Learning to Group and Order: Cross-Instance Self-Supervised RL for Vision-Centric MLLMs

Songming Yang[1] **Hong-Tao Yu**[1] **Xiu-Shen Wei**[1,2]*
[1]School of Computer Science and Engineering, Southeast University
[2]Key Laboratory of New Generation Artificial Intelligence Technology
and Its Interdisciplinary Applications, Southeast University
{yangsm,yuht,weixs}@seu.edu.cn

Abstract

Reinforcement learning (RL) post-training has become an important paradigm for improving multimodal large language models (MLLMs), especially with automatically verifiable rewards. Self-supervised visual pretext tasks provide such rewards without human annotations. However, existing methods mainly exploit within-instance structure, offering limited supervision for cross-instance discrimination and fine-grained comparison. We introduce **G**roup-and-**O**rder **S**elf-**S**upervised Reinforcement **L**earning (GO-SSL), a cross-instance jigsaw framework for vision-centric MLLM post-training. Given two visual instances, GO-SSL mixes their local elements and trains the model to group them by instance of origin while recovering the order within each group, coupling instance-level comparison with spatial, temporal, or geometric structure recovery. This paired formulation turns the same unlabeled data into richer verifiable supervision through diverse pairings, shuffles, and a random-to-hard curriculum. We instantiate GO-SSL mainly on cross-image jigsaw tasks and further extend it to 3D depth and video temporal ordering. Across diverse benchmarks, GO-SSL consistently improves over the base MLLM and prior self-supervised post-training baselines, with clear gains in fine-grained perception and spatial understanding. Further evaluations across model scales and families support its generality. These results suggest that cross-instance comparison provides a useful self-supervised RL signal for improving transferable visual understanding in MLLMs. Code is available at https://github.com/SEU-VIPGroup/GO-SSL.

1 Introduction

Multimodal large language models (MLLMs) have made remarkable progress in visual question answering and multimodal reasoning [1], as exemplified by LLaVA-NeXT [2], InternVL [3], and Qwen2.5-VL [4]. Nevertheless, their general-purpose capabilities do not always translate into reliable vision-centric understanding. Fine-grained recognition often requires distinguishing subtle local evidence [5–8], while spatial and relational tasks demand accurate reasoning about visual structure [9, 10]. Improving these abilities calls for learning signals that directly target how models inspect and organize visual information.

Post-training is a promising way to strengthen such capabilities without redesigning the underlying MLLM. Supervised fine-tuning can improve visual behavior, but collecting and annotating task-specific examples across diverse visual domains is costly. Reinforcement learning with verifiable rewards (RLVR) [11, 12] offers another route: a model can learn from feedback when the correctness

*Corresponding author. This work was supported by National Natural Science Foundation of China under Grant (62522602, 62272231), Basic Research Program of Jiangsu under Grant (BK20250073), the Fundamental Research Funds for the Central Universities (4009002401, 2242025K30024), and the Big Data Computing Center of Southeast University.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

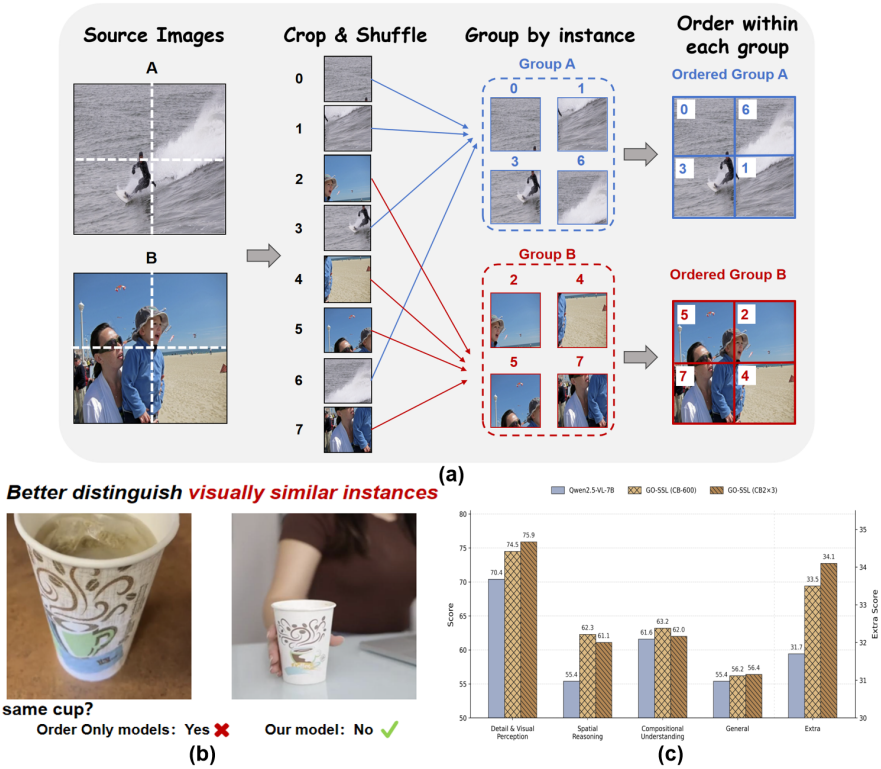


Figure 1: GO-SSL: task construction and visual transfer. **(a)** Shuffled patches from two images are grouped by source and reordered to recover each image's spatial layout. **(b)** A qualitative example of distinguishing visually similar object instances. **(c)** Category-wise benchmark averages for Qwen2.5-VL-7B and two GO-SSL variants, showing gains across all five categories.

of its output is automatically checkable. The challenge for vision-centric post-training is therefore to design tasks whose answers are verifiable while still requiring meaningful visual understanding.

Self-supervised learning (SSL) offers a natural solution to this challenge. Classical visual pretext tasks, including context prediction, rotation prediction, and jigsaw solving [13–15], derive supervision from input transformations rather than manual labels. Recent methods such as SSL4RL [16], Spatial-SSRL [17], and Visual Jigsaw [18] bring such tasks into RL post-training using verifiable rewards. However, jigsaw-style objectives largely recover spatial or temporal structure within a single source. They teach the model where an element belongs, but not which visual instance it belongs to. This leaves cross-instance discrimination without explicit supervision, even when distinguishing similar instances requires subtle local cues.

To make cross-instance comparison an explicit part of self-supervised learning, we propose *Group-and-Order Self-Supervised Reinforcement Learning* (GO-SSL). As shown in Fig. 1(a), GO-SSL interleaves local elements from two sources and asks the model to recover both their source membership and their order within each source. Grouping makes cross-instance comparison an explicit requirement, whereas ordering preserves the need to reconstruct meaningful visual structure. Both targets are known from task construction and can therefore be checked automatically. Figure 1(b) illustrates the type of fine-grained instance distinction that motivates this design.

Pairing also creates diverse comparative contexts from the same unlabeled pool. We exploit this flexibility through a random-to-hard curriculum that makes source discrimination more challenging. We study image jigsaw reconstruction as the primary setting and extend the same principle to cross-scene depth ranking and cross-video temporal ordering. Across image, 3D, and video benchmarks, GO-SSL improves upon the base MLLM and prior self-supervised post-training baselines; Fig. 1(c) summarizes the image-side gains. We additionally examine grouping and ordering ablations, model-family generalization, and transfer to targeted cross-instance discrimination tasks.

Our contributions are summarized as follows:

- We propose GO-SSL, a cross-instance self-supervised RL framework that converts unlabeled visual data into verifiable group-and-order tasks, coupling fine-grained instance discrimination with spatial, temporal, and geometric structure recovery.
- We show that pairing visual instances creates diverse comparative contexts from the same unlabeled pool, with random-to-hard curricula controlling source ambiguity and task granularity.
- We demonstrate improvements over prior self-supervised post-training baselines on image, 3D, and video benchmarks, and further evaluate generalization across model scales and families and transfer to targeted cross-instance discrimination tasks.

2 Related Work

Our work connects self-supervised RL post-training with vision-centric understanding. We review these directions with an emphasis on verifiable visual supervision.

Self-supervised RL post-training for MLLMs Self-supervised learning (SSL) derives supervision from intrinsic data structure, with classical tasks including context prediction, rotation prediction, and jigsaw solving [13–15]. Modern SSL employs masked reconstruction, contrastive learning, and self-distillation [19–24]. Fine-grained approaches further explore progressive augmentation [25] and asymmetric augmentation for image hashing [26]; related detection pretraining combines holistic-local contrastive learning with reconstruction [27]. Although effective for representation learning, these objectives are not directly aligned with MLLMs' discrete outputs.

Post-training and alignment methods, including RLHF, DPO, and RL with verifiable rewards (RLVR), have substantially improved large models [11, 12, 28, 29]. Many multimodal methods still emphasize text-centric reasoning, where images serve mainly as supporting evidence. To provide more direct visual feedback, SSL4RL constructs intrinsic verifiable rewards from self-supervised tasks [16], and Visual Jigsaw demonstrates the benefits of ordering-based supervision across images, videos, and 3D inputs [18]. These studies show how visual pretext tasks can supply automatically checkable RL targets, while existing jigsaw-style objectives mainly use within-source structural signals. GO-SSL instead makes cross-instance source discrimination explicit alongside structural ordering.

Vision-centric understanding Fine-grained recognition depends on discriminative local and global visual cues [30], yet fine-grained perception and spatial reasoning remain challenging for MLLMs. Benchmarks including ViewSpatial-Bench, Spatial457, and 3DSRBench highlight limitations in viewpoint-dependent and 3D reasoning [31–33]. Recent self-supervised methods address related capabilities through verifiable visual tasks. Spatial-SSRL [17] constructs patch reordering, flip recognition, inpainting, depth ordering, and relative 3D position tasks, while Visual Jigsaw [18] recovers structural order across different visual domains.

Other related directions include object-level correlation for few-shot segmentation [34], explanatory instructions for generalizing across vision tasks [35], global-local alignment for 3D scene completion [36], low-light video question answering [37], and image-text alignment for hyperspectral change detection [38]. These works underscore the value of local visual evidence, correspondence, and structural organization.

Unlike prior within-source jigsaw formulations, GO-SSL mixes local elements from two sources and jointly recovers their source membership and within-source order. This produces verifiable supervision that directly couples cross-instance discrimination with spatial, geometric, or temporal structure recovery. Varying the source pairs also creates comparative contexts with different levels of ambiguity from the same unlabeled visual pool.

3 Methodology

As shown in Fig. 2, Group-and-Order Self-Supervised Reinforcement Learning (GO-SSL) constructs verifiable cross-instance jigsaw tasks by mixing local elements from two visual instances and requiring MLLMs to recover both instance grouping and within-instance ordering. We instantiate GO-SSL as cross-image jigsaw, cross-scene depth ranking, and cross-video temporal ordering.

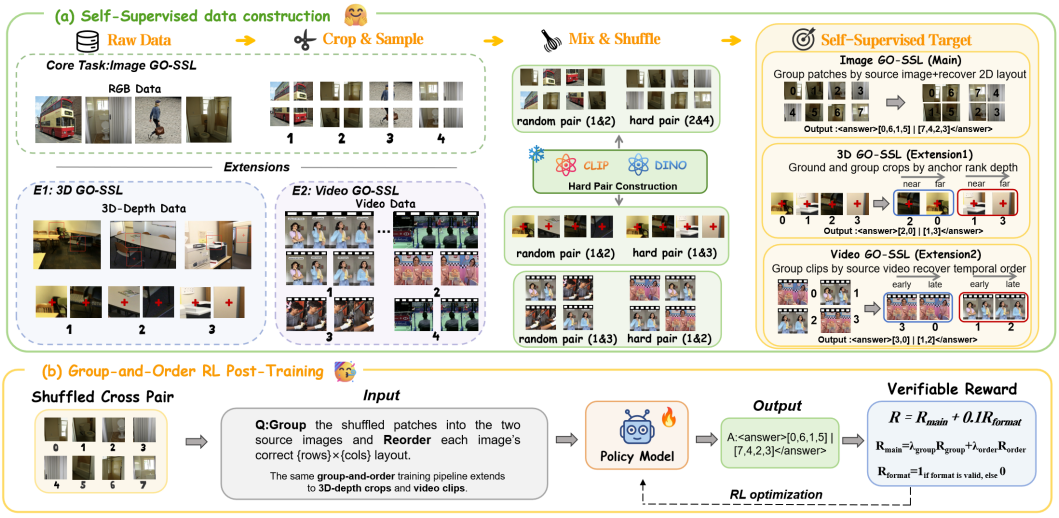


Figure 2: Overview of GO-SSL. **(a)** Self-supervised task construction: local elements from paired images, RGB-D scenes, or videos are shuffled and assigned verifiable group-and-order targets. **(b)** RL post-training: the MLLM predicts source groups and their internal spatial, depth, or temporal order, and a deterministic verifier provides rewards.

3.1 Group-and-Order Task Formulation

GO-SSL converts visual data into shuffled elements and asks the MLLM to recover their latent source structure. Given two visual sources x^A and x^B, we extract two element sets $\mathcal{E}^A = \{e_1^A, \ldots, e_K^A\}$, and $\mathcal{E}^B = \{e_1^B, \ldots, e_K^B\}$, where $\mathcal{E}^u$ denotes the K elements from source x^u, and each element can be an image patch, a local 3D crop, or a video clip. Each source also has an intrinsic ordered sequence $\bar{s}^A = (e_1^A, \ldots, e_K^A)$, and $\bar{s}^B = (e_1^B, \ldots, e_K^B)$, corresponding to spatial raster order for images, depth order from closest to farthest for 3D, and chronological order for videos.

