Frame Ordering by Evidence Source

SNU AI Challenge 2026 · Final Report — Team 에필로그 · JunHyeong Kwon (권준형)

1. What the Problem Actually Is

Four frames and one caption are given, and the original temporal order must be recovered. Scoring is Exact-Match: all four positions must be right, and getting three right scores zero.

It is a puzzle, but not a spatial jigsaw. A spatial jigsaw is solved at the **boundaries** — adjacent pieces continue each other's lines and colors, so matching locally gives the answer. A temporal jigsaw has no such cue. Four frames sampled from one video can be seconds apart, sometimes with a cut between them. There is no boundary to butt against.

Three separate things fix the order, and each comes from a different place.

One — the caption already states the order of events. The caption narrates the video's events in temporal order. Given "A man picks up a bucket and then carries it outside", the order is already in the text; what remains is deciding which frame belongs to which clause. This part is not order inference but **assignment**. The data confirms it: 203 groups of training samples share a byte-identical set of four frames, and across the 208 pairs they form only 8 share the same label — 96% take a different answer. The same pixels are ordered differently under different captions, so an approach that memorizes order from frames cannot work here.

Two — the caption goes silent within a scene. When two frames belong to the same event, the caption cannot separate them. In "A man bowls a strike", nothing in the sentence says whether the instant just after release precedes the instant just before impact. That is readable only from pixels — the ball's position, whether the pins still stand: the **direction of irreversible change**.

Three — sometimes the answer is to do nothing. For 15.5% of samples the given order is already correct (No_ordering). This is not a choice among 24 candidates but a question of whether to intervene at all, and effort spent on getting the order right actively damages this judgment (§4).

The three decisions rest on different evidence and fail in different ways. The design below follows from that.

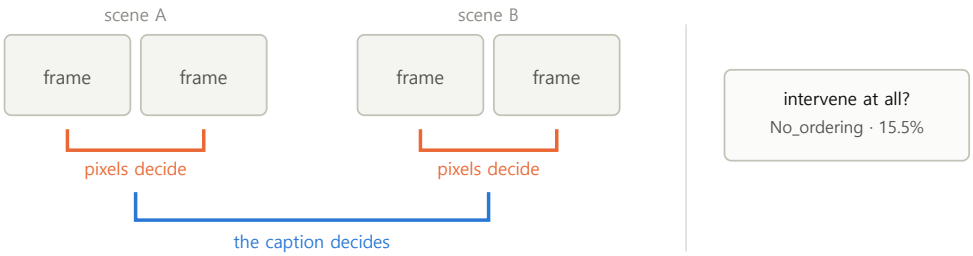

Figure 1. Within a single sample the three decisions are fixed by different evidence. Cross-scene order comes from the caption, within-scene direction from the pixels, and whether to intervene is a separate channel.

Relation to prior work. Order recovery is an old task. In self-supervised learning, predicting the order of shuffled frames has served as a pretext for representation learning [2,3], and more recently, partitioning and shuffling visual input and having the model emit the permutation in natural language has been proposed as MLLM post-training [4]. In the setting where a caption is also given, the closest work is Sort Story [1], which sorts jumbled image-caption pairs into a story; the standard decomposition there is an ensemble of unary (position) and pairwise (order) predictions.

Two things differ here. First, the caption is not attached per frame but is one sentence for all four, so **the frame-to-clause assignment is itself unknown**. Second, scoring is Exact-Match. These two differences dictate the design in the next section.

2. Approach: Score the Whole Answer Rather Than Split It

Having three evidence sources is not a reason to build three modules. Exact-Match demands **joint** accuracy over four positions, so splitting into per-position or per-pair subproblems collapses at the join even when each part is individually accurate. Measured: a pairwise verification rerank [6], the standard decomposition, fixed 3 rows and broke 12. And the frame-to-clause assignment requires seeing images and text together, so it cannot be split into a vision module and a language module either.

So the answer is not decomposed. There are only 24 permutations, which makes **exhaustive enumeration affordable**. Instead of having the model generate an answer, we compute the teacher-forced log-likelihood of each of the 24 candidate strings and take the argmax.

- **Format errors go to zero.** Generation can emit an invalid string; scoring cannot leave the candidate set.

- **A joint posterior is obtained.** The score vector spans all 24 candidates, so "what is second, and by how little" is preserved.
- **The margin (top-1 minus top-2) comes for free.** This connects directly to §1. Samples whose order the caption fixes have a large margin; samples that hinge on two frames within one scene have a margin near zero. The margin is therefore **a measurement of which evidence source is deciding this sample**, and both §4 and §5 build on it.

That MLE-trained likelihood does not by itself guarantee correct relative ranking among candidates is a recurring finding in summarization and generation, where it is corrected by training a ranking objective over a candidate set [7,8]. Here the candidate set is finite at 24 and the answer is unique, so the correction can be applied at inference time without training — exhaustive scoring, then only the identity adjustment of §4. Results from actually training a ranking loss appear in §8.

