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

The Better You Learn, The Smarter You Prune: Towards Efficient Vision-language-action Models via Differentiable Token Pruning

Titong Jiang[1,2]*, Xuefeng Jiang[1,3]*, Yuan Ma[1†], Xin Wen[1], Bailin Li[1],
Kun Zhan[1], Peng Jia[1], Yahui Liu[2], Sheng Sun[3], and Xianpeng Lang[1‡]

***Abstract*— We present LightVLA, a simple yet effective differentiable token pruning framework for vision-language-action (VLA) models. While VLA models have shown impressive capability in executing real-world robotic tasks, their deployment on resource-constrained platforms is often bottlenecked by the heavy attention-based computation over large sets of visual tokens. LightVLA addresses this challenge through adaptive, performance-driven pruning of visual tokens: It generates dynamic queries to evaluate visual token importance, and adopts Gumbel softmax to enable differentiable token selection. Through fine-tuning, LightVLA learns to preserve the most informative visual tokens while pruning tokens which do not contribute to task execution, thereby improving efficiency and performance simultaneously. Notably, LightVLA requires no heuristic "magic numbers" and introduces no additional trainable parameters, making it compatible with modern inference frameworks. Experimental results demonstrate that LightVLA outperforms different VLA models and existing token pruning methods across diverse tasks on the LIBERO benchmark, achieving higher success rates with substantially reduced computational overhead. Specifically, LightVLA reduces FLOPs and latency by 59.1% and 38.2% respectively, with a 2.6% improvement in task success rate. Meanwhile, we also investigate the learnable query-based token pruning method LightVLA* with additional trainable parameters, which also achieves satisfactory performance. Our work reveals that as VLA pursues optimal performance, LightVLA spontaneously learns to prune tokens from a performance-driven perspective. To the best of our knowledge, LightVLA is the first work to apply adaptive visual token pruning to VLA tasks with the collateral goals of efficiency and performance, marking a significant step toward more efficient, powerful and practical real-time robotic systems. Project site: *https://liauto-research.github.io/LightVLA*.**

I. Introduction

From large-scale industrial operations to personal healthcare and leisure activities, robotics have been reshaping nearly every facet of human life. Recently, robotics witnessed the rise of embodied intelligence as its newest technical leap, when artificial intelligence (AI) is introduced into robotics, thanks to the emergence of vision-language-action (VLA) models. VLA models can be regarded as a family of large vision-language models (VLMs) which directly translate visual information and language instructions into executable action policies. Leveraging the general knowledge and reasoning capabilities inherited from LLMs, VLA models have shown transformative potential in tackling complex robotic reasoning, planning and manipulation tasks [1]–[8].

[1] LiAuto Inc., [2] School of Vehicle and Mobility, Tsinghua University, [3] Institute of Computing Technology, Chinese Academy of Sciences
* Equal contributions.
† Project Lead.
‡ Corresponding author. Email: langxianpeng@lixiang.com

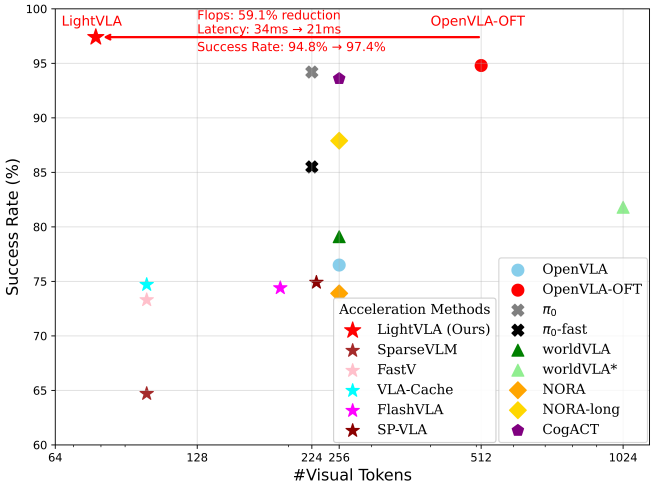

Fig. 1: LightVLA achieves better performance than common VLA models and acceleration methods with fewer visual tokens, yielding efficient computation and lower latency.

Unfortunately, the success of VLA models also comes with high computational complexity. As typical VLA models usually include a large language model (LLM) with billion-scale parameters, the expensive attention-based computational cost and high forward latency of VLA models hinder them from real-time applications on systems with computing constraints at edge devices such as household robots [9] and autonomous vehicles [10], [11]. As such, the acceleration techniques for VLA models plays a significant role in making VLA models more efficient and practical [12].

