정

항목	최종 설정	항목	최종 설정
베이스	Qwen3-VL-32B-Instruct	양자화	NF4 4-bit, double-quant
연산	BF16, 비전타워 BF16 유지	LoRA	r=32, LLM decoder linear
학습 파라미터	전체의 0.8%	listwise 비율	30%
학습률	1e-4	epoch	1 (8,871행)
메모리	gradient checkpointing ON	학습 시간	A100 80GB 약 19시간

4. Vision Encoder LORA - 최종성능 도약

4.1 동기: 순수 시각 추론이 병목이다

3절까지의 모델은 LoRA를 LLM decoder에만 적용하고 vision encoder는 완전히 freeze했다. 그러나 4장의 유사한 프레임을 순서 매기는 일은 본질적으로 프레임 간 미세한 시각 차이(fine-grained visual difference) 를 읽는 과제이며, error analysis 결과 caption에 시간 단서어가 전혀 없는 sample이 전체의 20.8% 에 달했다. 이런 sample은 언어 단서 없이 오직 시각 추론으로만 풀 수 있으므로, LLM 측만 fine-tuning하는 한 원리적으로 도달할 수 없는 병목이었다.

4.2 방법: freeze된 눈을 안전하게 열기

통합 학습으로 얻은 best 모델(3절, LB 0.90924)에서 이어서, vision transformer block의 linear layer 108개(LoRA tensor 216개)에 LoRA를 추가하고 LLM·vision LoRA를 함께 fine-tuning하였다.

4.3 구현상 핵심 이슈와 해결

Gradient checkpointing 사용 시 vision encoder의 입력(pixel values)이 gradient를 요구하지 않아, vision LoRA에 gradient가 전혀 흐르지 않는 문제가 있었다. 이는 본 학습 전 5분 규모의 사전 검증(smoke test)으로 포착하였다. Vision patch-embed 출력에 require-grad hook을 등록해 backward graph가 vision block을 통과하도록 수정하여, vision LoRA 216개 parameter 전부에 정상적으로 gradient가 흐름을 확인한 뒤 본 학습을 진행하였다.

4.3 결과

Vision LoRA는 시작점(통합 모델)의 validation accuracy 0.564에서 학습이 진행되며 최고 0.600(+3.6%p)까지 상승했고, 이 best checkpoint가 실제 Public LB 0.90924 → 0.92146 향상으로 이어졌다. Validation 점수의 향상이 leaderboard 향상으로 transfer됨을 확인한 것이다.

5. 24-way 제약 추론

추론 역시 학습과 동일하게 "24개 중 정답 고르기"로 통일하였다. 자유 생성 후 문자열을 파싱하는 대신 모든 유효 순열의 후보 완성 문장을 직접 채점하므로, 출력은 항상 유효한 순열이며 형식 오류나 파싱 실패가 발생하지 않는다.

6. 실험 결과와 분석

실험	학습/추론 설정	Public LB	결론
Trial 3	SFT-only	약 0.89	정답 가능도는 학습하지만 근접 후보 억제가 부족
Trial 4 Final	SFT 70% + Listwise 30% TTA k=3, alpha=0.5	0.90924	SFT-only 대비 약 +1.924%p
Trial 4 Variant	동일 모델, TTA k=5	0.90575	TTA 증가가 항상 성능 향상으로 이어지지 않음
Trial 4 (Vision)	통합 모델 + vision encoder Lora	0.92146	순수 시각 병목 해소

6.1 오답 유형에서 학습 목표로

오답은 주로 한 쌍의 앞뒤만 바뀌는 인접 스왑(약 36%)과 두세 쌍이 어긋나는 중간 수준의 뒤섞임(약 30%)으로 나타났다. 두 유형이 전체 오류의 약 66% 를 차지하므로, listwise 후보 집합도 이 분포를 반영해 인접 스왑 3개와 중간 후보 2개를 우선 포함하였다. 이는 오류 분석을 단순 진단으로 끝내지 않고 학습 데이터 구성에 직접 연결한 부분이다.