We mix all $2K$ elements and apply a random permutation to obtain a shuffled sequence $\mathcal{Z} = (z_0, z_1, \ldots, z_{2K-1})$, where z_j denotes the element at shuffled index j. For each source $u \in \{A, B\}$, let $s^u = (j_1^u, \ldots, j_K^u)$ be its ground-truth ordered sequence of shuffled indices. Here, j_i^u is the shuffled index of its i-th element, so $z_{j_i^u} = e_i^u$. Thus, s^u restores the intrinsic order of source x^u using indices in $\mathcal{Z}$.

The model is asked to output two ordered lists of shuffled indices:

$$\hat{y} = [\hat{s}_1 \mid \hat{s}_2],$$

$$\hat{s}_1, \hat{s}_2 \in \{0, \ldots, 2K-1\}^K.$$

Here, $\hat{y}$ is the model prediction, $\hat{s}_1$ and $\hat{s}_2$ are the two predicted source groups, and $[\hat{s}_1 \mid \hat{s}_2]$ denotes concatenation with a group delimiter. A valid output must use each shuffled index exactly once:

$$\mathrm{set}(\hat{s}_1) \cap \mathrm{set}(\hat{s}_2) = \emptyset,$$

$$\mathrm{set}(\hat{s}_1) \cup \mathrm{set}(\hat{s}_2) = \{0, \ldots, 2K-1\}.$$

For image and video GO-SSL, the two output lists are unordered at the group level, since the first predicted list may correspond to either source A or source B. We therefore evaluate both source assignments during reward computation and keep the better one, making the reward invariant to arbitrary group naming while still requiring source recovery and within-group ordering. In contrast, 3D GO-SSL uses explicit anchors to define group identities, so group-level swapping is not applied.

We use *answer-only* outputs, requiring the model to directly predict two ordered index lists inside `<answer></answer>` tags. These compact, permutation-like targets can be deterministically verified. Appendix B provides a comparison between direct-answer and chain-of-thought prompting.

Supervision multiplicity We quantify the source-level contexts available from a fixed pool $\mathcal{D} = \{x_i\}_{i=1}^n$. For a pretext constructor c, let ρ_c be the number of distinct sources in a training context.

The source-context multiplicity before stochastic transformations is

$$\mathrm{SCM}(\mathcal{C}; n) = \sum_{c \in \mathcal{C}} \binom{n}{\rho_c}. \tag{1}$$

Single-source jigsaw supplies n contexts (or approximately $T \cdot n$ for T task types), whereas pairwise GO-SSL permits $\binom{n}{2} = O(n^2)$ source pairs. We sample a finite subset with the random-to-hard curriculum; shuffling and patch granularity add further target diversity. Training recipes are compared in Appendix D.1.

The combinatorial count is a property of the task construction, not a measured sample-complexity or compute-efficiency guarantee; a fixed training budget uses only a subset of the available pairs.

3.2 Task Instantiations

The formulation above is modality-agnostic. The three GO-SSL tasks share the same general formulation, but differ in their source elements, within-source ordering criteria, and group-assignment rules. For each source $u \in \{A, B\}$, let $\mathcal{I}_u$ denote its shuffled index set. We use $\mathrm{argsort}_{j \in \mathcal{I}_u}$ to denote the sequence of shuffled indices sorted in ascending order by the specified criterion.

Cross-image group-and-order Our primary task is cross-image jigsaw. Given two images x^A and x^B, we first center-crop each image into a square and divide it into an $m \times n$ grid, yielding $K = m \cdot n$ patches per image. The patch order follows row-major order, from left to right and top to bottom. Let p_i^u denote the i-th patch of image x^u in this order. After the $2K$ patches from both images are shuffled into $\mathcal{Z}$, the ground-truth target is

$$y_{\mathrm{img}} = [s^A_{\mathrm{img}} \mid s^B_{\mathrm{img}}],$$
$$s^u_{\mathrm{img}} = \mathrm{argsort}^{\mathrm{lex}}_{j \in \mathcal{I}_u} \big(r(z_j), c(z_j)\big).$$

Here, $r(z_j)$ and $c(z_j)$ denote the original row and column positions of patch z_j, respectively. Lexicographic sorting recovers the row-major order of each source image, yielding s^u_{img}. The two groups in y_{img} are evaluated up to group-level swap invariance, as defined in the general formulation.

Unlike single-image jigsaw, ours requires the model to separate visually mixed sources before reconstructing each layout, encouraging cross-image discrimination beyond local continuity.

3D depth group-and-order We extend GO-SSL to RGB-D scenes. Each sample contains two anchor images x^A and x^B, and $2K$ shuffled local crops. For each anchor x^u, we select K marked regions with valid depth values and crop local patches around them. Let c_i^u denote the i-th crop after sorting the crops of anchor x^u from closest to farthest, *i.e.*, $d(c_1^u) < d(c_2^u) < \cdots < d(c_K^u)$, where $d(c_i^u)$ is the depth of the marked center point in crop c_i^u. After all crops are shuffled into $\mathcal{Z}$, the ground-truth target is

$$y_{\mathrm{3D}} = [s^A_{\mathrm{3D}} \mid s^B_{\mathrm{3D}}],$$
$$s^u_{\mathrm{3D}} = \mathrm{argsort}_{j \in \mathcal{I}_u} d(z_j).$$

Here, $d(z_j)$ denotes the depth of the marked center point in crop z_j. Sorting by depth yields s^u_{3D}, which contains crops from anchor x^u ordered from closest to farthest. The two groups in y_{3D} correspond to anchors A and B, respectively.

The model is not asked to compare absolute depth values across two different images; instead, the task combines cross-scene crop grounding with per-scene ordinal depth reasoning.

Video group-and-order We further instantiate GO-SSL for video understanding. Given two videos v^A and v^B, we sample K clips from each video in chronological order. Let q_i^u denote the i-th clip of video v^u in this order. After all $2K$ clips are shuffled into $\mathcal{Z}$, the ground-truth target is

$$y_{\mathrm{vid}} = [s^A_{\mathrm{vid}} \mid s^B_{\mathrm{vid}}],$$
$$s^u_{\mathrm{vid}} = \mathrm{argsort}_{j \in \mathcal{I}_u} \tau(z_j).$$

Here, $\tau(z_j)$ denotes the original temporal position of clip z_j within its source video. Sorting by this position yields s^u_{vid}, which recovers the chronological order of clips from video v^u. The two groups in y_{vid} are evaluated up to group-level swap invariance.

Compared with standard video ordering, cross-video G&O prevents the model from solving the task only through local temporal continuity. Because clips from two different videos are mixed together, it must simultaneously capture source-level coherence and temporal progression, which is expected to strengthen temporal reasoning, event continuity modeling, and broader video comprehension.

3.3 Reward Design

GO-SSL uses deterministic rewards because source identity and within-source order are known by construction. The total reward is

$$r = r_{\text{fmt}} + r_{\text{main}}, \tag{2}$$

where $r_{\text{main}} \leq 1$ measures group-and-order correctness, and $r_{\text{fmt}} = 0.1$ only when the output follows the required `<answer></answer>` format. Exact predictions receive full reward, while valid imperfect predictions receive discounted partial credit. The reward expressions in Fig. 2(b) schematically illustrate the component-level scores rather than the final training reward. The actual reward incorporates format scaling, group-assignment invariance, and the validity and partial-credit rules specified below.

Grouping and ordering reward The main reward evaluates source grouping and within-source ordering. For image and video GO-SSL, the two predicted groups are unlabeled, so we consider both mappings from predicted groups to ground-truth sources. Let $m : \{1, 2\} \to \{A, B\}$ be a bijection, where $m(g)$ denotes the source assigned to predicted group g. For a mapping m, the grouping score is

$$R_{\text{grp}}^{(m)} = \frac{1}{2K} \sum_{g=1}^{2} \sum_{i=1}^{K} \mathbf{1}[\text{src}(\hat{s}_{g,i}) = m(g)], \tag{3}$$

where $\hat{s}_{g,i}$ is the i-th shuffled index in predicted group $\hat{s}_g$, and $\text{src}(j) \in \{A, B\}$ is the ground-truth source identity of shuffled element z_j.

The ordering score measures position accuracy within each matched source:

$$\text{Acc}(a, b) = \frac{1}{K} \sum_{i=1}^{K} \mathbf{1}[a_i = b_i], \tag{4}$$

$$R_{\text{ord}}^{(m)} = \frac{1}{2} \sum_{g=1}^{2} \text{Acc}\left(\hat{s}_g, s^{m(g)}\right), \tag{5}$$

where s^A and s^B are the ground-truth ordered shuffled-index sequences, and a_i, b_i denote the i-th entries of two length-K index lists. We combine grouping and ordering as

$$R^{(m)} = \lambda_{\text{grp}} R_{\text{grp}}^{(m)} + \lambda_{\text{ord}} R_{\text{ord}}^{(m)}, \tag{6}$$

$$\tilde{R} = \max_{m \in \mathcal{M}} R^{(m)}, \tag{7}$$

where $\mathcal{M}$ contains the two mappings for image/video GO-SSL and the fixed anchor mapping for 3D GO-SSL; $\tilde{R}$ is the resulting best score.

Validity and partial credit Invalid outputs, including repeated, missing, out-of-range, or incorrectly sized indices, receive zero task reward. Exact correctness is evaluated under the same group-swap invariance as in Eq. (7). For valid predictions, we set $r_{\text{main}} = 1$ if the prediction is exactly correct, and otherwise assign a discounted partial reward $r_{\text{main}} = \gamma \tilde{R}$, where $\gamma \in (0, 1)$. This discount ensures that exact recovery is always preferred while still providing dense learning signals for difficult cross-instance puzzles.

3.4 Curriculum Learning Strategy

The difficulty of GO-SSL largely depends on how the two sources are paired. Unrelated sources make grouping relatively easy, while semantically similar sources or those sharing fine-grained visual patterns create more ambiguous and informative puzzles. We therefore adopt a random-to-hard curriculum, mainly instantiated in the image setting.

Random pairs In the first stage, two sources are sampled independently from the training set: $x^A, x^B \sim p_{\text{data}}(x)$ and $x^A \neq x^B$. It provides a stable warm-up, allowing models to learn the answer format, grouping behavior, and basic ordering mechanics before facing highly ambiguous examples.

Hard pairs After the model learns the task structure, we increase the difficulty with hard visual pairs. We consider two variants. **CLIP-Hard (CH)** pairs are retrieved as nearest neighbors in CLIP feature space by $\mathcal{N}_{\text{CLIP}}(x) = \text{TopK}_{x' \neq x} \text{sim}\big(\phi_{\text{CLIP}}(x), \phi_{\text{CLIP}}(x')\big)$, where $\phi_{\text{CLIP}}(\cdot)$ denotes CLIP image features and $\text{sim}(\cdot, \cdot)$ is cosine similarity. These pairs are semantically similar and thus make cross-image grouping more ambiguous.

Our main variant is **CLIP-Blind (CB)** pairing, inspired by the CLIP-blind image pairs introduced by Tong et al. [5]. We first retrieve CLIP-near candidates and then re-rank them with DINO to favor pairs that are close under CLIP but relatively different under DINO:

$$\text{score}(x, x') = \text{sim}_{\text{CLIP}}(x, x') - \lambda \, \text{sim}_{\text{DINO}}(x, x'). \tag{8}$$

Here, sim_{CLIP} and sim_{DINO} denote cosine similarities computed from CLIP and DINO features, respectively. A high score indicates that two images are semantically related but differ in finer visual structure, encouraging patch-level discrimination beyond coarse semantic cues.

Curriculum schedule The curriculum gradually shifts from random to hard pairs. For higher-complexity variants, we further increase the number of elements per source, *e.g.*, from 2×2 to 2×3 image patches.

3.5 RL Optimization

GO-SSL employs a GRPO-inspired group-relative policy objective [11]. For each prompt, the policy samples a group of candidate answers, each scored by our deterministic verifier. The resulting rewards are used to construct within-group relative advantages, encouraging higher-reward answers over lower-reward ones for the same visual puzzle.

Following the RL post-training setup of Visual Jigsaw [18], we omit explicit KL penalties and entropy regularization. In GO-SSL, each valid response must contain two ordered lists that use every shuffled index exactly once. Our deterministic verifier checks output validity and computes task rewards based on grouping and ordering correctness, while invalid permutations receive zero task reward. We use this no-KL/no-entropy setting throughout our GO-SSL experiments.

4 Experiments

Our evaluation first focuses on image-centric benchmarks, followed by extensions to 3D and video. Additional ablation studies examine the contributions of grouping, ordering, and source-pair construction.

4.1 Experimental Setup

We use Qwen2.5-VL-7B-Instruct [4] as the base MLLM and post-train it with LoRA-based RL on our answer-only GO-SSL tasks. For image GO-SSL, training samples are constructed from the COCO 2014 training split [39], where two source images are paired, their local patches are shuffled together, and the model is rewarded for recovering both source groups and within-source spatial order. We further instantiate the same group-and-order formulation on ScanNet [40] for 3D depth ordering and LLaVA-Video [41] for temporal video ordering. To increase task difficulty, we adopt a random-to-hard curriculum, where random image pairs are followed by CLIP-Hard or CLIP-Blind pairs mined with CLIP [42] and DINOv2 [43] features.

Our evaluation primarily uses VLMEvalKit [44] and follows official evaluation protocols whenever available. For fair comparison, all models are evaluated using the default settings of VLMEvalKit or the official evaluation protocols adopted in prior work. We also evaluate the released 7B checkpoints of Spatial-SSRL [17] and Visual Jigsaw [18] under the same pipeline. Key hyperparameters are chosen based on small validation sweeps on held-out data. Key training settings, hardware configuration, hard-pair construction, and benchmark descriptions are provided in Appendices C and D.