Many acceleration approaches for VLMs and VLA models have been explored in prior studies, including model quantization, layer skipping [12], token pruning [13]–[20] and lightweight model design [21]–[23]. Among these acceleration approaches, visual token pruning is of particular interest, as the dominant majority of input tokens for VLA models are visual tokens. Given the sparse nature of vision modality [14], visual tokens often convey little or redundant information, which brings prominent yet unnecessary computational burden for VLA models. Meanwhile, though visual token pruning has been relatively widely studied in VLMs [13]–[15], recent studies [24]–[26] show that these methods produce unsatisfactory performance when transferring to VLA models, as VLMs focuses on global semantics while specialized robotic tasks depends more on local semantics. Therefore, visual token pruning oriented for VLA models holds great potential while the related exploration is limited.

It is widely acknowledged that there is a trade-off relationship between efficiency and generalization performance for the VLA acceleration. Consequently, previous visual token pruning methods often choose efficiency as the priority at the acceptable cost of performance degradation. For example, EfficientVLA [12] first sets the retained token number as a hyperparameter, and then proposes several approaches to minimize the performance drop caused by token reduction. However, we argue that efficiency and performance are not intrinsically contradictory. Note that the sparsity of the visual input is not only contributing to computational inefficiency, but also damaging performance by introducing noises and diverting attention. As such, we propose that by eliminating the sparsity of visual inputs, efficiency and performance can be optimized simultaneously, breaking the efficiency-performance trade-off in VLA models. More specifically, we investigate the underlying sparsity of visual tokens in VLA models, and propose LightVLA, a performance-driven differentiable visual token pruning framework for efficient VLA. To evaluate the importance of visual tokens for task execution, LightVLA generates dynamic queries through the cross attention between visual tokens and task instruction tokens. Subsequently, each query selects a useful token in a differentiable manner using the Gumbel-softmax [27] technique. Following the fine-tuning paradigm, we take OpenVLA-OFT [7] as the foundation model and train LightVLA to distinguish and retain only informative visual tokens that contribute to overall performance. As shown in Fig. 1, LightVLA obtains state-of-the-art performance on the LIBERO benchmark with significantly fewer visual tokens. Compared to OpenVLA-OFT, LightVLA achieves a 59.1% reduction in total FLOPs and 2.6% improvement in task success rate, highlighting that efficiency and performance are collateral goals that can be achieved simultaneously. To further fill in the existing gap in visual token pruning for VLA models, we propose LightVLA* in Discussion (Section V), which is an efficient and effective variant of LightVLA which introduces learnable query as additional trainable parameters to guide the model to select informative tokens. To sum up, our contributions are outlined as follows:

- We empirically show that performance and efficiency can be collaterally optimized for VLA models.
- We propose LightVLA, a performance-driven differentiable visual token pruning framework for VLA models.
- Comprehensive experiments on the LIBERO benchmark demonstrate the state-of-the-art performance and efficiency of LightVLA compared to its foundation model and other previous models.
- To further fill in current gap in token pruning for VLA models, we propose LightVLA* an early exploration on learnable query-based token pruning, which also improves the performance and efficiency.

II. Related Work

Vision-language model (VLM). Vision-language models (VLMs) integrate vision and language modalities, extending the reasoning capabilities of large language models (LLMs) to process visual input. This integration is typically achieved by encoding images into hundreds of visual tokens aligned with text tokens [28]. Representative VLMs include LLaVa [28], BLIP-2 [29], InternVL [30], and Qwen-VL [31], whose parameter spaces typically range from 7B to 70B. Despite these strengths, VLMs are not inherently designed to directly generate task-specific robotic action policies, which leads to the birth and emergence of VLA models.

Vision-language-action (VLA) model. VLAs [1]–[4] extend VLMs for embodied intelligence to generate feasible action policies for complicated robotic tasks like manipulating, bridging the gap between perception and action. Similar to VLMs, representative VLA models (e.g. OpenVLA [2], π_0 [3] and CogACT [6]) typically tokenize image patches into hundreds of visual tokens, and then concatenate them with task tokens, from which actions are generated either as discrete tokens [2], [21], [32] or continuous values [6], [7]. Early VLA models like OpenVLA generate discrete action tokens to further yield the action policy in the auto-regressive way, while recent VLA models aim to generate continuous action tokens like CogACT and OpenVLA-OFT. Action chunking technique [7], [21], [22], which aims to predict a sequence of action policies, has also demonstrated potential to improve the overall performance. However, the billion-scale parameter and high inference cost of VLA models make them challenging to deploy in real-time low-latency robotic tasks. To optimize the computation overhead and latency, existing works often aim to design lightweight VLA models, such as TinyVLA [23], SmolVLA [22], and NORA [21]. Beyond model architectures, token pruning offers a promising direction for optimizing efficiency with fewer input tokens, yet remains underexplored in VLA studies.

