캡션이 지목한 토큰만 보는 단일 모델로 셔플된 4프레임을 시간순 재배열한다

Cross-Targeted FitPrune · One-Pass Score24 Head · Stackelberg Two-Time-Scale QLoRA

팀 윤가진손 (yhjs)
민윤홍(팀장) · 김가람 · 손석현 · 이진아

2026. 07. 27 · github.com/myunhh/SNU-AI-Challenge-ygjs

공개 LB 0.93193 단일 Qwen3-VL-32B + QLoRA

0.93193	32B · 4bit	50%	16회	0.42%
공개 리더보드 EM 무작위 기준선의 22.4배	Qwen3-VL-32B-Instruct NF4 base + bf16 LoRA	LLM에 넣기 전에 잘라내는 시각 토큰의 비율	샘플당 채점 횟수 균형 8뷰 × 2패스	학습 대상 134.3M 전체 320억 파라미터 중

그림 1 · 학습 경로 — 샘플 1개당 forward 1회 – Prune 적용X(SFT -> DPO) -> Prune 적용 O(SFT) 총 3회 학습함

프루닝은 LLM에 들어가기 전에 딱 한 번 일어난다. 점선 블록은 동결 구간이자 `@torch.no_grad()` 구간이라, **토큰 선택 자체에는 그래디언트가 흐르지 않는다.** 헤드와 본체는 서로 다른 학습률로 갱신된다.

그림 2 · 추론 경로 — 샘플 1개당 forward 16회, 가중치는 그대로

test-time augmentation - 가중치는 하나뿐이고 입력만 바꿔 16번 채점한다

표 1 최종 제출 시스템 구성

구성 요소	설정	선택 근거
베이스 · 양자화	Qwen3-VL-32B-Instruct · NF4 4bit base + 4bit QLoRA	32B를 24 GB 카드에 올리는 유일한 경로. 어댑터가 양자화를 통과하지 않아 학습과 서빙의 수치 조건이 일치한다
어댑터	LoRA r=16, α=32, dropout 0.05 · 언어 64층 × 7 projection = **448 모듈** · `134,217,728`	비전 타워는 bf16 고정, LoRA를 붙이지 않아 시각 정확도를 보존한다
분류 헤드	Linear(5120 → 24), bias 없음, fp32 · `122,880`	lm_head의 A~X 24개 행을 그대로 가져와 초기화한다
학습 대상	**Train dataset 전부, 파라미터는 32B의 0.42%**	**헤드는 이 중 0.09%뿐이지만 학습률은 5배를 받는다**
학습	1,500스텝 · 유효배치 16 · 본체 5e-5 / 헤드 2.5e-4 · cosine · KT λ=0.5	1200스텝 이후로는 loss가 움직이지 않고 수렴하기 때문에 1200스텝으로 제출
추론	균형 TTA8 + 풀토큰 stage-2(전 샘플) + 토큰 부스트 0.5 + 모션 항 0.3	추가적인 학습 없이 정확도를 올릴 수 있는 추론 방법

설계 원칙 **학습과 추론이 같은 함수를 부른다**

- **공유 경로.** 프롬프트 생성과 토큰 선택(prepare→keep_mask→forward_prepared)을 두 경로가 그대로 공유한다. 차이는 추론에만 있는 다중 뷰 집계뿐이다.
- **0에서 되돌아간다.** 모션·부스트·객체성·MMR은 가중치를 0으로 두면 이전 파이프라인을 비트 단위로 재현한다. 표 3·4의 모든 증분이 정확한 ablation이다.

주의 **프루닝은 미분되지 않는다**

- prepare()와 keep_mask()는 모두 @torch.no_grad()다. 어느 토큰을 남길지는 학습되는 규칙이 아니라 매 forward마다 새로 계산되는 **결정적 게이트**다.
- 역전파가 전하는 메시지는 "다른 토큰을 고르라"가 아니라 **"이 절반을 봤다는 전제에서 읽는 법(LoRA)과 채점법(헤드)을 고치라"**이다. 그래서 `motion_weight` 같은 선택 파라미터를 바꾸면 재학습이 필요하고, 표 3의 07-23 → 07-24 행이 그 정렬 효과를 측정한 것이다.