Table 1: Image-centric benchmark results. CB-600 uses 400 random 2×2 steps and 600 CLIP-Blind 2×2 steps; CB-2×3 replaces the last 200 steps with 2×3 patches. Avg. is the unweighted mean across all 14 benchmark metrics; bold denotes the best score.

Method	Detail Underst. & Vis. Percept.				Spatial Underst. & Reason.					Compos. Underst.	General		Extra		Avg.
	HRBench4K	HRBench8K	V*	MMStar	VSR	SpatialEval	CVBench2D	DA-2K	OmniSpatial	SugarCrepe++	SeedBench	NaturalBench	Hallusion	EMMA	
	test	test	test	cp	test	maze	test	test	test	test	test	G-acc	f-acc	code	
Qwen2.5-VL-7B	68.50	65.38	76.96	70.80	77.31	27.60	74.99	54.45	42.66	61.59	76.95	33.89	36.99	26.42	56.75
Spatial-SSRL [17]	70.00	69.88	77.49	72.40	79.36	36.40	76.10	59.28	44.29	62.94	76.78	34.16	38.15	29.26	59.04
Visual Jigsaw [18]	71.75	71.13	80.63	72.80	80.36	30.53	75.12	60.35	44.49	63.02	76.95	35.21	36.99	28.55	59.13
Image GO-SSL (CB-600)	71.50	69.88	81.68	74.80	**80.42**	**46.67**	76.74	**61.70**	**45.79**	**63.17**	**77.14**	35.26	**38.73**	28.36	60.85
Image GO-SSL (CB-2×3)	**74.00**	**72.37**	**82.20**	**75.20**	79.68	42.60	**76.82**	61.36	45.27	61.99	**77.14**	**35.63**	38.15	**30.14**	**60.90**
(Gain)	+5.50	+6.99	+5.24	+4.40	+2.37	+15.00	+1.83	+6.91	+2.61	+0.40	+0.19	+1.74	+1.16	+3.72	**+4.15**

4.2 Main Results on Image Cross Jigsaw

We first evaluate GO-SSL on image-centric multimodal benchmarks. Table 1 compares the base Qwen2.5-VL model, prior self-supervised post-training baselines, and our cross-image G&O variants.

Overall, GO-SSL achieves the strongest average performance among all compared models. Image GO-SSL (CB-2 $\times$ 3) reaches an average score of 60.90, improving the base model by 4.15 points and outperforming Spatial-SSRL and Visual Jigsaw by 1.86 and 1.77 points, respectively. The simpler Image GO-SSL (CB-600) variant also achieves a strong average score of 60.85, showing that the gains are not tied to a single curriculum setting. The improvements are broad but most visible on representative perception and spatial benchmarks. For example, Image GO-SSL (CB-2 $\times$ 3) improves performance on HRBench4K/8K [45] and V* [6], suggesting stronger detail-oriented visual perception. It also brings clear gains on SpatialEval-maze [46] and DA-2K [47], indicating that cross-image grouping and ordering benefit spatial and geometric reasoning. The gains on SugarCrepe++ [48] and Hallusion [49] further suggest that the learned visual grounding transfers beyond the training puzzle itself.

Compared with Visual Jigsaw, GO-SSL extends single-source ordering to cross-source group-and-order recovery, making cross-instance discrimination an explicit optimization target. Unlike Spatial-SSRL, which combines heterogeneous spatial pretext tasks, GO-SSL uses a unified objective that couples source discrimination with structural recovery.

We also note that GO-SSL uses a moderate image-side RL budget. Spatial-SSRL constructs 81,053 self-supervised QA pairs with an additional SFT warm-up, and Visual Jigsaw trains image jigsaw for 1000 steps with global batch size 256. By contrast, our image GO-SSL uses 1000 steps with global batch size 64, while extracting richer supervision from paired sources through joint grouping and ordering targets. More details are provided in Appendix D.1.

The gains also extend to Qwen2.5-VL-3B and LLaVA-OneVision-7B [50] (Appendix E.1). Targeted cross-instance evaluations further test object comparison, grouping, attribute/state discrimination, and local correspondence (Appendix G).

4.3 Cross-Modal Extensions to 3D and Video

Although our main focus is cross-image G&O, the same formulation can be applied whenever visual elements have both source identity and internal structure. We instantiate this idea on 3D RGB-D crops, where ordering is defined by depth, and on video clips, where ordering is defined by time.

Table 2 shows that 3D GO-SSL improves the average score by 8.04 points over the base model and outperforms both Spatial-SSRL and 3D Visual Jigsaw. The largest gains appear on DA-2K [47] and BLINK [51], suggesting stronger relative depth perception and 3D spatial grounding. Improvements

Table 2: Accuracy on 3D benchmarks. GO-SSL applies grouping and ordering to cross-scene depth.

Model	Omni3DBench	3DSRBench	BLINK	Static-EmbodiedBench	OmniSpatial	DA-2K	Avg.
	count	test	rd-acc	test	test	test	
Qwen2.5-VL-7B	17.14	57.42	77.42	46.50	42.66	54.45	49.27
Spatial-SSRL [17]	20.00	58.21	83.87	48.00	44.29	59.28	52.28
Visual Jigsaw(3D) [18]	5.71	58.13	82.26	47.00	45.99	71.56	51.78
3D GO-SSL	**24.29**	**58.35**	**91.13**	**49.00**	**46.18**	**74.90**	**57.31**
(Gain)	+7.15	+0.93	+13.71	+2.50	+3.52	+20.45	**+8.04**

Table 3: Evaluation results on video benchmarks under the 16-frame setting. Video GO-SSL extends group-and-order learning to cross-video temporal ordering and improves temporal/video understanding over baselines.

Model	AoTBench	TUNABench	VideoMME	TempCompass		VSI-Bench	Video-TT	Avg.
	vqa	test	wo subs	mc	yorn	test	test	
Qwen2.5-VL-7B	45.52	53.14	57.44	71.77	73.46	32.79	36.90	53.00
Visual Jigsaw(Video) [18]	51.67	55.10	58.07	73.10	74.68	35.39	38.90	55.27
Video GO-SSL	**55.75**	**55.45**	**58.84**	**73.29**	**76.20**	**35.55**	**39.30**	**56.34**
(Gain)	+10.23	+2.31	+1.40	+1.52	+2.74	+2.76	+2.40	**+3.34**

on Omni3DBench [52] and OmniSpatial [10] further indicate that the learned depth-aware grouping and ordering signal transfers beyond the direct depth-ranking task.

Table 3 shows that Video GO-SSL improves the average score by 3.34 points over the base model and 1.07 points over Video Visual Jigsaw. The strongest gain appears on AoTBench [53], which directly stresses temporal order and event progression. Gains on TempCompass [54] and Video-TT [55] further suggest better temporal reasoning and broader video comprehension.

The strongest empirical evidence in this paper remains the cross-image setting, while the 3D and video experiments study the generality of the group-and-order formulation. We keep the main text concise and provide additional 3D/video ablations, including order-only and Group+Order variants, in Appendix F.2, and video frame-number ablations in Appendix F.3.

4.4 Ablation Studies

This section examines the key design choices of GO-SSL through ablation studies. Table 4 lists each trained variant once, with the random G&O model used as the anchor for objective and pair-construction comparisons, and CLIP-Blind-600 for scale, granularity, and decoy comparisons. More ablation studies are provided in Appendix F.

Objective design We first examine the individual contributions of cross-instance grouping and spatial ordering. The group-only variant mixes patches from two source images and predicts their source membership without within-source ordering supervision. In contrast, the order-only variant performs single-image jigsaw reconstruction, recovering patch order without cross-instance grouping. Both variants improve over the Qwen2.5-VL baseline, showing that source discrimination and structural ordering provide useful self-supervised signals. The full G&O objective achieves a higher overall average, suggesting that the two components are complementary: grouping encourages cross-image comparison, while ordering strengthens local-to-global spatial reconstruction.

Hard-pair construction The next analysis investigates how paired-image selection affects training. A key advantage of GO-SSL's cross-instance formulation is that it naturally supports hard-pair construction: the difficulty of the grouping problem can be controlled simply by changing which two source images are paired. Compared with random pairing, CLIP-Hard improves spatial understanding, indicating that semantically similar pairs provide a stronger signal than random pairs. Under the same training budget, CLIP-Blind achieves better overall performance than CLIP-Hard. This is likely because CLIP-nearest pairs are mainly similar in global semantic or language-aligned space, while their local patch-level differences may still be easy to separate. CLIP-Blind pairs, which are close in

Table 4: Ablation studies of GO-SSL. Each trained variant is listed once. "D&V", "Spatial", "Comp.", "Gen.", and "Ext." denote detail understanding and visual perception, spatial understanding and reasoning, compositional understanding, general multimodal benchmarks, and extra benchmarks, respectively. Avg. is computed over the same 14 benchmark metrics as Table 1. Bold indicates the best result within each axis group.

Axis	Variant	Training setting	D&V	Spatial	Comp.	Gen.	Ext.	Avg.
Baseline	Qwen2.5-VL	–	70.41	55.40	61.59	55.42	31.71	56.75
	Spatial-SSRL	–	72.44	**59.09**	62.94	55.47	**33.71**	59.04
	Visual Jigsaw	–	**74.08**	58.17	**63.02**	**56.08**	32.77	**59.13**
Objective	Group only	Two images, 2×2; grouping only	71.08	55.66	62.12	55.52	32.07	57.14
	Order only	Single image, 2×2; ordering only	71.63	58.17	**62.74**	55.61	33.37	58.43
	G&O-Random†	Random pairs, 2×2; group + order	**74.88**	**59.83**	62.56	**56.39**	**34.34**	**60.19**
Pair / curriculum	CLIP-Hard-600	400 R + 600 CH (2×2)	74.64	61.01	62.12	**56.56**	33.58	60.43
	CLIP-Blind-400	400 R + 400 CB (2×2)	74.03	61.14	62.83	56.22	**34.22**	60.40
	CLIP-Blind-600‡	400 R + 600 CB (2×2)	74.47	**62.26**	**63.17**	56.20	33.54	60.85
	CLIP-Blind-2×3	400 R + 400 CB (2×2) + 200 CB (2×3)	**75.94**	61.15	61.99	56.39	34.15	**60.90**
Candidate pool	GO-SSL-Decoy-Explicit	CB candidates + 2 CB decoys; explicit R_{sel}	73.98	60.65	**62.83**	**56.50**	**34.70**	60.31
	GO-SSL-Decoy-Implicit	CB candidates + 2 CB decoys; implicit via $R_{\text{grp}} + R_{\text{ord}}$	**74.24**	**61.23**	62.71	56.29	34.04	**60.46**

R = random pairing; CH = CLIP-Hard; CB = CLIP-Blind. † G&O-Random is the shared anchor for objective and pair-construction comparisons. ‡ CLIP-Blind-600 is the shared anchor for curriculum scale, granularity, and decoy comparisons.

CLIP space but separated in DINO space, create a more informative ambiguity: the two images are semantically related, but their local visual structures require careful comparison.

Curriculum scale and granularity We further compare different hard-pair training lengths and jigsaw granularities. Increasing the CLIP-Blind stage from 400 to 600 steps improves the average performance, suggesting that the model continues to benefit from challenging cross-instance comparisons. Using a denser 2×3 layout yields the best detail-understanding and visual-perception score, showing that finer granularity can strengthen local detail sensitivity. However, it does not uniformly improve all categories, indicating a trade-off between richer structural supervision and higher optimization difficulty.

Decoy candidate expansion Finally, we investigate whether adding visually similar distractor patches can further strengthen GO-SSL. Both decoy variants add two CLIP-Blind distractor patches to the candidate pool. *GO-SSL-Decoy-Explicit* uses an additional selection reward R_{sel} to identify true patches, whereas *GO-SSL-Decoy-Implicit* relies only on R_{grp} and R_{ord}, so distractors are penalized through grouping or ordering errors. Compared with CLIP-Blind-600, decoy variants improve extra benchmark categories but not the overall average, suggesting that distractors may improve robustness to irrelevant candidates while increasing optimization difficulty. Thus, the simpler CLIP-Blind G&O formulation already captures most benefits with less reward-design complexity, cf. results in Appendix F.7.

5 Conclusion

In this work, we introduced GO-SSL, a self-supervised RL post-training framework that improves MLLMs via cross-instance group-and-order reasoning. By grouping visual elements from different sources and recovering their internal order, GO-SSL provides a verifiable, annotation-free signal that promotes fine-grained perception, cross-instance discrimination, and spatial understanding. Experiments on image benchmarks show consistent gains over baselines and prior self-supervised post-training methods, while extensions to 3D and video demonstrate the formulation's generality. These results suggest that effective self-supervised RL for MLLMs should design verifiable pretext tasks structurally aligned with transferable visual capabilities. Future work will improve GO-SSL's efficiency and study how the G&O principle scales to larger MLLMs and specialized visual domains.

References

[1] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *National Science Review*, 11(12):nwae403, 2024.

[2] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge. `https://llava-vl.github.io/blog/2024-01-30-llava-next/`, 2024.

[3] Zhe Chen, Jiannan Wu, Wenhai 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. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 24185–24198, 2024.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo 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, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9568–9578, 2024.

[6] Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 13084–13094, 2024.

[7] Hongtao Yu, Yuxin Peng, Serge Belongie, and Xiu-Shen Wei. Benchmarking large vision-language models on fine-grained image tasks: A comprehensive evaluation. In *Proc. Int. Conf. Learn. Representations*, 2026. URL `https://openreview.net/forum?id=cVc74MLspe`.

