Compressor-VLA: Instruction-Guided Visual Token Compression for Efficient Robotic Manipulation

Juntao Gao[1], Feiyang Ye[2,†], and Jing Zhang[1,†]
[1]*School of Information Science and Technology, Beijing University of Technology, Beijing, China*
[2]*Li Auto Inc.*
gaojt_2@emails.bjut.edu.cn, feiyang.ye.uts@gmail.com, zhj@bjut.edu.cn

***Abstract*—Vision-Language-Action (VLA) models have emerged as a powerful paradigm in Embodied AI. However, the significant computational overhead of processing redundant visual tokens remains a critical bottleneck for real-time robotic deployment. While standard token pruning techniques can alleviate this, these task-agnostic methods struggle to preserve task-critical visual information. To address this challenge, simultaneously preserving both the holistic context and fine-grained details for precise action, we propose Compressor-VLA, a novel hybrid instruction-conditioned token compression framework designed for efficient, task-oriented compression of visual information in VLA models. The proposed Compressor-VLA framework consists of two token compression modules: a Semantic Task Compressor (STC) that distills holistic, task-relevant context, and a Spatial Refinement Compressor (SRC) that preserves fine-grained spatial details. This compression is dynamically modulated by the natural language instruction, allowing for the adaptive condensation of task-relevant visual information. Experimentally, extensive evaluations demonstrate that Compressor-VLA achieves a competitive success rate on the LIBERO benchmark while reducing FLOPs by 59% and the visual token count by over 3x compared to its baseline. The real-robot deployments on a dual-arm robot platform validate the model's sim-to-real transferability and practical applicability. Moreover, qualitative analyses reveal that our instruction guidance dynamically steers the model's perceptual focus toward task-relevant objects, thereby validating the effectiveness of our approach.**

***Index Terms*—Vision-Language-Action Models, Robot Manipulation, Token Compression, Efficient Learning**

I. Introduction

The paradigm of Vision-Language-Action (VLA) models, which directly maps raw sensory inputs and language commands to low-level robot actions, represents a significant advancement in the pursuit of general-purpose robotic agents [1], [2]. Seminal works such as RT-1 [1], RT-2 [3], Gato [6], and more recent open-source efforts like OpenVLA [7] and Octo [8], have demonstrated that pre-training on large-scale, diverse multimodal datasets endows these models with remarkable generalization capabilities, simplifying the traditional, decoupled "perception-planning-control" pipeline [9], [10]. Typically, these architectures rely on a high-capacity Vision Transformer (ViT) for perception, a Large Language Model (LLM) for reasoning, and an action head for control. Crucially, the ViT backbone grounds the model in the physical world by deconstructing images into grids of patches and projecting them into long sequences of hundreds or even thousands of visual tokens.

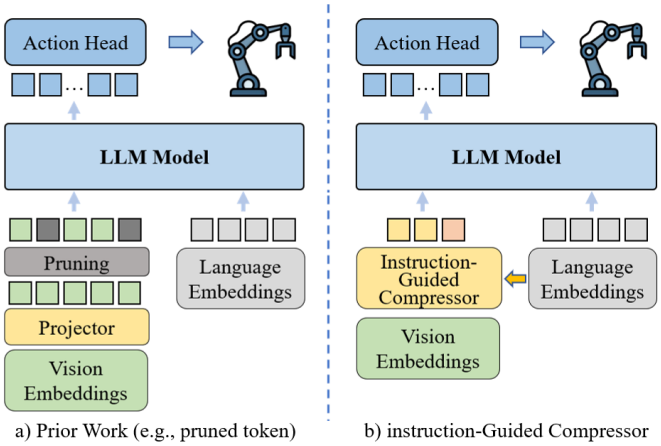

Fig. 1. Comparison of visual token reduction strategies. (a) Prior pruning-based methods remove low-scoring tokens directly. (b) Compressor-VLA forms a compact task-conditioned token set through two guided pathways, with STC for global context compression and SRC for local detail preservation.

This rich, high-dimensional representation encodes essential fine-grained details and complex spatial relationships, forming the perceptual bedrock upon which all subsequent planning and decision-making are built.