Visual token pruning. Token pruning has been successfully applied to diverse neural architectures ranging from the original transformer and ViT [33]–[35] to modern large-scale models like LLMs and VLMs [13]–[20]. Existing methods often pre-define a hyperparameter which determines a fixed number of retained visual tokens, necessitating large empirical exploration to select the optimal value of this hyperparameter. In the context of VLA models, visual token pruning methods optimized specifically for VLMs tend to underperform when extended to VLA models, as analyzed in previous works [24]–[26]. Existing works [12], [24]–[26] dedicated to VLA explore the potential of training-free visual token pruning with the guidance like the attention scores, which still relies on a fixed token pruning ratio. This compression ratio present two major limitations: it introduces a strong inductive bias, potentially limiting the model's adaptability to varying visual inputs and tasks, and this approach often faces an inevitable performance drop. Different from existing efforts, our proposed differentiable visual token pruning framework LightVLA dynamically selects necessary tokens for each task scenario, leading to improved performance while maintaining significant computational efficiency. On the other hand, some well-noted inference-oriented platforms like vLLM [36] and SGLang [37] are optimized for high-throughput generation and they do not expose intermediate

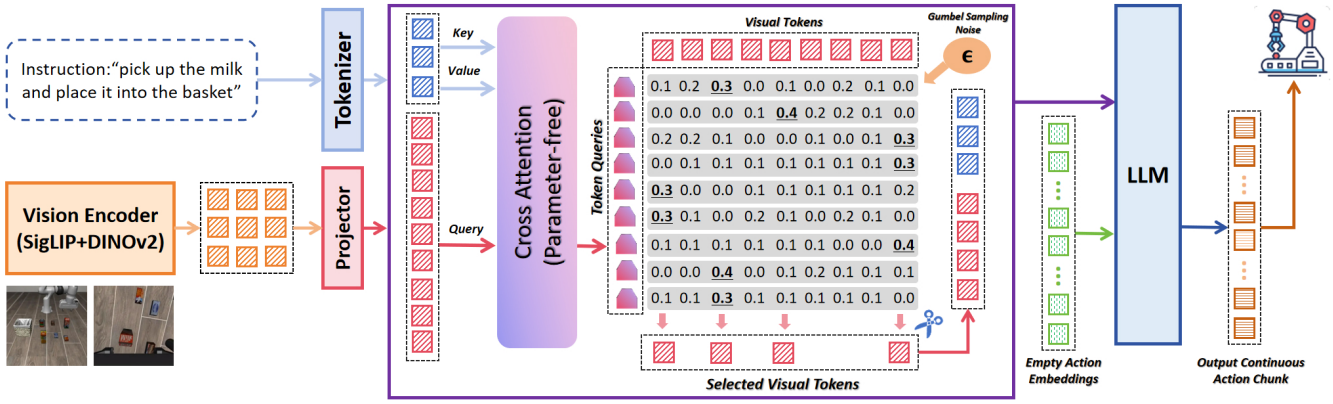

Fig. 2: Illustration of the proposed LightVLA framework. Gray regions indicate the use of Gumbel-softmax for differentiable token selection.

attention scores during inference. This makes traditional attention score-based token selection methods [12], [14] infeasible. LightVLA does not require the attention scores from from within LLM, which can be well supported by these platforms, facilitating practical real-world deployment.

III. Method

We introduce LightVLA, a differentiable visual token pruning framework for VLA acceleration. Unlike previous studies, our strategy is purely performance-driven, meaning that the only optimization goal is better performance. LightVLA enables the model to adaptively retain and discard tokens, during which time the model spontaneously learns to retain only useful tokens to maximize performance, thereby improving efficiency. Moreover, LightVLA requires no extra parameters, hyper-parameters or auxiliary losses, making it a universal framework compatible with most VLAs.