왜 캡션으로 시각 토큰을 고르는가

Fit and Prune: Fast and Training-free Visual Token Pruning forMulti-modal Large Language Models – AAAI 2025 논문 내용 활용
LEARNPRUNER: RETHINKING ATTENTION-BASEDTOKEN PRUNING IN VISION LANGUAGE MODELS – ICLR 2026 논문 내용 활용

문제 1 토큰이 비싸다

- 프레임 4장을 고해상도로 넣으면 샘플당 시각 토큰이 **약 4,400개**까지 늘어 32B 모델의 메모리와 속도를 압박한다.
- attention 비용은 토큰 수의 제곱에 비례한다. **절반을 버리면 비용은 1/4**이 된다.

단점 어느 프레임이 캡션의 어느 부분인지 모른다

- 캡션 전체를 한 벡터로 뭉치면 4장이 모두 같은 기준과 비교되어 이미지별 구분이 사라진다.
- 그렇다고 "이미지 i ↔ 캡션 조각 j"를 미리 짝지을 수도 없다. **그 대응 관계를 알아내는 것이 곧 우리가 풀어야 할 문제**이기 때문이다.

문제 2 배경이 예산을 독식한다

- 캡션의 장면 명사(pool, snow, ice)와 매칭되는 **대면적 배경**이 토큰 예산을 가져갔다.
- 정작 시간 순서를 가르는 소형 전경 물체(스키어, 공, 얼굴)가 밀려나 있었다. 이른바 **stuff-over-things**다.

해법 배정을 아예 풀지 않는다

- 캡션을 규칙 기반으로 4개 이벤트로 분해한 뒤, 각 시각 토큰을 **4개 이벤트 전부와** 교차 채점(4×4)하고 이벤트 축으로 **최댓값**을 취한다.
- 어떤 이벤트와든 강하게 반응하면 살아남는 **합집합 의미론**이라, 맞지 않는 이벤트가 준 낮은 점수는 자동으로 탈락한다. 대응 관계를 몰라도 된다.
- 분해는 접속사·문장부호 기반의 결정적 규칙으로만 한다. 모델 생성 텍스트를 학습에 넣는 것이 규정 위반이기 때문이고, 같은 함수를 학습·추론에 함께 써 불일치를 원천 차단한다.

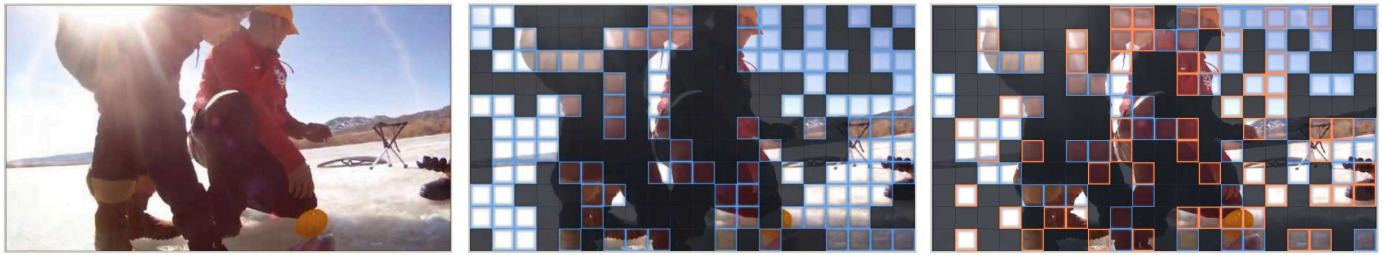
기회 캡션은 4장 전체를 서술한다

- 이 과제에서는 캡션에 4장 전체의 내용이 담겨 있음이 보장된다.
- 따라서 "캡션이 지목하는 것"을 기준으로 고르면 순서 판별 신호는 보존하면서 배경 노이즈만 덜어낼 수 있다. FitPrune(AAAI'25)이 보고한 **텍스트 조건부(cross) 신호가 시각 self 신호보다 우월**하다는 결과와 방향이 같다.