However, this high-dimensional perceptual reliance imposes significant computational demands. State-of-the-art VLAs typically employ Vision Transformers [11] to encode image frames into long token sequences, incurring substantial FLOPs and memory costs during LLM processing. This overhead leads to inference latencies that hinder real-time robot manipulation [12]. Furthermore, raw visual inputs contain task-irrelevant information (e.g., background clutter), which wastes resources and introduces noise that may degrade policy quality [13], [14].

To address this efficiency challenge, a common approach is token reduction, which typically involves a pruning layer that discards tokens based on task-agnostic importance scores [13]–[15]. While effective at reducing token count, this "hard" selection strategy risks losing crucial information. More importantly, this entire approach fails to leverage the language instruction to guide the compression process. This is suboptimal for robotics, where the relevance of visual cues is highly dynamic and dependent on the given instruction. Other related methods include token merging [16] and query-based compression [17], but they often share the same task-agnostic limitation.

In this work, we propose Compressor-VLA, a hybrid instruction-guided token compression framework for VLA models, as illustrated in Fig. 1(b). Instead of directly pruning tokens,

The work done during the collaboration with Li Auto Inc.
† Corresponding Author.

our key idea is to *reconstruct* a compact task-conditioned representation through two complementary pathways: the **Semantic Task Compressor (STC)**, which summarizes task-relevant global context, and the **Spatial Refinement Compressor (SRC)**, which preserves manipulation-relevant local details. This design enables more efficient and context-aware visual compression for downstream action generation.

Our main contributions are summarized as follows: (i) we propose Compressor-VLA, a hybrid instruction-conditioned token compression framework that integrates global and local pathways for efficient, task-oriented visual processing in VLA models; (ii) extensive experiments on the LIBERO benchmark and real-world dual-arm robot deployments demonstrate that our approach achieves a strong efficiency-performance trade-off, reducing FLOPs by 59% while maintaining a competitive 97.3% success rate; and (iii) we provide comprehensive qualitative and quantitative analyses validating the effectiveness of our instruction-guided compression strategy and the complementary roles of the two pathways.

II. Related Work

Vision-Language-Action Models. Drawing from pretrained vision, LLM, and VLM foundations, VLAs have emerged to process multimodal inputs-visual observations and language instructions to generate robotic actions for embodied tasks [1]–[7], [18]. RT-1 [1] and Octo [6] utilize a transformer-based policy integrating diverse data, such as robot trajectories across varied tasks, objects, environments, and embodiments. Recent studies [3], [7], [18] harness pretrained VLMs to produce robotic actions, tapping into world knowledge from large-scale vision-language datasets. For example, RT-2 [3] and OpenVLA [7] model actions as tokens within the language vocabulary, whereas RoboFlamingo [18] adds a separate policy head for action prediction. OpenVLA [7] and Octo [8], trained on large, aggregated datasets such as Open X-Embodiment [19], have become crucial benchmarks for the community. To mitigate catastrophic forgetting and better leverage the powerful priors of VLMs, methods like OTTER [12] freeze the language-aligned vision encoders, such as CLIP [20] and train a lightweight policy network on top of text-aware visual features. While architectural choices differ, all these models face the common challenge of efficiently processing long visual token sequences.

Efficient Token Processing. VLM inference can be optimized by reducing visual tokens, which dominate the input sequence. While efficient token processing has been explored in language models [21], [22], the higher redundancy in images makes visual token reduction for VLMs particularly effective. Common strategies include token pruning [13], [14], merging [16], and query-based aggregation [17]. Most prior VLA methods, such as VLA-Cache [23] and EfficientVLA [10], rely on pruning to shorten sequences. In contrast, our work utilizes query-based aggregation as an information bottleneck. Moreover, unlike task-agnostic prior work, our compression is task-conditioned, ensuring a goal-oriented filtering process rather than generic summarization.

Instruction-Conditioned Mechanisms. Modulating a network's behavior based on an external context is a powerful technique, ranging from high-level cross-attention [11] to fine-grained feature modulation. The latter, exemplified by methods like Feature-wise Linear Modulation (FiLM) [24] and Adaptive Instance Normalization (AdaIN) [25], uses a conditioning signal to generate affine transformation parameters (scale and shift) that are applied to a network's intermediate representations. In our work, we employ FiLM to design a lightweight and efficient instruction-conditioned mechanism, using the language instruction to guide the visual compression.