A. Problem Definition

A typical VLA can be decomposed into three components: A visual encoder with a projector f_v, a LLM backbone f_ϕ, and an action head (or de-tokenizer) f_a. The visual encoder encodes the input image or images X_I into L_v initial visual tokens $H_{v'} = f_v(X_I) \in \mathbb{R}^{L_v \times D'}$. The initial visual tokens are projected to $H_v = f_v(X_I) \in \mathbb{R}^{L_v \times D}$ and then concatenated with language tokens $H_l \in \mathbb{R}^{L_l \times D}$ and sent into the LLM backbone. Herein, D' and D denote the initial vision token embedding dimension and the text token embedding dimension, respectively. The action head f_a finally translates the output hidden states from f_ϕ into action policies oriented for robotic tasks.

The LLM outputs the final hidden states (i.e. output tokens) $H = f_\phi(H_v, H_l)$. The computational bottleneck mainly lies in the decoder layers of the LLM f_ϕ, which involves extensive attention-based computation of all visual tokens H_v and text tokens H_l. Eventually, the action head converts H of LLM into continuous action policies $A = f_a(H)$. Since $L_v >> L_l$ in most cases, one effective method to decrease the computation overhead is to introduce an efficient visual token pruner f_p. The objective of a specific visual token pruner f_p is to determine the pruned set of visual tokens $H'_v = f_p(H_v) \subseteq H_v$ to be kept, in order to reduce the computational cost of VLA without compromising its performance. Note that we keep the [CLS] token since it maintains the global visual information [38], and conduct pruning only on the patch-level visual tokens.

B. LightVLA

In previous studies [13]–[20], visual token pruners usually reduce the number of visual tokens to a pre-defined value L'_v. While simple and effective, this practice also comes with the risk of performance degradation due to information loss, especially when the task and scenario get complicated that L'_v visual tokens cannot convey sufficient information. As such, it is imperative for VLA models to dynamically determine H'_v according to its inputs so that the information loss can be further minimized.

In this study, we propose LightVLA, a novel query-based visual token pruning strategy that can adaptively distinguish informative visual tokens from H_v. As shown in Fig. 2, LightVLA employs a series of L_v token queries $Q = \{q_1, q_2, \cdots, q_{L_v}\}$, each responsible for selecting one useful visual token from all tokens $h_k = q_i(H_v)$. All visual tokens selected by the queries constitute the pruned set $H'_v = \{\, h_k \mid \exists\, q_i,\ h_k = q_i(H_v) \,\}$. In the extreme condition when each query chooses a unique token, all tokens are selected, thus $H'_v = H_v$. When multiple queries choose the same token, repeated tokens will not be selected, and therefore $H'_v \subset H_v$. The pruning process can be divided into three steps: Query generation, token scoring, and token selection.

*1) **Query Generation**:* While it is a common practice in prior VLM works to adopt learnable embeddings as queries [13], [29], it also introduces extra parameters into the model, rendering this approach less feasible in resource-constrained platforms that VLA is often implemented on. As such, we propose a parameter-free query generation method for better compatibility.

We note that the usefulness of a visual token can be reflected by the interactions between its visual information and the language instruction. For instance, when we give an instruction *pick up the milk and place it into the basket* as the text prompt, the VLA model should focus more on the

two semantic objects (i.e. milk and the basket) in the image instead of other less informative objects or background.

Therefore, queries can be generated via cross attention between visual tokens and language tokens.

$$Q = \text{softmax}(\frac{H_v H_l^T}{\sqrt{D}})H_l, \tag{1}$$

where $Q \in \mathbb{R}^{L_v \times D}$. Note that unlike traditional attention design, neither weight nor bias matrices is included in the query generation process for simplicity.

*2) **Token Scoring**:* In this step, each query estimates the usefulness of all tokens individually through token scoring.

$$S = \frac{QH_v^T}{\sqrt{D}} \tag{2}$$

Here, $S \in \mathbb{R}^{L_v \times L_v}$ is the score matrix, where each element $s_{i,j}$ denotes the score of the j-th token assigned by the i-th query.

*3) **Token Selection**:* To determine the pruned token set, each query selects the token with the highest score.

$$H_v' = \{h_k | k = \text{argmax}_j(s_{i,j}), j = 1, 2, \cdots, L_v\}. \tag{3}$$

However, during training, it is noted that the argmax operation is not differentiable. One solution proposed in previous studies [18] is to incorporate auxiliary loss on the token scores. However, the introduction of auxiliary loss not only complicates the training procedures, but may also lead to performance deterioration and gradient conflicts of different optimization objectives, and the ground truth of token scores is difficult to define.