인용 정확성 cross 단독은 논문의 권장 구성이 아니다

- FitPrune Table 5(GQA)는 베이스라인 `62.0` / Cross `60.1` / Self `55.4` / **Self+Cross `61.5`**다.
- 채택 근거인 "cross > self"는 유효하지만, 본문도 *"insufficient when used alone"*이라고 명시한다. 우리는 cross 계열만 쓰므로 **그 보완 자리를 객체성·모션(캡션과 무관한 순수 시각 신호)으로 메웠다.**

그림 3 · 동기의 실측 근거 — 같은 샘플(`diZi5g`)·같은 프레임·같은 예산(220 토큰 중 110개 = 50%)



(a) 원본 프레임
두 사람이 얼음 위 낚시 구멍 옆에 앉아 있다

(b) 순수 캡션 코사인 top-k
균일한 얼음·눈 배경이 선택을 독식

(c) 최종안 (객체성 + MMR + 모션 + boost)
사람·장비가 keep-set에 복귀

파란 테두리 칸만 LLM에 들어가고 어두운 칸은 버려진다. (b)는 `--objectness-weight 0 --mmr-lambda 0 --motion-weight 0`으로 실제 재현한 07-16 이전 경로다. 예산은 같은데 배치가 달라진다.

주황색 테두리 칸은 Prune되지 않은 토큰 중 점수가 높은 토큰의 상위 50%이다. 다시한번 Transformer에 들어가 사용됨

구현 원 논문에서 무엇을 버렸나

- 레이어별 스케줄, 이진탐색, LLM 내부 어텐션 통계 수집은 모두 버리고 **LLM 진입 직전 단 한 번의 임베딩 공간 행렬곱**으로 대체했다. 어텐션 구현을 건드리지 않아 SDPA 고속 경로가 유지되고, 프루닝 자체의 오버헤드는 사실상 없다.
- **FitPrune 저장소 코드는 한 줄도 쓰지 않았고 구조적으로 쓸 수도 없다.** 원저작은 `modeling_llama.py`를 개조해 매 레이어 어텐션 행렬을 실체화하지만, 우리는 `Qwen3VLTextModel`을 직접 호출하는 별개 경로에서 pre-LLM 단일 컷을 한다.
- 스코어링 모듈은 **모델 의존성이 0인 순수 텐서 로직**이라 GPU 없이 도는 단위 테스트 112개가 불변식을 고정한다. 4×4 교차 채점은 논문에 없는 자체 설계분이다(FitPrune은 이미지 1장 입력이라 이 문제 자체가 없다).

그림 4 · 프루닝 스코어링 상세 — 그림 1·2의 "FitPrune" 블록 내부

E1 "The camera pans left..." · **E2** "kneeling beside a fishing hole" · **E3** "shifts from a frontal to a rear view" · **E4** "a skier moves forward and jumps" · **OBJ** 객체성 (w=0.3) · **MOTION** 모션 (w=0.3)

밝을수록 고득점이며 모두 같은 프레임(`diZi5g img2`)이다. **E1~E4가 서로 다르게 생긴 것이 rank-1 퇴화를 고친 증거**다. 수정 전에는 육안으로 구별되지 않았다. E3처럼 "...as"로 어중간하게 끊긴 기계적 분할이 섞여도 max-pool의 합집합 의미론 덕에 선택이 무너지지 않는다. 오른쪽 두 장은 캡션을 보지 않는 순수 시각 신호로, 원논문에서 self-attention이 맡던 보완 역할을 대신한다.

