2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

SlotVLA: Towards Modeling of Object-Relation Representations in Robotic Manipulation

Taisei Hanyu[1,*], Nhat Chung[2,*,†], Huy Le[2], Toan Nguyen[2],
Yuki Ikebe[1], Anthony Gunderman[1], Duy Nguyen Ho Minh[3,7,8], Khoa Vo[1],
Tung Kieu[4], Kashu Yamazaki[5], Chase Rainwater[1], Anh Nguyen[6], Ngan Le[1,†]

***Abstract*— Inspired by how humans reason over discrete objects and their relationships, we explore whether compact object-centric and object-relation representations can form a foundation for multitask robotic manipulation. Most existing robotic multitask models rely on dense embeddings that entangle both object and background cues, raising concerns about both efficiency and interpretability. In contrast, we study object–relation-centric representations as a pathway to more structured, efficient, and explainable visuomotor control. Our contributions are two-fold. First, we introduce LIBERO+, a fine-grained benchmark dataset designed to enable and evaluate object-relation reasoning in robotic manipulation. Unlike prior datasets, LIBERO+ provides object-centric annotations that enrich demonstrations with box- and mask-level labels as well as instance-level temporal tracking, supporting compact and interpretable visuomotor representations. Second, we propose SlotVLA, a slot-attention–based framework that captures both objects and their relations for action decoding. It uses a slot-based visual tokenizer to maintain consistent temporal object representations, a relation-centric decoder to produce task-relevant embeddings, and an LLM-driven module that translates these embeddings into executable actions. Experiments on LIBERO+ demonstrate that object-centric slot and object-relation slot representations drastically reduce the number of required visual tokens, while providing competitive generalization. Together, LIBERO+ and SlotVLA provide a compact, interpretable, and effective foundation for advancing object–relation-centric robotic manipulation. Our full project is publicly available at https://slot-vla.github.io.**

I. Introduction

Recent advances in vision-language-action (VLA) modeling have significantly improved visuomotor control in robotics [1]–[4], integrating language conditions with visual cues to enable precise, multitask action prediction across numerous applications [5]. While many architectures such as OpenVLA [1], π_0 [2], ECoT [3], HPTs [4] have contributed to VLA pipelines, even employing a Large Language Model (LLM) for action decoding [1], [3], the vision encoder remains a critical bottleneck, as it serves as the perceptual foundation for action reasoning. In particular, pretrained encoders such as DINOv2 [6], SigLIP [7] are widely adopted in VLAs to produce a large number of visual tokens (e.g., 256 to 512), whose computational costs can become increasingly prohibitive as the number of visual tokens grows (as in Fig. 1a). While these embeddings are rich in information, they can entangle various information and even redundant background features, limiting interpretability and potentially obscuring task-relevant cues from the action decoder [8].

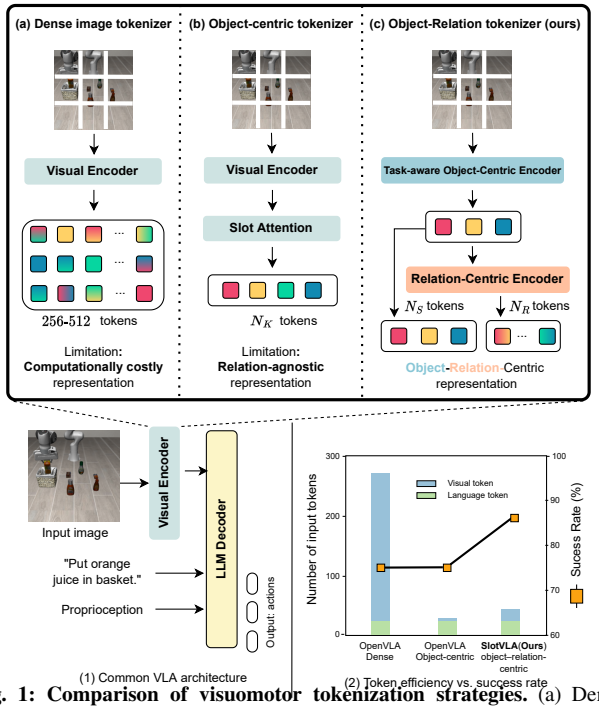

Fig. 1: Comparison of visuomotor tokenization strategies. (a) Dense tokenizers generate hundreds of tokens across the scene, leading to computationally costly representations. (b) Object-centric tokenizer yields N_K tokens, each representing an object. (c) Our object–relation-centric tokenizer yields N_S object tokens and N_R relation tokens, producing structured and efficient representations. Plot (2) shows that our method achieves higher success rates with fewer tokens compared to baselines on LIBERO-Goal.

Object-centric learning in computer vision has shown remarkable success in producing disentangled, interpretable representations that support generalization across tasks [9]–[12]. While promising, directly transferring object-centric methods to robotics is suboptimal due to the interactive nature of embodied manipulation tasks (shown in Fig. 1b of limited relational modeling among object slots). In robotic environments, scenes are often cluttered, and many objects are irrelevant to the current task. Simply increasing the number of slots does not help, as irrelevant objects or background elements may dominate the representations, preventing slots from consistently capturing meaningful entities [13], [14]

*These authors contributed equally.
†Correspondence: nhatchung14@gmail.com; thile@uark.edu
[1]University of Arkansas, USA [2]FPT Software AI Center, Vietnam
[3]University of Stuttgart, Germany [4]Aalborg University, Denmark
[5]Carnegie Mellon University, USA [6]University of Liverpool, UK
[7]German Research Center for Artificial Intelligence, Germany
[8]Max Planck Research School for Intelligent Systems, Germany

and confusing the action decoder. Therefore, instead of modeling all objects in the scene, the model must identify and focus on the task-relevant objects that the manipulator interacts with. Moreover, purely object-centric encodings fail to capture essential relational cues, most notably the gripper–object interactions. As a result, action decoders are forced to infer control parameters from incomplete relational signals. Unlike passive perception, robotic manipulation requires explicit modeling of both objects and their relations to the manipulator and surrounding scene [3], [15].

To facilitate the study and future research of object-relation representations for multitask robotic manipulation, we introduce **LIBERO+**, a fine-grained benchmark that extends LIBERO [16] with explicit object-focused annotations. Originally, while LIBERO provides a diverse set of manipulation tasks, like many existing datasets, it does not include detailed grounding at the object level. LIBERO+ fills this gap by adding box- and mask-level labels along with temporal tracking across RGB-D modalities, enriching demonstrations with structured information that supports object–relation reasoning. These annotations enable systematic design and evaluation of object–relation-centric visuomotor models and open opportunities for interpretable learning.