6.2 시각 병목에서 비전 LoRA로

Caption에 시간 단서가 없는 sample(20.8%)은 언어 단서만으로는 풀 수 없다. 이 관찰이 vision encoder LoRA(4절)의 직접적 근거가 되었으며, validation accuracy 0.564→0.600, Public LB 0.90924→0.92146의 향상으로 그 효과가 실증되었다. 즉 error analysis → 병목 특정 → 표적 개입(targeted intervention) 의 진단-처방 루프가 두 핵심 기여(Listwise, Vision LoRA)를 각각 이끌었다.

7. 효율성, 재현성 및 규정 준수

32B 규모의 VLM을 사용하지만 NF4 4-bit 양자화, LoRA, gradient checkpointing, KV cache 기반 후보 채점을 통해 계산량과 메모리 사용을 제어하였다. 통합 학습은 A100 80GB에서 약 19시간, 비전 LoRA 미세조정은 H100 NVL에서 수 시간이 소요되었고, 추론은 RTX 3090 24GB 환경에서 피크 약 20.6GB 로 실행된다. 테스트 819행 전체는 24시간 제한 안에 처리된다.

항목	측정/설정값	비고
학습 GPU 및 시간	A100 80GB, 약 19시간 + 비전 Lora 미세조정	1 epoch, train_full 8,871행
학습 중 모니터	Val_small 250행	실제 규칙 기준
추론 GPU	RTX 3090 24GB	대회 제출 환경
Peak VRAM	약 20.5GB	24GB 단일 GPU 내 실행
전체 테스트	819행, 24시간 이내	H100 기준 약 14초/행 실측
모델/제출 크기	LoRA 어댑터 약 1.1GB + 베이스 32B	전체 80GB 제한 이내
외부 API 비용	없음	외부 상용 API 미사용

7.1 재현 파이프라인

01	환경 구성	setup_box.sh로 의존성과 Qwen3-VL-32B 가중치를 준비한다.

02	데이터 분할 및 검증	make_split.py, test_perms.py, smoke_unified.py / smoke_vision.py로 seed 42 분할과 순열 수학, 두 학습 갈래 및 비전 LoRA gradient 흐름을 확인한다.
03	통합 학습	train_unified.py로 SFT와 listwise가 혼합된 단일 LoRA 어댑터를 학습한다.
04	비전 **Lora** 파인튜닝	train_vision.py로 통합 모델에서 이어 비전 인코더 LoRA를 학습한다.
05	제약 추론	infer.py에서 24-way scoring, TTA k=3, prior α=0.5를 적용해 submission.csv를 생성한다

7.2 규정 준수 요약

점검 항목	적용 내용	상태
학습 데이터	제공된 train 데이터만 사용	충족
외부 데이터	사용하지 않음	충족
앙상블	단일 LoRA 어댑터 사용	충족
추론 방식	24-way scoring + 허용된 TTA	충족
재현성	seed 42, 분할/단위테스트/실행 스크립트 제공	충족

8. 결론

본 방법의 출발점은 "multimodal 모델을 처음부터 만들기보다 공개된 pretrained parameter를 과제에 맞게 fine-tuning한다"는 판단이다. 이 전략은 32B VLM의 기존 representation을 활용하면서도 LoRA로 학습 범위를 약 0.8%까지 줄여 현실적인 계산 비용을 유지한다. 여기에 문제를 24-way 순열 선택으로 재정의하고, SFT와 listwise ranking을 통합 학습(joint training) 하여 정답의 likelihood와 근접 후보 대비 우위를 함께 학습하였다. 마지막으로, error analysis가 지목한 순수 시각 추론 병목을 vision encoder LoRA로 해소하여 최종 Public LB 0.92146을 달성했으며, 단일 RTX 3090에서 실행 가능한 inference pipeline을 구축하였다. 성능 향상의 매 단계가 데이터 기반 관찰(data-driven observation)에서 출발했다는 점이 본 접근의 핵심 특징이다.