입력 · 출력 · 학습, 세 지점에 같은 원리를 적용한다

Rethinking Neural Network Learning Rates: A Stackelberg Perspective - ICML 2026 논문 내용 활용

입력 무엇을 볼지 고르는 단계

- 최종 중요도는 세 신호를 이미지별 min-max 정규화 후 결합한 값이다. **캡션 코사인 0.4 + 객체성 0.3 + 모션 0.3.**
- 잔류 토큰은 단순 상위 절단이 아니라 greedy **MMR**로 고른다(`score − λ·relu(선택 토큰과의 최대 코사인)`, λ = 0.5). 중복 배경이 소수의 대표로 압축되고, 남은 예산이 새 정보를 담은 토큰으로 흐른다.

출력 24개를 한 번에 채점하는 단계

- 정답 공간이 24개로 닫혀 있는데 글자를 생성할 이유가 없다. 마지막 은닉 상태 위의 `Linear(5120 → 24)` 하나가 24개 순열 후보를 **단 한 번의 forward로 동시에 채점**한다.
- 초기 가중치는 무작위가 아니다. 기존 언어모델 head에서 정답 글자 A~X에 해당하는 24개 행을 그대로 가져와, **이미 검증된 채점 논리와 수학적으로 동일한 함수**에서 출발한다.
- 프루닝으로 입력 길이가 달라져도 **출력 차원은 항상 24**다. 뷰·패스마다 나온 확률을 곧바로 평균할 수 있고, 1위와 2위의 점수 차(**마진**)를 2차 의사결정에 재사용할 수 있다. 형식 붕괴나 파싱 실패도 구조적으로 없다.

학습 두 모듈을 다른 속도로 가르치는 단계

- 새로 붙인 헤드는 백지에 가깝고, QLoRA로 미세조정되는 본체는 이미 유능하다. 같은 학습률로 끌면 헤드가 못 따라오거나 본체가 망가진다.
- 게임이론의 **Stackelberg 리더-팔로워** 구조를 빌려 본체를 리더(느린 시간축), 헤드를 팔로워(빠른 시간축)로 두고 헤드 학습률을 본체의 **5배**로 잡았다.
- 헤드에만 걸리는 weight decay 0.1은 장식이 아니다. 목적함수가 헤드에 대해 **강볼록**이어야 한다는 이론적 가정을 성립시켜 팔로워의 최적 응답을 유일하게 만드는 수렴 조건이다.
- 두 값 모두 원문(*Rethinking Neural Network Learning Rates: A Stackelberg Perspective – ICML 2026*)이 실제로 쓴 설정이다. 배율 5는 Figure 2, 정규화 0.1은 실험 절 기재값이다.

실패 기록 1차 학습은 3분의 2를 버렸다

- 무작위 초기화로 시작한 1차 학습은 손실이 **ln(24) ≈ 3.178**(24지선다를 균등하게 찍는 상태) 근방에 정체하다 코사인 스케줄의 약 **65% 지점**에서야 탈출했다.
- 원인은 헤드 학습률이 작은 유효배치(스텝당 4샘플)의 그래디언트 노이즈에 비해 과도했던 것이다.
- 이때문에 2000스텝 학습 중 약 1300스텝을 낭비하며 직접적인 학습은 700스텝밖에 되지 않아 정확도가 낮았다.

처방 warm-start

- Prune되지 않은 데이터로 학습한 어댑터 위에 Prune된 데이터를 추가 학습하자라는 warm-start 아이디어로 시작
- 유효배치 4 → 16 확대와 **이미 검증된 어댑터에서의 warm-start**. 실측 궤적은 ce 10.9(1스텝) → 2.44(20스텝) → 1.2대 수렴으로, 병리적 플래토가 사라졌다.
- 베이스 모델·QLoRA 구성(r16/α32, 언어 64층 7 projection) · 라이브러리 버전이 동일함을 전수 확인했고, 전이 실패에 대비해 **10스텝 테스트 학습** (평균 ce < 3.0)를 러너에 내장했다.