Model. Qwen3-VL-32B-Instruct, quantized to 4-bit NF4 and adapted with LoRA (r=32, α=64, all-linear plus the vision merger). A backbone with interleaved multi-image input and M-RoPE was required for the frame-to-clause assignment, and 4-bit quantization lands at a measured 21.5 GB peak, so both training and inference fit a single 24 GB card.

Scoring implementation. The prompt of four images plus caption is shared by all 24 candidates, so it is prefilled once and the KV cache is reused. Reusing the cache manually misplaces M-RoPE positions because the continuation call carries no `image_grid_thw`; prefill positions are therefore obtained from `get_rope_index` and continuation positions computed linearly from the end of the prompt. Sequential and batched scoring were verified to agree on the argmax.

3. Training Where the Caption Is Silent

The decomposition in §1 also dictates what to train. Cross-scene order is already fixed by the caption, so there is little to gain from training it. What needs training is **the span where the caption is silent** — the direction between two frames inside one scene.

A leakage-free split comes first. Training samples are not independent: many are drawn from the same source video and therefore share frames. A random split would put near-copies on both sides and inflate validation. Every image is MD5-hashed, samples sharing any frame hash are merged by union-find into video-level groups, and the 1000-sample validation set is held out by whole group. Hashes appearing in five or more samples — black frames and common transitions — are excluded from the union, because they otherwise chain unrelated videos into one giant component. Every number reported below rests on this split; §7 shows how much the choice of validation set matters here.

Error analysis supports this. The model matches scenes to caption clauses well, but its accuracy on the direction of visually similar frame pairs (0.73) is 14 pp below dissimilar pairs (0.87). The training data was built accordingly.

1. **Sequence answer format.** The answer is unified as "the shown frame indices listed in temporal order". Autoregressive decoding then follows the caption's narrative order, and within-scene comparisons localize to adjacent positions.
2. **Same-scene pair drills.** SigLIP embeddings group each sample's four frames into the 2+2 partition maximizing within-pair similarity, and before/after QA is generated **only for frames inside the same pair** — 14k items mixed into the main task. Cross-scene pairs are deliberately not drilled, since the caption already fixes them. This is the single largest gain: an ablation without the drills (v4) scores val 0.530 against 0.556 with them (v3).
3. **Presentation-order shuffle augmentation.** At every step the frames are presented in a random permutation with the label transformed to match. Multi-image VLMs are known to have a position bias, where the same information yields different predictions depending on where an image sits [5]; in this task that bias changes the answer directly. The augmentation induces an order-equivariant representation and removes it.

4. Protecting the Third Decision

One problem recurred throughout. **Almost every intervention that raises ordering accuracy damages the No_ordering judgment.** Three seemingly unrelated incidents turned out to share a cause.

- **Augmentation manufactures false identities.** Shuffling the presentation order yields "exactly as given" with probability 1/24. Real shuffled samples never carry the identity answer, so these fake identities contaminate a channel worth 15.5% of the data. Fix: redraw whenever the shuffle lands on identity.
- **TTA destroys the definition of identity.** Averaging over shuffled presentation orders erases the very reference that "exactly as given" is defined against. Averaging all candidates together drops identity accuracy from 0.87 to 0.48. Fix: score the identity candidate from the canonical pass only and average the other 23 — a **split channel**.
- **Ranking-loss retraining collapses identity.** Retraining to widen the margin between candidates (rank-cal) dropped identity from 0.83 to 0.70 when no_ordering rows were excluded from training. Diagnosing the cause and including identity rows as gold restored identity to 0.943 from 0.868 — but ordering then regressed across every cohort, 0.539 to 0.470 (§8).

These are not three separate bugs. No_ordering is an intervention decision, not an ordering decision, so objectives, augmentations, and inference tricks tuned for ordering erode it by default. The final system isolates the channel explicitly:

identity-safe augmentation, split-channel TTA, and an identity bonus of +0.5 grid-searched on val (no entropy gate turned out best).

5. Spending Compute Where the Evidence Is Weak

Training reduces position bias but leaves a residual [5]. Scoring under 8 presentation orders and averaging (TTA8) cancels that residual and raises accuracy, but applied to all 819 samples it costs roughly 34 hours in RTX 3090 terms, over the 24-hour budget. This is where the margin from §2 pays off. A large margin means the caption has fixed the order and the answer is already settled, so rescoring cannot change it. A small margin means the sample hinges on a pixel judgment — exactly where TTA does change answers.