[8] Yuxin Peng, Zishuo Wang, Geng Li, Xiangtian Zheng, Sibo Yin, and Hulingxiao He. A survey on fine-grained multimodal large language models. *Chinese Journal of Electronics*, 35(2): 771–803, 2026. doi: 10.23919/cje.2025.00.336.

[9] Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023.

[10] Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. OmniSpatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*, 2025.

[11] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[12] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[13] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 1422–1430, 2015.

[14] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[15] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. Eur. Conf. Comp. Vis.*, pages 69–84, 2016.

[16] Xiaojun Guo, Runyu Zhou, Yifei Wang, Qi Zhang, Chenheng Zhang, Stefanie Jegelka, Xiaohan Wang, Jiajun Chai, Guojun Yin, Wei Lin, and Yisen Wang. SSL4RL: Revisiting self-supervised learning as intrinsic reward for visual-language reasoning. *arXiv preprint arXiv:2510.16416*, 2025.

[17] Yuhong Liu, Beichen Zhang, Yuhang Zang, Yuhang Cao, Long Xing, Xiaoyi Dong, Haodong Duan, Dahua Lin, and Jiaqi Wang. Spatial-SSRL: Enhancing spatial understanding via self-supervised reinforcement learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2026.

[18] Penghao Wu, Yushan Zhang, Haiwen Diao, Bo Li, Lewei Lu, and Ziwei Liu. Visual jigsaw post-training improves MLLMs. In *Proc. Int. Conf. Learn. Representations*, 2026. URL `https://openreview.net/forum?id=tBf2SUzfZw`.

[19] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[20] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 16000–16009, 2022.

[21] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. Int. Conf. Mach. Learn.*, pages 1597–1607, 2020.

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 9729–9738, 2020.

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Inf. Process. Syst.*, pages 21271–21284, 2020.

[24] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 9630–9640, 2021.

[25] Baofeng Tan, Xiu-Shen Wei, and Lin Zhao. Prototype-based contrastive learning with stage-wise progressive augmentation for self-supervised fine-grained learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4125–4134, 2025.

[26] Feiran Hu, Chenlin Zhang, Jiangliang Guo, Xiu-Shen Wei, Lin Zhao, Anqi Xu, and Lingyan Gao. An asymmetric augmented self-supervised learning method for unsupervised fine-grained image hashing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17648–17657, 2024.

[27] Chen-Long Duan, Yong Li, Xiu-Shen Wei, and Lin Zhao. Long-tailed object detection pre-training: Dynamic rebalancing contrastive learning with dual reconstruction. In *Advances in Neural Information Processing Systems*, volume 37, pages 40230–40253, 2024. doi: 10.52202/079017-1272.

[28] Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Inf. Process. Syst.*, pages 27730–27744, 2022.

[29] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Inf. Process. Syst.*, pages 53728–53741, 2023.

[30] Songming Yang, Jing Wen, and Bin Fang. FG-MoE: Heterogeneous mixture of experts model for fine-grained visual classification. *Pattern Recognition*, 175:113050, 2026. doi: 10.1016/j.patcog.2026.113050.

[31] Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. ViewSpatial-Bench: Evaluating multi-perspective spatial localization in vision-language models. *arXiv preprint arXiv:2505.21500*, 2025.

[32] Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M. de Melo, Jieneng Chen, and Alan Yuille. Spatial457: A diagnostic benchmark for 6D spatial reasoning of large multimodal models. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 24669–24679, 2025.

[33] Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso De Melo, and Alan Yuille. 3DSRBench: A comprehensive 3D spatial reasoning benchmark. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 6924–6934, 2025.

[34] Chunlin Wen, Yu Zhang, Jie Fan, Hongyuan Zhu, Xiu-Shen Wei, Yijun Wang, Zhiqiang Kou, and Shuzhou Sun. Object-level correlation for few-shot segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 23689–23699, 2025.

[35] Yang Shen, Xiu-Shen Wei, Yifan Sun, Yuxin Song, Tao Yuan, Jian Jin, He-Yang Xu, Yazhou Yao, and Errui Ding. Explanatory instructions: Towards unified vision tasks understanding and zero-shot generalization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 54515–54558, 2025. URL `https://proceedings.mlr.press/v267/shen25h.html`.

[36] Zhiwen Yang and Yuxin Peng. GaLa-2.5D: Global-local alignment with 2.5d semantic guidance for camera-based 3D semantic scene completion in autonomous driving. *Chinese Journal of Electronics*, 35(2):817–828, 2026. doi: 10.23919/cje.2025.00.297.

[37] Jie Yang, Miao Ma, Yutong Li, and Zhao Pei. VQALS: A video question answering method in low-light scenes based on illumination correction and feature enhancement. *Chinese Journal of Electronics*, 34(4):1300–1308, 2025. doi: 10.23919/cje.2023.00.403.

[38] Lanxin Wu, Jiangtao Peng, Bing Yang, Weiwei Sun, Xinran Luo, Wenhui Hou, and Wei Li. Contrastive learning for improved image-text alignment in hyperspectral change detection. *Chinese Journal of Electronics*, 35(4):1339–1350, 2026. doi: 10.23919/cje.2025.00.538.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. Eur. Conf. Comp. Vis.*, pages 740–755, 2014.

[40] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 5828–5839, 2017.

[41] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024.

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. Int. Conf. Mach. Learn.*, pages 8748–8763, 2021.

[43] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[44] Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. VLMEvalKit: An open-source toolkit for evaluating large multi-modality models. In *Proc. ACM Int. Conf. Multimedia*, pages 11198–11201, 2024.

[45] Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proc. Conf. AAAI*, pages 7907–7915, 2025.

[46] Jiayu Wang, Yifei Ming, Zhenmei Shi, Vibhav Vineet, Xin Wang, Yixuan Li, and Neel Joshi. Is a picture worth a thousand words? delving into spatial reasoning for vision language models. In *Advances in Neural Inf. Process. Syst.*, pages 75392–75421, 2024.

[47] Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. In *Advances in Neural Inf. Process. Syst.*, pages 21875–21911, 2024.

[48] Sri H Dumpala, Aman Jaiswal, Chandramouli Sastry, Evangelos Milios, Sageev Oore, and Hassan Sajjad. Sugarcrepe++ dataset: Vision-language model sensitivity to semantic and lexical alterations. In *Advances in Neural Inf. Process. Syst.*, pages 17972–18018, 2024.

[49] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 14375–14385, 2024.

[50] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy Visual Task Transfer. *Transactions on Machine Learning Research*, 2025. URL `https://arxiv.org/abs/2408.03326`.

[51] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. In *Proc. Eur. Conf. Comp. Vis.*, pages 148–166, 2024.

[52] Damiano Marsili, Rohun Agrawal, Yisong Yue, and Georgia Gkioxari. Visual agentic AI for spatial reasoning with a dynamic API. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 19446–19455, 2025.

[53] Zihui Xue, Mi Luo, and Kristen Grauman. Seeing the arrow of time in large multimodal models. *arXiv preprint arXiv:2506.03340*, 2025.

[54] Yuanxin Liu, Shicheng Li, Yi Liu, Yuxiang Wang, Shuhuai Ren, Lei Li, Sishuo Chen, Xu Sun, and Lu Hou. Tempcompass: Do video llms really understand videos? *arXiv preprint arXiv:2403.00476*, 2024.

[55] Yuanhan Zhang, Yunice Chew, Yuhao Dong, Aria Leo, Bo Hu, and Ziwei Liu. Towards video thinking test: A holistic benchmark for advanced video reasoning and understanding. In *Proc. IEEE Int. Conf. Comp. Vis.*, pages 20626–20636, 2025.

[56] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-Thought Prompting elicits reasoning in large language models. In *Advances in Neural Inf. Process. Syst.*, pages 24824–24837, 2022.

[57] Zifu Wang, Junyi Zhu, Bo Tang, Zhiyu Li, Feiyu Xiong, Jiaqian Yu, and Matthew B Blaschko. Jigsaw-r1: A study of rule-based visual reinforcement learning with jigsaw puzzles. *arXiv preprint arXiv:2505.23590*, 2025.

[58] Ryan Liu, Jiayi Geng, Addison J Wu, Ilia Sucholutsky, Tania Lombrozo, and Thomas L Griffiths. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. In *Proc. Int. Conf. Mach. Learn.*, pages 38489–38517, 2025.

[59] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Inf. Process. Syst.*, pages 27056–27087, 2024.

[60] 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. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *Advances in Neural Inf. Process. Syst.*, pages 87310–87356, 2024.

[61] Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. SEED-Bench: Benchmarking multimodal large language models. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 13299–13308, 2024.

[62] Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. NaturalBench: Evaluating vision-language models on natural adversarial samples. In *Advances in Neural Inf. Process. Syst.*, pages 17044–17068, 2024.

[63] Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. Can MLLMs reason in multimodality? EMMA: An enhanced multimodal reasoning benchmark. *arXiv preprint arXiv:2501.05444*, 2025.

[64] Jiahao Xiao, Jianbo Zhang, Bowen Yan, Shengyu Guo, Tongrui Ye, Kaiwei Zhang, Zicheng Zhang, Xiaohong Liu, Zhengxue Cheng, Lei Fan, Chu-yi Li, and Guangtao Zhai. Static and plugged: Make embodied evaluation simple. *arXiv preprint arXiv:2508.06553*, 2025.

[65] Fanheng Kong, Jingyuan Zhang, Hongzhi Zhang, Shi Feng, Daling Wang, Linhao Yu, Xingguang Ji, Yu Tian, Victoria W., and Fuzheng Zhang. TUNA: Comprehensive fine-grained temporal understanding evaluation on dense dynamic videos. In *Proc. Conf. Association for Computational Linguistics*, pages 1810–1839, 2025.

[66] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 24108–24118, 2025.

[67] Jihan Yang, Shusheng Yang, Anjali W. Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pages 10632–10643, 2025.

[68] Jianshu Zhang, Dongyu Yao, Renjie Pi, Paul Pu Liang, and Yi R. Fung. VLM^2-Bench: A Closer Look at How Well VLMs Implicitly Link Explicit Matching Visual Cues. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7510–7545, 2025. doi: 10.18653/v1/2025.acl-long.372.

[69] Jihyung Kil, Zheda Mai, Justin Lee, Arpita Chowdhury, Zihe Wang, Kerrie Cheng, Lemeng Wang, Ye Liu, and Wei-Lun Chao. MLLM-CompBench: A comparative reasoning benchmark for multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0906.

A Impact Statement

This paper presents the development, training, and evaluation of GO-SSL, a cross-instance self-supervised reinforcement learning framework for improving the vision-centric understanding of multimodal large language models. By constructing automatically verifiable group-and-order tasks from visual data, GO-SSL reduces reliance on task-specific human annotations, external model judges, and manually annotated question-answer pairs. This may make visual post-training more scalable and reproducible, especially for tasks requiring fine-grained perception, spatial organization, temporal ordering, and cross-instance comparison.

The proposed method may benefit downstream applications that rely on accurate visual understanding, such as embodied agents, robotics, assistive systems, and visual reasoning tools. At the same time, stronger vision-centric MLLMs may also be used in scenarios where incorrect predictions or overconfident visual judgments could cause practical harm. Examples include safety-critical decision support, surveillance-related applications, and real-world autonomous systems. Our work does not directly target these high-stakes deployment settings, and all experiments are conducted in research settings using existing datasets and benchmarks.

Since GO-SSL improves model capabilities through automatically constructed self-supervised tasks, its behavior still depends on the quality, diversity, and biases of the underlying visual data. Models trained with this framework should therefore be carefully evaluated for robustness, fairness, and domain-specific reliability before deployment. We encourage future work to study safeguards, failure modes, and responsible evaluation protocols for self-supervised RL post-training of MLLMs.

B Answer-Only versus Chain-of-Thought Prompting

Reasoning-based prompting and visual jigsaw optimization have been explored in prior work [56, 57], while explicit deliberation does not necessarily improve every task [58]. We further compare answer-only (AO) prompting and reasoning-based prompting on several vision-centric benchmarks using Qwen2.5-VL-3B and Qwen2.5-VL-7B. These benchmarks cover image, 3D, and video understanding, allowing us to examine whether explicit reasoning is consistently beneficial across different visual domains. For AO evaluation, we use a direct-answer prompt, *e.g.*, "Answer with the option letter directly." For CoT evaluation, we instead ask the model to reason before answering, *e.g.*, "Think step by step before answering. The last line of your response should be exactly in the following format: Answer: `LETTER`, where `LETTER` is one of the options." As shown in Table 5, enabling chain-of-thought (CoT) reasoning does not consistently improve performance over direct answer prediction. CoT leads to clear drops for both 3B and 7B models, suggesting that explicit reasoning may introduce noisy intermediate steps for perception-heavy tasks where the answer can often be inferred directly from visual evidence.

The gap is more pronounced for the smaller 3B model. This observation suggests that smaller MLLMs may not always benefit from generated reasoning traces, as their intermediate rationales can be less reliable and may amplify errors rather than correct them. These preliminary results do not imply that CoT is generally ineffective; rather, they motivate our choice of answer-only supervision for jigsaw-style SSL-RL, where the target is directly verifiable and unnecessary intermediate reasoning may introduce additional noise.

C Benchmark Descriptions

We evaluate GO-SSL on a broad set of image, 3D, and video benchmarks. The benchmark suite covers detail understanding and visual perception, spatial reasoning, compositional understanding, general multimodal capability, hallucination robustness, depth-aware 3D reasoning, 3D spatial understanding, video understanding, and related capabilities.