보조손실 얼마나 틀렸는지를 가르친다

- 교차 엔트로피는 "틀렸다"만 알려준다. 인접 두 프레임만 뒤바뀐 예측과 순서를 완전히 뒤집은 예측이 같은 벌점을 받는다.

$$\mathcal{L} = \mathrm{CE}(p, y) + \lambda \cdot \mathbb{E}_{c\sim p}[\,\mathrm{KT}(c, y) / 6\,],\ \lambda = 0.5$$

- 정답이 one-hot일 때 이 항은 **정확히 0**이라 CE의 최적점과 충돌하지 않고, 균등 분포에서는 정확히 0.5가 된다(S_4의 평균 Kendall 거리 3을 최댓값 6으로 정규화).
- CE가 가리키는 지점은 그대로 두고 그 주변의 **오차 기하만 재배치**한다.

노출 정책 잘린 입력과 온전한 입력을 모두 보여준다

- 학습 시 샘플마다 확률 **0.75**로만 프루닝을 적용해 모델이 두 종류의 입력을 모두 보게 한다. 추론에서 두 패스를 모두 수행하는 stage-2가 학습 분포 안에서 작동하게 하는 장치이지, 정규화 목적의 드롭아웃이 아니다.
- **숨은 대칭.** 학습 루프 첫 줄의 균등 순열 증강과 추론의 균형 8뷰 TTA는 목적이 같다. 슬롯 위치 편향 제거다. 학습은 **기댓값 수준**에서, 추론은 **대수적으로 정확히** 상쇄한다.
- 추론시에도 Prune한 데이터로 1차 forward 하고, Prune하지 않은 데이터로 2차 forwar를 진행하여 Prune된 데이터와 전체 데이터를 모두 보여준다.

표 2 최종 학습 레시피

항목	값
본체 / 헤드 학습률	5e-5 / 2.5e-4 (비율 **5×**, 원문 Figure 2 설정) · 헤드 weight decay 0.1
유효 배치 · 스텝	16 · 1,500스텝 — train 9,535건 전량 기준 약 2.5 epoch – loss, acc변화율을 고려하려 1200스텝으로 최종 선정
스케줄	cosine, warmup 20 (다항 감쇠는 총 LR 예산 부족으로 기각)
보조손실	기대-Kendall, λ = 0.5 — one-hot에서 0이라 CE 최적점과 무충돌
학습 시 프루닝	확률 0.75 적용, keep 50% — pruned / full 두 분포에 동시 노출
증강	균등 순열 증강 — 정답 순열의 사전분포를 평탄화

모든 값이 1차 본런의 실측 진단에서 유도됐다.

가중치를 바꾸지 않고 얻어낸 증분

규정은 **모델 앙상블을 전면 금지**하는 반면 **테스트타임 증강은 명시적으로 허용**한다. 추가 정확도는 단일 가중치에 입력만 바꿔 여러 번 질의하는 방식으로만 얻어야 한다. 아래 네 기법은 모두 프루닝 아이디어를 서빙 단계로 연장한 것이다.

(a) 균형 TTA8 — 위치 편향의 정확한 상쇄

- 슬롯 위치 자체에 편향이 생길 위험이 있는데, 흔한 대응인 무작위 뷰 평균은 편향을 **기댓값 수준에서만** 상쇄한다.
- 클라인 4원군 V_4 = {e, (01)(23), (02)(13), (03)(12)}는 4개 뷰에 걸쳐 각 입력이 각 슬롯을 **정확히 한 번씩** 방문하는 라틴 방진을 이룬다.
- 최종 시스템은 이를 이면군 **D_4**(S_4의 Sylow-2 부분군)로 확장한 8뷰를 쓴다. 각 슬롯을 정확히 두 번씩 방문해 같은 상쇄 성질을 유지하면서 표본을 두 배로 늘린다. 실측에서 **단일 최대 증분(+1.22%p)**이었다.

