

Diagnosing Order Dependence in Vision-Language Task-Progress Judgment with Equal-State Supervision

Can Wang¹, Xiaoda Yang², Xueyi Zhang³,
Menglan Tang¹, Jingyang Xue¹, Shenzhou Gao¹, Jiahe Zhang¹,
Xiaoran Geng¹, Jiangning Zhang², Hongyu Yuan¹, Peijian Zhang¹,
Xiaobin Hu³, Jianwen Lou^{2†}

¹ Qingdao University ² Zhejiang University

³ School of Computing, National University of Singapore
jianwen.lou@zju.edu.cn [†]Corresponding author

Abstract

Vision-Language Models (VLMs) are increasingly applied to embodied task settings, yet it remains unclear whether their task-progress judgments are grounded in physical state comparison or in input-order shortcuts. We study a diagnostic two-frame setting: given two robot manipulation states and a task goal, the model must decide which state is closer to completion, or whether the two states are effectively equal. We observe a strong Forward-Backward Performance Gap: models often answer correctly in the natural forward order, but fail when the same two physical states are shown with their image slots swapped. This reversal is not intended to mimic the dominant deployment order of robot execution; rather, it is a counterfactual diagnostic for testing whether the selected physical state remains consistent under input-slot inversion. To reduce this order dependence, we introduce *Equal-State* supervision as a zero-point calibration signal, training models to output *equal* when no task-relevant progress difference is present instead of defaulting to a directional answer. From real manipulation videos, we construct the Embodied Causal-Temporal Reasoning (ECTR) dataset and propose Tri-Stage Weighted Alignment (TSWA), a staged fine-tuning procedure that strengthens spatial description, task-conditioned comparison, and answer calibration in sequence. On Qwen2.5-VL-7B, TSWA reduces GAP from 60.50% to 5.14% and raises overall accuracy from 44.66% to 80.19%. This comes with a small decrease in forward accuracy, but large gains in backward accuracy, equal-state recognition, and pairwise consistency, suggesting that TSWA replaces order-dependent correctness with more calibrated state comparison.

1 Introduction

Embodied agents need to judge whether a physical state advances a task toward completion during robot execution. We study a minimal diagnostic

form of this ability: given two visible robot manipulation states and a task goal, a VLM must decide which state is closer to completion, or whether the two states are effectively equal under the task goal.

Natural robot executions usually present observations in forward temporal order. However, high accuracy in this natural order alone does not show that a model understands task progress: it may simply exploit the shortcut that the later-presented or naturally ordered frame is more complete. We therefore introduce a counterfactual forward-backward diagnostic. For each dynamic state pair, we evaluate both the natural order and the input-slot-swapped order. If the model truly compares physical states against the task goal, the selected physical state should remain the same even though the output slot label changes under input-slot reversal.

We use *progression bias* as a descriptive hypothesis for the observed asymmetry. Because many video and robot demonstrations are observed in natural temporal order, a VLM may learn an association between input order and task progress. We do not claim that order bias is the only source of error; rather, the forward-backward diagnostic isolates whether a model’s answer is stable under a controlled input-order intervention.

To reduce this order dependence, we introduce *Equal-State* supervision as a zero-point calibration signal. *Equal-State* examples provide a controlled no-change anchor: when the two inputs are identical under the same task goal, any directional answer is unsupported by visual evidence. This anchor teaches the model that task-progress comparison is not always a forced choice between *img1* and *img2*. In combination with forward-backward dynamic pairs, it calibrates the boundary between task-relevant progress and no task-relevant progress under the same goal.

To make this idea trainable and evaluable, we construct the Embodied Causal-Temporal Reason-

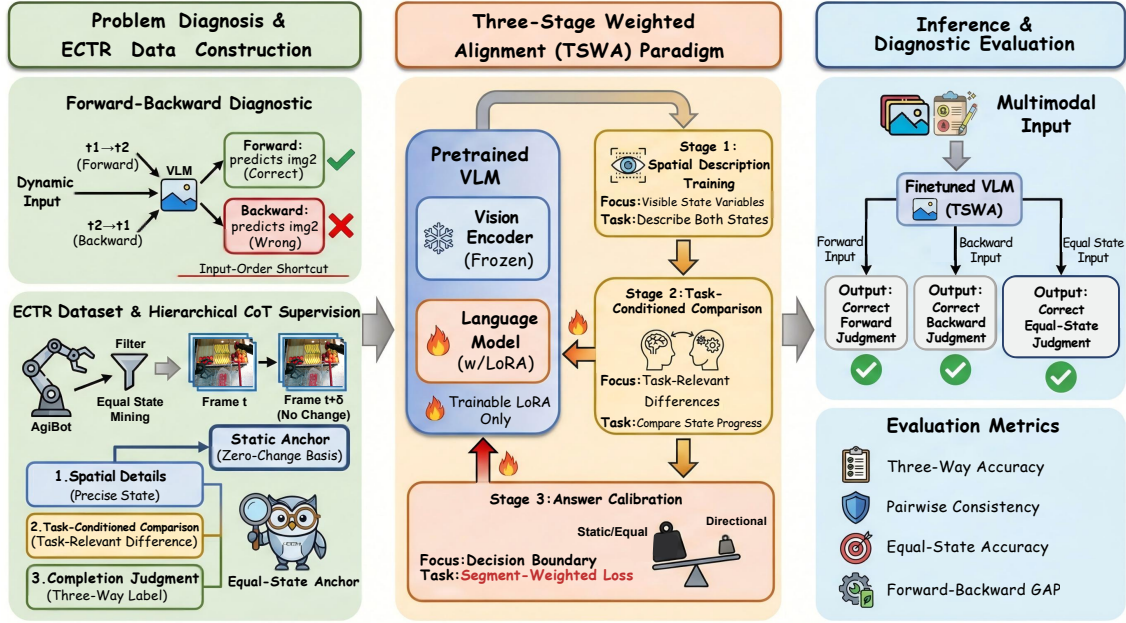


Figure 1: **Overview of the ECTR-TSWA framework.** (Left) Forward-backward reversal is used as a counterfactual diagnostic for order dependence in two-frame task-progress judgment. (Middle) TSWA follows an observe-compare-calibrate curriculum: spatial description, task-conditioned comparison, and answer calibration. (Right) Evaluation combines three-way accuracy, GAP, and pairwise consistency to test whether the model selects the same closer-to-completion physical state under input-slot reversal.

ing (ECTR) dataset from real robot manipulation videos with structured supervision. Each example contains two frames, a task instruction, and a three-way label from $\{img1, img2, equal\}$. In addition to the final label, ECTR provides structured supervision for spatial description, task-conditioned comparison, and completion judgment, covering forward, backward, and equal-state cases.

We further propose Tri-Stage Weighted Alignment (TSWA), a lightweight staged fine-tuning procedure built on ECTR rather than a new model architecture. Its stages follow the structure of the diagnostic task: observe the visible state variables, compare task-relevant differences under the goal, and calibrate the final three-way answer. TSWA is implemented with LoRA on the 7,308-example ECTR training split.

Contributions. Our main contributions are as follows:

- We introduce the Forward-Backward Performance Gap as a counterfactual diagnostic for order dependence in two-frame task-progress judgment. The backward setting is not treated as the dominant deployment order, but as a slot-reversal test of whether a model se-

lects the same closer-to-completion physical state.

- We construct ECTR, a diagnostic dataset from real robot manipulation videos, providing structured CoT-style supervision over spatial description, task-conditioned comparison, and completion judgment across forward, backward, and equal-state examples.
- We propose TSWA, a lightweight staged fine-tuning procedure. Experiments show that TSWA substantially reduces order dependence and improves pairwise consistency; ablations indicate that spatial description and task-conditioned comparison provide the main gains, while answer calibration offers a smaller but consistent refinement.

2 Related Work

2.1 Task-Progress Judgment and Order Dependence in VLMs

VLMs have expanded from static image understanding to video and multi-frame spatiotemporal reasoning (Fateh et al., 2024), yet recent benchmarks show systematic errors when temporal con-

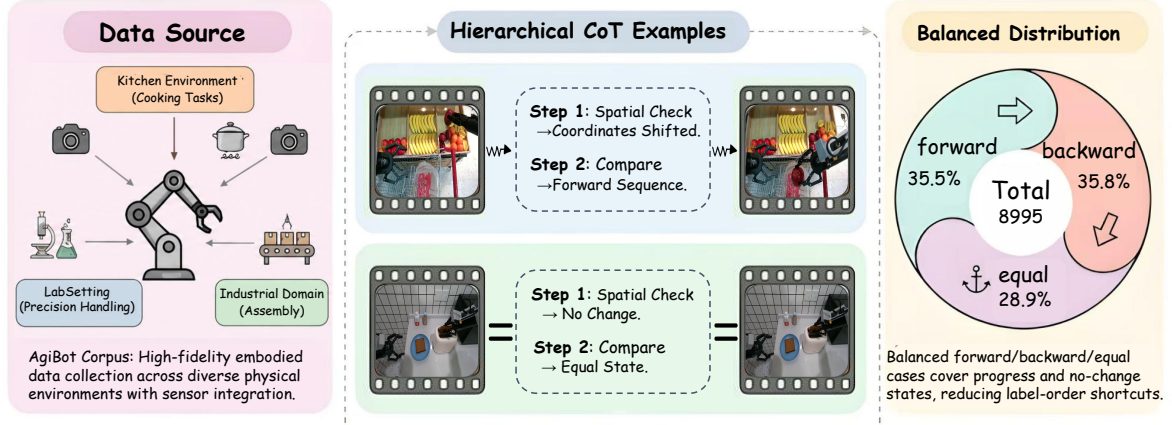


Figure 2: **ECTR dataset pipeline.** (Left) AgiBot data covering kitchen, lab, and industrial scenarios. (Middle) Hierarchical CoT supervision with spatial description, task-conditioned comparison, and three-way judgment (simplified two-step examples shown; full annotation in Section 3). (Right) Approximately balanced class distribution.

text must be combined with visual evidence (Xie et al., 2025; Liu et al., 2024). A common pathology is reliance on shallow cues—single-frame bias or text-side language priors—rather than genuine temporal evidence (Liu et al., 2024; Li et al., 2024b); a representative failure is temporal hallucination, where models produce directional judgments even when visual evidence is insufficient. Prior work mitigates such errors through negative samples, order perturbation, contrastive learning, or decoding strategies (Wu et al., 2025; Bansal et al., 2023; Wei et al., 2018), but often targets event order or action recognition over longer videos. A related asymmetry appears in language models trained on “A is B” that fail to generalize to “B is A” (Berglund et al., 2024); ours is a visual analogue, where the same physical event yields different judgments depending on frame order. We instead study a targeted diagnostic setting: whether a VLM makes consistent task-progress judgments when the same state pair is presented forward and reversed, deciding which frame is closer to completion or whether the two show no task-relevant difference under the task goal.