III. Method

A. Overall Architecture and Data Flow

To address the efficiency bottleneck in VLA models, we introduce a hybrid instruction-guided compressor. This compression module replaces the standard projector connecting the vision encoder and the LLM backbone, as illustrated in Fig. 2.

The proposed token compression module operates on two primary inputs: the visual features $X \in \mathbb{R}^{N \times D}$ from a pre-trained vision encoder, where N and D represent the number of vision tokens and embedding dimension, respectively, and the language instruction embeddings from the VLA's own LLM backbone. To accommodate variable-length instructions, we apply mean pooling across the sequence of language embeddings to derive a single fixed-size vector, L_{pooled}. This vector subsequently functions as the conditioning signal for instruction guidance. Instead of a separate pre-trained language encoder (e.g., CLIP), we leverage the LLM's internal embeddings to reduce parameters and utilize the rich semantic information already present in pre-trained large-scale models.

The data flows through two parallel pathways. This hybrid architecture can balance two competing and equally critical requirements in robotic manipulation: high-level semantic understanding and low-level spatial precision. A separate global compressor, similar to standard Perceiver-style models, excels at creating a holistic scene summary but risks removing the fine-grained spatial details necessary for precise actions. The proposed token compression model resolves this challenge by creating a hybrid architecture, where the two components handle different complementary tasks:

- The **Semantic Task Compressor (STC)** provides the strategic context (*what to do and where to go*) by generating a globally-contextualized summary $Z_G \in \mathbb{R}^{k \times D}$.
- The **Spatial Refinement Compressor (SRC)** supplies the tactical precision (*how to act*) by producing a spatially-aware local representation $Z_L \in \mathbb{R}^{N' \times D}$.

The final compressed sequence, $Z = \text{Concat}([Z_G; Z_L])$, integrates the outputs of STC and SRC as input for the LLM to generate robot actions. The specific designs of STC and SRC are detailed below.

B. Semantic Task Compressor

The goal of the STC is to distill the visual scene into a compact, fixed-length summary that is dynamically conditioned on the language instruction. It employs a query-based aggregation

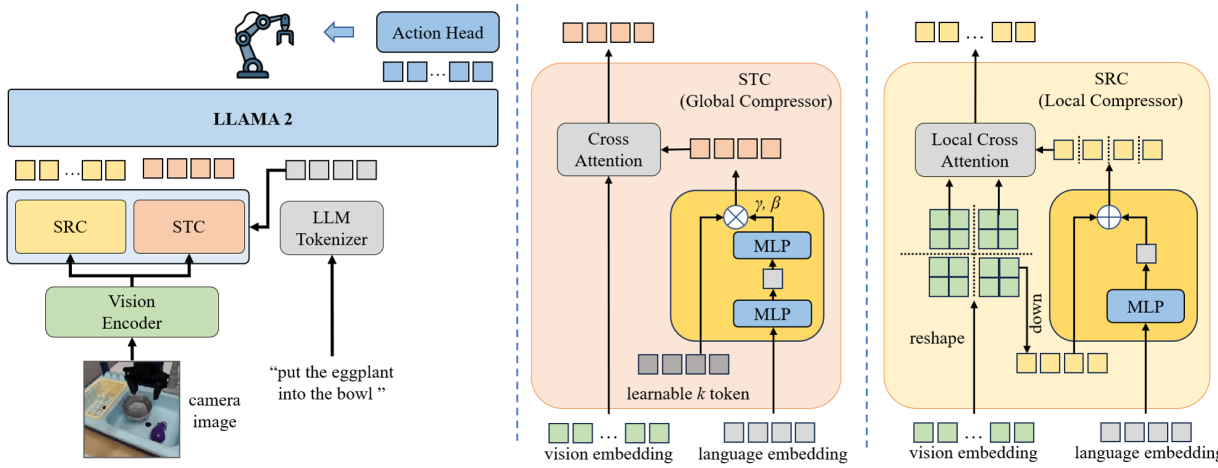

Fig. 2. The architecture of the proposed Compressor-VLA. The module features two instruction-guided parallel pathways. The Semantic Task Compressor (STC) uses language to modulate its queries, while the Spatial Refinement Compressor (SRC) infuses language information directly into local visual tokens.