(b) 풀토큰 stage-2 — 버린 증거의 회수

- 프루닝 패스로 24차원 점수를 얻은 뒤, 같은 뷰의 **이미 준비된 텐서**에 토큰을 자르지 않은 forward를 한 번 더 수행해 함께 집계한다. 비전 인코딩을 다시 계산하지 않아 비용이 낮다.
- 과거에 기각한 "같은 정보를 다르게 물어보기"류 캐스케이드와 다르다. 질문이 아니라 **정보를 추가**하기 때문이다.
- 저마진 샘플만 승격시키는 캐스케이드 모드도 있으나, 24시간 예산에 여유가 있어 전 샘플 적용을 택했다. 실측으로 **58%의 샘플에서 마진이 증가**했다(중앙값 0.980 → 0.983).

(c) 토큰 부스트 — 프루닝 점수의 재사용

- 살아남은 토큰 중 상위 50%를 LLM 입력 시퀀스에 한 번 더 복제한다. 원본과 같은 m-RoPE 좌표를 주므로 위치 의미론이 훼손되지 않는다. Qwen3-VL의 좌표는 "순서"가 아니라 "화면 격자 위치"를 뜻하는 **의미 좌표**라 복제본이 같은 좌표를 갖는 것이 옳다.
- 잘려나간 토큰을 되살리는 것이 아니라 **이미 학습된 분포 안에서 상위 토큰의 비중만 재조정**하는 서빙 전용 기법이다. 비용은 샘플당 13.27초 → 14.59초(+10%).

(d) 모션 항 — 캡션이 나르지 않는 신호

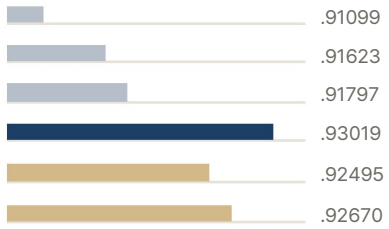
- 캡션 없이 이미지만으로도 정답률이 무작위의 5배로 나온 진단을 계기로, 캡션과 상보적인 물리 신호를 프루닝 점수에 더했다.
- 각 토큰이 나머지 3개 프레임의 같은 격자 위치와 얼마나 다른지를 **원시 잔차 노름**으로 재면 정지 배경은 0에 가깝고 움직이는 전경은 높다. 사전 검증에서 객체성과 스피어만 **+0.693**, 캡션 코사인과는 **+0.011**로 상보성 가설을 지지했다.
- 4장의 토큰 격자가 일치할 때만 정의된다(학습셋의 약 78%). 불일치 시에는 억지로 보간하지 않고 2항 블렌드로 폴백한다. **모션 가중치 0인 경우와 완전히 동일한 코드 경로**다.

그림 5 뷰를 늘리는 것이 아니라 균형을 맞추는 것

V_4 · 4뷰 — 각 색이 각 열에 정확히 1회

	S1	S2	S3	S4
e	1	2	3	4
(01)(23)	2	1	4	3
(02)(13)	3	4	1	2
(03)(12)	4	3	2	1

뷰 수 대비 점수



8뷰에서 정점을 찍고 24뷰(전체 S_4)에서 꺾인다. "원본과 먼 홀순열이 노이즈를 넣는다"는 가설은 짝순열만 모은 12뷰(A_4)로 격리 검증해 **기각**됐다. **원인은 확정하지 못했고 실측만 보고한다.**

표 3 최종 트랙의 제출 이력

일자	구성	Public 점수	Δ	해석
07-23	SFT 단독, 모션 항 미적용, 균형 TTA8 + 풀토큰 stage-2	0.92670	—	기준선
07-24	+ 추론 시 모션 항 0.3 (학습 설정과 정렬)	0.93019	+0.35%p	학습-서빙 설정 일치의 효과
07-24	**+ 토큰 부스트 0.5 ← 최종 제출물**	**0.93193**	**+0.17%p**	**재학습 없는 서빙 전용 증분**

세 행 모두 동일한 단일 어댑터·헤드·TTA8·stage-2를 쓰며, 차이는 프루닝과 서빙 설정뿐이다.

표 4 기법별 기여도

서로 다른 시점의 격리 측정치가 섞여 있어 단순 합산은 성립하지 않는다.