2.2 Task-Oriented State Comparison in Embodied AI

Task-progress estimation and state comparison are widely studied in robotics, video understanding, and embodied intelligence, via trajectory ranking, goal-state matching, learned reward representations, temporal contrastive learning, or VLM-based scoring (Schroeder et al., 2025; Ma et al., 2023). These are useful for long-horizon modeling

but many implicitly use temporal order as supervision, assuming later states are closer to success—an assumption that is brittle in real manipulation with failures, reversals, interruptions, or local perturbations. We focus on a minimal diagnostic unit: given two concrete states and a task instruction, determine which is closer to completion. By introducing equal-state examples, we further require recognizing when no task-relevant change has occurred, rather than forcing every difference into a directional progress label.

2.3 Structured CoT Supervision and Multi-Stage Alignment

Visual Chain-of-Thought and structured intermediate supervision improve multimodal reasoning (Li et al., 2025; Shao et al., 2024; Wei et al., 2022), helping models organize visual evidence and reduce shallow shortcuts (Geirhos et al., 2020). However, most CoT-style datasets target static image understanding or general VQA, rarely supervising task-conditioned state comparison or no-change decisions. Multi-stage alignment and curriculum learning progress from simpler perceptual tasks to complex reasoning (Bengio et al., 2009; Chen et al., 2025). TSWA follows this direction but targets a narrower failure mode—inconsistent two-frame task-progress judgment—separating perception, comparison, and calibration, while Equal-State supervision supplies a zero-point reference for insufficient visual evidence. Adjacent multimodal work has studied audio-video generation, multimodal data augmentation, cross-modal generation evaluation, and pro-

gressive training for embodied hallucination mitigation (Yang et al., 2026a, 2024, 2026c,b); our focus is a diagnostic benchmark for order dependence in task-progress state comparison.

3 Dataset

To support this diagnostic, ECTR contains 8,995 samples covering three types of two-frame task-progress judgments: forward, backward, and equal-state. It is built from real robot manipulation videos drawn from AgiBot World (AgiBot-World-Contributors et al., 2025), a large-scale embodied corpus spanning five deployment domains with over 100 one-to-one replicated real-life scenarios, including kitchen, laboratory, and industrial settings. This breadth lets ECTR sample task-progress comparisons across diverse objects, layouts, and manipulation goals rather than a single narrow environment. A representative gallery is provided in Appendix B.8. Through trajectory-level splitting and three-layer CoT annotation, ECTR provides both final labels and structured supervision for spatial description, task-conditioned comparison, and completion judgment in one unified format.

3.1 Task Definition and Data Construction

Each ECTR sample consists of two images, a task instruction, and a three-way label from {img1, img2, equal}. The label indicates whether img1 is closer to completion, img2 is closer, or the two states are indistinguishable in task progress. We downsample raw videos to reduce frame-level noise and select temporally separated action states for dynamic pairs. For equal-state samples, we intentionally duplicate one frame as both inputs, using exact identity as a controlled zero-point rather than as a complete model of real static scenes. Under literally unchanged evidence, any directional prediction is necessarily a false motion prediction, making this setting a clean probe for the model’s tendency to hallucinate progress. Noisier near-static pairs, which differ at the pixel level but not in task state, are a harder regime left to future work (see Limitations). The dataset contains 8,995 samples and is approximately balanced across the three classes.

Trajectory-level split protocol. To eliminate event-level data leakage, we split the data at the trajectory level: each original robot manipulation video is assigned as a whole to exactly one of

train, validation, or test, with all forward, backward, and equal-state pairs from that video kept in the same split. This rules out train–test leakage from adjacent-frame sampling and ensures all test episodes are unseen during training. The final train, validation, and test sets contain 7,308, 844, and 843 samples, respectively.

3.2 Hierarchical Supervision and Annotation Workflow

ECTR organizes supervision into three levels. The first level, spatial description, records the positions, poses, contact relationships, and local scene layout of the objects in both frames, providing direct visual evidence for the later judgment. The second level, task-conditioned comparison, analyzes the state differences between the frames and explains how those differences relate to the task goal. The third level, completion judgment, outputs the final three-way label. This design explicitly decomposes task-progress judgment into a process of observing, comparing, and then judging, so that during training the model sees not only final answers, but also intermediate state representations and task-conditioned comparison logic.

For annotation, GPT-5 generates spatial descriptions and comparison reasoning chains; the final label is assigned by rule on monotonic-success trajectories (later state $\rightarrow$ img2, earlier $\rightarrow$ img1, duplicate $\rightarrow$ equal), so the supervision target is unaffected by GPT-5 generation bias. Five graduate students reviewed CoT text for factual and logical errors; full quality-control statistics are reported in Appendix B.

4 Method

A single-stage training objective would entangle three distinct learning problems: perceiving fine-grained object states, comparing task-conditioned progress across frames, and calibrating the final decision boundary under ambiguity. These problems are separable because order bias can enter at different levels—a model may fail to detect relevant visual differences, compare the wrong physical attributes, or overcommit to a directional prediction despite insufficient evidence. TSWA targets each failure mode in sequence rather than leaving them to be resolved jointly.

TSWA is a lightweight staged fine-tuning procedure rather than a new model architecture. The pipeline decomposes training into three con-

nected sub-tasks: spatial description training, task-conditioned comparison training, and answer calibration. The backbone is kept frozen throughout, and LoRA (Hu et al., 2021) is used for parameter-efficient fine-tuning. The three stages are executed sequentially on the same ECTR training set, with each later stage continuing from the previous checkpoint rather than training from scratch.

4.1 Spatial Description Training

The first stage aims to improve the model’s perception of fine-grained state differences. Many VLMs, when processing image pairs, invoke high-level semantics too early and overlook key information such as coordinates, poses, and contact relationships. We therefore restrict the task to pure observation: given an image pair and a task instruction, the model only needs to describe the state of the objects in each frame, without making any directional judgment. The supervision corresponds to the spatial-description layer of ECTR. By decoupling observation from judgment, this stage pulls the model’s attention back to visible evidence, establishing a reliable state representation for later comparison and calibration stages.

4.2 Task-Conditioned Comparison Training

After Stage 1 has produced a more stable state representation, Stage 2 teaches the model how to map visible state differences to task-progress judgments. We introduce a structured CoT sequence that asks the model to follow a perceive–compare–judge chain. Given an image pair, the model first extracts spatial details from the two frames, then identifies which differences are task-relevant—for example whether an object is closer to the target, whether a container is closer to being emptied, or whether a gripper is closer to being closed—and finally outputs a three-way label. For equal-state samples, the model must infer *equal* from the fact that no significant task-relevant difference is observed. This reasoning path requires the model to draw a conclusion from the absence of evidence, rather than defaulting to a directional judgment. The role of this stage is to turn the visual representations from Stage 1 into an interpretable cross-frame comparison process under the task goal.

4.3 Answer Calibration and Segment-Weighted Loss

Stage 3 does not add new visual evidence; it refines the final answer boundary on top of the per-

ception and comparison abilities learned in Stages 1 and 2. At this point the model may already describe object states and compare them, but still overcommit to a directional answer when the evidence is insufficient. We therefore introduce a token-level segment-weighted loss that assigns increasing weight along the response: the spatial-description segment receives a base weight, the task-conditioned-comparison segment receives a medium weight, and the final completion label receives the highest weight.

The weighting is asymmetric across *response segments*, not across answer classes. Equal-state samples do not receive a separate class multiplier. Instead, all samples place their final answer token under the largest segment weight. As a result, final-decision mistakes become more salient during calibration, including false directional predictions on equal-state anchors and incorrect slot predictions on dynamic forward/backward pairs. The loss is defined as:

$$\mathcal{L}_{\text{TSWA}} = -\frac{\sum_{i=1}^N w_i \cdot \log P(y_i | y_{<i}, I)}{\sum_{i=1}^N w_i \cdot \mathbb{I}(y_i \neq \text{pad})} \quad (1)$$

where the three segment weights are $\lambda_1 = 1.0$, $\lambda_2 = 3.0$, and $\lambda_3 = 10.0$:

$$w_i = \begin{cases} \lambda_1, & y_i \in \text{spatial-desc. segment,} \\ \lambda_2, & y_i \in \text{task-conditioned-} \\ & \text{comparison segment,} \\ \lambda_3, & y_i \in \text{final-judgment segment.} \end{cases}$$

The normalization by the sum of token weights keeps the loss scale stable when the final segment receives a larger weight. This keeps intermediate reasoning supervision while making the final output the strongest calibration target. Section 5.3 quantifies its contribution relative to ordinary continued fine-tuning.

5 Experiments

5.1 Experimental Setup

Evaluation metrics. ECTR evaluates three complementary aspects of calibrated task-progress judgment: directional state selection, no-change recognition, and order-invariant pairwise consistency. We use the Forward-Backward Performance Gap (GAP) as the core metric for measuring imbalance between forward and backward

Model	Bwd ($\uparrow$)	Fwd ($\uparrow$)	Equal ($\uparrow$)	Overall ($\uparrow$)	GAP ($\downarrow$)
Proprietary Models					
GPT-4o-mini (OpenAI, 2024)	36.60	100.00	11.60	54.72	63.40
GPT-4o (OpenAI, 2024)	53.70	99.80	25.60	64.49	46.10
GPT-o3 (OpenAI, 2025)	28.20	97.30	50.40	59.86	69.10
Gemini-1.5-Pro (Gemini Team, Google, 2024)	77.70	87.70	70.13	79.68	10.00
Doubao-1.5-Pro (ByteDance Seed, 2025a)	55.50	82.70	85.61	73.11	27.20
Seed-1.6 (ByteDance Seed, 2025b)	56.90	100.00	52.33	72.21	43.10
Open-Source Models (2D Perception)					
NVILA-8B (Liu et al., 2025)	58.90	41.90	50.49	50.40	17.00
PaliGemma (Beyer et al., 2024)	4.10	84.20	16.32	37.54	80.10
Qwen2.5-VL-7B (Bai et al., 2025)	23.20	83.70	16.67	44.66	60.50
InternVL-8B (Chen et al., 2024)	14.90	83.30	26.64	43.77	68.40
DeepSeek-VL2 (Wu et al., 2024)	55.20	84.50	51.32	65.42	29.30
LLaVA-OneVision-7B (Li et al., 2024a)	22.80	82.10	4.90	41.07	59.30
Open-Source Models (3D Perception)					
3D-LLM (Hong et al., 2023)	28.40	35.80	13.86	27.72	7.40
LL3DA (Chen et al., 2023)	28.90	42.00	58.43	41.00	13.10
Chat-Scene (Huang et al., 2024)	38.20	45.40	37.79	40.84	7.20
Video-3D LLM (Zheng et al., 2025)	68.90	31.60	66.64	54.15	37.30
Ours (TSWA)					
Ours (S1: Spatial Description)	59.25	87.23	71.43	72.84	27.98
Ours (S1+S2: Full CoT)	72.41	79.44	87.68	78.77	7.03
Ours (Full TSWA)	74.92	80.06	88.67	80.19	5.14

Table 1: **Main results on ECTR.** Accuracy (%) for backward/forward/equal classes, overall accuracy over all test examples, and Forward-Backward Gap (GAP; lower better). Forward accuracy alone can be misleading: several models obtain high forward accuracy while failing under slot reversal or equal-state cases.

queries. GAP is defined as the absolute difference between forward accuracy and backward accuracy on the same physical event. In addition to GAP, we report accuracy for the three labels and overall accuracy over all test examples. GAP and overall accuracy are complementary: a low GAP indicates balanced behavior across frame orders but not necessarily correct ones, since symmetric failure in both directions also yields a small GAP (e.g., the 3D-perception baselines in Table 1). We therefore report the two together. We additionally report Pairwise Consistency (PC) on the dynamic forward-backward pairs. Whereas GAP is an aggregate difference between two separately computed accuracies, PC is an instance-level metric: for each physical state pair we issue both the forward query and the backward query—the same two states with their image slots swapped—and count the pair as consistent only when the model answers *both* queries correctly. PC is the percentage of dynamic pairs meeting this criterion, so a model that gives the same slot-based answer in both orders, or that predicts *equal*, fails the pair. PC is thus a strictly harder bar than either forward or backward accuracy alone on the same pair. Cru-

cially, $GAP = |Acc_{fwd} - Acc_{bwd}|$ is aggregated separately over all forward and backward examples, whereas PC is an instance-level criterion on each physical pair; a small GAP therefore does not imply high PC. Besides symmetric failure in both orders, the model may answer one subset of pairs correctly forward and a different subset correctly backward, giving $Acc_{fwd} \approx Acc_{bwd}$ ($GAP \approx 0$) while almost no pair is correct in both orders (low PC). We therefore report GAP and PC together.

