Technical Report for Text-Guided Image Sequence Reconstruction

Yubeen Jonggang Celebration Gaegang Party
SNU AI Challenge 2026
Yubeen Bae, Sowon Kim, Soyoun Kim
soyoun9176@gmail.com

Abstract

We reconstruct the temporal order of four shuffled images conditioned on a natural-language description. Because the output space contains only $4! = 24$ permutations, we adapt a single Qwen3.5-27B vision-language model to emit a compact four-digit order and use structured test-time computation. The final method evaluates four cyclic views, restores all predictions to one coordinate system, and applies mode voting, exact Kemeny consensus, and selective teacher-forced negative log-likelihood (NLL). We also evaluate Attractor, a separately submitted iterative method that repeatedly reorders the current arrangement until a fixed point or cycle closure. Both methods seek more robust predictions than single-view Greedy without exhaustively evaluating all 24 input arrangements. Using the same fine-tuned checkpoint, the public leaderboard scores are 0.94764 for Greedy, 0.94938 for Attractor, and 0.95113 for Cyclic4; all three private scores are 0.93089, and re-running the submitted pipeline on different hardware changes at most 2 of 819 answers.

1 Introduction

Given four shuffled images and a sentence, the task is to recover the unique chronological image order. Solving it requires more than matching objects across modalities: the model must infer state changes, physical causality, human intent, and a globally consistent event sequence.

The task admits several useful formulations. Viewed autoregressively, it resembles in-context next-item prediction, $[\text{text}] \to [I_1] \to \cdots \to [I_4]$; however, the next output is one of the input images rather than a token from a fixed vocabulary, so such a decoder would need a dynamic image vocabulary built from the four input image embeddings. Because $|S_4| = 24$[1], the same task can instead be written as 24-way classification or as retrieval, where the sentence is a query and each candidate image sequence is a key. Our implementation keeps the pretrained language decoder and represents this structured output with four image-index tokens.

Visual storytelling studies sequence-level coherence across image collections [5], and pretrained vision-language systems reuse strong multimodal representations with limited task-specific adaptation [8]. We adapt an open-weight Qwen3.5-27B vision-language model (VLM) using LoRA/QLoRA rather than training visual and language components from scratch [1, 4]. The vision encoder remains frozen and only low-rank updates on language linear layers are trained.

Our main contribution is a single-checkpoint inference procedure that reduces sensitivity to arbitrary input indices. The model predicts four cyclic views; the predictions are mapped to a common coordinate system and combined by mode, exact Kemeny consensus, and selective NLL tie-breaking. We call this procedure *Cyclic4*, distinguish the one-pass *Greedy* baseline, and separately evaluate

[1] S_4 is the set of all permutations of four elements.

Attractor, which repeatedly reorders the current arrangement. Both are designed to capture most of the benefit of aggregating over all 24 input arrangements at a small fraction of its inference cost. The experiments retain only ablations that support the final choices: LoRA rank, training checkpoint, and test-time computation. On the leaderboard, with the fine-tuned checkpoint fixed across methods, Cyclic4 improves the recorded public score from 0.94764 to 0.95113 relative to Greedy.

2 Problem setup

Each example contains a sentence s and shuffled images I_{1234}. Let $p = (p_1, p_2, p_3, p_4) \in S_4$ denote chronological order, where p_t is the original image index at time t. The required answer is the inverse permutation $a = p^{-1}$, so $a_{p_t} = t$.

We use exact-match accuracy, $\mathrm{EM} = \frac{1}{N}\sum_{n=1}^{N} \mathbf{1}[\hat{a}_n = a_n]$; an example is correct only when all four positions match.

Two fine-tuned artifacts are used throughout. The *full-training checkpoint* is the original LoRA-fine-tuned Qwen3.5-27B model behind all leaderboard submissions, scored on the hidden competition test set. The *holdout-trained adapter* is a reproduced LoRA adapter trained after excluding the `locked954` split; it supports the controlled inference ablation on the 937 examples remaining after the applied data filters.