Image side For image-side evaluation, we consider five complementary categories. First, we use HRBench4K/8K [45], V* [6] and MMStar [59] to measure detail-understanding and visual-perception, where models need to recognize subtle visual evidence, high-resolution details, and small objects in cluttered scenes. Second, we evaluate spatial understanding with VSR [9], SpatialEval [46], CV-Bench-2D [60], DA-2K [47], and OmniSpatial [10], covering spatial relations, 2D layout, object

Table 5: Comparison of answer-only and reasoning-based evaluation prompting. Reasoning "×" denotes direct answer-only prompting, while Reasoning "✓" denotes CoT-style prompting. "Avg." is computed over the listed benchmarks.

Models	Reasoning	Benchmark						Avg.
		V*	VSR	OmniSpatial	SugarCrepe++	DA-2K	TUNABench	
3B Models								
Qwen2.5-VL-3B	×	75.39	77.27	45.08	61.26	52.03	50.35	60.23
Qwen2.5-VL-3B	✓	67.54	68.34	40.05	50.49	54.16	48.11	54.78
7B Models								
Qwen2.5-VL-7B	×	76.96	77.31	42.66	61.59	54.45	53.14	61.02
Qwen2.5-VL-7B	✓	79.06	71.53	41.48	56.02	54.64	49.23	58.66

counting, relative depth, and complex spatial reasoning. Third, we use SugarCrepe++ [48] to evaluate compositional understanding, especially whether models can distinguish subtle changes in objects, attributes, and relations. Finally, we include SEEDBench [61] and NaturalBench [62] for general multimodal understanding, and HallusionBench [49] and EMMA [63] for hallucination-related robustness and broader multimodal reasoning.

3D side For 3D-side evaluation, we use Omni3DBench [52], 3DSRBench [33], BLINK [51], StaticEmbodiedBench [64], OmniSpatial [10], and DA-2K [47]. These benchmarks evaluate different aspects of 3D-aware perception and reasoning. Omni3DBench and 3DSRBench focus on object-centric and scene-level 3D spatial understanding, while DA-2K directly tests relative depth judgment. BLINK includes perception tasks such as relative depth estimation, visual correspondence, and multi-view reasoning, which are closely related to 3D-aware visual perception. StaticEmbodiedBench further evaluates embodied spatial understanding from static scenes, and OmniSpatial provides a broader testbed for complex spatial reasoning.

Video side For video-side evaluation, we use AoTBench [53], TUNA-Bench [65], Video-MME [66], TempCompass [54], VSI-Bench [67], and Video-TT [55]. These benchmarks cover complementary video abilities. AoTBench and TempCompass emphasize temporal directionality, event order, and fine-grained temporal reasoning. Video-MME and TUNA-Bench evaluate broader video question answering and temporal understanding under diverse video scenarios. VSI-Bench focuses on spatial memory and reasoning in egocentric videos, while Video-TT evaluates robust temporal and video reasoning.

D Experimental Setup Details

Base model and training We use Qwen2.5-VL-7B-Instruct [4] as the base MLLM for all main experiments. All GO-SSL models are post-trained with LoRA-based reinforcement learning. The LoRA rank is set to $r = 32$ and the LoRA scaling factor is set to $\alpha = 64$. We optimize the model with a GRPO-inspired group-relative RL objective using answer-only outputs, where the model is required to place the final group-and-order prediction inside `<answer></answer>` tags.

For image and 3D GO-SSL, we use a rollout group size of $G = 8$ and a global batch size of 64. For video GO-SSL, due to the larger memory cost of video inputs, we use $G = 4$ and a global batch size of 32. Unless otherwise specified, the reward weights are set to $\lambda_{\text{grp}} = 0.3$ and $\lambda_{\text{ord}} = 0.7$, the learning rate is set to 2×10^{-5}, and the partial reward discount is set to $\gamma = 0.3$. These key hyperparameters are selected based on small-scale validation experiments. The main experiments are conducted on 8 NVIDIA A100 GPUs. All rewards are automatically computed from the self-supervised group-and-order targets, requiring no human annotations, external model judges, or additional supervised labels.

For the default GO-SSL task construction, we use a 2×2 patch grid for each image in image GO-SSL, sample 4 local crops for each anchor scene in 3D GO-SSL, and divide each video into 4 temporal clips for video GO-SSL. These choices are made to balance task difficulty and training stability. Empirically, overly simple tasks provide weak supervision and lead to limited downstream generalization, while overly difficult tasks make the group-and-order objective hard to learn and may introduce noisy rewards.

Training data For image GO-SSL, we construct cross-image jigsaw samples from the COCO 2014 training split [39]. Each sample contains two source images. We crop and partition each source image into local patches, shuffle the patches from both sources together, and train the model to group patches by source image and recover the row-major spatial order within each group.

For 3D GO-SSL, we use RGB-D data from ScanNet [40]. Each sample contains two anchor images and shuffled local crops. The model first assigns each crop to its corresponding anchor image and then ranks the crops within each anchor from closest to farthest according to the depth of the marked center point. For video GO-SSL, we use LLaVA-Video [41]. Each sample contains clips from two source videos. The model is trained to group clips by source video and recover the chronological order within each video.

Curriculum learning We adopt a random-to-hard curriculum for image GO-SSL. In the warm-up stage, two source images are sampled randomly from the training set, which stabilizes format learning and basic group-and-order behavior. In the hard-pair stage, we introduce visually or semantically similar image pairs to make source grouping more ambiguous and encourage fine-grained discrimination.

We consider two hard-pair variants. *CLIP-Hard* pairs are mined using OpenAI CLIP ViT-B/32 features by retrieving nearest neighbors in CLIP feature space. These pairs are semantically similar and make grouping harder than random pairing. *CLIP-Blind* pairs are constructed by further filtering CLIP-near candidates with Facebook DINOv2-Base features. This favors pairs that are close under CLIP but differ under DINOv2, reducing coarse semantic shortcuts and encouraging patch-level discrimination. Unless otherwise specified, our main GO-SSL variants for image, 3D, and video use the same two-stage schedule: 400 steps with random source pairing followed by 600 steps with CLIP-Blind hard pairing.

Evaluation protocol We mainly use VLMEvalKit [44] for evaluation and follow its standardized protocols whenever a benchmark is supported. For benchmarks not directly covered by VLMEvalKit, we follow the official evaluation scripts or the evaluation code released by related work. For some benchmarks whose official evaluation uses a completion-style protocol, we use a strict direct-answer prompt to ensure unambiguous answer extraction and consistent comparison across models. For HallusionBench, we report f-acc using VLMEvalKit's built-in rule-based answer extraction and matching. For all compared models, including Qwen2.5-VL-7B-Instruct, Spatial-SSRL [17], Visual Jigsaw [18], and our GO-SSL variants, we use the default evaluation settings of VLMEvalKit or the official repository. In the result tables, "test" typically denotes the default overall accuracy reported by the official evaluation protocol, unless otherwise specified. Metric names such as "G-acc" follow the official outputs, *e.g.*, Group accuracy.

We download the released 7B checkpoints of Spatial-SSRL and Visual Jigsaw from Hugging Face and evaluate them under the same evaluation pipeline as our models. Detailed benchmark descriptions and metrics are provided in Appendix C.

D.1 Comparison of Training Recipes

We further compare the training recipes of GO-SSL with two closely related self-supervised RL post-training methods, Spatial-SSRL and Visual Jigsaw. As summarized in Table 6, this comparison covers modality, task form, supervision multiplicity, the use of extra SFT, tunable parameters, and RL training budgets. The comparison documents differences in the published training recipes. Because parameter tuning, rollout groups, and training data differ, it is not a strictly controlled attribution study; the objective ablations in Table 4 provide a more direct comparison of grouping and ordering.

As defined in Sec. 3.1, supervision multiplicity measures how many distinct source-level verifiable contexts can be induced from n raw visual sources before stochastic transformations such as shuffling, cropping, masking, or prompt variations. Under this view, a single-source jigsaw task provides n source-level contexts, while a multi-task single-source SSL pipeline with T pretext types provides approximately $T \cdot n$ contexts. In contrast, GO-SSL uses pairwise source construction and can draw from $\binom{n}{2}$ cross-instance contexts. This does not change the amount of raw data, but increases the density and diversity of supervision that can be extracted from the same training pool.

Table 6: Comparison of self-supervised RL post-training recipes. Supervision multiplicity follows the definition in Sec. 3.1.

Method	Modality	Task form	Supervision multiplicity	Extra SFT	Tuning params	RL steps	Global batch	Rollout group
Spatial-SSRL [17]	Image / RGB-D	Multi-task single-source SSL	$\approx T \cdot n$ $(T = 5)$	Yes	Full	360	128	5
Visual Jigsaw-Image [18]	Image	Single-source ordering	n	No	Full	1000	256	16
Visual Jigsaw-3D [18]	RGB-D	Single-source ordering	n	No	Full	800	128	16
Visual Jigsaw-Video [18]	Video	Single-source ordering	n	No	Full	1000	128	16
GO-SSL-Image	Image	cross-instance group + order	$\binom{n}{2}$	No	LoRA	1000	64	8
GO-SSL-3D	RGB-D	cross-instance group + order	$\binom{n}{2}$	No	LoRA	1000	64	8
GO-SSL-Video	Video	cross-instance group + order	$\binom{n}{2}$	No	LoRA	1000	32	4

Table 7: Generalization across model scales and families. D&V denotes detail understanding and visual perception; Comp., General, and Extra denote the corresponding benchmark categories. LLaVA-OV abbreviates LLaVA-OneVision.

Model	D&V	Spatial	Comp.	General	Extra	Avg.
Qwen2.5-VL-3B						
Qwen2.5-VL-3B	68.63	53.02	61.26	49.45	29.39	54.18
Spatial-SSRL-3B	70.08	55.66	60.94	50.22	32.33	56.05
GO-SSL-3B	**70.78**	**56.37**	**61.31**	**50.69**	**32.51**	**56.62**
LLaVA-OneVision-7B						
LLaVA-OV-7B	66.83	51.79	**56.74**	56.42	23.88	53.12
LLaVA-OV-7B + GO-SSL	**69.65**	**52.77**	56.60	**56.65**	**25.01**	**54.46**

Table 6 highlights several practical differences. Spatial-SSRL constructs self-supervised QA pairs from multiple heterogeneous tasks, including patch reordering, flip recognition, inpainting, depth ordering, and relative 3D position prediction. While this design provides diverse supervision, it also requires an additional SFT cold-start stage and performs full-parameter tuning during RL post-training. Visual Jigsaw adopts a simpler recipe by using single-source ordering tasks for images, 3D data, and videos, but each raw source is mainly used as an independent ordering instance, and the reported training uses full-parameter tuning with larger rollout groups and global batch sizes.

GO-SSL keeps the task family unified while changing the source interaction structure. Instead of reconstructing the structure within a single source, GO-SSL pairs two sources, mixes their local elements, and requires the model to recover both source grouping and within-source order. This makes cross-instance discrimination an explicit part of the verifiable reward. As shown in Table 6, GO-SSL uses LoRA-based RL post-training, avoids extra SFT warm-up, and uses smaller global batch sizes than Visual Jigsaw, especially in the video setting. Overall, the paired formulation allows GO-SSL to obtain rich supervision from a relatively simple and lightweight training recipe through different source pairings, shuffles, and hard-pair curricula.

E Generalization and Transfer Analyses

E.1 Generalization across Model Scales and Families

The main experiments use Qwen2.5-VL-7B-Instruct. To examine whether GO-SSL depends on a particular model scale or backbone family, we additionally evaluate it on Qwen2.5-VL-3B and LLaVA-OneVision-7B. For consistency, the category averages follow the same 14 image-side benchmarks used in the main experiments.

As shown in Table 7, on Qwen2.5-VL-3B, GO-SSL improves the base model by 2.44 points in average performance and outperforms Spatial-SSRL-3B across all five capability categories. The improvement therefore persists when the backbone is reduced from 7B to 3B parameters.

Table 8: 3D evaluation on LLaVA-OneVision-7B (LLaVA-OV-7B). GO-SSL improves all six benchmark scores, indicating cross-backbone transfer. The benchmark subsets and evaluation metrics are consistent with Table 2.

Model	Omni3D Bench	3DSRBench	BLINK	Static-EmbodiedBench	Omni Spatial	DA-2K
LLaVA-OV-7B	56.00	57.42	62.10	36.50	44.36	55.03
LLaVA-OV-7B + GO-SSL-3D	**58.67**	**58.81**	**67.74**	**39.50**	**45.27**	**63.15**

Table 9: Cross-domain transfer evaluation of GO-SSL variants. Image GO-SSL and 3D GO-SSL are evaluated on both image-side and 3D-side benchmarks. The results show that each variant is strongest in its own training domain, while still transferring non-trivially to the other domain.

Model	Training Domain	Image-side Benchmarks					3D-side Benchmarks		
		HRBench4K	HRBench8K	V*	VSR	SugarCrepe++	DA-2K	BLINK	Omni3DBench
		test	test	test	test	test	test	rd-acc	count
Qwen2.5-VL-7B	–	68.50	65.38	76.96	77.31	61.59	54.45	77.42	17.14
Image GO-SSL	Image	71.50	69.88	81.68	80.42	63.17	61.70	81.45	21.43
(Gain)		+3.00	+4.50	+4.72	+3.11	+1.58	+7.25	+4.03	+4.29
3D GO-SSL	3D	69.38	67.88	80.10	78.95	62.14	74.90	91.13	24.29
(Gain)		+0.88	+2.50	+3.14	+1.64	+0.55	+20.45	+13.71	+7.15

GO-SSL also transfers to a different model family. On LLaVA-OneVision-7B, it improves four of the five capability categories and raises the overall average from 53.12 to 54.46. Together, these results indicate that the benefit of cross-instance group-and-order supervision is not specific to either the Qwen2.5-VL architecture or a single model scale.