strategy, using a small set of k learnable queries, $Q \in \mathbb{R}^{k \times D}$, where $k \ll N$. We modulate these queries using Feature-wise Linear Modulation (FiLM) [24]. Specifically, a semantic representation of the task, denoted by E_L, is computed by passing the pooled language vector through a dedicated MLP:

$$E_L = \text{MLP}_{\text{STC}}(L_{pooled}) \tag{1}$$

Then, a small MLP generates per-query affine transformation parameters (scale γ and shift β):

$$\gamma, \beta = \text{MLP}_{\text{FiLM}}(E_L) \tag{2}$$

$$Q_{\text{con}} = \gamma \odot Q + \beta \tag{3}$$

These conditioned queries then interact with the full set of visual tokens X via cross-attention to produce the global summary Z_G:

$$Z_G = \text{Attention}(Q = Q_{\text{con}}, K = X, V = X) \tag{4}$$

Our approach ensures that task instructions shape the information bottleneck from the outset. By employing FiLM, we non-linearly adapt the STC's learnable queries, which function as abstract "concept detectors", to prioritize scene features most relevant to the current task.

C. Spatial Refinement Compressor

To compensate for the potential spatial information loss in the global pathway, the goal of SRC is to reduce token count while preserving fine-grained, task-relevant local details. It operates on non-overlapping local windows of the 3D feature $X' \in \mathbb{R}^{H \times W \times D}$, where H, W, and D represent the height, width, and embedding dimension, respectively.

For each window containing tokens $X_w \in \mathbb{R}^{w \times w \times D}$, we first generate a representative "raw" query, denoted by q_{raw}, by downsampling and reshaping the original visual features. This query is then modulated by the language instruction. A transformed instruction embedding is produced from the pooled language vector via a separate MLP, $E'_L = \text{MLP}_{\text{SRC}}(L_{pooled})$, and added to the raw query. This conditioned query, denoted by q_w, then attends to the original, unmodified window tokens to produce a task-aware summary z_w:

$$q_{raw} = \text{Downsample}(X_w) \tag{5}$$

$$q_w = q_{raw} + E'_L \tag{6}$$

$$z_w = \text{Attention}(Q = q_w, K = X_w, V = X_w) \tag{7}$$

The operator $\text{Downsample}(\cdot)$ conducts downsampling and flattens to the input vector. The final local representation Z_L is the concatenation of all z_w. This design efficiently infuses task-relevance into the query process, guiding the model to preserve local details pertinent to the instruction. For the SRC, the primary goal is to preserve local detail with minimal distortion. We therefore opt for a simpler direct injection of the language embedding into the query. This linear shift acts as a gentle "hint" to the local query, subtly biasing its attention towards task-relevant features within its window without aggressively transforming the representation, thereby safeguarding the high-fidelity spatial information required for precise manipulation.

IV. Experiments

We evaluate Compressor-VLA on both simulation benchmarks and real-world scenarios.

A. Simulation Setup

Environment and Baselines. We conduct experiments on the **LIBERO** benchmark [26], a comprehensive suite featuring a simulated Franka Panda arm across four task suites (Spatial, Object, Goal, Long) with diverse scenes and instructions. We evaluate Compressor-VLA against two categories of methods. First, we consider **General-purpose VLAs**. We utilize OpenVLA-OFT [27] as our primary baseline to isolate compression effects, alongside representative high-performing models CogACT [28] and $\pi 0$ [29]. Second, we benchmark against **Efficient VLAs** that represent distinct acceleration paradigms. Specifically, SP-VLA [30] and SpecPrune-VLA [33] employ hierarchical model scheduling or speculative pruning strategies. SparseVLM [32] and FastV [31] focus on adaptive token sparsification, whereas VLA-Cache [23] utilizes temporal feature caching.

TABLE I

PERFORMANCE ON LIBERO-SPATIAL, LIBERO-OBJECT, LIBERO-GOAL, AND LIBERO-LONG TASK SUITES. THE BEST RESULTS ARE HIGHLIGHTED IN **BOLD**. "COMPRESSED TOKENS" DENOTES THE NUMBER OF VISUAL TOKENS FED TO THE LLM AFTER COMPRESSION; "AVG." AND "MIN." INDICATE AVERAGE RETAINED AND MINIMUM REQUIRED TOKEN COUNTS, RESPECTIVELY.

