Recovering Chronological Order from Shuffled Video Frames

Permutation-Scored Vision–Language Modeling with Label-Grounded Retrieval

Kim JaeJun Ryu Uijae

Team: Hobby
Technical Report

1 Task and the metric it imposes

We are given a one-sentence caption and four frames from a short video in random order, and must output the ordering that returns them to chronological sequence. Scoring is exact match (EM) over the whole four-tuple: all four positions must be right, so three-of-four scores the same as none. With $4! = 24$ orderings chance is 4.2%, and in the training data the most common single answer is the identity ordering $[1, 2, 3, 4]$ — 1,478 of 9,535 samples (15.5%), exactly the captions marked as having no temporal ordering — so a method must clear EM = 0.155 just to beat "always guess sorted." This report is organized around what the *training* data demanded and how each demand was met; the first demand comes from the metric itself: whatever the model does, it must emit a valid permutation and be judged on whole-tuple correctness, which shapes the decoder (Section 4).

2 What the training data demanded

We built the method around four properties of the training set, each measured on the provided labels; each dictates a later choice.

(i) The order lives in the caption, not just the pixels. Retraining with every caption replaced by a generic placeholder dropped calibration EM from 302/500 to 141/500 (−32.2 points). *Demand:* the model must reason over the four images *and* the caption's compositional meaning jointly — not over frames alone.

(ii) The label is narrative order, not the video's timeline. Two training samples can hold the same four frames yet order them differently, because each caption tells a different story over the same shots. A frame's absolute position in one sample therefore does not transfer to another; only within-shot adjacency does — adjacent-in-time frames keep their local order across samples at 98.6% fidelity, versus 80.8% for frames from different scenes. *Demand:* any lookup must use adjacency, never absolute position.

(iii) Frames recur heavily across samples. A union–find over SHA-256 digests of the (non-flat) frame images collapses the 9,535 samples into 4,980 groups, one a single connected component of 4,293 samples (∼45%), the rest mostly singletons (4,742). *Two demands:* a random train/val split would place near-duplicate samples on both sides and overstate accuracy, so we evaluate on a grouped split (Section 7); and the labeled frames are themselves a reusable signal — a sample whose frames appear in the labeled data can have its order partly recovered by lookup (Section 6).

(iv) One sample in six is already sorted. The identity ordering is 15.5% of the data and coincides with the no-ordering captions. It is both the baseline to beat (0.155) and, as we found, the model's dominant failure mode: when unsure it over-predicts "already sorted."

3 Choosing the model: which family, and why Qwen

The task is to assign a chronological order to four images given a caption, and the choice of model family turned on that.

Why not an encoder. Contrastive/embedding vision models (CLIP-style, SigLIP, DINOv2) give per-image features but cannot assign a probability to an ordering, and they use the caption only as a global embedding, not as compositional instruction. We tested this: frozen DINOv2, SigLIP2, and DeBERTa-text features into a small MLP fell *below* the 0.155 identity floor on the honest split (Section 8). Features alone do not carry ordering.

Why a generative VLM. What demand (i) needs is an autoregressive vision–language model that ingests four images and text jointly and assigns a teacher-forced likelihood to a candidate output — which is exactly what lets us score orderings (Section 4).

Why Qwen3-VL. Among generative VLMs we chose Qwen3-VL because its properties map onto the constraints: open weights released before the competition cutoff; native multi-image input in a single prompt; a size ladder (4B, 8B, 27B) that let us measure the capacity axis under one recipe; and 4-bit quantization that fits even the 27B on a single 24 GB RTX 3090 for both training and inference, within the compute budget. The 27B variant has a hybrid attention stack — of 64 layers, 16 full-attention and 48 gated-delta (causal by construction) — a fact that later rules out inference-time attention edits.