To address the modeling gap, we propose an object-relation-centric paradigm for robotic VLA models. Concretely, we introduce **`SlotVLA`**, a framework that employs slot attention with a task-aware filter to extract only relevant object representations and a relation encoder to capture their interactions, enabling relational reasoning under strict token budgets. As shown in Fig. 1c, the **`SlotVLA`** framework relies on an Object-centric Encoder that disentangles object features into a compact set of slots and filters them for task relevance, yielding concise and interpretable representations. The Relation Encoder then models their interactions, including those involving the robot gripper and the background context. Finally, both object- and relation-centric tokens are decoded into precise control actions.

In summary, our contributions are two-fold:

- **LIBERO+** dataset: a fine-grained benchmark emphasizing object–relation reasoning with RGB-D input and instance-level annotations for robotic manipulation.
- **`SlotVLA`**: an object-relation-centric VLA framework that combines object-centric slots with relation-centric tokens. The object slots capture disentangled entities from the environment and are filtered for task relevance, while the relation tokens (e.g., gripper-object interactions) encode task-aware interactions. Together, they yield a compact and interpretable representation for action decoding.

II. Related Works

VLA Learning in Robotic Manipulation. VLA learning has emerged as a powerful paradigm for instruction-following agents in various embodied tasks, including 3D scene reconstruction, navigation, cross-embodiment transfer, and, most notably, robotic manipulation [1]–[4]. As visual perception is a critical bottleneck, many recent architectures such as OpenVLA [1], π_0 [2], ECoT [3], HPTs [4] have leveraged strong pretrained vision encoders, like DINOv2 [6], SigLIP [7], that are effective at capturing diverse features. However, they can generate dense representations that intermingle object positions, affordances, and backgrounds, making it challenging for action decoders to isolate task-relevant signals [17]. Diverging from the recent VLA literature, our research considers a novel low-token perspective towards performing manipulation tasks. *Inspired by the remarkable success of object-centric learning in computer vision [9], [10], we explore whether a few rich semantic slots (i.e., object slots and object-relation slots) can offer a more efficient and interpretable foundation for robotic manipulation while isolating task-relevant objects out.*

Vision Token Reduction. Vision-language models (VLM) process numerous tokens, especially in multi-view [1] and video reasoning [8]. To reduce memory and computational costs, several token compression methods have been introduced. Approaches such as Token Merging [18], PruMerge [19], and TokenPacker [20] reduce redundancy by aggregating tokens, while models such as Qwen-VL [21] and MQT-LLaVA [22] employ Q-Former [23] or resampler modules [4] to construct fixed-length visual representations. However, these methods emphasize general-purpose compression rather than extracting task-relevant and interpretable structures, limiting their applicability to robotics. Meanwhile, slot-based learning has recently been leveraged to focus on modular, object-centric structures in visual reasoning [9], [10], [24], [25], using just a few semantic slots. *Unlike existing works, our `SlotVLA` extends the slot attention mechanism to robot manipulation tasks, particularly by improving upon object-centric slots to form object-scene interaction semantics, preserving their relational features for multitask manipulation control.*

Object-Centric Representation in Robot Manipulation This strategy is often realized by explicitly encoding objects as bounding boxes [26], keypoints, or poses [27], using grounding techniques to feed these tokens into the model [28]. More recently, object-centric information has also been expressed in text prompts with explicit coordinates [29], guiding the model's attention toward task-relevant entities through multimodal grounding. While this provides strong supervision and explicit grounding, it ties performance to the quality of external annotations and limits flexibility in capturing unannotated or relational aspects of the scene [30], [31]. In contrast, slot attention discovers object slots directly from raw inputs, enabling flexible and generalizable object representations. *Building on this, our SlotVLA formulation unifies object-centric slots with relation-centric tokens, where slots capture disentangled entities filtered for task relevance, and relation tokens encode task-aware interactions (e.g., gripper–object). This yields a compact, interpretable representation that enhances robot action decoding.*

III. LIBERO+: Data Curation

We build LIBERO+ as an extension of the LIBERO benchmark suite [16]. While LIBERO is well-suited for evaluat-

ing high-level visuomotor learning, it lacks explicit object-level supervision. To address this gap, LIBERO+ introduces finer-grained, object-centric annotations designed to support compact, interpretable, and low-token VLA representations. As LIBERO is a task suite built on top of robosuite [32], a non-trivial challenge lies not only in grounding natural language descriptions to object entities in the simulator, but also in representing each object holistically as a single "object" is often composed of multiple underlying assets (e.g., meshes, textures, physical handles), which are fragmented into disjointed parts, reducing both interpretability and utility for fine-grained visuomotor reasoning. To address this, we manually align low-level asset names with specific objects and label their natural linguistic references, ensuring consistency across entire trajectories. In addition, we unify semantic object masks and disambiguate multiple instances of the same object by leveraging preassigned asset names, producing coherent and interpretable masks that also form bounding boxes. Then, in order to provide task-relevant, active objects, we include task-specific naturalistic nouns for each demonstration to filter for relevant object labels. Specifically, as seen in Fig. 2, LIBERO+ augments the original demonstrations with both **box-level** and **mask-level** object annotations, as well as **instance-level temporal ID tracking** across both RGB and Depth modalities, as follows:

- `Bounding boxes` provide 2D spatial anchors that localize objects, serving as entry points for token extraction.
- `Object masks` offer pixel-level segmentations aligned with bounding boxes, preserving object boundaries and preventing feature entanglement with background pixels.
- `Instance-level temporal IDs` maintain consistent object identities across frames (e.g., basket1, plate1, plate2), enabling objects to be temporally tracked throughout a sequence and supporting long-horizon reasoning.
- `Depth map`, restricted to masked regions, provides per-pixel depth signals that encode occlusions and relative distances, which are critical for gripper–object reasoning.
- `Task-relevant objects` provide the set of objects explicitly mentioned in the task description. For example, given the task "robot put the bowl on top of the cabinet", the identified task-relevant objects are robot, bowl, and cabinet.

LIBERO+ includes four subsets, derived from the original LIBERO suite: LIBERO-Goal, LIBERO-Object, LIBERO-Spatial, and LIBERO-Long. Each subset is curated to emphasize different aspects of manipulation, while consistently providing structured object–relation annotations to support fine-grained reasoning. For temporally consistent action supervision, we retain LIBERO's native action labels but introduce a filtering step to remove redundant *no-op* actions.