Model	Spatial SR (%)	Object SR (%)	Goal SR (%)	Long SR (%)	Avg. SR (%)	FLOPs (T)	Compressed Tokens
OpenVLA-OFT [27]	97.6	98.4	97.9	94.5	97.1	3.95	512
CogACT [28]	97.2	98.0	90.2	88.8	93.6	-	512
$\pi 0$ [29]	96.8	98.8	95.8	85.2	94.2	-	512
SP-VLA [30]	75.4	85.6	84.4	54.2	74.9	3.10	229 (avg.)
FastV [31]	96.8	81.0	96.4	73.0	86.8	3.18	256
SparseVLM [32]	96.8	94.2	97.6	93.6	95.6	3.04	256
SpecPrune-VLA [33]	98.2	96.3	**97.7**	94.0	96.6	1.70	197 (avg.)
VLA-Cache [23]	**99.0**	97.7	97.4	93.6	97.0	3.28	212 (min.)
Compressor-VLA	98.8	**99.2**	96.4	**94.8**	**97.3**	**1.62**	160

Metrics and Implementation. We report the Success Rate (SR) to evaluate task performance. To explicitly measure computational **efficiency**, we utilize FLOPs and compressed token counts. Implementation follows OpenVLA-OFT [27], training on 8 Nvidia A100 GPUs for 150,000 gradient steps with a global batch size of 64. We employ LoRA [34] (rank 32) to fine-tune the vision encoder and LLM backbone, while the compressors and action head are fully fine-tuned. The learning rate decays from $5e^{-4}$ to $5e^{-5}$. Specifically for our compression modules, we set the global queries $k = 16$ and the local window size $w = 2$.

Results & Analysis. Table I highlights that Compressor-VLA significantly reduces computational overhead without compromising accuracy. Compared to the baseline, we reduce FLOPs by 59% and tokens by over 3×, while slightly improving the success rate (97.3% vs. 97.1%). When benchmarking against efficient VLAs, our advantages are twofold. First, unlike **VLA-Cache**, which relies on reuse mechanisms that yield only marginal FLOPs reduction (3.28T), our structural compression achieves a much lower computational cost (1.62T). Second, compared to aggressive pruning methods like **SpecPrune-VLA** (1.70T) and **SP-VLA**, our method maintains a competitive success rate comparable to the full-sized model, avoiding the accuracy degradation observed in those approaches.

B. Ablation Study

Component Analysis. We analyze component contributions in Table II, comparing our full model (**STC+SRC**) against single-stream variants (**STC/SRC-Only**), an unconditioned baseline (**No Guidance**), and a modulation variant (**STC+SRC-FiLM**) where SRC uses FiLM. Results validate our design choices. The full architecture achieves the highest average success rate (97.3%). Removing guidance drops performance to 96.3%, confirming task-aware compression is necessary. Drops in single-stream models (95.9% and 95.5%) demonstrate that neither global context nor local detail alone suffices; the dual-pathway is essential. Crucially, 'STC+SRC-FiLM' yields lower performance (97.0%) than our default. This suggests simple additive injection is more effective than FiLM for SRC, preserving high-frequency details with minimal distortion.

TABLE II

ABLATION ON COMPRESSOR COMPONENTS AND GUIDANCE ON LIBERO SUITES. BEST RESULTS ARE HIGHLIGHTED IN **BOLD**.

Config	Spat.	Obj.	Goal	Long	Avg.	FLOPs	Tok.
STC+SRC	**98.8**	**99.2**	96.4	**94.8**	**97.3**	1.62	160
STC+SRC-FiLM	98.4	99.0	96.0	94.4	97.0	1.62	160
No Guidance	98.6	99.0	93.8	93.8	96.3	1.43	160
STC-Only	96.0	97.6	**96.8**	93.0	95.9	**0.76**	**32**
SRC-Only	97.6	98.2	94.2	92.2	95.5	1.20	128

Hyperparameter Sensitivity. Table III illustrates the trade-off between global compactness (k) and local granularity (w). For global queries, performance saturates at $k = 16$; increasing to $k = 32$ incurs higher FLOPs (1.83T) without accuracy gains, leading us to select $k = 16$ as the efficient default. In contrast, the model is highly sensitive to local window size. The smallest window ($w = 2$) yields the best result (94.8%), while larger windows ($w = 4, 8$) cause drops. This confirms minimizing the compression window is critical for preserving spatial details necessary for precise manipulation.