4 Decoding: scoring permutations, not generating them

The metric's demand (Section 1) is met in the decoder. The obvious route — generate the ordering as text and parse it — has a failure mode EM punishes: free generation can emit an invalid answer (a repeated index, a missing frame, malformed brackets), and any repair is a guess. Instead we *score* orderings: for each sample we enumerate all 24 permutations and compute the model's teacher-forced log-probability of emitting exactly that ordering given the caption and four frames; the prediction is the arg max. The output is a valid permutation by construction — nothing to parse, no invalid-output case — and the problem reduces to 24 cheap scoring passes sharing one visual and textual prefix. The model never has to *write* an answer, only to *prefer* the right one, which is what the metric rewards. It also constrains what can be layered on top: any technique that reshapes the target sequence (e.g. a chain-of-thought before the ordering) breaks the 24-way comparison, because the orderings stop being scored on equal footing (Section 8).

5 Fine-tuning Qwen

Full fine-tuning of a 27B model does not fit 24 GB, so we adapt it with QLoRA: base weights 4-bit NF4 and frozen, a LoRA adapter (rank 16, $\alpha = 32$, dropout 0.05) on the attention and MLP projections, vision tower frozen — 79.7M trainable parameters (0.29%). The target is the chronological index list as a short JSON array, with cross-entropy on the ordering tokens only, and we mix shuffled and reversed presentations into training so the model cannot read the answer off input position. To honor demand (ii) the target is the caption's narrative order. The submitted adapter is trained on all 9,535 samples; the recipe numbers we report come from a sibling that excludes the calibration slice. Fine-tuning variants we tried and dropped are collected in Section 8.

6 Retrieval from the labeled frames

Demands (ii) and (iii) motivate a second stage. Because frames recur across the labeled data and within-shot adjacency is highly consistent, we can recover part of the order by lookup. For an input sample we match each frame against the labeled training frames by 64-bit difference hash (Hamming distance $\leq \tau = 4$, fuzzy so re-compression does not break the link), read the adjacency constraints its matched frames carry in the training labels, and skip samples whose four frames are functionally duplicated. We then filter the model's 24 candidates to those consistent with the constraints (allowing either orientation of an ambiguous chain) and keep the highest-scoring survivor; if none survives we fall back to the model's arg max. This uses training labels only, so it is legal under the rules; it is deterministic and not an ensemble — it overrides the model only where a training-grounded constraint disagrees, and never invents an ordering the model did not rank.

7 What needed hyperparameter tuning

A few choices genuinely needed tuning, and we made all of them on a 500-sample calibration slice carved from training — never on the validation set (spent once per track as a final verdict) and never against the leaderboard.

- **Training length.** Rather than assume a schedule we measured the learning curve on calibration: EM rose $277 \to 295 \to 302$ over three epochs (cosine, annealed to zero); a six-epoch run reached $274 \to 300 \to 302$ and tied the three-epoch model at its endpoint (McNemar $+36/-36$, $p = 1.0$). Three epochs is the convergence point, so we stopped there.
- **Adapter capacity.** We settled on LoRA rank 16 / $\alpha = 32$; larger rank saturated without measurable gain.
- **Retrieval threshold τ.** Tuned by leave-one-out on calibration (excluding each sample's own identity to imitate the held-out setting); the gain is stable across τ, and $\tau = 4$ is the operating point.
- **Retrieval policy.** The adjacency-only rule, the functional-duplicate skip, and the forward-or-reverse survivor filter were all fixed on calibration.

The discipline mattered: several post-hoc "improvements" looked positive on calibration and vanished on validation (Section 8), which is why validation was reserved as a single verdict.

8 Results and analysis

Three readings. *Leakage:* the same feature-MLP swings from 0.26 to 0.07 between splits, so we trust only the grouped column for the model's own ability, and it confirms demand (iii). *Capacity:* on the honest split the ladder is monotone — feature-MLP < 4B < 8B — and 27B extends it, though modestly (+1.9 points on mixed val), and at 4-bit 27B we are at the practical limit for 24 GB. *Retrieval:* validated once on held-out (on the 8B line, $0.570 \to 0.620$, $+52/-2$, McNemar $p = 1.65 \times 10^{-13}$). A within-train split necessarily under-counts the frame recurrence the retrieval feeds on, so this +5-point figure is a conservative lower bound on its deployed value; the submitted system's leaderboard result is 0.91972.

8.1 What did not work

Most of the project was ruling levers out, and several were rejected independently in both of our experiment lines — which is itself evidence.