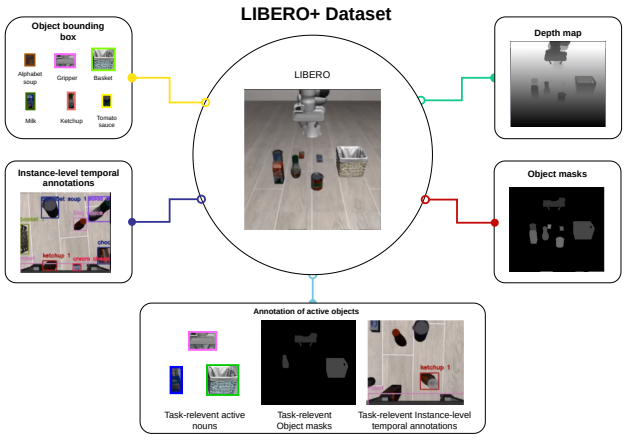

Fig. 2: Overview of the LIBERO+ dataset.

This refinement reduces idle-frame redundancy and sharpens the alignment between annotated objects and action-relevant dynamics. As a result, LIBERO+ yields compact yet semantically rich representations, where object-centric slots are directly grounded in action-relevant cues. This design ensures that LIBERO+ serves as a challenging and comprehensive testbed for advancing efficient, interpretable, and object-relation-centric VLA reasoning. The statistical summary of LIBERO+, constructed from four subsets: LIBERO-Object (L-Object), LIBERO-Goal (L-Goal), LIBERO-Spatial (L-Spatial), and LIBERO-Long (L-Long), is in Table I.

TABLE I: Statistics of LIBERO+. TR corresponds to Task-relevant objects.

Statistics	LIBERO+			
	L-Object	L-Goal	L-Spatial	L-Long
# Tasks	10	10	10	10
# Object Layouts	1	1	10	9
# Objects	12	7	11	29
# TR Objects	2–3	2–3	3–4	3–4
# Total Frames	72,063	54,779	47,253	84,896
# Total BBoxes	570,328	374,692	510,985	487,333
# TR BBoxes	285,912	130,814	221,468	257,105

IV. Methodology

A. Overview

The goal of `SlotVLA` is to achieve token-efficient visuomotor reasoning by transforming dense visual embeddings into compact object–relation representations. Conventional encoders produce hundreds of tokens (typically 256–512), which are computationally expensive for downstream reasoning. Object-centric representations address this overhead by focusing on discrete objects, but they overlook relational cues such as gripper–object interactions.

In contrast, our approach compresses the dense input into object-centric representations and applies a task-aware filtering mechanism that discards irrelevant objects in the environment. This yields a clearer representation with only a few slots (around 4), corresponding to the 2–4 task-relevant objects observed across LIBERO tasks (Table I). Additionally, the object slots are augmented with a small number of relation tokens to explicitly capture interactions among objects and with the manipulator, resulting in a representation

that is both compact and relationally expressive. The overall architecture of this framework is illustrated in Fig. 3.

B. Problem Formulation

Formally, let $\mathbf{V}_t = \{\mathbf{v}_t^1, \ldots, \mathbf{v}_t^N\}$ denote a N dense set of visual tokens/patches from the feature map encoded by a visual encoder from the image $\mathbf{I}_t$ at time t and $\mathbf{P} = \{\mathbf{p}^1, \ldots, \mathbf{p}^M\}$ represents the embeddings of language tokens of task description. Our goal is to learn a function $g_\phi(\cdot)$ that gives semantically-rich compact representation as:

$$\{\mathbf{S}_t, \mathbf{R}_t\} = g_\phi(\mathbf{V}_t, \mathbf{P}) \tag{1}$$

where $\mathbf{S}_t \in \mathbb{R}^{N_S \times d}$ denotes the object-centric representations with N_S tokens, and $\mathbf{R}_t \in \mathbb{R}^{N_R \times d}$ denotes the relation-centric representations that capture interactions among slots and with dense features using N_R tokens. Importantly, $N_S + N_R \ll N$, enabling downstream reasoning tasks to be carried out in a token-efficient manner. Given such visual token representations along with task embedding $\mathbf{P}$ and proprioception embedding $\mathbf{o}_t$, our final goal is to train an action decoding function $f_\theta(\cdot)$ that predicts action logits A_t over possible actions that a robot can take to solve the task:

$$A_t = f_\theta(\mathbf{S}_t, \mathbf{R}_t, \mathbf{P}, \mathbf{o}_t) \tag{2}$$

C. Task-Aware Object-Centric Encoder

As task scaling requires open-vocabulary object extraction at test time, we aim to reduce redundancy in dense tokens by emphasizing task-relevant representations. To this end, we design a slot-based encoder that compresses dense visual tokens into compact object-centric tokens, guided by language-based task filtering. The encoder comprises two modules: *(1) slot attention with temporal consistency*, which extracts object-centric slots from dense features and maintains their updates over time; and *(2) task-aware slot filtering*, which selects the slots most relevant to the manipulation task.

1) Slot Attention with Temporal Consistency: On top of the visual encoder, we employ slot attention [13] to map dense visual patches $\mathbf{V}_t \in \mathbb{R}^{N \times d}$ into a set of learnable slots $\tilde{\mathbf{S}}_t \in \mathbb{R}^{N_{\tilde{S}} \times d}$. This iterative attention process, implemented with a GRU [33], captures modular semantics [34], [35] and produces object-centric tokens:

$$\begin{aligned}
\tilde{\mathbf{a}}_{i,j} &= \frac{e^{\mathbf{a}_{i,j}}}{\sum_l e^{\mathbf{a}_{i,l}}}, \quad \text{where} \quad \mathbf{a} = \tfrac{1}{\sqrt{d}} k(\mathbf{V}_t) q(\tilde{\mathbf{S}})^\top \\
\mathbf{w}_{i,j} &= \frac{\tilde{\mathbf{a}}_{i,j}}{\sum_l \tilde{\mathbf{a}}_{l,j}} \\
\tilde{\mathbf{S}}'_t &= \mathrm{GRU}(\texttt{inputs} = \mathbf{w}^\top v(\mathbf{V}_t), \texttt{states} = \tilde{\mathbf{S}}_t).
\end{aligned} \tag{3}$$

where linear transformation heads $q(\cdot)$, $k(\cdot)$, $v(\cdot)$ are used to map learnable slots $\tilde{\mathbf{S}}_t$ and frame-wise feature maps $\mathbf{V}_t$.