TABLE III

ABLATION ON HYPERPARAMETERS k AND w ON THE LIBERO-LONG TASK SUITE. BEST RESULTS ARE HIGHLIGHTED IN **BOLD**.

Hyperparam.	Value	Long	FLOPs	Tokens
Global Queries	$k = 8$	94.2	**1.51**	**144**
	$k = 16$	**94.8**	1.62	160
	$k = 32$	**94.8**	1.83	192
Local Window	$w = 2$	**94.8**	1.62	160
	$w = 4$	92.6	0.86	64
	$w = 8$	91.4	**0.70**	**40**

C. Real-World Evaluation

To validate sim-to-real transfer, we deployed Compressor-VLA on a Mobile ALOHA robot equipped with dual 7-DoF Piper arms. The perception system utilizes a multi-view setup: wrist-mounted cameras on the arms and a third-view RGB camera. We designed two tasks (Fig. 3): (1) **Spatial Awareness** ("Put X into bucket"), evaluating precise localization with diverse objects (simulated toys, yellow banana, green pepper, purple eggplant) over 100 trajectories per object. (2) **Semantic Understanding** ("Stack Tower of Hanoi"), testing multi-step

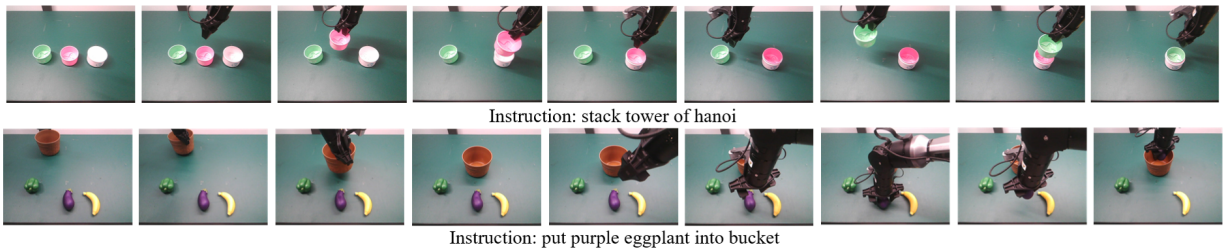

Fig. 3. Execution examples on real-world tasks.

logical reasoning, for which we collected 10 trajectories per relative configuration. The model was fine-tuned from OpenVLA checkpoints for 100,000 steps using RGB inputs from both views. We use a learning rate of $5e^{-4}$, which is decayed to $5e^{-5}$ after 50,000 steps.

TABLE IV
REAL-WORLD TASK SUCCESS RATE ON MOBILE ALOHA.

Model	Spatial Awareness (SR %)	Semantic Understand (SR %)
OpenVLA-OFT	91.7 (22/24)	76.7 (23/30)
Compressor-VLA	**100** (24/24)	**83.3** (25/30)

Results and Efficiency. As shown in Table IV, Compressor-VLA achieves competitive real-world success rates on both tasks, supporting the sim-to-real applicability of the proposed compression strategy. We further evaluated computational efficiency on the robot hardware. Compared to the baseline (∼123ms), Compressor-VLA significantly reduces inference latency to 73.4–85.6ms. This 1.43–1.67× speedup outperforms efficient alternatives like SparseVLM (∼96.3ms) and VLA-Cache (∼97.8ms). Notably, the proposed STC and SRC modules incur negligible overhead (0.87ms). This efficiency extends to training, reducing the total duration from 32 to 24 hours on 8×A800 GPUs.

D. Qualitative Analysis

We present qualitative results to validate our design principles: instruction-guided attention and the STC-SRC synergy.

Instruction-Conditioned Attention (Fig. 4). To verify dynamic guidance, we visualize STC attention maps for the same initial scene under different commands. As shown in Fig. 4, in **Task 1** ("put the alphabet soup..."), the model focuses on the soup can. Crucially, in **Task 2** ("put the cream cheese box..."), attention shifts to the cheese box. This confirms two capabilities: (1) **Goal-Directed Filtering**: The model effectively isolates task-relevant objects. (2) **Predictive Focus**: The attention acts as a temporally-aware filter, anticipating the *immediate* sub-goal (e.g., the first object to grasp) rather than attending to all mentioned items simultaneously. Additionally, the consistent low-level attention on the gripper indicates stable proprioceptive awareness.