We use *test-time computation* (TTC) for additional inference-time model evaluations. *Test-time augmentation* (TTA) refers specifically to fixed, label-preserving input views and restored-prediction aggregation; Cyclic4 is therefore TTA within TTC. Attractor is adaptive, iterative TTC, whereas Greedy uses neither. Exhaustive TTA would present all 24 input arrangements and aggregate the restored predictions, at 24 decodes per example; both TTC strategies here approximate that aggregate at a fraction of its cost.

3 Method

3.1 Direct four-token adaptation

The final training manifest contains 9,386 examples. We load Qwen3.5-27B with NF4 double quantization and BF16 computation, freeze the vision encoder and base parameters, and train LoRA modules on the language-model projections `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. The adopted configuration uses rank 32, $\alpha = 64$, and dropout 0.05. The prompt presents the sentence and four images and requests only the chronological four-digit permutation. Uniform causal cross-entropy is applied to exactly these four answer tokens; no auxiliary loss, sample weight, or position weight is used. Images use `max_pixels` $= 409600$.

Training runs for two epochs (4,694 optimizer steps) with batch size 4, fused AdamW, learning rate 10^{-4}, cosine decay, warmup ratio 0.03, weight decay 0.01, and seed 20260717. The first epoch uses a deterministic ID-hash-based input permutation; the second samples from the remaining 23 permutations. Rank 32 and the two-epoch budget are supported by the large-backbone selection experiments in Section 4; no 27B-specific rank sweep is claimed.

3.2 Cyclic4

Rather than decode all 24 input arrangements, Cyclic4 restricts the views to the cyclic subgroup of order four, and reaches for the more expensive consensus stages only on the examples where those four views disagree. The final inference procedure uses deterministic greedy decoding and the four cyclic views

$$\mathcal{V} = \{1234, 2341, 3412, 4123\}. \tag{1}$$

For view $v = (v_1, \ldots, v_4)$, a local prediction $r^{(v)}$ is restored to original indices as $\tilde{p}^{(v)} = (v_{r_1^{(v)}}, \ldots, v_{r_4^{(v)}})$. A unique mode is accepted immediately. If the mode is tied, all 24 permutations are evaluated by total Kendall distance [7]; the minimum-cost set is the exact Kemeny consensus [6]. Only if several Kemeny optima remain do we select the candidate with minimum four-view, full-vocabulary teacher-forced NLL. Because all candidates contain four tokens, sum and mean NLL induce the same ordering. Throughout this subsection, $q \in S_4$ denotes a candidate chronological

sequence of original image indices; q^{-1} converts that sequence to the competition's image-wise rank vector.

Algorithm 1 Cyclic4 for one example

Require: sentence s, images I_{1234}, model M

$\mathcal{P} \leftarrow \emptyset$
for all $v \in [1234, 2341, 3412, 4123]$ **do**
  $r^{(v)} \leftarrow \textsc{Greedy4}(M, s, I_v)$
  $\tilde{p}^{(v)} \leftarrow (v_{r_1^{(v)}}, \ldots, v_{r_4^{(v)}})$
  add $\tilde{p}^{(v)}$ to $\mathcal{P}$
$U \leftarrow \arg\max_{q \in \mathrm{unique}(\mathcal{P})} \mathrm{count}_{\mathcal{P}}(q)$
if $|U| = 1$ **then**
  $q \leftarrow$ sole element of U
else
  $K \leftarrow \arg\min_{q \in S_4} \sum_{p \in \mathcal{P}} d_K(q, p)$
  if $|K| = 1$ **then**
    $q \leftarrow$ sole element of K
  else
    $q \leftarrow \arg\min_{c \in K} \sum_{v \in \mathcal{V}} \mathrm{NLL}(M, s, I_v, v^{-1}(c))$
return q^{-1}

3.3 Attractor

Attractor maintains a current chronological arrangement $c_t \in S_4$ along a single deterministic trajectory initialized at $c_0 = 1234$. At iteration t, the images are shown in order c_t and the model returns a local permutation r_t. The next arrangement is

$$c_{t+1} = (c_{t,r_{t,1}}, c_{t,r_{t,2}}, c_{t,r_{t,3}}, c_{t,r_{t,4}}). \tag{2}$$

If $r_t = 1234$, the current arrangement is treated as a fixed point and returned. Because decoding is deterministic, revisiting an arrangement closes a cycle; the implementation returns the cycle-closing endpoint and caps the trajectory at 24 iterations. The terminal chronological arrangement is inverted to produce the competition's image-wise rank vector. Because the trajectory halts as soon as it reaches a fixed point or closes a cycle, the number of decodes adapts per example instead of being fixed at 24.