To ensure *temporal consistency* in object identity, slots are initialized through a *slot carryover mechanism*,

$$\tilde{\mathbf{S}}_t^{(0)} = \begin{cases} \mathrm{RandomInit}(), & t = 0 \\ \tilde{\mathbf{S}}_{t-1}^{(T)}, & t > 0, \end{cases} \tag{4}$$

where T is the number of refinement steps per frame. Thus, slots start randomly at $t = 0$ and are propagated from the previous timestep otherwise.

2) Task-Aware Slot Filter: While leveraging slots is practical for robotic manipulation, not all slots are relevant to a given task. We therefore introduce a *task-aware slot filter* to select N_S slots from $N_{\tilde{S}}$ using bidirectional cross-attention (BCA) [36], followed by a transformer layer (Trans), by estimating the relevance scores of the tokens with respect to the task description. Specifically, the task-aware slot filter computes the relevant scores π_t at time t between the object-centric slots $\tilde{\mathbf{S}}'_t$ from the GRU and the embeddings of language tokens $\mathbf{P}$:

$$\pi_t = \mathrm{Trans}(\mathrm{BCA}(\tilde{\mathbf{S}}', \mathbf{P})). \tag{5}$$

Then, we retain N_S most relevant slots during training and inference with a $\mathrm{Top}_k(\cdot)$ function and filter out the rest. Given the the object-centric slots $\tilde{\mathbf{S}}'_t$, we extract a refined subset $\mathbf{S}_t = \{\tilde{\mathbf{s}}_t^1, \ldots, \tilde{\mathbf{s}}_t^{N_S}\}$ to serve as task-aware object-centric tokens, where $N_S \leq N_{\tilde{S}}$, based on their scores:

$$\mathbf{S}_t = \{\tilde{\mathbf{s}}_t^i \mid i \in \mathrm{Top}_k(\pi_t)\}. \tag{6}$$

D. Relation-Centric Encoder

In robotic tasks, understanding of interactions between objects is crucial for planning grasps and manipulations. Although object-centric tokens localize items of interest, they do not inherently capture relationships among objects (e.g., the gripper and objects). Specifically, we model relations using learnable queries, $\tilde{\mathbf{R}}_t$. These queries integrate information from dense visual patches ($\mathbf{V}_t$) and object-centric tokens ($\mathbf{S}_t$) using a Cross-Attention Block (CAB), which consists of multi-head cross-attention and feed-forward layers. The CAB applies visual conditioning to process the patches and slot-based conditioning to process the object tokens.

$$\mathbf{R}_t = \mathrm{CAB}(\mathrm{CAB}(\tilde{\mathbf{R}}_t, \mathbf{V}_t), \mathbf{S}_t). \tag{7}$$

E. Action Decoding

Inspired by [8], we employ LoRA [37] to integrate the object and relation tokens into a VLA framework that leverages an LLM action decoder, which we denote as $\mathrm{AD}_{\theta_{\text{LoRA}}}(\cdot)$. The task-aware object-centric tokens $\mathbf{S}_t$ and the relation tokens $\mathbf{R}_t$ are concatenated together with the language embeddings $\mathbf{P}$ from the task description and the robot proprioception $\mathbf{o}_t$ to form a multimodal input sequence. Then, we define the action prediction as a classification problem, where each control dimension is discretized via binning [1] or tokenization [38].

$$a_t = \underset{k}{\mathrm{argmax}}\, \mathrm{AD}_{\theta_{\text{LoRA}}}([\mathbf{S}_t\ ; \mathbf{R}_t\ ; \mathbf{P}\ ; \mathbf{o}_t]) \tag{8}$$

where $[;]$ denotes the concatenation operation, and the action a_t is obtained by greedy decoding over action logits A_t. Recall that $N_S + N_R \ll N$; thus, this approach is much token efficient than using $[\mathbf{V}_t; \mathbf{P}, \mathbf{o}_t]$ as input.

F. Objective Functions

We adopt a two-stage training strategy: first, we supervise the Task-aware Object-Centric Encoder (Section IV-C), and then we fine-tune the Relation-Centric Encoder (Section IV-D) jointly with the Action Decoding (Section IV-E).

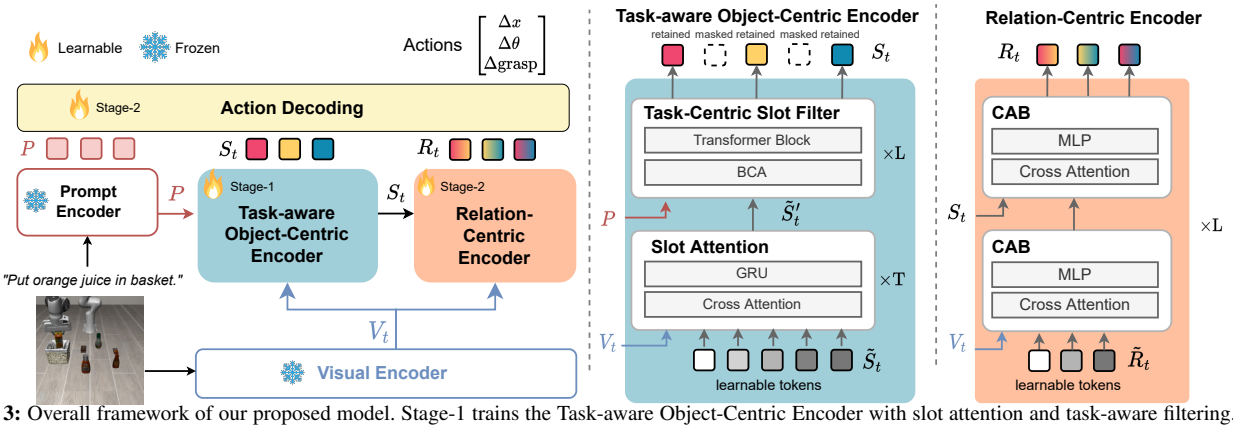

Fig. 3: Overall framework of our proposed model. Stage-1 trains the Task-aware Object-Centric Encoder with slot attention and task-aware filtering. Stage-2 freezes Stage-1 parameters and introduces the Relation-Centric Encoder, enabling relational reasoning for final action decoding.