Hybrid Architecture Synergy (Fig. 5). We visualize the distinct roles of the two modules using **Task 10** ("put the yellow and white cup into the microwave"). The **STC** acts as a high-level planner, maintaining a broad focus on the global positions of the cup and the microwave to encode the semantic layout. In contrast, the **SRC** exhibits fine-grained spatial awareness; its attention is concentrated specifically on the **mug's handle**, a critical geometric detail required for a stable grasp. This validates our design: STC provides the semantic "where", while SRC refines the tactical "how".

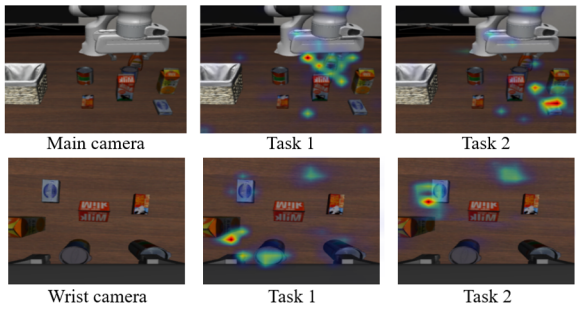

Fig. 4. STC attention visualization for the same scene. The model dynamically focuses on the **soup** in **Task 1** (Left) and shifts to the **cheese box** in **Task 2** (Right) based on the instruction.

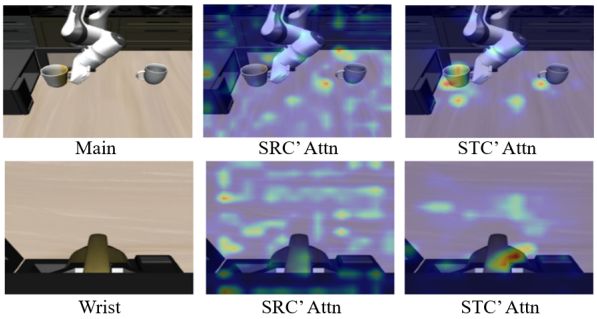

Fig. 5. Visualization of the hybrid architecture's synergy on Task 10 ("Put the yellow and white cup into the microwave...") in LIBERO-10.

V. CONCLUSIONS

We present Compressor-VLA, a token compression framework for VLA models. Our framework performs instruction-guided visual token compression by combining a Spatial Refinement Compressor for spatial detail and a Semantic Task Compressor for contextual summary. Extensive experiments on simulation tasks demonstrate that Compressor-VLA significantly reduces computational load while maintaining competitive task success rates. Experiments on real-world tasks further support the practical applicability of our method. We show the effectiveness of our hybrid architecture and validate our design choices through ablation studies. Our work shows that by intelligently filtering visual data based on task relevance at an early stage, we can achieve more efficient and robust robotic agents. Current validation is conducted on LIBERO

and one real-world robot platform; extending evaluation to additional benchmarks and robot platforms is an important direction for future work.

Acknowledgements. The work is supported by the Beijing Natural Science Foundation (No. L247025).

REFERENCES

[1] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, and others. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[2] Zane Durante, Ran Gong, Bidipta Sarkar, Naoki Wake, Rohan Taori, Paul Tang, Shrinidhi Lakshmikanth, Kevin Schulman, and others. An interactive agent foundation model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3652–3662, 2025.

[3] Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, and others. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pages 2165–2183. PMLR, 2023.

[4] Lei Xiao, Jifeng Li, Juntao Gao, Feiyang Ye, Yan Jin, Jingjing Qian, Jing Zhang, Yong Wu, and Xiaoyuan Yu. AVA-VLA: Improving Vision-Language-Action models with Active Visual Attention. *arXiv preprint arXiv:2511.18960*, 2025.

[5] Xin Yan, Zhenglin Wan, Feiyang Ye, Xingrui Yu, Hangyu Du, Yang You, and Ivor Tsang. HBVLA: Pushing 1-Bit Post-Training Quantization for Vision-Language-Action Models. *arXiv preprint arXiv:2602.13710*, 2026.