기법	Δ	비고	판정	근거
모델 규모 8B BF16 → 32B 4bit (통제 실험)	+3.73%p	데이터·레시피·프롬프트 고정, 모델만 교체	채택	규모 이득이 4bit 양자화 손실을 압도
TTA 도입 (1뷰 → 3뷰)	+3.81%p	격리 측정, 신뢰구간 +1.8 ~ +5.8	채택	슬롯 위치 편향 완화
균형 TTA4 → 균형 TTA8 (D_4)	+1.22%p	단일 변경 기준 최대 증분	채택	편향의 정확한 상쇄 + 표본 2배
프루닝 모션 항 (학습·추론 정렬)	+0.35%p	표 3 · 819건 중 11행만 변경	채택	캡션과 독립적인 전경 신호 보강
토큰 부스트 0.5	+0.17%p	서빙 전용, 재학습 없음 · 7행 변경	채택	상위 토큰의 어텐션 비중 재조정
균형 TTA8 → 전체 24뷰 (S_4)	−0.35%p	뷰를 더 늘려도 이득 없음	기각	균형이 개수보다 중요
DPO 재적용 (독립 2회)	열세	인접 스와프 hard-negative · 미제출	기각	기대-Kendall 항과 목적 중복

정답 라벨 없이 무엇을 확인했나

평가셋의 정답은 공개되지 않으므로 정확도를 직접 분해할 수 없다. 대신 Score24 헤드가 내놓는 **마진**(1위−2위 확률차)과 **설정 간 예측 이동**을 이용해 세 가지를 확인했다. D-1·D-3은 최종 제출 구성의 `progress.jsonl` 819행 전수, D-2는 표 3의 세 구성 간 비교다.

D-1 모델은 대부분 매우 확신한다

margin	0~.1	.1~.3	.3~.6	.6~.9	.9~1
	45	53	50	88	**583**

- 819건 중 **583건(71%)이 margin > 0.9**. 저마진(<0.3)은 **98건 = 12.0%**뿐이다.
- 세 설정의 저마진 집합은 자카드 **86%**로 겹친다. 어떤 설정을 쓰든 어려운 문제는 대체로 같은 문제들이고, 따라서 서빙 기법이 움직일 수 있는 행의 상한이 **98건으로 구조적으로 정해져 있다.**

D-2 기법이 확신한 예측을 흔들지 않았다

변경	뒤집힌 행	변경 전 마진	마진<0.3
모션 0 → 0.3	11	0.014	**11/11 = 100%**
부스트 0 → 0.5	7	0.005	**7/7 = 100%**
(참고) 안 뒤집힌 행	801~808	0.983	전체 중 12%

- 뒤집힌 18건이 전부 마진 < 0.3이고 중앙값은 0.005~0.014로 **결정 경계에 딱 붙은 샘플**들이다. 전체에서 저마진은 12%뿐인데 변경분은 100%가 거기 몰렸다.
- 잘 작동하는 정련 기법이 보여야 할 정확한 거동이며, **"서빙 트릭이 멀쩡한 예측을 흔든다"는 위험이 실측으로 배제**된다.

D-3 긴 캡션은 메모리는 비싸고 정확도는 쉽다

구간	캡션 평균	마진 중앙값	저마진 비율
0~577행	154자	0.989	**16.8%**
578~819행	459자	0.976	**0.4%**

- 저마진 98건 중 **97건이 짧은 캡션 구간**에 있다. 캡션이 길수록 정보가 많아 모델이 확신한다.
- 그런데, cache 누적으로 인해 자원을 가장 많이 사용해서 OOM이 발생한다. 이는 GPU를 비우고 모델을 재실행하면 이전의 cache가 사라져 해결된다.
- (test 데이터를 볼 수 없어 데이터 출처가 둘일 가능성이 있어 인과는 단정하지 않는다.)