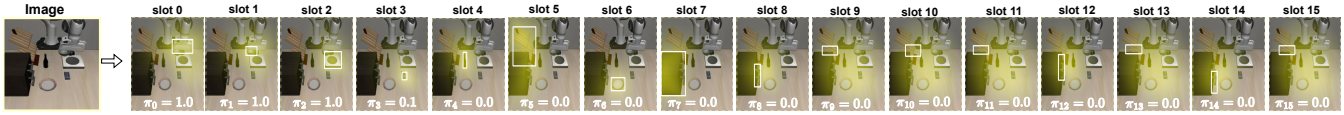

Fig. 4: Slot decomposition result. Task query: "*Put the bowl on the stove*". Task-relevant slots correctly bind to objects, while irrelevant slots scatter.

1) Stage – 1 Training: We first train the Task-aware Object-Centric Encoder with two objectives that include: (a) slot-attention supervision and (b) task-aware slot filter supervision (Fig. 3: Stage-1).

$$\mathcal{L}_{\text{slot-enc}} = \lambda_{\text{slot-attn}}\,\mathcal{L}_{\text{slot-attn}} + \lambda_{\text{track}}\,\mathcal{L}_{\text{track}} + \lambda_{\text{int}}\,\mathcal{L}_{\text{int}}. \quad (9)$$

where, given $N_{\tilde{S}}$ predicted slots and N_G ground-truth objects per frame with the requirement of $N_G < N_{\tilde{S}}$, we align predicted slots with ground-truth objects at each timestep via Hungarian matching [39] using box-based costs.

(a) Slot Attention Supervision: We supervise the slot attention with the bounding boxes, objectness, and masks:

$$\mathcal{L}_{\text{slot-attn}} = \lambda_{\text{box}}\,\mathcal{L}_{\text{box}} + \lambda_{\text{obj}}\,\mathcal{L}_{\text{obj}} + \lambda_{\text{seg}}\,\mathcal{L}_{\text{seg}}. \quad (10)$$

Here, $\mathcal{L}_{\text{box}}$ is a standard DETR-style box loss [40], $\mathcal{L}_{\text{obj}}$ is a binary cross-entropy (BCE) loss that assigns 1 to matched slots and 0 to unmatched ones, thereby encouraging slots to activate only when corresponding to real objects. $\mathcal{L}_{\text{seg}}$ is a pixel-wise BCE that enforces finer alignment between predicted and ground-truth instance masks. Note that the supervising signals are coming from our introduced LIBERO+ dataset. Then, we enforce temporal consistency with a tracking loss, aligning slots with the same object across frames using bounding box and mask annotations:

$$\mathcal{L}_{\text{track}} = -\sum_{(i,t)} \log \frac{\sum_{(i',t')\in\mathcal{P}(i,t)} \exp\Big(\text{sim}(\mathbf{s}_t^i, \mathbf{s}_{t'}^{i'})/\tau\Big)}{\sum_{(j,t'')\in\mathcal{P}(i,t)\cup\mathcal{N}(i,t)} \exp\Big(\text{sim}(\mathbf{s}_t^i, \mathbf{s}_{t''}^{j})/\tau\Big)}, \quad (11)$$

with $\mathbf{s}_t^i = (\tilde{\mathbf{S}}_t^{(T)})_i$, where $\mathcal{P}(i,t)$ denotes slots of the same object across nearby frames (positives), and $\mathcal{N}(i,t)$ denotes slots from other objects or different videos (negatives).

(b) Task-Aware Slot Filter Supervision: We directly supervise the task relevence score π_t^i for each slot i from Eq. 5 with a class-imbalanced BCE:

$$\mathcal{L}_{\text{int}} = \frac{1}{N}\sum_{i,t} w(\hat{u}_{i,t})\,\text{BCE}(\hat{u}_{i,t}, \pi_t^i), \quad (12)$$

where $\hat{u}_{i,t} \in \{0,1\}$ indicates whether slot i is relevant to the instruction at time t. The weight function $w(\cdot)$ up-weights positive labels to address class imbalance (e.g., $w(1) = 2.0$, $w(0) = 1.0$ in our experiments).

2) Stage – 2 Training: We used cross-entropy (CE) loss between the predicted action logits A_t before the argmax operation and the ground-truth one-hot action label $\hat{A}_t$ to train the Action Decoder together with Relation-Centric Encoder (Fig. 3: Stage-2).

$$\mathcal{L}_{CE} = -\sum_{t=1}^{L} \hat{A}_t \log A_t, \quad (13)$$

where L is the total number of action steps. Note that Stage-1 parameters are frozen at this stage.

V. EXPERIMENTS

A. Benchmarks and Evaluation Strategies

We conduct experiments on LIBERO+ benchmark (see details in Section III). The diversity across its four subsets—in terms of layouts, object counts, and task complexity—enables us to evaluate the models' ability to balance spatial precision, relational reasoning, and temporal consistency.

Baselines & Compared Methods. We adopt *OpenVLA* [1] as the initial baseline, as it offers a strong visuomotor policy but is computationally challenging. Our study emphasizes visual tokenization strategies, which are orthogonal to the choice of pretrained visuomotor backbones; accordingly, we limit our comparisons to OpenVLA as a representative reference. In summary, OpenVLA provides a fair starting point for evaluating our object- and object-relation–based approaches for the following comparison: *OpenVLA* [1]:

TABLE II: Benchmark on LIBERO+ consisting of four subsets from LIBERO. Highest results are **bolded**, second-highest are underlined. No. Token indicates the number of tokens used. The tasks, numbered from 1 to 10, are specific to each subset and sorted alphabetically.