Evaluation protocol and prompt control.

Baseline models are evaluated zero-shot, and all trained variants are evaluated under the same CoT-style prompt with *equal* as an explicit output option and greedy decoding for reproducibility; the full template is provided in Appendix A.

Implementation details. The backbone is Qwen2.5-VL-7B-Instruct (Bai et al., 2025) with LoRA (Hu et al., 2021) (rank = 32, alpha = 64); full hyperparameters and the training schedule are reported in Appendix A.

5.2 Main Results on ECTR

Table 1 reports the overall performance of different models on ECTR. Existing VLMs exhibit substantial instability in two-frame task-progress judgment. For example, Qwen2.5-VL-7B achieves 83.70% forward accuracy but only 23.20% backward accuracy, yielding a GAP of 60.50%. Similar imbalances appear across both proprietary and open-source models: GPT-o3 reaches 97.30% forward but 28.20% backward, and InternVL-8B reaches 83.30% forward but 14.90% backward. These results indicate that natural-order accuracy alone is insufficient for evaluating task-progress judgment, since a model may answer forward examples correctly while failing the slot-reversal diagnostic.

Our full TSWA model improves overall accuracy from 44.66% to 80.19% over the Qwen2.5-VL-7B backbone, while reducing the forward-backward GAP from 60.50% to 5.14%.

Forward accuracy and calibrated reliability.

Forward-order accuracy decreases from 83.70% in the zero-shot backbone to 80.06% after Full TSWA. We view this as an important trade-off rather than hiding it: the backbone’s high forward accuracy is paired with very low backward and equal-state accuracy, suggesting that part of its forward success comes from an input-order shortcut. More specifically, most of the forward drop occurs when the model moves from zero-shot prediction to the state-comparison stages (83.70%→79.44% at S1+S2), while Stage-3 calibration partially recovers forward accuracy (79.44%→80.06%) and further reduces GAP. TSWA therefore gives up a small amount of shortcut-driven forward advantage, but gains much larger improvements in backward accuracy (23.20%→74.92%), equal-state accuracy (16.67%→88.67%), and PC (12.23%→69.28%). The goal of TSWA is not to maximize natural-order accuracy alone, but to reduce order-dependent correctness and improve calibrated state comparison.

Pairwise Consistency (Table 2) confirms this instance-level effect. The backbone answers both orders correctly for only 12.23%, whereas TSWA reaches 69.28% through the staged progression (S1 54.55%, S1+S2 63.32%). Since PC is computed only on dynamic pairs and counts *equal* predictions as incorrect, the gain reflects improved slot-reversal reasoning rather than a shift toward the equal label on dynamic pairs.

Setting	GAP (↓)	PC (↑)
Qwen2.5-VL-7B (zero-shot)	60.50	12.23
S1	27.98	54.55
S1+S2	7.03	63.32
Full TSWA	5.14	69.28

Table 2: Pairwise Consistency (PC, %) on 319 dynamic forward-backward pairs alongside GAP. A pair counts only when both queries are answered correctly; *equal* on a dynamic pair fails. PC rises as GAP falls.

Equal-state failures in instruction-aligned models.

Several instruction-aligned models still fail on equal-state examples despite near-saturated forward accuracy and an explicit *equal* option in the prompt (GPT-4o: 99.80% forward, 25.60% equal; GPT-4o-mini: 100.00% forward, 11.60% equal), suggesting a directional-progress prior rather than an underspecified output space.

5.3 Controlled Analysis

Could Stage 3 reduce GAP simply by recalibrating the answer boundary, without genuine comparison ability? To isolate this, we continue from the S1+S2 checkpoint under three conditions identical in data, epochs (6), and learning rate (1.5×10^{-5}): stopping at **S1+S2**; **calibration-only** ordinary fine-tuning (all token weights 1.0); and **Full TSWA** with the segment-weighted loss.

Table 3 shows that S1+S2 already yields 78.77 overall / 7.03 GAP, while S3 alone reaches only 63.70 / 9.02 (Section 6). Ordinary continued training narrows GAP to 5.77; segment-weighted calibration goes further to 5.14 with gains distributed across all three classes (backward 73.67 → 74.92, forward 79.44 → 80.06, equal 88.18 → 88.67)—confirming that Stage 3 contributes genuine calibration, not the primary capability gain. The improvement is spread across labels rather than concentrated on equal-state samples alone, consistent with Stage 3’s role as a decision-boundary calibrator rather than a new comparison module.

5.4 Cross-Dataset and Cross-Architecture Generalization

External benchmark transfer evaluation. To test whether the model merely adapts to the distribution of ECTR, we evaluate it on the external MMSI-Bench benchmark (Yang et al., 2025). The TSWA-finetuned 7B model reaches 30.10% overall accuracy, a +4.20-point gain over its own Qwen2.5-VL-7B backbone. Because cur-

rent VLMs cluster within a narrow low-accuracy range on this benchmark (e.g., GPT-4o 30.30%, Claude-3.7-Sonnet 30.20%, and Qwen2.5-VL-72B 30.70%), we interpret this result as a modest external-transfer check rather than as evidence of broad cross-benchmark superiority. Improvements are slightly higher on position relation (40.00%) and attribute measurement (39.06%), broadly consistent with the goal of Stage 1. Additional MMSI-Bench details are provided in Appendix C.2.

Cross-architecture generalization. For cross-architecture testing, we apply Stage 2 to five additional open-source VLMs and evaluate the full TSWA pipeline on InternVL-8B and DeepSeek-VL2; complete results are in Appendix C. Across the five additional backbones, Stage 2 reduces GAP after task-conditioned comparison training (e.g., InternVL-8B 68.4%→16.7%, PaliGemma 80.1%→23.3%). Together with the Qwen2.5-VL main-backbone result, this shows reduced order dependence across six open-source architectures. Full TSWA further improves InternVL-8B and DeepSeek-VL2, reducing GAP from 16.72 to 13.88 and from 12.33 to 10.44, respectively, with consistent overall-accuracy gains across both backbones.

6 Ablation Studies

6.1 Effectiveness of Multi-Stage Training

S3 alone achieves only 63.70 overall / 9.02 GAP; partial combinations S1+S3 (69.63 / 12.40) and S2+S3 (69.99 / 8.66) both fall short of S1+S2 (78.77 / 7.03), confirming that spatial grounding followed by task-conditioned comparison is the primary source of capability. Notably, S1+S3 produces a larger GAP (12.40) than S3 alone (9.02), indicating that perception training without task-conditioned comparison can sharpen the forward bias rather than correct it. Full S1+S2+S3 further improves to 80.19 / 5.14; Stage 3 is most effective once both perception and comparison are in place. The ordering matters: spatial grounding must precede comparison, which in turn anchors the final calibration—reversing or skipping stages reopens the gap, as the anomalously high GAP of S1+S3 (12.40, above even S3 alone) makes explicit.

6.2 Synergy of CoT Components

We ablate the three components of the structured CoT supervision: spatial description (S), compar-

Method	Bwd	Fwd	Equal	Overall	GAP
<i>Controlled Calibration Analysis</i>					
S1+S2	72.41	79.44	87.68	78.77	7.03
S1+S2+S3 (uniform)	73.67	79.44	88.18	79.36	5.77
Full TSWA	74.92	80.06	88.67	80.19	5.14
<i>Effectiveness of Multi-Stage Training</i>					
S3 only	52.35	61.37	85.22	63.70	9.02
S1+S3	60.50	72.90	78.82	69.63	12.40
S2+S3	60.50	69.16	86.21	69.99	8.66
S1+S2	72.41	79.44	87.68	78.77	7.03
S1+S2+S3 (Full)	74.92	80.06	88.67	80.19	5.14
<i>Synergy of CoT Components</i>					
S (Spatial)	59.25	87.23	71.43	72.84	27.98
R (Comparison)	37.93	59.50	61.58	51.84	21.57
C (Judgment)	23.20	70.40	62.07	50.53	47.21
S+R	65.52	72.59	60.10	66.90	7.07
S+R+C (Full CoT)	74.92	80.06	88.67	80.19	5.14
<i>Impact of Equal-State Constraint</i>					
w/o Equal-State	54.55	66.67	23.65	51.72	12.12
Full Model	74.92	80.06	88.67	80.19	5.14

Table 3: **Controlled and ablation studies on ECTR.** Overall denotes accuracy over all test examples. Blocks ablate calibration, multi-stage structure, CoT components, and equal-state constraint.

ison reasoning (R), and completion judgment (C). S alone leaves a GAP of 27.98 despite strong forward accuracy; R-only and C-only both perform worse on average, confirming that comparison logic and judgment labels require grounded spatial descriptions. S+R reduces GAP to 7.07; adding C further improves overall accuracy to 80.19 and equal-state accuracy to 88.67. The three components are complementary: S anchors the visual state, R organizes the comparison, and C aligns the chain with the final decision. Notably, C alone is the weakest single component (50.53 overall), yet adding it on top of S+R yields the largest equal-state gain (60.10→88.67).

6.3 Role of the Equal-State Constraint

Removing equal-state samples sharply reduces equal accuracy (88.67→23.65) and overall accuracy (80.19→51.72), and also hurts both directional classes (backward 74.92→54.55; forward 80.06→66.67). Thus, equal-state supervision is not merely an auxiliary identity-detection class: it provides a no-change reference that stabilizes the decision boundary across all three labels.

Overall, these ablations show that TSWA closes the gap through complementary spatial grounding, task-conditioned comparison, and equal-state calibration rather than a simple label-prior shift, sup-

porting the separation of perception, comparison, and calibration over a single monolithic objective.