E.2 Cross-Backbone Generalization on 3D Benchmarks

Beyond image-centric evaluation, we further investigate whether GO-SSL transfers to 3D visual understanding with a different backbone. Specifically, we apply the same cross-scene group-and-order formulation to LLaVA-OneVision-7B and evaluate the resulting model on 3D benchmarks.

As shown in Table 8, LLaVA-OneVision-GO-SSL-3D improves the original backbone on all six 3D benchmarks. The improvements are particularly notable on BLINK and DA-2K, which emphasize fine-grained spatial comparison and depth-aware reasoning. These results further suggest that the effectiveness of GO-SSL originates from the general group-and-order formulation rather than from a specific backbone architecture.

E.3 Cross-Domain Transfer

We further evaluate whether GO-SSL learns transferable visual skills beyond its training domain. Here, Image GO-SSL refers to our CLIP-Blind-600 variant in Table 4. As shown in Table 9, Image GO-SSL and 3D GO-SSL both improve over the base model on image-side and 3D-side benchmarks. Each variant performs best on the domain that matches its training signal: Image GO-SSL obtains stronger gains on image-side benchmarks such as HRBench, V*, and VSR, while 3D GO-SSL shows larger improvements on depth- and spatially-oriented benchmarks such as DA-2K, BLINK, and Omni3DBench. At the same time, the off-domain results remain consistently positive, indicating that the group-and-order formulation does not merely overfit to a modality-specific puzzle. Instead, cross-instance grouping and within-source ordering appear to learn general skills such as local evidence comparison, structural organization, and spatial grounding, which transfer across visual domains.

F Additional Ablation Studies

We provide additional ablations in Fig. 3 to further analyze the design choices of GO-SSL. These studies cover three aspects: the relative weighting between grouping and ordering rewards, the robustness of Video GO-SSL under different frame budgets, and the effect of adding cross-instance grouping beyond order-only training in 3D and video domains.

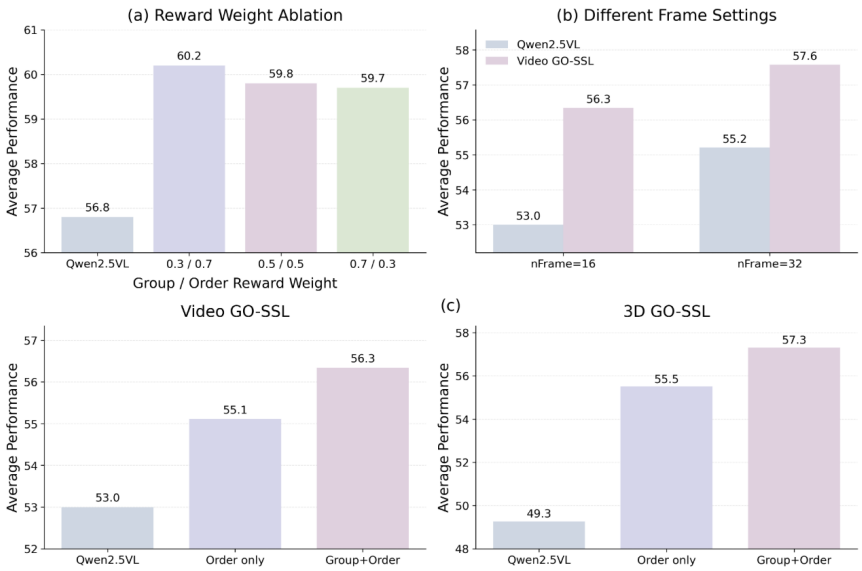


Figure 3: Additional ablation studies. (a) Reward weight ablation on the image-side GO-SSL objective. (b) Video GO-SSL under different input-frame settings. (c) Order-only versus full Group+Order variants for video and 3D GO-SSL.

F.1 Effect of Group-and-Order Reward Weights

GO-SSL optimizes two complementary objectives: grouping elements from the same source and recovering their within-source order. To study their relative contribution, we vary the reward weights of the group and order terms while keeping the same training setting. As shown in Fig. 3(a), all three weight configurations clearly outperform the Qwen2.5-VL baseline, confirming that the gains do not depend on a single narrow choice of reward weighting.

To isolate the effect of reward weighting, all three variants are trained with 1000 random-pair steps and no hard-pair mining. Among the three variants, the order-heavy setting $(\lambda_{\mathrm{grp}}, \lambda_{\mathrm{ord}}) = (0.3, 0.7)$ achieves the highest average across the same 14 benchmark metrics used in Table 1, consistent with our preliminary validation sweep. This suggests that spatial ordering provides a particularly strong transfer signal for image-side visual understanding, likely because recovering patch structure requires attending to local details, object parts, and their layout consistency. The balanced and group-heavy settings remain competitive, indicating that cross-source grouping is also beneficial. Overall, these results support our design choice of jointly optimizing grouping and ordering, with a slightly larger weight on the ordering reward.

F.2 Cross-Modal Ablations on 3D and Video

We further ablate whether the Group-and-Order formulation remains useful beyond static images. For both 3D and video, we compare an order-only variant with the full Group+Order variant. The order-only variant removes cross-instance grouping and trains the model to recover the intrinsic order within a single source: depth ordering for 3D crops and temporal ordering for video clips. The Group+Order variant follows the full GO-SSL formulation, where elements from two sources are mixed and the model must first separate source membership and then recover the within-source order.

As shown in Fig. 3(c), the order-only variants already improve over the base Qwen2.5-VL model, confirming that depth ordering and temporal ordering themselves provide useful self-supervised signals. However, the full Group+Order variants further improve the average performance in both domains. For video, order-only improves the average from 53.0 to 55.1, while Group+Order further improves it to 56.3. For 3D, order-only improves the average from 49.3 to 55.5, and Group+Order further reaches 57.3. These results support the benefit of the paired group-and-order formulation over the single-source ordering variants under our training setup.

For 3D, the improvement suggests that coupling depth ordering with cross-scene crop grounding encourages the model to compare local regions across different visual contexts, rather than only

Table 10: Comparison of feature-space pairing strategies under identical 1,000-step training budgets. CLIP-near + DINO-far corresponds to CLIP-Blind-600 in the main experiments. Differences are computed as DINO-near + MAE-far minus CLIP-near + DINO-far.

Pairing Strategy	D&V	Spatial	Comp.	Gen.	Extra	Avg.
CLIP-near + DINO-far	**74.47**	**62.26**	**63.17**	56.20	33.54	**60.85**
DINO-near + MAE-far	74.22	61.12	62.69	**56.43**	**34.38**	60.49
Difference	-0.25	-1.14	-0.48	+0.23	+0.84	-0.36

learning ordinal depth cues within one scene. For video, the improvement indicates that cross-video grouping helps the model distinguish event-level coherence across different videos while recovering temporal order within each video. Overall, these results support the central design of GO-SSL: grouping and ordering are complementary, and their combination provides richer verifiable supervision than ordering alone.

F.3 Video GO-SSL under Different Frame Settings

We also study whether Video GO-SSL remains effective under different input-frame budgets. As shown in Fig. 3(b), Video GO-SSL consistently improves over the Qwen2.5-VL baseline under both 16-frame and 32-frame evaluation settings. With 16 frames, the average performance improves from 53.0 to 56.3. With 32 frames, the average performance improves from 55.2 to 57.6. This indicates that the learned temporal group-and-order ability transfers across different video sampling budgets.

The gain is larger under the 16-frame setting, where the model has fewer visual observations and therefore benefits more from improved temporal organization. Under the 32-frame setting, the baseline becomes stronger due to richer visual input, but Video GO-SSL still provides a clear improvement. These results suggest that Video GO-SSL does not merely exploit a specific frame configuration, but improves temporal/video understanding in a way that remains robust across input-frame settings.

F.4 Why Does CLIP-Blind Outperform CLIP-Hard?

We provide an intuitive explanation for why CLIP-Blind pairing leads to stronger downstream transfer than CLIP-Hard pairing in Table 4. CLIP-Hard retrieves nearest neighbors in CLIP feature space, making paired images semantically similar. However, CLIP similarity is often dominated by global language-aligned concepts, such as object category, scene type, style, or common attributes. For a patch-level group-and-order task, such semantic similarity does not necessarily create the most informative ambiguity: two images may be close in CLIP space while still being separable by simple local cues such as color, texture, or layout.

CLIP-Blind pairing instead keeps images close in CLIP space while penalizing high DINO similarity. Since DINO features tend to capture visual structure and local appearance beyond high-level semantic alignment, this re-ranking favors pairs that are semantically related but differ in finer visual organization. Such pairs encourage the model to move beyond coarse semantic cues and compare local evidence, object parts, boundaries, and spatial layouts. Therefore, CLIP-Blind does not merely make the grouping task harder; it creates a more useful type of ambiguity, where paired images are semantically related but locally discriminative. This better matches the downstream capabilities targeted by GO-SSL, including fine-grained perception, cross-instance comparison, and spatial understanding. This interpretation is consistent with our results: CLIP-Hard improves spatial understanding over random pairing, while CLIP-Blind achieves stronger overall transfer.

F.5 Analysis of Alternative Hard-Pair Constructions

We investigate how different feature-space criteria affect hard-pair construction and downstream transfer. Our default CLIP-Blind strategy, referred to here as *CLIP-near + DINO-far*, favors pairs with high CLIP similarity but relatively low DINO similarity. As an alternative, we consider *DINO-near + MAE-far*, which favors pairs that are similar in DINO feature space but dissimilar in MAE feature space.

Table 11: Effect of progressively finer image layouts. All variants share 400 random 2×2 steps followed by 400 CLIP-Blind 2×2 steps; only the layout of the final 200 CLIP-Blind steps differs. Avg. is computed over 14 benchmarks.

Variant	Final 200 steps	D&V	Spatial	Comp.	Gen.	Extra	Avg.
CLIP-Blind-600	2×2	74.47	**62.26**	**63.17**	56.20	33.54	60.85
CLIP-Blind-2×3	2×3	75.94	61.15	61.99	56.39	34.15	60.90
CLIP-Blind-3×3	3×3	**76.19**	61.11	61.95	**56.53**	**34.26**	**60.99**

As shown in Table 10, both strategies improve downstream performance but exhibit different transfer profiles. CLIP-near + DINO-far performs better on detail perception, spatial reasoning, and compositional understanding, whereas DINO-near + MAE-far achieves slightly higher scores on the General and Extra categories. These results suggest that different feature-space pairing criteria provide complementary training signals.

Overall, CLIP-near + DINO-far remains our preferred strategy, achieving the higher overall average and stronger results on the visual capabilities most closely related to the group-and-order objective.

F.6 Effect of Increasing Patch Granularity

The main experiments use 2×2 patches and progressively introduce a 2×3 layout. We further examine whether GO-SSL continues to benefit from finer patch granularity by extending the final curriculum stage to 3×3 patches.

As the layout becomes finer, detail understanding and visual perception improve consistently from 74.47 to 75.94 and 76.19. The overall average also increases from 60.85 to 60.90 and 60.99. This suggests that finer patch granularity provides additional supervision for local visual discrimination.

The overall gains nevertheless remain modest, while spatial and compositional performance slightly decrease. A finer layout exposes more local visual details but also introduces stronger patch-boundary fragmentation, a larger output space, and a more difficult grouping-and-ordering problem. We therefore use the simpler 2×2 setting as the default and regard denser layouts as a complementary way to increase task granularity.

We also experimented with using 3×3 layouts from the beginning of training. Each sample then contains $2 \times 9 = 18$ shuffled patches, compared with 8 patches for 2×2 and 12 patches for 2×3. This substantially increases visual-token cost and combinatorial difficulty, making direct 3×3 training less stable under the same budget. The progressive curriculum is therefore preferable.

F.7 Analysis on Enlarged Candidate Pool with Decoys

We further investigate whether enlarging the candidate pool with additional decoy patches can provide a stronger self-supervised signal. Starting from the 2×2 CLIP-Blind setting, we add two CLIP-Blind distractor patches to the candidate pool. The model is still required to output two ordered groups corresponding to the two true source images, while ignoring irrelevant decoy patches.

We consider two variants. *GO-SSL-Decoy-Explicit* introduces an additional selection reward R_{sel} to directly supervise whether the model selects the true jigsaw patches before grouping and ordering:

$$R_{\mathrm{decoy}} = \lambda_{\mathrm{sel}} R_{\mathrm{sel}} + \lambda_{\mathrm{grp}} R_{\mathrm{grp}} + \lambda_{\mathrm{ord}} R_{\mathrm{ord}}. \tag{9}$$

In contrast, *GO-SSL-Decoy-Implicit* removes R_{sel} and keeps only the original group-and-order reward:

$$R_{\mathrm{decoy}} = \lambda_{\mathrm{grp}} R_{\mathrm{grp}} + \lambda_{\mathrm{ord}} R_{\mathrm{ord}}. \tag{10}$$

In this case, selecting a decoy is penalized only implicitly through errors in grouping or ordering.