So everything is scored in one pass, and **only the samples with margin < 2 (148, i.e. 18%)** are rescored under TTA8 and merged. The rescoring moves the argmax on 42 of them, and 41 rows change in the final submission after the identity bonus; the split-channel TTA itself is worth +3.7 pp measured on val.

configuration	3090-equivalent	24 h budget	accuracy
TTA8 on all	~34 h	exceeded	best
single pass on all	~4.4 h	met	−3.7 pp
margin routing (adopted)	**~10.6 h**	**met**	**TTA8-level**

The routing criterion is the decomposition of §1 rather than the budget — the margin identifies which evidence source is deciding each sample — and meeting the budget followed from it.

6. Results

stage	Public LB
4B QLoRA baseline	0.66492
8B + sequence format + shuffle augmentation	0.87085
32B + same-scene drills (v3) + 24-permutation scoring	~0.92
+ margin-routed TTA + identity calibration	**0.92495**
final submission (same pipeline)	**0.91972**

Calibration was grid-searched over 1000 val samples. Uncalibrated EM is 0.5670; an identity bonus of 0.5 is optimal at 0.5730, where the no_ordering cohort reaches EM 0.8590 (n=156) against 0.5201 for the rest (n=844). That gap is exactly the difficulty difference between the two decisions described in §1 and §4.

7. Error Analysis: Why Val and Test Disagree

Where the errors are. 59% of val errors come from "which two frames form the first half". 47.8% of hard failures concentrate on captions containing zero temporal connectives — that is, the remaining errors are the §1-Two type, where the caption does not fix the order and the decision must be made from pixels alone.

Refuting our own hypothesis. An early observation that "91% of errors involve the first or last frame" motivated endpoint-specific methods. Building a null model showed that under a uniform-wrong-answer assumption the expected figure is already 95.7%, and the measured 92.6% falls below it. It was a combinatorial artifact rather than a signal, and the whole family was dropped.

The metric misbehaves. A group-split val drawn from train moves differently from the actual test set. Cutting both by the same criterion — how many frames are exact duplicates of train frames — gives the following.

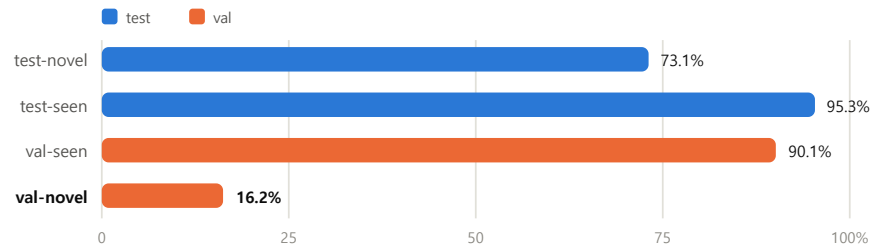

Figure 2. Share of samples with confidence margin > 2, by cohort. Under the same definition of "novel", only val-novel drops to 16.2%.

cohort (frames exactly duplicated in train)	n	margin > 2	EM
test-novel (0 frames)	431 (52.6%)	**73.1%**	(labels withheld)

test-seen (4 frames)	358 (43.7%)	95.3%	(labels withheld)
val-seen (4 frames)	181	90.1%	0.939
val-novel (0 frames)	**610**	**16.2%**	**0.369**

The same "novel" cohort is a high-confidence population in test and a coin flip in val. The decomposition explains why. 20.6% of test captions (169 of them) run past 69 words and are synthetic narratives with explicit temporal scaffolding — "begins by … then … finally". These are the type where §1-One fixes the order completely, and in that cohort the margin exceeds 2 for 98% of samples. Val, derived from train, structurally contains none of them. So val-novel is the hardest slice, where only §1-Two remains, and test-novel is not.

The practical corollary: a sub-1 pp improvement on val does not guarantee transfer to public. A recalibration that went 6–0 on val reversed to −4 samples on public. The public score itself covers 573 samples (70%), so it cannot resolve differences below ±0.7 pp. All subsequent decisions were made on "no identity regression + EM on the test-like cohort + no regression in the high-confidence band" instead.

8. Rejected Hypotheses

Failures the decomposition predicted. The two below were expected to fail before they were run, and were run as tests of the thesis.

hypothesis	result	what the decomposition says
lookup over a train precedence graph	EM 0.156	cross-scene order is caption-conditional — identical frames take different answers (§1-One)
reverse scoring P(caption | order)	EM 0.095	the caption is not recoverable from the order; information flows one way only

Failures it did not predict. These were filtered by measurement against criteria registered in advance.