System	eval set	EM
identity / majority baseline	any	0.155
feature-MLP (DINOv2 / SigLIP2)	random split	0.26–0.27
feature-MLP (same)	grouped holdout	0.07–0.08
Qwen3-VL 4B + LoRA	grouped holdout	0.30
Qwen3-VL 8B + LoRA	grouped holdout	0.327
Qwen3-VL 8B + LoRA	mixed val	0.540
Qwen3-VL 8B + LoRA + retrieval	mixed val	0.620
Qwen3-VL 27B + LoRA	mixed val	0.589
Qwen3-VL 27B + LoRA + retrieval	**mixed val / public LB**	**0.636 / 0.91972**

Table 1: Exact match (chance 0.042). "Grouped holdout" splits by shared-frame group; "mixed val" does not. The submitted system is the last row.

- **Feature fusion under an honest split.** DINOv2, DeBERTa-text, and SigLIP2 features into an MLP all fell below the identity floor on the grouped holdout; the text stream added nothing over vision alone (this is why the model is a generative VLM, Section 3).
- **Re-interrogating the same model adds nothing.** A clause-conditioned pairwise reranker gave $\Delta\text{EM} = +0.000$ (McNemar $p = 1$, bootstrap CI $[-0.013, +0.013]$); reweighting the 24 scores with hand priors and re-asking pairwise both moved validation only within noise. The 24 log-probabilities already encode the model's full judgment.
- **RL did not beat supervised fine-tuning.** Three GRPO arms (vanilla, DAPO, DAPO with a trainable vision tower) all failed a pre-registered gate (ordered-subset EM $\geq$ base $+0.04$): best arm 0.338 vs base 0.336. A pair-auxiliary scout beat SFT by only $+0.005$. Margin and listwise objectives were null-to-negative.
- **Chain-of-thought supervision collapsed the scoring.** Writing a reasoning trace before the ordering dropped epoch-1 calibration to 182/500 (vs 277): with a long trace the teacher-forced probability of the final ordering became degenerate (top-1 log-probability ≈ 0 for all 500 samples, a 15-nat gap to the next candidate), so the 24-way comparison collapsed into reading back one greedy trajectory. Generative reasoning and permutation scoring are incompatible.
- **Vision unfreezing hurt** (≈ -3.6 points, sample-matched), and **inference-time attention surgery is dead for this backbone**: a bidirectional frame-attention patch tied the baseline (296 vs 300, $p = 0.59$), because 48 of 64 layers are gated-delta and structurally causal, so no mask edit reproduces the dense-attention effect it assumes.

8.2 Error analysis

The model's residual errors are dominated by over-predicting the identity ordering (demand (iv)) — it guesses "already in order" when unsure. Four independent attempts to move this (re-scoring priors, salience tokens, presentation-order curricula, margin training) were all flat or negative, which points to the error being information-limited by the caption rather than a perception or calibration bug. The remaining hard cases are short clips with near-identical or symmetric frames (fades to black, duplicated slides, stand–squat–stand) whose order is genuinely underdetermined — an exact-match ceiling below 1.0 that no module removes.

9 Strengths, limitations, and cost

Strengths. Permutation scoring makes every output a valid answer with no parsing and aligns the objective with the metric. The grouped split gave an honest yardstick that stopped us chasing leakage-inflated baselines. The retrieval stage is a deterministic, label-grounded correction with a strong test behind it and is not an ensemble. The whole system trains and runs offline on one 24 GB GPU; producing all required predictions runs in 6.7 hours against the 24-hour limit at 22.8 GB peak.

Limitations. The model's honest novel-source generalization is modest ($\sim$0.33 at 8B), and the retrieval stage's benefit depends on the frame recurrence measured in demand (iii) — a property of the provided data — which a within-train split understates, so we lean on the held-out lower bound. Capacity, the one lever that still moves the model, is effectively exhausted at 4-bit 27B on 24 GB; rank and epoch increases saturated. The honest reading is that the model is near its ceiling for this data, and the result rests on a validation design careful enough to tell real gains from leakage and a retrieval signal used within the rules.