As shown in Table 4, both decoy variants remain competitive but do not consistently improve over the simpler CLIP-Blind curriculum. GO-SSL-Decoy-Explicit achieves a 60.31 overall average and obtains the best external benchmark average among the decoy variants, reaching 34.70. This suggests that explicit candidate filtering can help distractor-aware recognition. GO-SSL-Decoy-Implicit slightly improves the overall average to 60.46, with stronger detail understanding and visual

Table 12: Targeted evaluation of cross-instance discrimination. GO-SSL denotes the CB-2×3 variant in Table 1. All values are accuracies (%).

Model	OC-Cpr	OC-Grp	VAW	State	StateAttr	Vis. Corr.
Qwen2.5-VL-7B	62.22	54.00	27.94	12.12	22.77	30.81
Visual Jigsaw	66.67	54.00	61.76	72.73	65.35	41.28
GO-SSL	**71.94**	**62.00**	**75.00**	**84.85**	**78.22**	**43.60**

perception (74.24) and spatial understanding (61.23) than the explicit variant. This indicates that implicit distractor rejection may better preserve the original structural reconstruction signal.

Nevertheless, both decoy variants remain below CLIP-Blind-600 and CLIP-Blind-2×3 in overall average. These results suggest that decoy-based candidate expansion is a promising but non-trivial extension. Rather than simply increasing difficulty, it changes the task from closed-set group-and-order reconstruction to candidate selection plus reconstruction, which may partially shift optimization away from structural layout recovery. We therefore regard it as an objective-level extension rather than a strictly controlled single-factor ablation.

G Targeted Evaluation of Cross-Instance Discrimination

The main benchmark suite evaluates broad visual perception, spatial reasoning, and multimodal understanding. To more directly examine the central capability targeted by GO-SSL, we additionally evaluate models on benchmark subsets whose task definitions explicitly require comparison, grouping, correspondence, or attribute/state discrimination across visual instances.

We consider three complementary benchmark families. VLM^2-Bench [68] evaluates explicit matching of visual cues across images: OC-Cpr requires cross-image object comparison, while OC-Grp requires grouping images that contain corresponding objects. For MLLM-CompBench [69], we report fine-grained visual comparison results under the VAW, State, and StateAttr metrics. Finally, the Visual Correspondence subset of BLINK [51] evaluates whether models can match the same local visual point across different images or viewpoints.

GO-SSL achieves the strongest performance on all reported metrics. Compared with Visual Jigsaw, it improves OC-Cpr and OC-Grp by 5.27 and 8.00 points, respectively; VAW, State, and StateAttr by 13.24, 12.12, and 12.87 points; and Visual Correspondence by 2.32 points.

Importantly, none of these evaluations uses the Group-and-Order output format encountered during training. The improvements therefore cannot be explained by adaptation to the training puzzle itself. Instead, they provide more direct evidence that cross-instance group-and-order supervision transfers to the capabilities it is designed to encourage: cross-instance object association, fine-grained comparative discrimination, source grouping, and local visual correspondence.

H Case Analysis and Visualization

H.1 Task Examples

Figure 4 illustrates representative GO-SSL task instances across image, 3D, and video domains. For images and videos, the model groups shuffled elements by their source and recovers the spatial or temporal order within each group. For 3D inputs, the model assigns local crops to their corresponding anchor scenes and orders them by the depth of the marked center points.

H.2 Visualization Analysis

To better understand how GO-SSL affects visual grounding, we conduct an occlusion-based sensitivity analysis on binary image-text verification examples. For each image-question pair, we ask the model whether the image matches a textual description, and estimate which regions support the predicted answer by masking local image regions.

Concretely, we divide each image into an 8×8 grid and mask one cell at a time with a gray patch. For a target answer $a \in \{\texttt{Yes}, \texttt{No}\}$, we compute its normalized probability over the two candidate

answers before and after masking cell (r, c). The occlusion sensitivity score is defined as

$$s_{r,c} = \max\left(p(a \mid I, q) - p(a \mid I_{\setminus(r,c)}, q), 0\right), \tag{11}$$

where q is the textual query and $I_{\setminus(r,c)}$ denotes the masked image. We normalize the resulting score map, resize it to the original resolution, and overlay it on the input image. Unless otherwise specified, we visualize the sensitivity map with respect to each model's own predicted answer.

Figure 5 compares Qwen2.5-VL-7B, Visual Jigsaw, Spatial-SSRL, and GO-SSL. In the first example, which asks whether the image matches the description "house at the end", GO-SSL correctly focuses on the relevant house region, while the baseline models attend to broader or distracting areas. In the second example, which requires judging whether the "moon is on the top of the castle", GO-SSL highlights the moon and castle roof region and predicts the correct answer. By contrast, the other models either fail to localize the relevant evidence or make incorrect predictions despite attending to partially related regions.

These results suggest that GO-SSL encourages more localized and relation-aware visual grounding. The grouping objective promotes discrimination between visually similar local regions across images, while the ordering objective encourages local-to-global spatial reasoning. Together, they help the model rely more on fine-grained visual evidence rather than only global scene priors.

H.3 Qualitative examples on different tasks

Figure 6 shows qualitative examples across diverse spatial understanding and reasoning tasks, including perspective taking, dynamic reasoning, spatial interaction, and complex spatial logic. GO-SSL gives correct predictions in these cases where the base model and prior self-supervised post-training baselines often fail. These examples suggest that cross-instance group-and-order training does not merely improve the model on the synthetic jigsaw objective, but also transfers to downstream problems that require fine-grained visual grounding and relational spatial understanding.

The key intuition is that GO-SSL repeatedly forces the model to distinguish local elements from different visual sources and then organize them into coherent structures. This encourages the model to compare subtle visual evidence, reason over object-level relations, and reduce reliance on coarse scene priors.

We further provide qualitative examples for the 3D and video extensions in Figures 7 and 8. For 3D-oriented tasks, 3D GO-SSL improves predictions on physical contact reasoning, object counting, and relative depth estimation, suggesting that the learned group-and-order objective transfers to geometric scene understanding. For video tasks, Video GO-SSL better handles action intention, final-state tracking, and dynamic scene changes, indicating that temporal group-and-order training can benefit downstream video understanding.

In addition, some benchmarks require not only direct visual recognition but also simple reasoning over the visual context, such as checking whether all instances belong to the same character, reading a map-label relation, or inferring an event from an image sequence. As shown in Figure 9, GO-SSL can also produce more reliable answers in such cases, suggesting that the learned group-and-order ability provides useful inductive bias beyond low-level spatial matching.

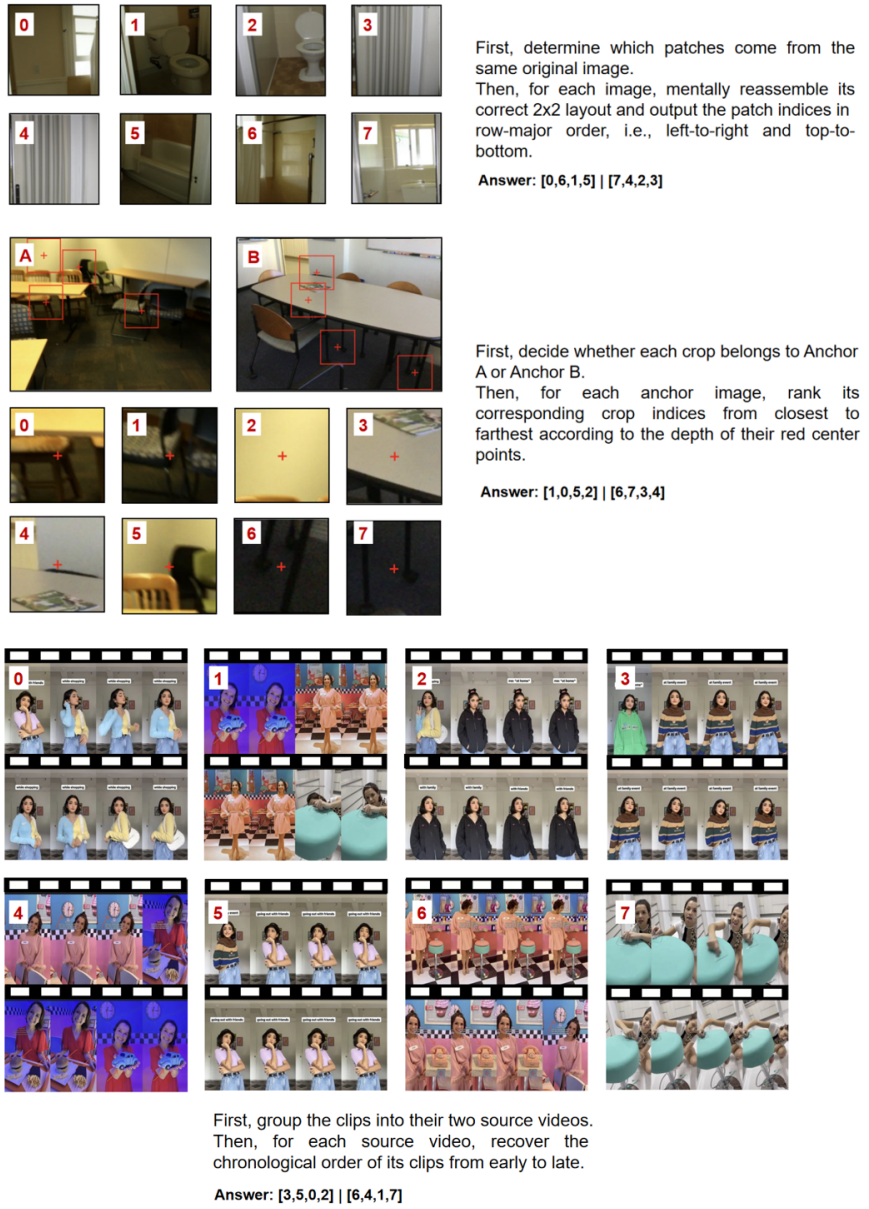


Figure 4: GO-SSL task examples across image, 3D, and video domains.

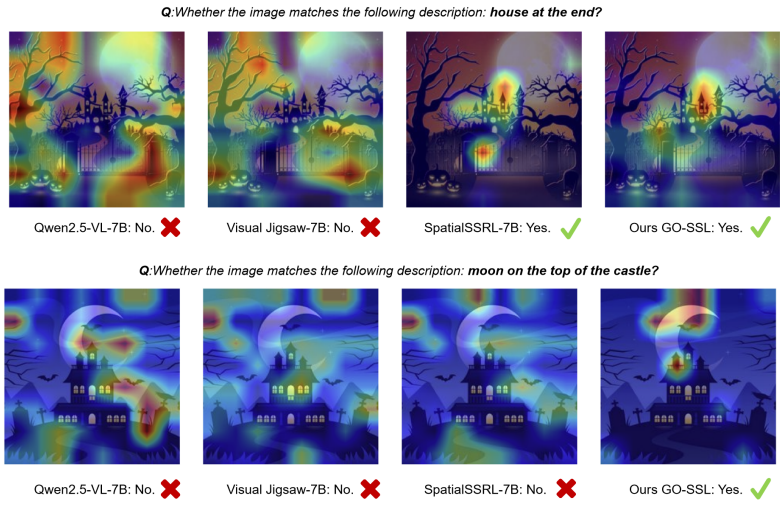


Figure 5: Qualitative comparison of occlusion-based sensitivity maps on fine-grained spatial matching examples. Compared with Qwen2.5-VL-7B and Visual Jigsaw, GO-SSL better localizes the visual evidence relevant to the queried description and produces correct answers. It also corrects a failure case of Spatial-SSRL in the second example, where object saliency alone is insufficient for precise spatial-relation judgment.

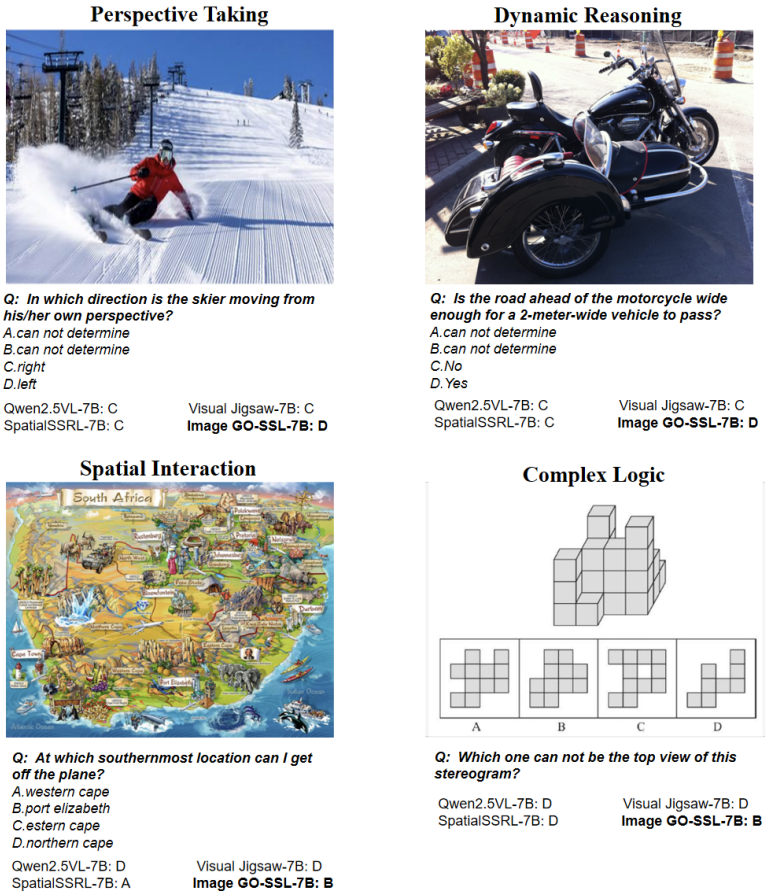


Figure 6: Qualitative examples on image-side tasks. GO-SSL gives correct predictions on perspective taking, dynamic reasoning, spatial interaction, and complex logic, where the base model and prior self-supervised post-training baselines often fail.