7 Conclusion

This paper identifies the Forward-Backward Performance Gap as a counterfactual diagnostic for order dependence in two-frame task-progress judgment. On ECTR, TSWA improves Qwen2.5-VL-7B overall accuracy from 44.66% to 80.19%, reduces GAP from 60.50% to 5.14%, and raises pairwise consistency from 12.23% to 69.28%. Ablations show that spatial grounding and task-conditioned comparison provide the main gains, while answer calibration refines the final boundary. Overall, slot reversal and equal-state anchors encourage calibrated state comparison over order-dependent shortcuts. More broadly, our results suggest that high natural-order accuracy alone is an unreliable indicator of task-progress understanding, motivating order-invariant comparison as an explicit training target.

Limitations

Although TSWA substantially reduces forward-backward imbalance, several limitations remain. First, equal-state pairs are constructed by duplicating a single frame. This design intentionally provides a controlled zero-point for diagnosing false directional predictions under literally unchanged visual evidence, but it is not a complete benchmark for near-static non-identical states with camera noise, illumination changes, object jitter, or task-irrelevant motion. Second, intermediate spatial descriptions and comparison reasoning chains are generated by GPT-5 and then manually reviewed; while final labels are rule-assigned and audited, part of the gain may still reflect teacher-guided descriptive patterns. Third, the method assumes a stable camera viewpoint, limiting applicability in continuous egomotion settings. Fourth, dynamic pairs come from successful monotonic-progress trajectories; non-monotonic executions with retries or failed grasps, and tasks with memory dependence, fall outside the pairwise formulation—residual proxy noise may also remain in visually ambiguous pairs despite our label audit. Fifth, TSWA slightly reduces natural forward-order accuracy on the main backbone. This reflects a trade-off between preserving order-specific forward performance and improving order-invariant reliability; future work should

examine this trade-off in online progress monitoring and policy-evaluation settings. Sixth, although we verify cross-architecture transfer on six open-source VLMs, these backbones are at the 7–8B scale and do not include the latest vision-language models (such as the recent Qwen3-VL series); validating TSWA on more recent and more capable backbones is left to future work, and we expect the perceive–compare–calibrate supervision to remain applicable as backbones improve, since it targets the order-dependence failure mode rather than any backbone-specific capacity. It nonetheless remains open whether stronger backbones, which already exhibit lower zero-shot GAP, leave enough headroom for staged alignment to yield comparable relative gains. Finally, three-stage training introduces overhead over single-stage fine-tuning; exploring more unified frameworks is a natural direction.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. U23A20311.

References

- AgiBot-World-Contributors, Qingwen Bu, Jisong Cai, Li Chen, Xiuqi Cui, Yan Ding, Siyuan Feng, Shenyuan Gao, Xindong He, Xuan Hu, Xu Huang, Shu Jiang, Yuxin Jiang, Cheng Jing, Hongyang Li, Jialu Li, Chiming Liu, Yi Liu, Yuxiang Lu, and 33 others. 2025. AgiBot World Colosseo: A large-scale manipulation platform for scalable and intelligent embodied systems. *arXiv preprint arXiv:2503.06669*.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*.
- Hritik Bansal, Yonatan Bitton, Idan Szepktor, Kai-Wei Chang, and Aditya Grover. 2023. VideoCon: Robust video-language alignment via contrast captions. *arXiv preprint arXiv:2311.10111*.
- Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. [Curriculum learning](#). In *Proceedings of the 26th International Conference on Machine Learning*, pages 41–48.
- Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and

- Owain Evans. 2024. The reversal curse: LLMs trained on “A is B” fail to learn “B is A”. In *ICLR*.
- Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, and 16 others. 2024. PaliGemma: A versatile 3B VLM for transfer. *arXiv preprint arXiv:2407.07726*.
- ByteDance Seed. 2025a. Doubao-1.5-Pro. https://team.doubao.com/en/special/doubao_1_5_pro.
- ByteDance Seed. 2025b. Seed1.6 technical report. https://seed.bytedance.com/zh/seed1_6.
- Menglan Chen, Xianghe Pang, Jingjing Dong, Wen-Hao Wang, Yaxin Du, and Siheng Chen. 2025. VLMGuard-R1: Proactive safety alignment for VLMs via reasoning-driven prompt optimization. *arXiv preprint arXiv:2504.12661*.
- Sijin Chen, Xin Chen, Chi Zhang, Mingsheng Li, Gang Yu, Hao Fei, Hongyuan Zhu, Jiayuan Fan, and Tao Chen. 2023. LL3DA: Visual interactive instruction tuning for omni-3D understanding, reasoning, and planning. *arXiv preprint arXiv:2311.18651*.
- Zhe Chen, Jiannan Wu, Wenhao Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*.
- Fawad Javed Fateh, Umer Ahmed, Hamza Khan, M. Zeeshan Zia, and Quoc-Huy Tran. 2024. TemporalVLM: Video LLMs for temporal reasoning in long videos. *arXiv preprint arXiv:2412.02930*.
- Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673.
- Gemini Team, Google. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.
- Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 2023. 3D-LLM: Injecting the 3D world into large language models. In *NeurIPS*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.
- Haifeng Huang, Yilun Chen, Zehan Wang, Rongjie Huang, Runsen Xu, Tai Wang, Luping Liu, Xize Cheng, Yang Zhao, Jiangmiao Pang, and Zhou Zhao. 2024. Chat-Scene: Bridging 3D scene and large language models with object identifiers. *arXiv preprint arXiv:2312.08168*.
- Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. 2025. OmniSpatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*.
- Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. LLaVA-OneVision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.
- Lingxiao Li, Yifan Wang, Xinyan Gao, Chen Tang, Xiangyu Yue, and Chenyu You. 2025. VisReason: A large-scale dataset for visual chain-of-thought reasoning. *arXiv preprint arXiv:2511.17731*.
- Shicheng Li, Lei Li, Yi Liu, Shuhuai Ren, Yuanxin Liu, Rundong Gao, Xu Sun, and Lu Hou. 2024b. VI-TATECS: A diagnostic dataset for temporal concept understanding of video-language models. In *ECCV*.
- Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Shihuo Chen, Xu Sun, and Lu Hou. 2024. TempCompass: Do video LLMs really understand videos? In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8731–8772.
- Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, Xiuyu Li, Haotian Tang, Yunhao Fang, Yukang Chen, Cheng-Yu Hsieh, De-An Huang, An-Chieh Cheng, Jinyi Hu, Sifei Liu, and 6 others. 2025. NVILA: Efficient frontier visual language models. In *CVPR*, pages 4122–4134.
- Yecheng Jason Ma, Vikash Kumar, Amy Zhang, Osbert Bastani, and Dinesh Jayaraman. 2023. LIV: Language-image representations and rewards for robotic control. In *ICML*.
- OpenAI. 2024. GPT-4o system card. Technical report, OpenAI.
- OpenAI. 2025. OpenAI o3 and o4-mini System Card. <https://openai.com/index/o3-o4-mini-system-card/>.
- Philip Schroeder, Ondrej Biza, Thomas Weng, Hongyin Luo, and James Glass. 2025. ROVER: Recursive reasoning over videos with vision-language models for embodied tasks. *arXiv preprint arXiv:2508.01943*.
- Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. 2024. Visual CoT: Advancing multimodal language models with a comprehensive

- dataset and benchmark for chain-of-thought reasoning. *arXiv preprint arXiv:2403.16999*.
- Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. 2024. [Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs](#). *arXiv preprint arXiv:2406.16860*.
- Donglai Wei, Joseph J. Lim, Andrew Zisserman, and William T. Freeman. 2018. Learning and using the arrow of time. In *CVPR*, pages 8052–8060.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*.
- Chang-Hsun Wu, Kai-Po Chang, Yu-Yang Sheng, Hung-Kai Chung, Kuei-Chun Wang, and Yu-Chiang Frank Wang. 2025. SEASON: Mitigating temporal hallucination in video large language models via self-diagnostic contrastive decoding. *arXiv preprint arXiv:2512.04643*.
- Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, and 8 others. 2024. DeepSeek-VL2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*.
- Qinghongbing Xie, Zhaoyuan Xia, Feng Zhu, Lijun Gong, Ziyue Li, Rui Zhao, and Long Zeng. 2025. GTR-Bench: Evaluating geo-temporal reasoning in vision-language models. *arXiv preprint arXiv:2510.07791*.
- Siyan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. 2025. MMSI-Bench: A benchmark for multi-image spatial intelligence. *arXiv preprint arXiv:2505.23764*.
- Xiaoda Yang, Xize Cheng, Jiaqi Duan, Hongshun Qiu, Minjie Hong, Minghui Fang, Shengpeng Ji, Jialong Zuo, Zhiqing Hong, Zhimeng Zhang, and Tao Jin. 2024. [AudioVSR: Enhancing video speech recognition with audio data](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15352–15361, Miami, Florida, USA. Association for Computational Linguistics.
- Xiaoda Yang, Yuxiang Liu, Kaiwen Zheng, Yuan Liu, Yibo Lai, Shengpeng Ji, Kai Jiang, Jianfei Chen, Xiaobin Hu, Shuicheng Yan, Jintao Zhang, Jun Zhu, and Zhou Zhao. 2026a. [TurboT2VA: Fast large-scale text-to-video-audio generation via score-regularized consistency distillation](#). *Preprint*, arXiv:2608.24674.
- Xiaoda Yang, Shuai Yang, Can Wang, Jingyang Xue, Menglan Tang, Checheng Yu, Xunzhe Zhou, Sashuai Zhou, Tao Jin, Lixin Yang, Xiangyu Yue, and Zhou Zhao. 2026b. A progressive training strategy for vision-language models to counteract spatio-temporal hallucinations in embodied reasoning. *arXiv preprint arXiv:2604.10506*.
- Xiaoda Yang, Majun Zhang, Changhao Pan, Nick Huang, Yuguang Yang, Fan Zhuo, Pengfei Zhou, Jin Zhou, Sizhe Shan, Shan Yang, Miles Yang, Yang You, and Zhou Zhao. 2026c. TMD-Bench: A multi-level evaluation paradigm for music-dance co-generation. *arXiv preprint arXiv:2605.01809*.
- Duo Zheng, Shijia Huang, and Liwei Wang. 2025. Video-3D LLM: Learning position-aware video representation for 3D scene understanding. In *CVPR*.

A Full Experimental Details

A.1 Model Architecture and Hyperparameters

We use Qwen2.5-VL-7B-Instruct (Bai et al., 2025) as the vision-language backbone. We use LoRA (Hu et al., 2021) for parameter-efficient fine-tuning and inject trainable adapters into all linear modules, including q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, and down_proj. The LoRA hyperparameters are set to rank $r = 32$, alpha $\alpha = 64$, and dropout rate = 0.05.

Training uses the AdamW optimizer with a cosine-annealing learning-rate schedule. To maintain high-resolution visual inputs under memory constraints, we use gradient accumulation so that all stages have an effective global batch size of 128. All training uses BF16 mixed precision and gradient checkpointing to maximize computational throughput. The full TSWA training pipeline was run on a single NVIDIA A100 GPU and took approximately 30 hours in total, corresponding to about 30 A100 GPU-hours.