hypothesis	verdict
GRPO with a similarity-weighted Kendall reward [4]	generation-policy gains do not transfer to discriminative scoring; identity damaged 0.83 → 0.75
ranking-loss retraining (SLiC-style [7], rank-cal v2)	the cause of identity collapse was diagnosed and fixed (0.868 → 0.943), but ordering regressed across all cohorts (§4)
arrow-of-time discriminator [9]	confirmed that a VLM reads temporal direction at 0.77 from an isolated frame pair, but combining six pairs does not beat the main scorer
partition-membership drills (v5)	mechanism confirmed at +2.8 pp partition accuracy, but with identity damage; the undamaged fix requires full retraining
native-video reordering verifier	net −24 on a fixed 96-sample set; packing still frames carries no motion signal
candidate-conditioned interpretation + cross-likelihood	interpretations recover their own candidate only 47% < 90% of the time; insufficient visual grounding

The negative results agree with each other. The remaining errors sit in §1-Two — the span where the caption is silent and only a pixel judgment is left — and are not reachable by inference-time methods on the same input with the same model. This agrees with prior work: recent work shows that large multimodal models read the irreversibility of time poorly and proposes **training** with a reverse reward as the remedy [9], and framing jigsaw order recovery as a post-training task [4] sits on the training side as well. Our measurements reconfirm that diagnosis from the inference side — though the training interventions we did try (GRPO, ranking loss) failed on this task because of the identity-channel erosion in §4.

9. Strengths and Limitations

Cost and resource profile.

peak VRAM, 4-bit inference	21.5 GB — fits one 24 GB card, no offloading
latency	3.9 s/sample single pass (H100); 819 samples in ~0.9 h
full inference, as submitted	~10.6 h in RTX 3090 terms, inside the 24 h budget
training	~10 GPU-hours cumulative, including the rejected experiments of §8
shipped artifacts	one LoRA adapter, 1.08 GB; no ensemble, no second model
external API spend	none — no commercial API used for preprocessing or augmentation

Strengths. Exhaustive 24-permutation scoring removes format errors and yields a confidence margin as a by-product, and that same quantity is what selects the training target, allocates inference compute, and protects the identity channel. The

submission CSV reproduces byte-for-byte from the published script chain.

Limitations. First, the §1-Two span was never solved. The decomposition localizes the problem and quantifies it through the val-novel cohort, but none of the interventions in §8 raised accuracy there. Second, the §7 conclusion that val does not represent test only changed how experiments were judged; a representative validation set was never built. Third, the 3090 inference times are bandwidth-scaled from H100 measurements. Fourth, a head-to-head backbone comparison was not run within the time budget.

Next. The priority is training aimed specifically at §1-Two. Explicit partition training carries a break-even at 0.83 probe accuracy (currently 0.74) and is dropped below it. Building a synthetic-scaffolding validation set that mimics the test distribution is the most direct way to fix the instrument problem of §7.

Reproduction. The submission CSV reproduces byte-for-byte through the chain below; details are in the repository README.

```
python scripts/prepare_data.py && python scripts/make_v3_data.py           # group split -> seq format + same-scene
drills
python scripts/train_sft.py --train-merger --out runs/32b-lora-v3          # 32B QLoRA SFT (r=32, 2 epochs)
python scripts/eval_infer.py --adapter runs/32b-lora-v3 --test \
    --out artifacts/test_scores_32b.json                                   # 24-permutation scoring (single pass)
python scripts/route_lowmargin.py --scores artifacts/test_scores_32b.json \
    --margin 2.0 --out artifacts/tta_route_ids.json                        # route margin < 2
python scripts/eval_infer.py --adapter runs/32b-lora-v3 --test --tta 8 --tta-split \
    --ids artifacts/tta_route_ids.json --out artifacts/test_scores_32b_tta.json
python scripts/merge_tta.py --base artifacts/test_scores_32b.json \
    --tta artifacts/test_scores_32b_tta.json --out artifacts/test_scores_32b_merged.json
python scripts/make_final_clean.py --scores artifacts/test_scores_32b_merged.json \
    --bonus 0.5 --out submissions/final_clean.csv
```

References

[1] Agrawal et al., *Sort Story: Sorting Jumbled Images and Captions into Stories*, EMNLP 2016. [2] Misra et al., *Shuffle and Learn: Unsupervised Learning using Temporal Order Verification*, ECCV 2016. [3] Lee et al., *Unsupervised Representation Learning by Sorting Sequences*, ICCV 2017. [4] Wu et al., *Visual Jigsaw Post-Training Improves MLLMs*, arXiv:2509.25190. [5] Tian et al., *Identifying and Mitigating Position Bias of Multi-image Vision-Language Models*, arXiv:2503.13792. [6] Qin et al., *Large Language Models are Effective Text Rankers with Pairwise Ranking Prompting*, arXiv:2306.17563. [7] Zhao et al., *Calibrating Sequence Likelihood Improves Conditional Language Generation*, ICLR 2023. [8] Liu et al., *BRIO: Bringing Order to Abstractive Summarization*, ACL 2022. [9] Xue et al., *Seeing the Arrow of Time in Large Multimodal Models*, arXiv:2506.03340.