	LIBERO-Goal			LIBERO-Spatial			LIBERO-Object			LIBERO-Long		
	OpenVLA	OC	ORC	OpenVLA	OC	ORC	OpenVLA	OC	ORC	OpenVLA	OC	ORC
Average	0.77	0.77	**0.86**	**0.72**	0.48	0.60	0.70	0.90	**0.91**	**0.56**	0.12	0.31
No. Token	256 (1×)	4 (64×)	20 (13×)	256 (1×)	4 (64×)	28 (9×)	256 (1×)	4 (64×)	28 (9×)	256 (1×)	4 (64×)	28 (9×)
GFLOPs	2,112 (1×)	561 (4×)	697 (3×)	2,112 (1×)	568 (4×)	723 (3×)	2,112 (1×)	568 (4×)	723 (3×)	2,112 (1×)	568 (4×)	723 (3×)
Task 1	0.60	0.90	0.70	0.82	0.70	0.65	0.75	0.85	0.95	0.75	0.50	0.20
Task 2	0.95	0.50	0.75	0.95	0.90	0.20	0.70	0.80	0.80	0.90	0.00	0.00
Task 3	0.70	0.90	1.00	0.82	0.00	0.40	0.85	1.00	1.00	0.55	0.10	0.40
Task 4	0.50	0.75	1.00	0.92	0.75	0.85	0.45	1.00	1.00	0.55	0.10	0.15
Task 5	0.95	1.00	0.95	0.70	0.80	0.90	0.95	1.00	0.70	0.40	0.15	0.40
Task 6	0.90	0.95	1.00	0.85	0.70	0.75	0.60	0.95	0.95	0.75	0.05	0.65
Task 7	0.75	0.35	0.85	0.88	0.70	0.65	0.45	0.90	1.00	0.40	0.10	0.55
Task 8	0.90	1.00	0.50	0.73	0.15	0.45	0.80	0.95	0.85	0.60	0.05	0.20
Task 9	0.90	0.60	0.90	0.82	0.00	0.80	0.50	0.85	0.95	0.35	0.00	0.10
Task 10	0.52	0.85	1.00	0.62	0.00	0.35	0.70	0.70	0.90	0.40	0.15	0.40

TABLE III: Ablation study on Task-Aware Slot Filtering. Object-centric slots (OC) and object–relation-centric slots (ORC) are compared. ✓ indicates that filtering is included, while ✗ indicates that filtering is not included. The tasks, numbered from 1 to 10 are specific to each subset and sorted alphabetically.

	LIBERO-Goal				LIBERO-Spatial				LIBERO-Object				LIBERO-Long			
	OC		ORC		OC		ORC		OC		ORC		OC		ORC	
Language	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓	✗	✓
Average	0.77	0.77	0.72	**0.86**	0.53	0.48	**0.60**	**0.60**	0.76	0.90	**0.91**	**0.91**	0.11	0.07	0.12	**0.31**
Task 1	0.75	0.90	0.60	0.70	0.75	0.70	0.80	0.65	0.95	0.85	0.75	0.95	0.05	0.15	0.50	0.20
Task 2	1.00	0.50	0.40	0.75	0.70	0.90	0.20	0.20	0.40	0.80	0.85	0.80	0.00	0.00	0.00	0.00
Task 3	0.95	0.90	0.70	1.00	0.15	0.00	0.75	0.40	1.00	1.00	0.85	1.00	0.25	0.00	0.10	0.40
Task 4	0.60	0.75	0.75	1.00	0.20	0.75	0.95	0.85	1.00	1.00	1.00	1.00	0.05	0.00	0.10	0.15
Task 5	1.00	1.00	1.00	0.95	0.90	0.80	0.70	0.90	0.85	1.00	0.85	0.70	0.10	0.15	0.15	0.40
Task 6	0.90	0.95	0.85	1.00	0.55	0.70	0.50	0.75	0.95	0.95	0.95	0.95	0.25	0.05	0.05	0.65
Task 7	0.35	0.35	0.15	0.85	0.40	0.05	0.55	0.65	1.00	0.90	1.00	1.00	0.00	0.00	0.10	0.55
Task 8	0.40	1.00	0.80	0.50	0.70	0.15	0.80	0.45	0.80	0.95	1.00	0.85	0.05	0.00	0.05	0.20
Task 9	0.90	0.60	1.00	0.90	0.50	0.00	0.80	0.80	0.45	0.85	1.00	0.95	0.10	0.00	0.00	0.10
Task 10	0.80	0.85	0.80	1.00	0.40	0.00	0.40	0.35	0.15	0.70	0.80	0.90	0.30	0.35	0.15	0.40

TABLE IV: Ablation study on number of object tokens.

Method	**4 tokens**	**8 tokens**	**16 tokens**
OC	0.77	0.65	0.77
ORC	0.86	0.74	0.72

dense-token policy baseline with full token input, *Object-Centric Slots (OC)*: independent object representations without explicit modeling of interactions (i.e. `SlotVLA` without Relation Encoder), and *Object-Relation-Centric Slots (ORC)*: our proposed object-relational tokenization that captures object and object-context interactions (i.e. `SlotVLA`).

Evaluation Configuration. We use 16 slots for L-Goal (simpler layouts) and 24 slots for L-Object, L-Spatial, and L-Long. After filtering, only 4 task-relevant object slots are retained, with relation slots matched to the initial object slots. This setup enables fair comparison without filtering and highlights the limitation of L-Long (up to 29 objects), which cannot be fully covered unless all task-relevant slots are retained. Training is performed with a batch size of 64 on 3 A100 GPUs for 50k iterations.

Evaluation Protocol. Each method is trained jointly across tasks within a subset and evaluated on 20 rollouts per task. We report average success rate per subset, along with ablation studies on slot scaling and temporal tracking.

B. Main Results

Table II reports success rates across the four subsets in LIBERO+. We can observe that OC yields comparable or slightly higher averages than OpenVLA in L-Goal (0.77) and L-Object (0.90), suggesting that compact object-centric slots can support reasoning when tasks are primarily object-driven with simple layouts. ORC provides additional gains in L-Goal (0.86) and L-Object (0.91) by explicitly encoding relational structure, though its advantage is less clear in L-Spatial (0.60) and it underperforms OpenVLA in L-Long (0.31 vs. 0.56). Thus, while relation-centric modeling is beneficial for object-focused tasks, its impact is more limited for spatially complex and long-horizon settings. A key dimension of this comparison is the trade-off between token number and computational cost. Slot-based models reduce the token count by more than an order of magnitude compared to OpenVLA (e.g., 4–28 slots vs. 256 dense tokens). This efficiency helps explain their competitiveness on L-Goal and L-Object, since these subsets involve only a few active objects where compact slots capture the essential cues without wasted computation. However, the same compactness becomes a limitation on L-Spatial and L-Long, where the available object slots (4) are so much fewer than the number of distinct objects (up to 29 in L-Long) that would also necessitate better relational modeling. Otherwise, not all objects can be represented, and fine-grained spatial layouts remain hard to capture, leading to degraded performance compared to the dense baseline. The key takeaway is that slot-based tokenization offers strong efficiency gains but its effectiveness depends on task complexity. OC and ORC perform well on simpler tasks (L-Goal, L-Object) by