A.2 TSWA Training Protocol

We instantiate the TSWA paradigm as a decaying-learning-rate process. Training is divided into three progressive stages with different epoch counts and learning rates:

- **Stage 1 (spatial description training):** 3 epochs, $\eta = 5 \times 10^{-5}$. This stage addresses inherent spatial-perception blind spots through a pure observation task.
- **Stage 2 (task-conditioned comparison training):** 6 epochs, $\eta = 3 \times 10^{-5}$. This stage injects the full structured CoT sequence to build cross-frame comparison logic.
- **Stage 3 (answer calibration):** 6 epochs, $\eta = 1.5 \times 10^{-5}$. This stage uses the segment-weighted loss to refine the final answer boundary.

A.3 Segment-Weighted Loss Configuration

Stage 3 uses a token-level segment-weighted loss. Given a target sequence

$$Y = \{y_1, y_2, \dots, y_N\},$$

we assign each supervised token y_i a segment weight w_i according to whether it belongs to the

spatial-description segment, the task-conditioned-comparison segment, or the final completion-judgment segment. The loss is the same as Equation 1:

$$\mathcal{L}_{\text{seg}} = -\frac{\sum_{i=1}^N w_i \log P(y_i | y_{<i}, I)}{\sum_{i=1}^N w_i \cdot \mathbb{I}(y_i \neq \text{pad})}.$$

The normalization by the sum of non-padding token weights keeps the loss scale stable even when the final segment receives a larger weight.

The segment weights are

$$w_i = \begin{cases} \lambda_1, & y_i \in \text{spatial-desc. segment,} \\ \lambda_2, & y_i \in \text{task-conditioned-} \\ & \text{comparison segment,} \\ \lambda_3, & y_i \in \text{final-judgment segment.} \end{cases}$$

The term *asymmetric* refers to this unequal weighting across response segments rather than to class-specific multipliers. In particular, equal-state examples do not receive a separate class weight. Instead, all examples place their final answer token under the largest segment weight λ_3 . As a result, final-decision mistakes become more salient during Stage 3, including false directional predictions on equal-state anchors and incorrect slot choices on dynamic forward/backward pairs.

This design matches the role of Stage 3 as answer-boundary calibration. It does not introduce new visual evidence and is not intended to be the primary source of capability gain. Rather, after Stages 1 and 2 have established spatial grounding and cross-frame comparison, Stage 3 concentrates supervision on the final mapping from the reasoning chain to the output label.

A.4 Default CoT Prompt Template

All training and evaluation instances use a unified multimodal instruction-following format. Each instance contains two images, denoted as img1 and img2, a task instruction, and the following fixed CoT-style prompt template. The template asks the model to first describe the spatial state of each image, then produce a task-conditioned comparison, and finally output a three-way label from {img1, img2, equal}.

```
You are a spatial perception
expert.
Please observe carefully and
describe the two images (img1
and img2) separately.
Task name: {task_instruction}
```

Description points:

1. Provide detailed descriptions of *img1* and *img2*.
2. Based on the above descriptions, analyze which image (*img1* or *img2*) has a state that is closer to the final stage of task completion. If the states in both images are essentially the same or nearly identical, then respond with *equal*.
3. Provide the final answer directly.

Output format:

spatial details: [Separately describe the spatial details of both images in detail]
 reason: [Analyze why one image is closer to task completion based on the description]
 closer to completion: [img1/img2/equal].

Since all models share the same prompt template and output space, the comparisons in Table 1 are made under uniform prompt conditions.

B Data Construction and Quality Control

B.1 Progress-Order Assumption and Scope

ECTR is built around a goal-conditioned state-comparison problem rather than a raw temporal-order prediction problem. For each robot trajectory, let

$$\mathcal{E}_j = (L_j, \{I_{j,1}, I_{j,2}, \dots, I_{j,T_j}\})$$

denote an episode with task instruction L_j and T_j observed frames. Each frame $I_{j,t}$ is treated as an observation of an underlying physical state $s_{j,t}$. We define task progress as a latent goal-conditioned scalar

$$\rho_{L_j}(s_{j,t}) \in [0, 1],$$

where a larger value means that the state is closer to completing task L_j . For two physical states $s_{j,u}$ and $s_{j,v}$, the ideal progress-comparison target is

$$z^*(s_{j,u}, s_{j,v}; L_j) = \arg \max_{s \in \{s_{j,u}, s_{j,v}\}} \rho_{L_j}(s).$$

This definition is over physical states, not over input positions. Therefore, if the same two frames are swapped between *img1* and *img2*, the physically closer state should remain unchanged, while the output slot label changes.

Directly annotating $\rho_{L_j}(s_{j,t})$ for every frame would require dense human progress scoring and

may introduce subjective noise. We therefore instantiate the progress order only on quality-controlled successful demonstrations whose execution is monotonic at the trajectory level:

$$\rho_{L_j}(s_{j,1}) \leq \rho_{L_j}(s_{j,2}) \leq \dots \leq \rho_{L_j}(s_{j,T_j}).$$

Under this monotonic-success assumption, temporal index is used as an operational proxy for progress:

$$u < v \Rightarrow \rho_{L_j}(s_{j,u}) \leq \rho_{L_j}(s_{j,v}).$$

This proxy is used only after filtering out trajectories with visible resets, failed grasps, corrective motions, or ambiguous task-state changes. It is not intended to claim that time is always equivalent to progress in arbitrary robot videos; rather, it provides a controlled construction rule for successful monotonic demonstrations.

The resulting task remains a visual state-comparison problem. Given two input images and a task instruction,

$$q = (L_j, \text{img1}, \text{img2}),$$

the model must output one of

$$y \in \{\text{img1}, \text{img2}, \text{equal}\}.$$

The first two labels indicate which input slot contains the physical state closer to completion, while *equal* indicates that the two inputs provide no task-relevant evidence of progress difference. This formulation excludes tasks whose progress cannot be inferred from the two visible states alone, such as tasks requiring hidden execution history, memory of previous identical actions, or non-visual success conditions.

B.2 Frame Downsampling and Pair Construction

This section details how raw trajectories are converted into ECTR training and evaluation instances. The construction has four steps: trajectory-level split, temporal downsampling, bidirectional dynamic-pair construction, and equal-state anchor construction.

Trajectory-level split. Before any pair is generated, the original trajectory set $\mathcal{T}$ is split into disjoint trajectory subsets:

$$\mathcal{T} = \mathcal{T}_{\text{train}} \dot{\cup} \mathcal{T}_{\text{val}} \dot{\cup} \mathcal{T}_{\text{test}}.$$

All pairs derived from the same raw trajectory remain in the same subset. This prevents leakage from adjacent frames or from forward/backward variants of the same physical event.

Ten-fold temporal downsampling. Raw robot videos contain many adjacent frames with nearly identical visual content, as well as short-term sensor jitter and occasional motion blur. We therefore apply a fixed ten-fold temporal downsampling before pair construction. For an episode

$$\mathcal{E}_j = (L_j, \{I_{j,1}, I_{j,2}, \dots, I_{j,T_j}\}),$$

we retain the subsequence

$$\tilde{\mathcal{E}}_j = (L_j, \{\tilde{I}_{j,1}, \tilde{I}_{j,2}, \dots, \tilde{I}_{j,M_j}\}),$$

where

$$\tilde{I}_{j,m} = I_{j,1+10(m-1)}, \quad m = 1, \dots, M_j,$$

and

$$M_j = \left\lfloor \frac{T_j - 1}{10} \right\rfloor + 1.$$

If the terminal frame I_{j,T_j} is not already included by the stride, we additionally keep it to preserve the final task-completion state. This downsampling reduces redundant neighboring frames while retaining the coarse state evolution needed for task-progress comparison.

Dynamic pair candidate set. From each downsampled monotonic-success trajectory, we construct temporally separated dynamic candidates. Let $1 \leq u < v \leq M_j$ index two selected downsampled frames. Under the progress-order proxy in Appendix B.1, the later state $\tilde{I}_{j,v}$ is closer to task completion than $\tilde{I}_{j,u}$. The physical answer is therefore

$$z_{j,u,v}^* = \tilde{I}_{j,v}.$$

We first form the unordered physical pair

$$p_{j,u,v} = (L_j, \tilde{I}_{j,u}, \tilde{I}_{j,v}, z_{j,u,v}^*), \quad u < v.$$

Pairs with severe blur, occlusion, ambiguous task-state change, or non-monotonic execution are removed during quality control.

Forward and backward query construction. For each retained physical pair $p_{j,u,v}$, we create two slot-ordered queries. The forward query follows the natural temporal order:

$$x_{j,u,v}^{\text{fwd}} = (L_j, \text{img1} = \tilde{I}_{j,u}, \text{img2} = \tilde{I}_{j,v}, y = \text{img2}).$$

The backward query swaps the two input slots while keeping the same physical states and task instruction:

$$x_{j,u,v}^{\text{bwd}} = (L_j, \text{img1} = \tilde{I}_{j,v}, \text{img2} = \tilde{I}_{j,u}, y = \text{img1}).$$

Thus, the physical answer remains $\tilde{I}_{j,v}$ in both queries, but the correct slot label changes from *img2* to *img1*. This construction is the core diagnostic mechanism of ECTR: a model that relies on presentation order will answer inconsistently across x^{fwd} and x^{bwd} , whereas a model that compares the two physical states should recover the same closer-to-completion state under both input orders.

Equal-state anchor construction. In addition to dynamic pairs, ECTR includes explicit no-change anchors. For a selected downsampled frame $\tilde{I}_{j,u}$, we construct

$$x_{j,u}^{\text{eq}} = (L_j, \text{img1} = \tilde{I}_{j,u}, \text{img2} = \tilde{I}_{j,u}, y = \text{equal}).$$

These anchors intentionally use identical visual evidence. They are not meant to exhaustively model all real static or near-static cases; instead, they provide a controlled zero-point in which any directional answer is necessarily a false motion prediction. This makes it possible to test whether a model defaults to a directional choice even when no task-relevant visual change is present.

Final dataset after filtering and balancing. Let $\mathcal{X}^{\text{fwd}}$, $\mathcal{X}^{\text{bwd}}$, and $\mathcal{X}^{\text{eq}}$ denote the retained forward, backward, and equal-state sets after filtering:

$$\mathcal{X}_{\text{ECTR}} = \mathcal{X}^{\text{fwd}} \cup \mathcal{X}^{\text{bwd}} \cup \mathcal{X}^{\text{eq}}.$$

We keep the three label types approximately balanced to avoid a dominant-class shortcut. The final dataset contains 8,995 examples, with 7,308 / 844 / 843 samples in the train / validation / test splits. Each example is serialized as a multimodal instruction-following instance containing two image paths, the task instruction, the fixed CoT prompt, spatial-detail supervision, task-conditioned comparison, and the final label from $\{\text{img1}, \text{img2}, \text{equal}\}$.

B.3 Interpretation of Equal-State Anchors

The equal-state construction deliberately uses exact duplicate frames. This design choice isolates a minimal and unambiguous no-change condition: when the two visual inputs are literally identical under the same task instruction, the correct answer cannot be a directional progress label. Therefore, equal-state anchors serve as a controlled diagnostic for false motion prediction.