[6] Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gomez Colmenarejo, Alexander Novikov, Gabriel Barth-Maron, Mai Gimenez, Yury Sulsky, Jackie Kay, Jost Tobias Springenberg, and others. A generalist agent. *arXiv preprint arXiv:2205.06175*, 2022.

[7] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, and others. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[8] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, and others. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[9] Hongkuan Zhou, Xiangtong Yao, Yuan Meng, Siming Sun, Zhenshan Bing, Kai Huang, and Alois Knoll. Language-conditioned learning for robotic manipulation: A survey. *arXiv preprint arXiv:2312.10807*, 2023.

[10] Yantai Yang, Yuhao Wang, Zichen Wen, Luo Zhongwei, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. EfficientVLA: Training-Free Acceleration and Compression for Vision-Language-Action Models. *arXiv preprint arXiv:2506.10100*, 2025.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, volume 30, 2017.

[12] Huang Huang, Fangchen Liu, Letian Fu, Tingfan Wu, Mustafa Mukadam, Jitendra Malik, Ken Goldberg, and Pieter Abbeel. Otter: A vision-language-action model with text-aware visual feature extraction. *arXiv preprint arXiv:2503.03734*, 2025.

[13] Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20857–20867, 2025.

[14] Cheng Yang, Yang Sui, Jinqi Xiao, Lingyi Huang, Yu Gong, Chendi Li, Jinghua Yan, Yu Bai, and others. Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19803–19813, 2025.

[15] Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9392–9401, 2025.

[16] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token Merging: Your ViT but Faster. In *International Conference on Learning Representations*, 2023.

[17] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, and others. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, volume 35, pages 23716–23736, 2022.

[18] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, and others. Vision-Language Foundation Models as Effective Robot Imitators. In *The Twelfth International Conference on Learning Representations*, 2024.

[19] Quan Vuong, Sergey Levine, Homer Rich Walke, Karl Pertsch, Anikait Singh, Ria Doshi, Charles Xu, Jianlan Luo, Liam Tan, Dhruv Shah, and others. Open x-embodiment: Robotic learning datasets and rt-x models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[20] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and others. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[21] Zihang Dai, Guokun Lai, Yiming Yang, and Quoc Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *Advances in neural information processing systems*, volume 33, pages 4271–4282, 2020.

[22] Xin Huang, Ashish Khetan, Rene Bidart, and Zohar Karnin. Pyramidbert: Reducing complexity via successive core-set based token selection. *arXiv preprint arXiv:2203.14380*, 2022.

[23] Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. Vla-cache: Towards efficient vision-language-action model via adaptive token caching in robotic manipulation. *arXiv preprint arXiv:2502.02175*, 2025.

[24] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[25] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[26] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, volume 36, pages 44776–44791, 2023.

[27] Moo Jin Kim, Chelsea Finn, and Percy Liang. Fine-tuning vision-language-action models: Optimizing speed and success. *arXiv preprint arXiv:2502.19645*, 2025.

[28] Qixiu Li, Yaobo Liang, Zeyu Wang, Lin Luo, Xi Chen, Mozheng Liao, Fangyun Wei, Yu Deng, Sicheng Xu, and others. Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation. *arXiv preprint arXiv:2411.19650*, 2024.

[29] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, and others. pi_0: A Vision-Language-Action Flow Model for General Robot Control. *arXiv preprint arXiv:2410.24164*, 2024.

[30] Ye Li, Yuan Meng, Zewen Sun, Kangye Ji, Chen Tang, Jiajun Fan, Xinzhu Ma, Shutao Xia, Zhi Wang, and Wenwu Zhu. SP-VLA: A Joint Model Scheduling and Token Pruning Approach for VLA Model Acceleration. *arXiv preprint arXiv:2506.12723*, 2025.

[31] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. 2024.

[32] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and others. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

[33] Hanzhen Wang, Jiaming Xu, Jiayi Pan, Yongkang Zhou, Guohao Dai. Specprune-vla: Accelerating vision-language-action models via action-aware self-speculative pruning. *arXiv preprint arXiv:2509.05614*, 2025.

[34] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and others. Lora: Low-rank adaptation of large language models. *ICLR*, volume 1, pages 3, 2022.

[35] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.