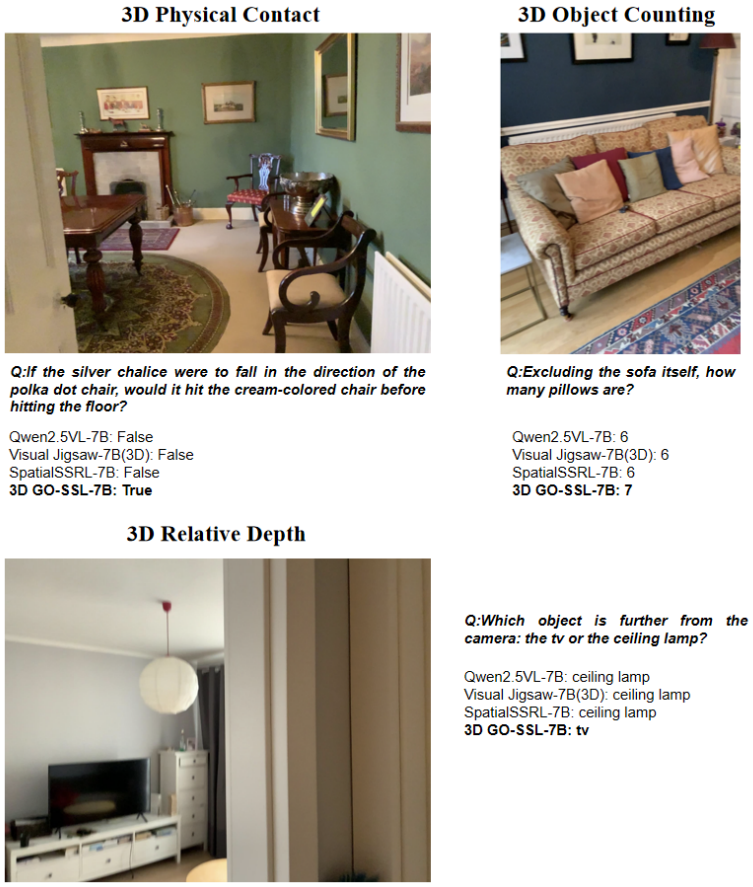


Figure 7: Qualitative examples on 3D-oriented visual understanding tasks. 3D GO-SSL gives correct predictions on physical contact reasoning, object counting, and relative depth estimation, where the base model and prior self-supervised post-training baselines often fail.

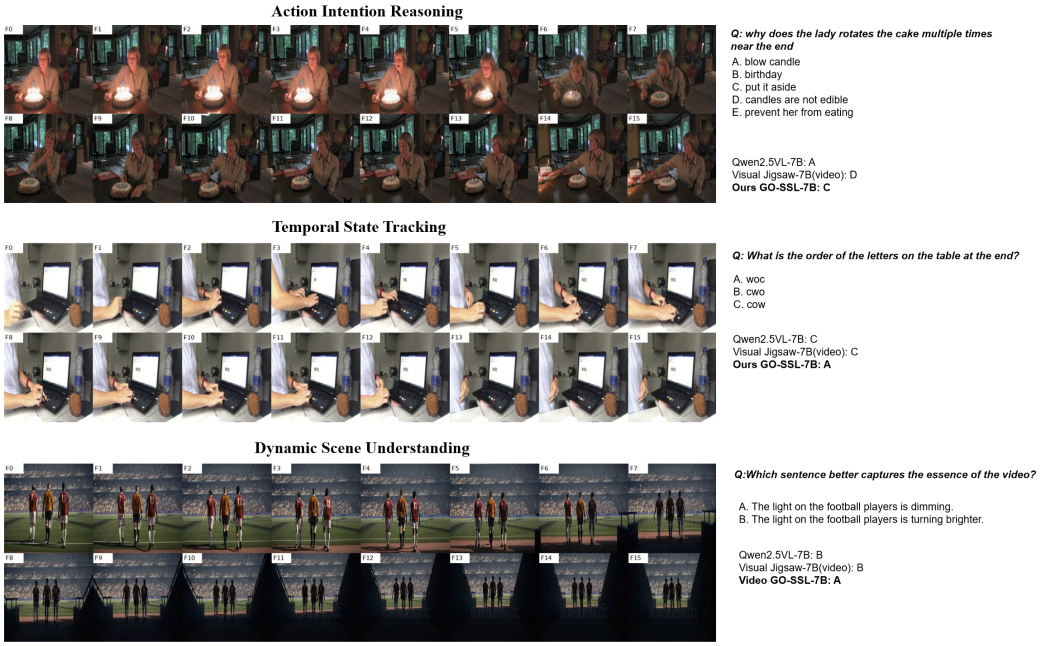
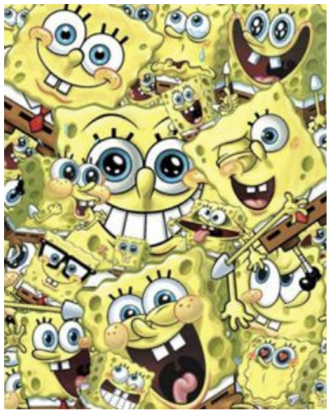


Figure 8: Qualitative examples on video understanding tasks. Video GO-SSL correctly answers questions requiring action intention reasoning, final-state tracking, and dynamic scene understanding, while the base model and video jigsaw baseline fail in these cases.

Q:*Does this figure only contain SpongeBob?*



Qwen2.5VL-7B:

No, the figure does not only contain SpongeBob. The image is a collage of various characters from the "SpongeBob SquarePants" series, including SpongeBob, Patrick Star, and other characters.

SpatialSSRL-7B:

No, the image does not only contain SpongeBob. It features multiple instances of SpongeBob SquarePants, along with other characters from the same animated series. The background and other elements in the image also include other characters and elements from the show.

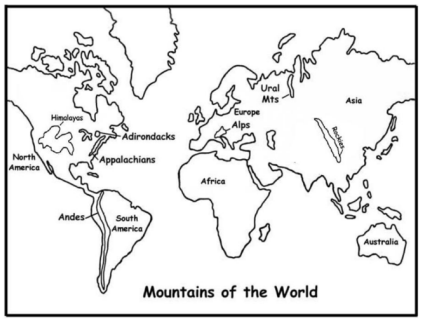
Visual Jigsaw-7B:

No, the image does not only contain SpongeBob. The image is a collage of various characters, including SpongeBob, but it also includes other characters from the same animated series. The characters are depicted in different poses and expressions, creating a lively and dynamic composition.

Ours GO-SSL-7B:

Yes, the figure in the image only contains SpongeBob. The image is a collage of various expressions and poses of SpongeBob SquarePants, a character from the animated television series "SpongeBob SquarePants."

Q:Based on the map, are the Rocky Mountains located in North America?



Qwen2.5VL-7B:

Yes, based on the map, the Rocky Mountains are located in North America. The map shows the Rocky Mountains in the western part of North America, which is consistent with their actual geographical location.

SpatialSSRL-7B:

Yes, based on the map, the Rocky Mountains are located in North America.

Visual Jigsaw-7B:

Yes, based on the map, the Rocky Mountains are located in North America. The map shows the Rocky Mountains in the western part of North America, specifically in the United States.

Ours GO-SSL-7B:

No, the Rocky Mountains are not located in North America. According to the map, the Rocky Mountains are labeled in Asia, specifically in the region that corresponds to the eastern part of the continent.

Q:According to the positive sequence of the images, does the man open the door?



Qwen2.5VL-7B:

Yes. Based on the sequence of images provided, it appears that the man is interacting with a door. In the first image, he is standing in front of the door, looking at it. In the second image, he is seen from the side, and it looks like he is reaching towards the door, possibly to open it. In the third image, he is seen from behind, and it seems that he has opened the door, as he is now standing in a different position, which suggests he has moved past the door. Therefore, according to the positive sequence of the images, it does appear that the man opens the door.

SpatialSSRL-7B:

Yes. Based on the sequence of images provided, it appears that the man is opening the door. In the first image, the man is standing in front of the door, facing it. In the second image, the man is slightly turned, and the door is partially open, suggesting that he is in the process of opening it. In the third image, the man is further turned, and the door is more open, indicating that he has continued to open it. Therefore, the sequence of images suggests that the man is indeed opening the door.

Visual Jigsaw-7B:

Yes, according to the sequence of images, the man appears to open the door. The first image shows the man standing in front of the door, the second image shows the man turning to face the door, and the third image shows the man opening the door.

Ours GO-SSL-7B:

No. The sequence of images provided does not show the man opening a door. The images depict the man standing in front of a door, but there is no visible action of opening the door. The sequence shows the man from different angles, but the door remains closed in all the images.

Figure 9: Qualitative examples of visually grounded reasoning across diverse scenarios. Compared with the base model and prior self-supervised post-training baselines, GO-SSL gives more reliable answers on instance-level recognition, spatial reasoning, and video understanding, suggesting benefits beyond low-level spatial matching.

I Prompt Templates

Prompt for Image Cross SSL

You are given n shuffled image patches that come from two **different images**.
Each original image was divided into a 2x2 grid, producing **k patches per image**.
Here are the patches, each tagged with an index reflecting the current shuffled order in which they are shown:

Patch 0: `<image>`
Patch 1: `<image>`
Patch 2: `<image>`
...
Patch n-1: `<image>`

Task:
First, determine which patches come from the same original image.
Then, for each image, mentally reassemble its correct 2x2 layout and output the patch indices in row-major order, *i.e.*, left-to-right and top-to-bottom.

Output format:
- Output **two lists of length 4**, separated by "|".
- Each list should contain the patch indices of one source image in the correct 2x2 layout.
- Use each patch index exactly once across the two lists.
- Output **ONLY** the final answer within `<answer></answer>` tags.

Answer format example:
`<answer>[2,0,3,1] | [7,4,6,5]</answer>`

Prompt for 3D-Depth Cross SSL

You are given two anchor images and several shuffled local crops.
Each anchor image contains k **red-boxed regions**. In each region, the **red center marker** indicates the exact point whose depth should be compared.
Each crop is taken from one marked region in one of the anchor images and contains the same red center marker.

Images:

Anchor A: `<image>`
Anchor B: `<image>`
Crop 0: `<image>`
Crop 1: `<image>`
Crop 2: `<image>`
...
Crop n-1: `<image>`

Task:
First, decide whether each crop belongs to Anchor A or Anchor B.
Then, for each anchor image, rank its corresponding crop indices from closest to farthest according to the depth of their red center points.
Do not rank crops by the average depth of the whole crop or the entire red-boxed region.
Only the depth of the red center marker should be used.

Output format:
- Output two lists separated by "|".
- The left list corresponds to Anchor A, and the right list corresponds to Anchor B.
- Each list must contain exactly k crop indices, ordered from closest to farthest.
- Each crop index must be used exactly once across the two lists.
- Output **ONLY** the final answer within `<answer></answer>` tags.

Answer format example:
`<answer>[2,5,0] | [4,1,3]</answer>`

Prompt for Video Cross SSL

You are given n shuffled video clips that come from two **different videos**.
Each original video contributes **k clips**.

Clips in current order:

Clip 0: `<video>`
Clip 1: `<video>`
Clip 2: `<video>`
...
Clip n-1: `<video>`

Task:
First, group the clips into their two source videos.
Then, for each source video, recover the chronological order of its clips from early to late.

Output format:
- Output two lists separated by "|".
- Each list must contain exactly k clip indices.
- The indices refer to the current shuffled clip order and are 0-based.
- Each clip index must be used exactly once across the two lists.
- Output **ONLY** the final answer within `<answer></answer>` tags.

Answer format example:
`<answer>[2,5,1] | [0,4,3]</answer>`

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims are stated in the abstract and introduction, and are supported by the method in Sec. 3, experiments in Sec. 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification:We discuss limitations in Appendix D, Appendix F.6 and future directions in the Sec. 5.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: This paper is primarily empirical and does not present theoretical results or formal proofs.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The task formulation, reward design, training setup, evaluation protocol, and additional prompts are provided in Sec. 3, Sec. 4, and the Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The datasets used in our experiments are available from their original providers under their respective licenses and access conditions. We describe the data construction, training, and evaluation details in Sec. 4 and Appendix D. The code and accompanying documentation are publicly available at `https://github.com/SEU-VIPGroup/GO-SSL`.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Training details, data sources, hyperparameters, hard-pair construction, and evaluation protocols are described in Sec. 4 and the Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars due to the high computational cost of large-scale MLLM RL training and evaluation, but report results across diverse benchmarks and ablations.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The model size, batch size, rollout group size, and training steps are reported in Sec. 4, with additional compute details provided in the Appendix D.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics, and the work uses public datasets with automatically generated self-supervised targets.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss potential positive impacts, risks, and responsible-use considerations in the Impact Statement in the Appendix A.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A].

    Justification:The paper does not release a high-risk model, scraped dataset, or generative system with a direct misuse pathway.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite the datasets, models, and evaluation tools used in our experiments throughout the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification:Yes. The paper does not introduce a new manually annotated dataset. GO-SSL task construction, prompts, rewards, training settings, and evaluation protocols are documented in the paper and appendix, and the code and accompanying documentation are publicly available at `https://github.com/SEU-VIPGroup/GO-SSL`.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The core method post-trains multimodal large language models with self-supervised reinforcement learning, as described in Sec. 3 and Sec. 4.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.