This construction should not be interpreted as a claim that all static or weak-change real-world cases are solved by identity detection. Near-static non-identical pairs may contain camera noise, illumination changes, small object jitter, or task-irrelevant motion; these cases are harder and are left to future work. However, the main forward-backward consistency evidence in ECTR does not rely on duplicate-frame identity. Pairwise Consistency is computed only on dynamic forward/backward pairs, and a prediction of *equal* on those dynamic pairs is counted as incorrect. Therefore, improvements in Pairwise Consistency measure whether the model can identify the same closer-to-completion physical state under input-slot reversal, independently of the duplicate-frame cue used for equal-state anchors.

B.4 Annotation Protocol

Each sample is annotated along three progressive dimensions:

Level 1: Spatial description (perceptual anchoring). This level enforces fine-grained visual grounding, including precise identification of key objects (static anchors and task-relevant entities), spatial topology (detailed positional relationships such as “the liquid is on the left” or “the cup is above the center”), and robot dynamics (explicit descriptions of arm posture and gripper state).

Level 2: Comparison reasoning (logical internalization). This segment bridges perception and judgment: (i) task-goal alignment, restating the goal (e.g., “empty the water”) as the baseline; (ii) comparative state analysis, comparing *img1* and *img2* to identify spatiotemporal differences; and (iii) motion-to-goal mapping, linking physical changes to task progress.

Level 3: Completion judgment (zero-point calibration). This is the final anchor (*img1*, *img2*, or *equal*). The *equal* label serves as a “zero-point

calibration” signal to suppress false directional prediction.

B.5 GPT-5 Annotation Prompt

For reproducibility, we include the annotation prompt used to obtain the GPT-5-generated spatial descriptions and comparison reasoning. The placeholders {*task_instruction*}, {*img1*}, and {*img2*} were filled for each pair. The final evaluation label is still assigned by the construction rules in Appendix B.2; the generated text was manually reviewed as described in Appendix B.6.

```
You are an expert annotator
for robot manipulation state
comparison.
You will be given two images,
img1 and img2, and a task
instruction.
Task name: {task_instruction}
Generate a structured
annotation with three fields:
spatial_details, reason, and
closer_to_completion.
1. spatial_details:
Describe img1 and img2
separately and in detail. For
each image, describe: the
position and state of the target
object; the position of the
robot arm or end-effector; the
relative position between the
robot arm and the target object;
the gripper opening/closing
state; and other task-relevant
contact, pose, or spatial
relations. Focus only on
visible evidence.
2. reason:
First sentence: state the task
goal.
Second sentence: compare
the two images in terms of
target-object displacement
or state change, robot-arm
position, robot-target
relative position, gripper
opening/closing state, and
any other task-relevant visual
changes.
Third sentence: based on
the comparison, decide which
image is closer to task
completion. If the two images
show no task-relevant progress
difference, choose equal.
3. closer_to_completion:
Output exactly one tag from
[img1/img2/equal].
Output format:
spatial_details: [detailed
description of img1 and img2]
reason: [task goal; comparison
of task-relevant visual
differences; decision rationale]
```

```

closer_to_completion:
[img1/img2/equal]

```

B.6 Quality Control Pipeline

Five graduate students reviewed GPT-5-generated CoT text under a unified review guideline. The guideline covers two dimensions: (i) factual consistency of spatial descriptions, where reviewers check whether object positions, poses, and contact relationships described by GPT-5 match the image content; and (ii) logical coherence of comparison reasoning, where reviewers verify whether the reasoning chain can naturally support the final task-progress judgment. Reviewers corrected sentences with descriptive errors and removed samples whose reasoning was clearly invalid. In addition, because motion blur affects both visual evidence and reasoning reliability, samples with severe blur were removed during the review stage. Each sample was independently reviewed by two of the five reviewers, and disagreements were resolved through discussion to consensus.

Monotonic-success filtering. Before pair construction, trajectories were filtered to retain those whose execution is monotonic at the trajectory level, so that temporal order is a valid proxy for task progress (Appendix B.1). Trajectories containing visible resets, failed grasps, or corrective motions were excluded from dynamic-pair generation.

Label audit. To verify the temporal-order proxy, we manually audited a random sample of 200 dynamic pairs and checked whether the later state is indeed closer to task completion under the goal. The proxy held for 94% of the audited pairs; pairs that failed the check were removed or relabeled.

Table 4 summarizes the candidate, removal, correction, and audit counts across the full quality-control process.

B.7 Representative Data Instances

Each ECTR example is stored as a JSON-style multimodal instruction-following instance. The format follows the common chat-template structure used by VLM instruction tuning. Each instance contains a `messages` field, a `target` field, and a unique `sample_id`.

Input message. The `messages` field contains one user message with a mixed multimodal content list. The first two content items are images,

Category	Count
Initial candidate pairs	10,810
Removed: severe blur	627
Removed: occlusion	301
Removed: ambiguous task state	265
Removed: non-monotonic trajectory	424
Removed: invalid reasoning	198
Removed: total	1,815
Corrected: spatial description	537
Corrected: comparison reasoning	319
Corrected: both	174
Dynamic-label audit (size / valid rate)	200 / 94%
Retained samples	8,995

Table 4: Quality-control and label-audit statistics. Removal counts sum to 1,815, leaving 8,995 retained samples; the audit covers 200 dynamic pairs with a 94% valid rate. Corrected samples are revised in place and remain in the retained set.

corresponding to the two candidate states to be compared. They are explicitly marked by

$$\text{image_id} \in \{\text{img1}, \text{img2}\}.$$

The third content item is a text prompt. It specifies the model role, inserts the task instruction, and asks the model to describe both images, compare their task progress, and output one of three labels: *img1*, *img2*, or *equal*. In short, the input side contains two image states, a task instruction, and a fixed CoT-style prompt.

Target supervision. The `target` field stores the supervised response. It is divided into three aligned components, matching the three supervision levels used by ECTR.

Field	Meaning
<code>spatial details</code>	Separate visual descriptions of <i>img1</i> and <i>img2</i> , including task-relevant objects, robot-arm pose, gripper state, contact relations, object position, and local scene layout.
<code>reason</code>	Task-conditioned comparison explaining why one state is closer to completion, or why the two states show no task-relevant progress difference.
<code>closer to completion</code>	Final three-way label from $\{\text{img1}, \text{img2}, \text{equal}\}$.
<code>sample_id</code>	Unique identifier for the example.

Table 5: JSON-style fields used by each ECTR instance.

The `spatial details` field anchors perception in visible evidence, the `reason` field teaches task-conditioned state comparison, and the final label provides the decision target used for accuracy, GAP, and pairwise consistency evaluation.

Forward, backward, and equal examples. For a forward dynamic example, *img1* is the earlier state and *img2* is the later state, so the final label is typically *img2*. For the corresponding backward example, the same two physical states are swapped between the two input slots, so the correct slot label becomes *img1*. For an equal-state example, the same frame is used for both *img1* and *img2*, and the final label is *equal*. Therefore, the JSON format keeps the physical comparison task fixed while allowing the slot label to change with input order.

Schema summary. For readability, we summarize the schema below without printing full image paths or long reasoning strings:

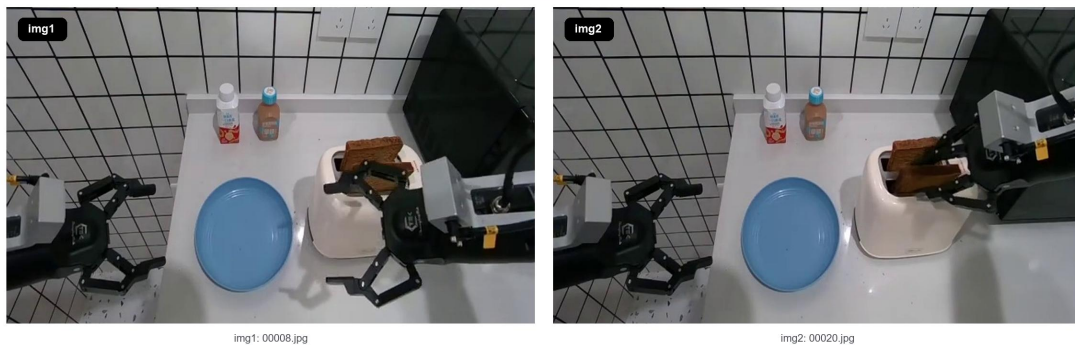
```
{
  "messages": [
    {
      "role": "user",
      "content": [
        {"type": "image",
         "image": image path,
         "image_id": "img1"},
        {"type": "image",
         "image": image path,
         "image_id": "img2"},
        {"type": "text", "text":
         fixed CoT prompt with task
         instruction}
      ]
    },
    {
      "target": {
        "spatial details":
         description of img1 and img2,
        "reason": task-conditioned
         comparison,
        "closer to completion":
         "img1" / "img2" / "equal"
      },
      "sample_id": unique example
       identifier
    }
  ]
}
```

Full serialized examples. To make the supervision format directly inspectable, we provide one full serialized example for each label type: forward, backward, and equal-state. Each example shows the two image inputs, the task-conditioned prompt, the structured spatial and reasoning supervision, and the final completion label.

Forward

natural temporal order; img2 is closer to completion

sample 00050 | target = img2



```
data.json sample
{
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "type": "image",
          "image": "testset/367/652814/Pick up the whole wheat toast from the front card slot white toaster with right arm_/00008.jpg",
          "image_id": "img1"
        },
        {
          "type": "image",
          "image": "testset/367/652814/Pick up the whole wheat toast from the front card slot white toaster with right arm_/00020.jpg",
          "image_id": "img2"
        },
        {
          "type": "text",
          "text": "You are a spatial perception expert. Please observe carefully and describe the two images (img1 and img2) separately. Task name: Pick up the whole wheat toast from the front card slot white toaster with right arm. Description points: 1. Provide detailed descriptions of img1 and img2. 2. Based on the above descriptions, analyze which image (img1 or img2) has a state that is closer to the final stage of task completion. If the states in both images are essentially the same or nearly identical, then respond with equal. 3. Provide the final answer directly. Output format: [Spatial details: [Separately describe the spatial details of both images in detail]] Reason: [Analyze why one image is closer to task completion based on the description] Closer to completion: [img1/img2/equal]."
```

Figure 3: **Full forward ECTR example.** The input follows the natural temporal order, and the later state in *img2* is closer to task completion.