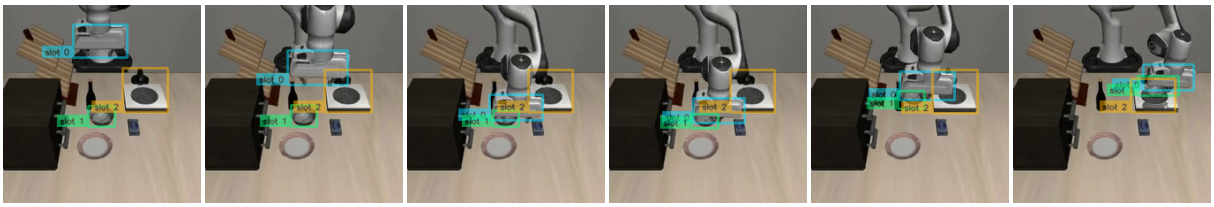

Fig. 5: Trajectory demonstration in simulation from exocentric views. Task query: "Put the bowl on the stove".

TABLE V: Ablation study on the effect of temporal consistency.

Method	Temporal Consistency ✗	✓
OC	0.38	0.77
ORC	0.40	0.86

focusing on task-relevant objects and gripper positions. OC, however, struggles with changing layouts and many objects (L-Spatial, L-Long), failing especially when filtered to only four slots. ORC better captures relational reasoning but still falters when object sets grow large, as in L-Long. Thus, while compact slots are efficient and competitive, scaling to complex layouts, long horizons remains an open challenge.

C. Task-Aware Slot Filtering Ablation

We ablated the slot filtering in Sec. IV-C.2 with two models:

- **With Slot Filtering (default):** only a small number of slots are passed downstream. OC retains 4 object slots; ORC uses 4 object slots plus 16–24 relation slots depending on the subset (Sec. V-A).
- **Without Slot Filtering:** all extracted slots are used (L-Goal 16, L-Object 24, L-Spatial 24, L-Long 24).

Table III shows that both OC and ORC perform well on subsets of simple layouts (L-Goal, L-Object), where reasoning depends on picking and placing a few task-relevant objects. Performance drops on L-Spatial and L-Long, which involve changing layouts and many more objects. In L-Long, OC remains somewhat feasible with 24 slots (0.11) but collapses when filtered to 4 (0.07), showing the limits of non-relational modeling. ORC, though better for relational reasoning, also suffers with only 24 slots when scenes contain up to 29 objects, as its module cannot recover unless task-relevant slots are consistently included. We can observe here that by filtering for irrelevant objects while focusing on relevant ones, ORC can perform more decently on L-Long. Overall, this indicates that slot-based models are effective in object-centric settings with simple object layouts, but scaling to complex, long-horizon tasks would require both relevant object coverage and strong relational reasoning.

D. Ablations and Insights

Our ablation study is conducted on L-Goal for both OC and ORC with task-aware slot filtering enabled.

Number of object slots. Table IV reports results on L-Goal when varying the number of object slots. For OC, performance is the same at 4 and 16 tokens (0.77) but drops at 8 tokens (0.65), suggesting that simply scaling slot numbers does not yield consistent gains, but rather adds noise to the system because of increased number of irrelevant objects. For ORC, performance is strongest at 4 tokens (0.86) and decreases slightly as more slots are added (0.74 at 8, 0.72 at 16). This indicates that relational encoding enables the use of a small slot budget, but adding more slots may dilute relational grounding or introduce distractors in simpler tasks like L-Goal, where only a few objects are active.

Temporal consistency. Table V evaluates the effect of temporal consistency in action decoding. Without it, both OC (0.38) and ORC (0.40) degrade sharply, even falling below the dense OpenVLA baseline (0.77 on L-Goal). With consistency, performance improves markedly (0.77 for OC, 0.86 for ORC). The larger gap for OC shows its sensitivity to identity drift, while ORC is more robust but still affected. These results highlight that slot-based models depend not only on representation quality but also on stable slot identities; without consistency, the Action Decoder must re-ground objects at every frame, undermining task success.

E. Qualitative Results.

We provide qualitative analysis on the L-Goal task "Put the bowl on the stove".

Slot decomposition. Fig. 4 shows how an input image is decomposed into slots. Each slot specializes to a distinct object or region, assigning high attention to task-relevant objects such as the gripper (slot 0), bowl (slot 1), or stove (slot 2). Irrelevant background regions receive negligible weights, indicating that the slot-based encoder factorizes scene into interpretable entities while filtering distractors.

Trajectory prediction with stable slots over time. Fig. 5 shows the predicted trajectories of task-relevant objects over time. Slots maintain consistent identities across frames (e.g., slot 0/1 track the gripper and bowl, slot 2 the stove), demonstrating temporal consistency for both dynamic and static objects. This stability enables the action decoder to generate coherent predictions as the robot grasps the bowl and places it on the stove, highlighting how slot-based representations capture object semantics while preserving identity over time.

VI. Discussion

As the first investigation of slot-based VLA, our results highlight both its potential and limitations for visuomotor reasoning. Compact slots in OC and ORC reduce token count and FLOPs by 3–4× compared to dense baselines while maintaining competitive accuracy on simpler tasks (L-Goal, L-Object), revealing an efficiency–performance trade-off when only a few objects matter. Increasing slot numbers does not guarantee gains: OC accuracy stagnates or declines, and ORC weakens as additional slots dilute relational grounding. Temporal consistency is also critical; without it, both models degrade substantially, indicating that stable slot identities are essential for action decoding.

VII. Conclusion

We introduced **LIBERO+**, an object-centric benchmark, and `SlotVLA`, a slot-based framework for structured visuomotor control. By using compact object and relation slots, our method reduces token usage by an order of magnitude compared to dense-token baselines while maintaining competitive performance. The results suggest that object-relation-centric tokenization provides an effective balance of efficiency, interpretability, and performance for multitask robotic manipulation. We hope LIBERO+ and `SlotVLA` serve as a foundation for future research on structured and trustworthy embodied AI.

Limitations. `SlotVLA` performs well in simple scenes, but scaling to complex layouts may require finer relational modeling with greater relational slot capacity. While full supervision enables trustworthy deployment, it is annotation-intensive and is more difficult to scale with real data.

Broader Impacts. `SlotVLA`'s slot-based design improves interpretability and efficiency, supporting safer embodied AI under limited compute, as object–relation grounding and structured supervision can further enhance trustworthy and collaborative human-AI systems, e.g. collision avoidance.