4 Experiments

Training choices are selected on the fixed `locked954` split. TTC is compared using the holdout-trained adapter on the filtered 937-example holdout. Leaderboard evaluation uses the full-training checkpoint and hidden competition data.

4.1 LoRA rank and checkpoint selection

Table 1 contains the closest controlled large-backbone pilots used to set the final training recipe. At the matched epoch-1 checkpoint, rank 32 exceeds rank 64 by 13 examples. Within the rank-32 run, epoch 2 improves by 29 examples over epoch 1, while a low-learning-rate epoch-3 continuation does not improve further. We therefore adopted rank 32 and two epochs for the final 27B run. These pilots support the choice but do not establish a universal or 27B-specific optimum.

4.2 Test-time computation and leaderboard evaluation

The controlled comparison merges the holdout-trained adapter into the BF16 base model on an A100 SXM 80 GB GPU, with FlashAttention 2, batch size 32, `max_pixels` $= 409600$, and deterministic greedy decoding; for Attractor only the adapter path was substituted, preserving the original submitted implementation. All three leaderboard submissions use the full-training checkpoint and differ only in their inference procedures, so their score differences are inference-method deltas. Table 2 reports both evaluations.

Table 1: Training choices on `locked954`. Comparisons are made only within each pilot family.

Choice	Pilot family	Candidate	Correct/954
LoRA rank	32B uniform CE, epoch 1	r32 / α64	**529**
		r64 / α128	516
Checkpoint	32B uniform CE, r32	epoch 1	529
		epoch 2	**558**
		epoch 3*	556

*Low-learning-rate continuation.

Table 2: Inference-procedure comparison. Holdout EM uses the holdout-trained adapter on 937 examples (correct counts in parentheses; runtimes are operational references); public and private scores use the full-training checkpoint on the hidden test set.

Inference	Holdout EM (937)	A100 min	Public	Private
Greedy	0.6158 (577)	10.4	0.94764	0.93089
Attractor	**0.6478 (607)**	14.9	0.94938	0.93089
Cyclic4	0.6457 (605)	44.9	**0.95113**	0.93089

Attractor produces no malformed outputs: 817 trajectories reach a fixed point and 120 a cycle-closing endpoint, none reaches the 24-iteration fallback, most finish in two iterations (644), and the maximum is six. Both strategies stay far below the cost of exhaustive TTA, at 4.3 times the Greedy runtime for Cyclic4 and 1.4 times for Attractor against a roughly 24-fold decoding cost.

On the controlled holdout, Attractor exceeds Cyclic4 by two correct examples. The methods differ on 128 examples: Attractor alone is correct on 33, Cyclic4 alone on 31, and both are incorrect with different answers on 64. On the public leaderboard, Attractor improves over Greedy by 0.00174, while Cyclic4 improves over Greedy by 0.00349 and exceeds Attractor by 0.00175.

The two evaluations therefore rank Attractor and Cyclic4 differently: a selection rule based only on controlled holdout EM would favor Attractor, whose public score is 0.00175 below that of Cyclic4. The rankings arise under different adapters and data, and the identical private scores provide no observed private-set separation among the three procedures.

4.3 Cross-hardware determinism

We re-executed the submitted Cyclic4 pipeline that produced the 0.95113 public score on the 819-example competition test set. The adapter, the four views, greedy decoding, `max_pixels`, and SDPA attention were fixed. Against that original submission R, let d_{ans} denote the number of different rank vectors out of 819. We executed four logically identical runs on a replacement A100 host (Run $A_{1:4}$, the same hardware specifications as submission run, but different server instance), and on RTX 3090 server with weight placement (Run B, RTX 3090 sm_86 with CPU offloading). Run $A_{1:4}$ shared the same `submission.csv` choices, although the result was different from the original submission. Table 3 compares the original choices R, replacement A100 experiment $A_{1:4}$, and RTX 3090 experiment. Both alter only 2 of 819 answers, so $|\mathrm{EM}(X) - \mathrm{EM}(R)| \leq 0.244$ points.($X \in \{A_{1:4}, B\}$)

Table 3: Cross-hardware determinism of the submitted Cyclic4 pipeline, measured against the original A100 submission R on 819 test examples.