Backward

input slots are reversed; img1 is closer to completion

sample 00021 | target = img1



img1: 00133.jpg

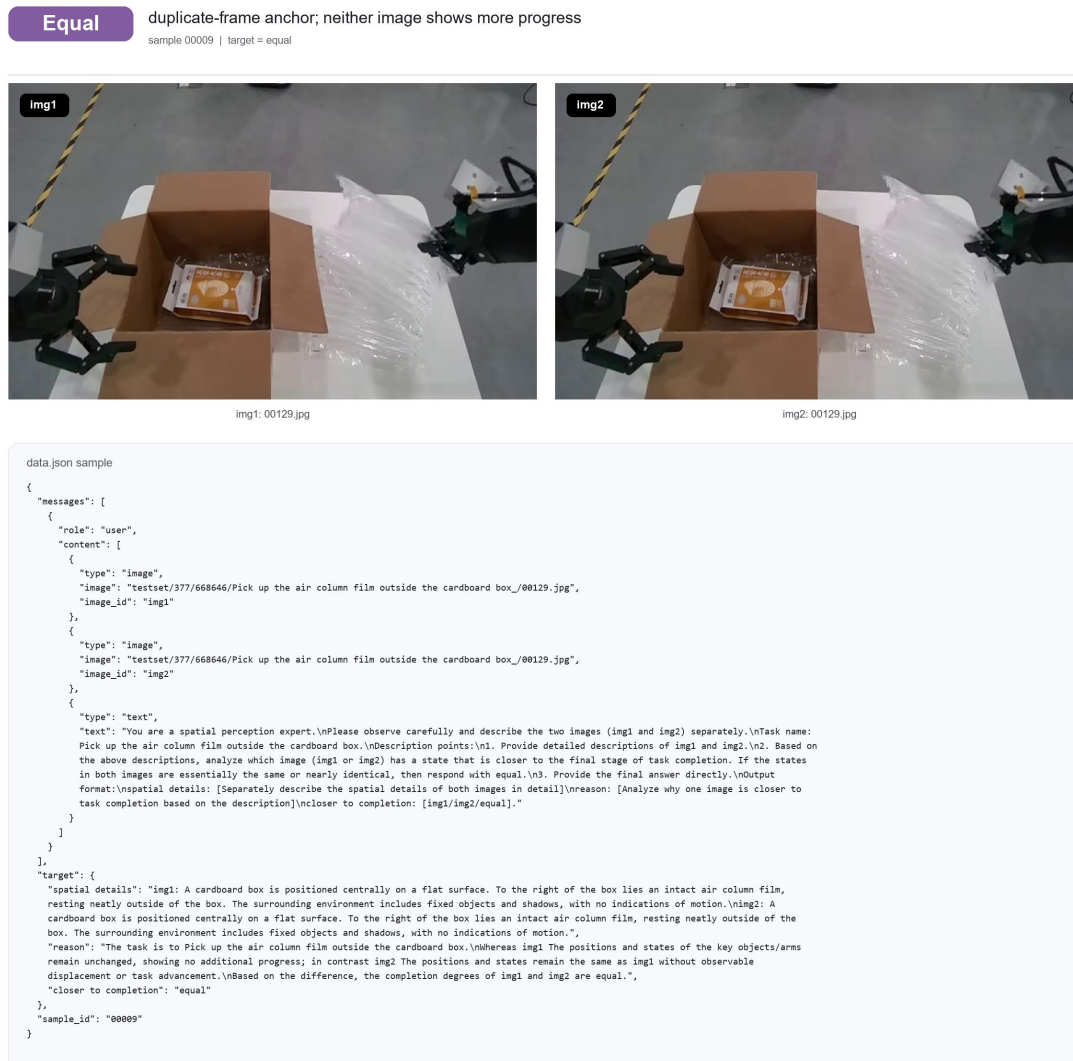


img2: 00123.jpg

data.json sample

```
{
  "messages": [
    {
      "role": "user",
      "content": [
        {
          "type": "image",
          "image": "testset/384/688220/Push the misplaced book on the shelf to the end with your right arm_/00133.jpg",
          "image_id": "img1"
        },
        {
          "type": "image",
          "image": "testset/384/688220/Push the misplaced book on the shelf to the end with your right arm_/00123.jpg",
          "image_id": "img2"
        },
        {
          "type": "text",
          "text": "You are a spatial perception expert. Please observe carefully and describe the two images (img1 and img2) separately. Task name: Push the misplaced book on the shelf to the end with your right arm. Description points: 1. Provide detailed descriptions of img1 and img2. 2. Based on the above descriptions, analyze which image (img1 or img2) has a state that is closer to the final stage of task completion. If the states in both images are essentially the same or nearly identical, then respond with equal. 3. Provide the final answer directly. Output format: [Spatial details: [Separately describe the spatial details of both images in detail] Reason: [Analyze why one image is closer to task completion based on the description] Closer to completion: [img1/img2/equal]."
```

Figure 4: **Full backward ECTR example.** The input slots are reversed relative to natural temporal order, so the physically later state appears in *img1*; the correct label is therefore *img1*.



B.8 Scene Coverage Gallery

Figure 6 provides a representative scene gallery for the AgiBot source data used by ECTR. The gallery is not a complete enumeration of all scenes. Instead, it selects 35 representative scenes from the 100+ one-to-one replicated real-life scenarios in AgiBot World (AgiBot-World-Contributors et al., 2025). We group them into five functional categories: Kitchen & Food Access, Cooking/Sink/Beverage, Tabletop Object Handling, Packing & Logistics, and Garment & Household.

The purpose of this gallery is to make the coverage of the source distribution visually inspectable. The selected scenes include food shelves, refrigerators, sinks, pouring tables, tabletop grasping, storage bins, grocery bins, carton packing, clothes hanging, garment ironing, and navigation aisles. These categories differ in object scale, workspace layout, target relation, and manipulation goal. Consequently, task-progress judgment cannot be reduced to a single environment-specific cue.

Each thumbnail is labeled with its AgiBot scene identifier and a short functional name. Forward, backward, and equal-state examples are sampled across all five functional categories, so the equal-state anchors are not restricted to a single visual domain. Table 6 summarizes the displayed subset.

Functional Category	Scenes shown
Kitchen & Food Access	7
Cooking, Sink & Beverage	7
Tabletop Object Handling	7
Packing & Logistics	7
Garment & Household	7
Total	35

Table 6: Scene counts in the representative gallery. These scenes are a displayed subset of the full AgiBot scene pool used to construct ECTR, not a replacement for the full corpus statistics.

C Complete Cross-Architecture and External-Benchmark Results

C.1 Complete Six-Backbone Cross-Architecture Generalization Results

Table 7 reports complete cross-architecture results. The goal of this experiment is twofold. First, we test whether the forward-backward imbalance

is specific to Qwen2.5-VL or appears across different VLM families. Second, we test whether the structured task-conditioned-comparison supervision transfers beyond the main backbone.

The zero-shot rows show that the imbalance is widespread but model-dependent. Some models are strongly biased toward the forward order, such as PaliGemma, InternVL-8B, LLaVA-OneVision, and Qwen2.5-VL-7B. Others have smaller GAP values but also lower overall accuracy, indicating that low GAP alone is insufficient for reliable task-progress reasoning. This is why we report three-class accuracy and overall accuracy together with GAP.

After Stage 2 task-conditioned comparison training, the five additional open-source architectures show a reduction in forward-backward imbalance. Together with the Qwen2.5-VL main-backbone result, this suggests that the perceive-compare-judge supervision format is not tied to one model family. The full TSWA pipeline is additionally evaluated on InternVL-8B and DeepSeek-VL2, where Stage 3 further improves overall accuracy and reduces GAP. These results support cross-architecture transfer of the training design, while also showing that the magnitude of Stage-3 calibration varies across backbones.

C.2 Complete MMSI-Bench Results

We use MMSI-Bench (Yang et al., 2025) as an external transfer check rather than as a direct leaderboard comparison. MMSI-Bench differs from ECTR in both task format and data distribution: it focuses on multi-image spatial intelligence rather than two-frame robot task-progress judgment. Therefore, we interpret performance on this benchmark as evidence of partial transfer of spatial-comparison ability, not as evidence that TSWA broadly solves external multi-image reasoning.

Table 8 reports the overall accuracy. The TSWA-finetuned 7B model reaches 30.10%, improving over its own Qwen2.5-VL-7B backbone by 4.20 points. At the same time, absolute accuracies on MMSI-Bench remain clustered in a narrow low-accuracy range for current VLMs. We therefore emphasize the within-backbone improvement rather than direct ranking against proprietary or much larger models.

The result is consistent with the design of TSWA: Stage 1 strengthens spatial description and object-relation grounding, while Stage 2 trains ex-

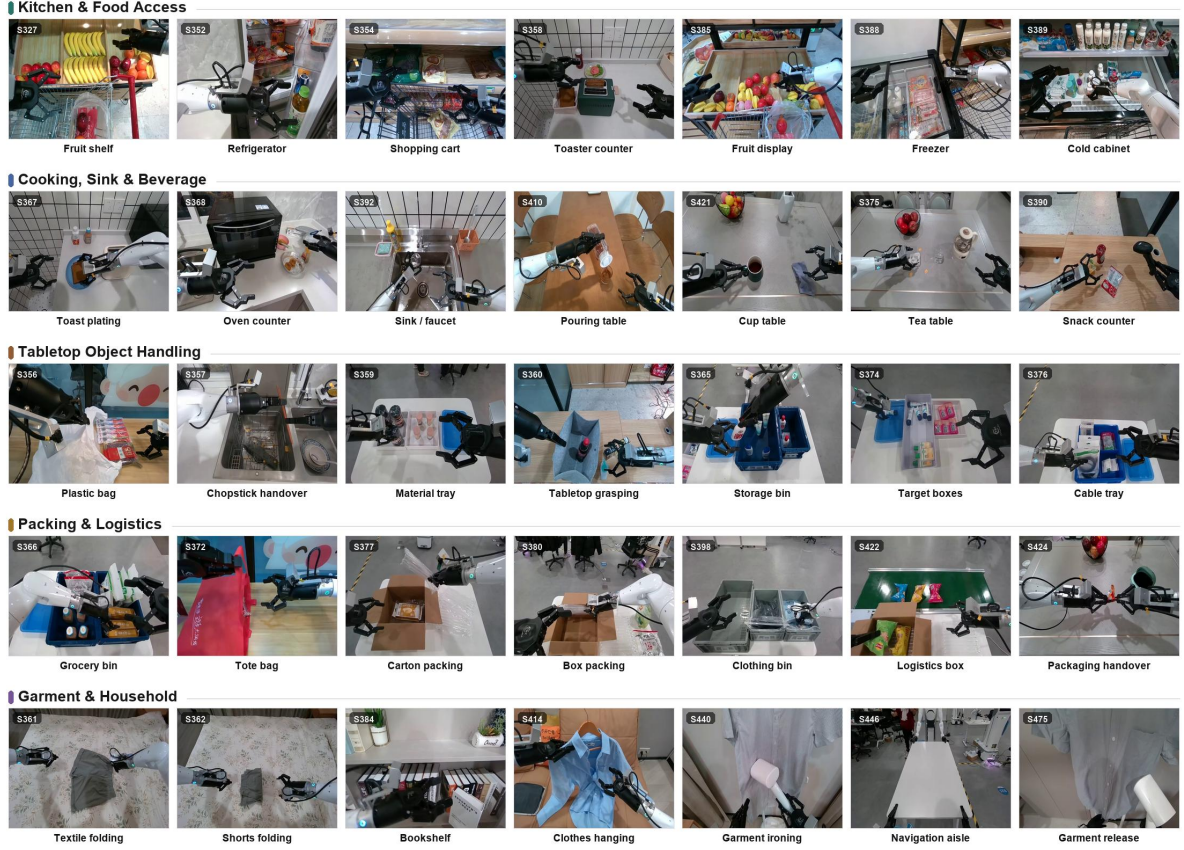


Figure 6: **Representative scene gallery.** Thirty-five representative scenes sampled from the 100+ AgiBot World scenarios used by ECTR. The gallery is grouped into five functional categories, each containing seven displayed scenes. The thumbnails illustrate diversity in objects, layouts, and manipulation goals; they are a displayed subset rather than the full scene list.

plicit cross-frame comparison. These abilities are useful beyond ECTR, but MMSI-Bench remains a challenging external benchmark with limited absolute accuracy. We view the result as supporting evidence for transfer beyond the ECTR distribution.