기각 그럴듯했지만 데이터로 걸러낸 것들

- **DPO 재적용 —** Prune 미적용 학습에서 명확한 성공(+0.87%p)이었기에 같은 아이디어를 얹었으나, SFT 단독보다 나빴다.
- 진단 중 **DPO 600/800/1000 세 체크포인트의 예측이 완전히 동일**함을 발견했다. 마진 손실이 이미 맞힌 샘플의 마진만 넓힐 뿐 결정 경계를 넘지 못한 것이다. 기대-Kendall 항과 **목적이 이중으로 겹쳤다**는 것이 가설이다.
- **이벤트별 배정 프루닝 — 데이터 게이트에서 기각.** "4개 이벤트를 4개 프레임에 1:1 배정"하는 확장안을 검토했으나, 학습 캡션의 **94%가 자연 절 4개 미만**(1절 35%, 2절 45%)이라 "4이벤트"의 상당수가 기계적 분할이었다. 전제가 성립하지 않아 **구현 전에** 기각했다.
- LearnPruner 논문의 의 LLM 중간 2차 컷도 원 논문 수치가 89%+ 극한 구간이라 미탑재다.

표 5 최종 구성의 실측 자원 사용량

추론 RTX 3090 24 GB · 학습 RTX Pro 6000 1장

항목	실측	제한 및 여유	비고
추론 VRAM 피크	`23.63 GiB / 23.99 GiB`	24 GB 카드 가용량의 **98.5%**	-
추론 시간 (819건)	약 7.2시간	-	TTA8 × 2패스 포함, 부스트로 +10%
모델 총 용량	약 21 GB (NF4 base 20.4 GB + 어댑터 0.5 GB)	제한 80 GB의 26%	단일 가중치 세트
학습 연산	RTX Pro 6000 1장 × 약 13.5시간	학습 대상 134.3M = 32B의 0.42%	-
외부 API 비용	0원	학습·추론·전처리 전 구간 미사용	-

● 장점

- **규정과 정합적인 구조.** 단일 가중치 세트 하나만 쓰고, 여러 번의 forward는 입력만 바꾼 것이라 TTA 범주다. 제공된 train 9,535건 외 외부 데이터를 쓰지 않았고, 캡션 분해가 규칙 기반이라 모델 생성 텍스트가 학습에 들어가지 않는다. 외부 상용 API 0건, 베이스는 2026-05-31 이전 공개 오픈소스다.
- **정확한 ablation이 가능한 구현.** 표 3·4의 모든 증분이 "다른 것은 전혀 바뀌지 않았다"는 보장 위에서 측정됐고, 그림 3의 (b)도 그 덕에 실제로 재현해 렌더링했다.
- **재현성 장치.** 추론 코드와 어댑터 가중치를 검증 스크립트와 함께 공개 저장소로 배포하고 완전 오프라인 실행을 확인했다. 단위 테스트 112개가 순열 규약·프루닝 불변식·집계 로직을 고정.
- **안전한 정련.** D-2가 보이듯 서빙 기법의 개입이 전부 저마진 구간에 국한돼 확신 있는 예측을 훼손하지 않는다.

● 한계

- **메모리 여유가 2% 미만이다.** 과거 구성(3뷰 + 저마진 캐스케이드)의 피크 19.9 GiB에서 여유가 크게 줄었다. 뷰별 순차 처리 개선을 설계해 두었으나 시간 부족으로 반영하지는 않았다.
- **평가셋 후반부에서 OOM이 반복된다.** 원인은 단편화가 아니라 평가셋 구조였다. Test의 후반부를 기점으로 캡션 평균이 154자에서 459자로 계단식 증가한다(서로 다른 두 실행이 같은 지점에서 크래시한 것이 단서). 프로세스를 재시작하면 이전에 쌓인 Cache가 없어져 OOM이 나지 않고 정상 작동한다
- **모션 항이 데이터의 약 22%에서 작동하지 않는다.** 4장의 원본 해상도가 달라 격자가 어긋나면 "같은 위치"가 정의되지 않는다. 이때는 2항 블렌드로 폴백한다.