References

[1] M. J. Kim, K. Pertsch *et al.*, "Openvla: An open-source vision-language-action model," in *CoRL*, 2025.

[2] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[3] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine, "Robotic control via embodied chain-of-thought reasoning," in *CoRL*, 2024.

[4] L. Wang, X. Chen, J. Zhao, and K. He, "Scaling proprioceptive-visual learning with heterogeneous pre-trained transformers," in *NeurIPS*, 2024.

[5] Y. Ma, Z. Song, Y. Zhuang, J. Hao, and I. King, "A survey on vision-language-action models for embodied ai," *CoRR*, vol. abs/2405.14093, 2024.

[6] M. Oquab, T. Darcet *et al.*, "DINOv2: Learning robust visual features without supervision," *TMLR*, 2024.

[7] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *ICCV*, 2023.

[8] T. Tian, B. Li, X. Weng, Y. Chen, E. Schmerling, Y. Wang, B. Ivanovic, and M. Pavone, "Tokenize the world into object-level knowledge to address long-tail events in autonomous driving," in *CoRL*, 2024.

[9] J. Jiang, F. Deng, G. Singh, M. Lee, and S. Ahn, "Slot state space models," in *NeurIPS*, 2024.

[10] C. Kung, S. Lu, Y. Tsai, and Y. Chen, "Action-slot: Visual action-centric representations for multi-label atomic activity recognition in traffic scenes," in *CVPR*, 2024.

[11] Y. Zhu, A. Joshi, P. Stone, and Y. Zhu, "Viola: Imitation learning for vision-based manipulation with object proposal priors," in *CoRL*, 2023.

[12] J. Shi, J. Qian, Y. J. Ma, and D. Jayaraman, "Composing pre-trained object-centric representations for robotics from "what" and "where" foundation models," in *ICRA*, 2024.

[13] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf, "Object-centric learning with slot attention," in *NeurIPS*, 2020.

[14] N. Heravi, A. Wahid, C. Lynch, P. Florence, T. Armstrong, J. Tompson, P. Sermanet, J. Bohg, and D. Dwibedi, "Visuomotor control in multi-object scenes using object-aware representations," *arXiv preprint arXiv:2205.06333*, 2022.

[15] W. Cai, Y. Ponomarenko, J. Yuan, X. Li, W. Yang, H. Dong, and B. Zhao, "Spatialbot: Precise spatial understanding with vision language models," *arXiv preprint arXiv:2406.13642*, 2024.

[16] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," in *NeurIPS*, 2023.

[17] Y. Lin, A. Zeng, S. Song, P. Isola, and T. Lin, "Learning to see before learning to act: Visual pre-training for manipulation," in *ICRA*, 2020.

[18] C. Tran, D. MH Nguyen, M.-D. Nguyen, T. Nguyen, N. Le, P. Xie, D. Sonntag, J. Y. Zou, B. Nguyen, and M. Niepert, "Accelerating transformers with spectrum-preserving token merging," in *NeurIPS*, 2025.

[19] Y. Shang, M. Cai, B. Xu, Y. J. Lee, and Y. Yan, "Llava-prumerge: Adaptive token reduction for efficient large multimodal models," *arXiv preprint arXiv:2403.15388*, 2024.

[20] W. Li, Y. Yuan, J. Liu, D. Tang, S. Wang, J. Qin, J. Zhu, and L. Zhang, "Tokenpacker: Efficient visual projector for multimodal llm," *arXiv preprint arXiv:2407.02392*, 2024.

[21] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[22] W. Hu, Z.-Y. Dou, L. H. Li, A. Kamath, N. Peng, and K.-W. Chang, "Matryoshka query transformer for large vision-language models," *NeurIPS*, 2024.

[23] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *ICML*, 2023.

[24] R. Qian, S. Ding, and D. Lin, "Rethinking image-to-video adaptation: An object-centric perspective," *CoRR*, vol. abs/2407.06871, 2024.

[25] K. Vo, T. Phan, K. Yamazaki, M. Tran, and N. Le, "Henasy: Learning to assemble scene-entities for interpretable egocentric video-language model," in *NeurIPS*, 2025.

[26] C. Devin, P. Abbeel, T. Darrell, and S. Levine, "Deep object-centric representations for generalizable robot learning," in *ICRA*, 2018.

[27] S. Tyree, J. Tremblay, T. To, J. Cheng, T. Mosier, J. Smith, and S. Birchfield, "6-dof pose estimation of household objects for robotic manipulation: An accessible dataset and benchmark," in *IROS*, 2022.

[28] P. Li, Y. Wu, Z. Xi, W. Li, Y. Huang, Z. Zhang, Y. Chen, J. Wang, S.-C. Zhu, T. Liu *et al.*, "Controlvla: Few-shot object-centric adaptation for pre-trained vision-language-action models," *arXiv preprint arXiv:2506.16211*, 2025.

[29] J. Wen, Y. Zhu, M. Zhu, J. Li, Z. Xu, Z. Che, C. Shen, Y. Peng, D. Liu, F. Feng *et al.*, "Object-centric instruction augmentation for robotic manipulation," in *ICRA*, 2024.

[30] A. Agostini and D. Lee, "Efficient state abstraction using object-centered predicates for manipulation planning," *arXiv preprint arXiv:2007.08251*, 2020.

[31] W. Goodwin, S. Vaze, I. Havoutis, and I. Posner, "Semantically grounded object matching for robust robotic scene rearrangement," in *ICRA*, 2022.

[32] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín, "robosuite: A modular simulation framework and benchmark for robot learning," *CoRR*, vol. abs/2009.12293, 2020.

[33] K. Cho, B. van Merrienboer *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*, 2014.

[34] J. Xu, S. De Mello, S. Liu, W. Byeon, T. Breuel, J. Kautz, and X. Wang, "Groupvit: Semantic segmentation emerges from text supervision," in *CVPR*, 2022.

[35] B. Jia, Y. Liu, and S. Huang, "Improving object-centric learning with query optimization," in *ICLR*, 2023.

[36] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang *et al.*, "Grounded language-image pre-training," in *CVPR*, 2022, pp. 10 965–10 975.

[37] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," in *ICLR*, 2022.

[38] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "Fast: Efficient action tokenization for vision-language-action models," *arXiv preprint arXiv:2501.09747*, 2025.

[39] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, 1955.

[40] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020.