Observations. The main comparison is the within-backbone change from Qwen2.5-VL-7B to Qwen2.5-VL-7B+TSWA. The improvement suggests that spatial grounding and cross-image comparison learned on ECTR provide partial transfer to a different multi-image spatial benchmark. However, the absolute score remains close to the broader VLM cluster on MMSI-Bench, so we avoid interpreting this result as broad cross-benchmark superiority.

C.3 Additional Robustness and Transfer Checks

This section reports additional checks added after review. They examine whether the gains are tied to duplicated equal-state frames, a specific answer

format, a specific temporal distance between compared frames, or the ECTR evaluation distribution.

Equal-state perturbations. To test whether equal-state recognition relies on exact pixel identity, we apply task-preserving perturbations to one image in each equal-state pair while keeping the underlying task state unchanged. Table 9 shows that TSWA remains robust under brightness/contrast changes, crop-resize, Gaussian noise, and JPEG compression.

We further check whether TSWA simply predicts *equal* more often. On dynamic pairs, the correct answer is directional, so predicting *equal* is always incorrect. Table 10 shows that TSWA lowers the equal-prediction rate on dynamic pairs while improving backward accuracy and pairwise consistency.

Prompt and temporal-gap robustness. Table 11 evaluates TSWA under alternative evaluation prompts and answer-label formats. Perfor-

Model	Setting	Backward	Forward	Equal	Overall	GAP
NVILA-8B (Liu et al., 2025)	Zero-shot	58.90	41.90	50.49	50.40	17.00
	+ Stage 2	68.65	74.14	69.95	71.06	5.49
PaliGemma (Beyer et al., 2024)	Zero-shot	4.10	84.20	16.32	37.54	80.10
	+ Stage 2	55.17	78.50	60.10	65.24	23.33
InternVL-8B (Chen et al., 2024)	Zero-shot	14.90	83.30	26.64	43.77	68.40
	+ Stage 2	65.83	82.55	69.95	73.19	16.72
	+ Full TSWA	71.47	85.36	77.34	78.17	13.88
LLaVA-OneVision (Li et al., 2024a)	Zero-shot	22.80	82.10	4.90	41.07	59.30
	+ Stage 2	60.19	81.31	62.56	68.80	21.12
DeepSeek-VL2 (Wu et al., 2024)	Zero-shot	55.20	84.50	51.32	65.42	29.30
	+ Stage 2	71.47	83.80	72.41	76.39	12.33
	+ Full TSWA	73.67	84.11	76.35	78.29	10.44
Qwen2.5-VL-7B (ours) (Bai et al., 2025)	Zero-shot	23.20	83.70	16.67	44.66	60.50
	+ Full TSWA	74.92	80.06	88.67	80.19	5.14

Table 7: **Complete cross-architecture generalization results.** Overall denotes accuracy over all test examples. Stage 2 task-conditioned comparison training reduces GAP across the five additional open-source VLMs, and the Qwen2.5-VL-7B main-backbone result is shown with the full TSWA pipeline. Full TSWA is additionally applied to InternVL-8B and DeepSeek-VL2, where it further improves overall accuracy and reduces GAP.

Model	Overall Acc. (%)
Blind GPT-4o (text only)	22.70
Qwen2.5-VL-7B (backbone)	25.90
Seed1.5-VL	29.70
Claude-3.7-Sonnet	30.20
GPT-4o (OpenAI, 2024)	30.30
Qwen2.5-VL-72B (Bai et al., 2025)	30.70
Ours: 7B + TSWA	30.10

Table 8: **MMSI-Bench overall accuracy.** The TSWA-finetuned 7B model improves over its own Qwen2.5-VL-7B backbone by +4.20 points. Because MMSI-Bench accuracies for current VLMs cluster in a narrow range, we use this result as a modest external-transfer check rather than as a direct ranking claim.

mance remains stable under direct-answer prompting, A/B/Same labels, and a short comparison prompt. Table 12 groups dynamic test pairs by temporal gap. TSWA improves backward accuracy and pairwise consistency across short-, medium-, and long-gap pairs, including fine-grained short-gap comparisons.

Additional external benchmarks. In addition to MMSI-Bench, we evaluate on OmniSpatial (Jia et al., 2025) and CV-Bench from Cambrian-1 (Tong et al., 2024). Table 13 reports categories where TSWA improves over its Qwen2.5-VL backbone, together with the benchmark averages. These results provide additional evidence that the learned comparison ability transfers beyond the ECTR prompt and data distribution.

Equal-State Probe	Qwen	TSWA
No perturbation	16.67	88.67
Bright/contrast	9.36	84.23
Crop-resize	6.40	79.31
Gaussian noise	6.89	77.83
JPEG compression	9.85	81.77
Avg. perturbed	8.13	80.79

Table 9: **Task-preserving equal-state perturbations.** Values are equal-state accuracy. Perturbations are applied to one image while preserving the task state.

Model	Bwd	PC	Equal Rate
Qwen	23.20	12.23	9.40
TSWA	74.92	69.28	4.30

Table 10: **Equal-prediction rate on dynamic pairs.** Predicting *equal* is wrong for all dynamic pairs; TSWA improves directional comparison while reducing this error mode.

Benchmark	Model	Motion	Locate	Geo.Str.	Geo.Reas.	Ego	Hypo.	Avg.
OmniSpatial	Qwen	35.09	45.33	40.00	29.42	64.51	37.35	45.02
	TSWA	42.86	53.33	66.67	38.46	75.00	66.67	53.44

Table 13: **Additional OmniSpatial transfer check.** We report the six OmniSpatial categories where TSWA improves over the Qwen2.5-VL backbone, plus the overall average across all eight categories.

Evaluation Format	Acc.	GAP	PC
Original	80.19	5.14	69.28
Direct answer	78.41	5.62	65.73
A/B/Same labels	76.75	6.84	62.88
Short comparison	80.07	4.91	70.81

Table 11: **Prompt and label-format robustness of TSWA.** Acc. denotes overall accuracy and PC denotes pairwise consistency.

Gap	#Pairs	Qwen Bwd	TSWA Bwd	Qwen PC	TSWA PC
Short	107	14.95	67.29	10.28	64.49
Medium	107	31.78	81.31	14.02	72.90
Long	105	22.86	74.29	12.38	70.48

Table 12: **Temporal-gap analysis on dynamic test pairs.** Short-gap pairs are more fine-grained, while long-gap pairs often span larger state transitions.

Model	Omni3D	Depth	CV-Bench Avg.
Qwen	75.17	70.83	75.07
TSWA	76.42	79.17	75.27

Table 14: **Additional CV-Bench transfer check.** TSWA improves Omni3D, Depth, and the overall average over its Qwen2.5-VL backbone.

C.4 Training Dynamics and Ablation Trajectories

Figure 7 visualizes the ablation results in the overall-accuracy–GAP space. This view is useful because a lower GAP alone does not necessarily imply better reasoning: a model can obtain a small GAP by failing symmetrically in both directions. The ideal direction is therefore toward higher overall accuracy and lower forward–backward imbalance.

The trajectory shows three patterns. First, spatial-description training moves the model upward in overall accuracy but leaves a visible GAP, indicating that better perception alone does not fully remove order bias. Second, adding comparison reasoning moves the model sharply toward lower GAP, showing that explicit comparison logic is the main source of order-invariant progress judgment. Third, answer calibration provides a smaller but consistent refinement once the perception and comparison stages are already in place.

Table 15 reports the checkpoint-level evolution across the three training stages.

Analysis. Stage 1 reaches high forward accuracy but remains imbalanced. Stage 2 improves backward accuracy substantially and reduces GAP

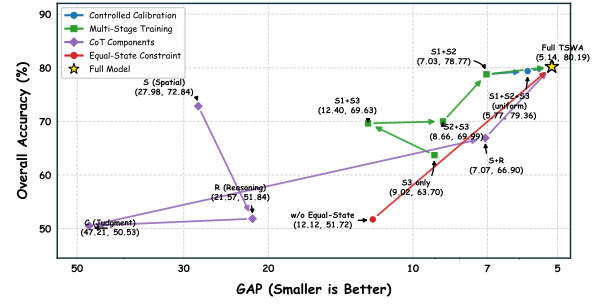


Figure 7: **Ablation trajectories in the overall-accuracy–GAP space.** The plot summarizes staged training, CoT-component ablations, and equal-state removal. The desired direction is higher overall accuracy with lower GAP.

from 27.98 to 7.03. Stage 3 then gradually improves the decision boundary: backward accuracy increases from 72.41 to 74.92, equal-state accuracy rises from 87.68 to 88.67, and GAP decreases from 7.03 to 5.14. These trends match the main claim that the primary capability gain comes from spatial grounding and task-conditioned comparison, while segment-weighted answer calibration adds a smaller final improvement.

Appendix summary. Together, the appendix details the implementation, data construction, quality control, cross-architecture transfer, external benchmark check, and training dynamics of ECTR–TSWA. The additional analyses support three conclusions: the forward–backward gap is widespread across VLM backbones; most of the TSWA gain comes from spatial grounding and task-conditioned comparison; and equal-state supervision functions as a controlled no-change anchor rather than a dominant-label shortcut.

Stage	Checkpoint	Bwd (%)	Fwd (%)	Equal (%)	Overall (%)	GAP ↓
<i>Stage 1: Spatial Description Training</i>						
S1	ckpt-600	58.31	86.60	70.44	72.00	28.30
	ckpt-800	57.68	86.92	70.94	72.00	29.24
	ckpt-1000 (final)	59.25	87.23	71.43	72.84	27.98
<i>Stage 2: Task-Conditioned Comparison Training</i>						
S2	ckpt-2200	71.16	78.50	87.19	77.82	7.34
	ckpt-2400	72.10	78.82	87.68	78.41	6.72
	ckpt-2600	71.79	79.13	87.19	78.29	7.34
	ckpt-2742 (final)	72.41	79.44	87.68	78.77	7.03
<i>Stage 3: Answer Calibration</i>						
S3	ckpt-2400	73.67	79.75	88.18	79.48	6.08
	ckpt-2600	74.29	80.06	88.67	79.95	5.77
	ckpt-2742	74.61	80.06	88.67	80.07	5.45
	ckpt-2800 (final)	74.92	80.06	88.67	80.19	5.14

Table 15: **Checkpoint-level performance evolution across training stages.** Overall denotes accuracy over all test examples. Stage 1 mainly improves spatial grounding, Stage 2 sharply reduces the forward–backward gap through task-conditioned comparison, and Stage 3 provides a smaller but consistent answer-calibration gain.