Runs	Difference	d_{ans} (819)
among $A_{1:4}$	none	0
R vs. $A_{1:4}$	host	2 (0.24%)
R vs. B	host, GPU architecture, offloading	2 (0.24%)

We observed that executing the pipeline across different server instances yields a slight variance of 1–2 rows, despite having identical hardware specifications. However, repeated executions on the

exact same server are highly deterministic. This suggests the discrepancy is an inherent characteristic of the hardware environments.

5 Conclusion and discussion

Using the full-training checkpoint, Cyclic4 improves the recorded public score from 0.94764 for Greedy to 0.95113, at 4.3 times the Greedy inference cost and well below the 24-fold cost of exhaustive aggregation. The training ablations support rank 32 over rank 64 in the matched large-backbone pilot and select the epoch-2 checkpoint. On the controlled holdout, which uses the holdout-trained adapter, Attractor instead obtains the highest EM; the two evaluations use different fine-tuned artifacts and data, so their rankings need not agree. Re-running the submitted pipeline on a different host and GPU architecture changes at most 2 of 819 answers.

Attractor and rule-compliant verification. No separate binary verifier was trained for Attractor. Such a verifier would require correct-order and shuffled-order fine-tuning subsets and combination of their inference outputs, conflicting with the competition rule against multiple fine-tunings of one model on dataset partitions with subsequent aggregation. Each reported evaluation therefore uses one fine-tuned model and combines only its own inference outputs.

Thinking models and reinforcement learning. We also explored *thinking models*, which emit an intermediate reasoning trace before the four-digit answer, and reinforcement learning (RL) as a way to optimize them. Despite instances of degeneration—reasoning traces endlessly looping without reaching a conclusion—a preliminary EM evaluation on 200 examples demonstrated promising baselines: 53% for Qwen3.5-9B, 57% for Qwen3.5-27B AWQ, and 45% for Gemma-4-26B FP16. These are strong enough to apply RL techniques such as GRPO [9], but the approach was ultimately skipped under the massive computational constraints it would impose. Supervised adaptation on generated reasoning traces was likewise not pursued, because the competition restriction on data creation or modification using generative models left the admissibility of such traces unclear.

Unimplemented considerations. Several architecture-level formulations were considered but not evaluated: dynamic-vocabulary autoregression, 24-way classification, cross-encoder retrieval over all 24 text–sequence pairs [2], a transformer without positional encoding and with 24 ordered readouts [10], and graph diffusion over text–image and image–image edges [3]. Each requires a custom scoring or training objective beyond parameter-efficient adaptation of the existing VLM. Training a model entirely from scratch was rejected for a different reason: the task relies heavily on common-sense reasoning, and acquiring robust, generalized common-sense knowledge inherently requires massive-scale, diverse pretraining datasets. Building that contextual world knowledge from the provided, limited dataset alone would be highly impractical and prone to severe overfitting, so fine-tuning a pretrained VLM was the more rational and efficient strategy, leveraging the common-sense and multimodal priors already embedded in the model.

References

[1] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *arXiv:2305.14314*, 2023.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, 2019.

[3] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020.

[4] Edward J. Hu et al. LoRA: Low-rank adaptation of large language models. *arXiv:2106.09685*, 2021.

[5] Ting-Hao K. Huang et al. Visual storytelling, 2016. arXiv:1604.03968.

[6] John G. Kemeny. Mathematics without numbers. *Daedalus*, 88(4):577–591, 1959.

[7] Maurice G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938.

[8] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv:2301.12597*, 2023.

[9] Zhihong Shao et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.

[10] Ashish Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017.