To overcome this obstacle, we adopt the Gumbel-softmax sampling technique [27] to make the argmax operation differentiable. This technique allows the process of sampling from discrete tokens to be differentiable in the backward process, which guides the VLA model to learn to select most informative visual tokens. Specifically, we convert the score matrix $S \in \mathbb{R}^{L_v \times L_v}$ into an indicator matrix $I \in \mathbb{R}^{L_v \times L_v}$.

$$S' = S + \epsilon \tag{4}$$

$$S_{\text{soft}} = \text{softmax}_j(S') \tag{5}$$

$$S_{\text{hard}} = \text{one-hot}(\text{argmax}_j(S')) \tag{6}$$

$$I = S_{\text{hard}} + S_{\text{soft}} - S_{\text{soft}}^{SG} \tag{7}$$

where S' is the score matrix injected with Gumbel sampling noise $\epsilon \in U(0, \alpha)$, S_{soft} and S_{hard} are the soft and hard scores, one-hot is the one-hot function, and SG indicates the stop gradient operation.

Note that unlike the original Gumbel-softmax operation where $\epsilon \in U(0, 1)$, we gradually decrease the intensity level of sampling noise as training progresses by decaying the noise upper bound α. We also provide corresponding ablation study in Section IV-D. This design encourages model to better explore more diverse token selection schemes in the early learning stage, and helps the model stabilize in the final stage. With the indicator matrix I, the pruned set can be obtained by

$$H_v' = IH_v^T \tag{8}$$

Here, since I is an indicator matrix, H_v' only contains the tokens selected by the queries. Moreover, the gradient of I equals to the gradient of S'. Thus, the queries can be correspondingly optimized in an end-to-end manner with the gradient descent. Notably, for inference, we follow the direct argmax operation to pick up visual tokens selected by queries without the Gumbel noise.

Moreover, we notice that the LLM backbone relies heavily on the position IDs of visual tokens to understand the spatial relationship. Therefore, the position IDs are retained during the token selection process.

IV. Experiments

A. Experimental Settings

Dataset. We evaluate LightVLA on the LIBERO benchmark [39], which features a Franka Emika Panda arm in simulation with demonstrations containing camera images, robot state, task annotations, and delta end-effector pose actions. We use four task suites including LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-Long, each of which provides 500 expert demonstrations across 10 tasks. We test LightVLA on all tasks for 50 trials (i.e. 500 trials in total for each suite) to assess policy generalization to different spatial layouts, object selection, task goals, and long-horizon planning tasks. We report the success rate (%) on each task suite.

Baselines. For comparison, we compare the proposed LightVLA with baselines of two types, where the first type is diverse VLA model design and the second type considers applying token pruning on existing VLA models:

- *Diverse VLA models*: OpenVLA [2], π_0 series [3], [4], NORA series [21], SmolVLA [22], OpenVLA-OFT [7], CogACT [6]. WorldVLA with 256 visual tokens and WorldVLA* with 1024 visual tokens [32].
- *Token pruning methods*: FlashVLA [24], SP-VLA [25], VLA-cache [26], FastV [14], SparseVLM [15].

Note that SparseVLM and FastV are token pruning methods which are firstly proposed for VLMs yet used on the OpenVLA backbone following [26].

Implementation details. All experiments are conducted on 8 Nvidia® H20 GPUs. We use the open-sourced OpenVLA-OFT [7] as the foundation model which consists of the two-branch vision encoder (DINOv2 [40] and SigLIP [41]), LLaMA-2-7B [42] as the language model backbone. The LoRA [43] technique with rank 32 is applied on the entire model, including the vision encoder, LLM backbone and action head, for fine-tuning. We initialize the foundation model with the open-sourced OpenVLA-OFT checkpoints on HuggingFace, and then the model is fine-tuned for 40,000 gradient steps in total and we decay the learning rate from $5e-4$ to $5e-5$ after 30,000 gradient steps. The batch size on each device is 8, resulting in a global batch size of 64.

B. Main Experiments

We evaluate our proposed LightVLA against other baselines constructed by diverse foundation model architectures on the LIBERO benchmark in Table I. The results of

TABLE I: Experimental results on LIBERO benchmark. TP denotes token pruning. AR, FM and PD denote different action decoding (i.e. generation) paradigms including auto-regressive, flow matching, and parallel decoding. The average number and standard deviation of retained visual tokens of LightVLA are denoted under the success rate, respectively. * Our reproduced results, slightly different from the original paper [7] due to hardware discrepancy.

Method	Scale	TP	Decoding	Backbone	Spatial SR(%)	Object SR(%)	Goal SR(%)	Long SR(%)	Avg.
OpenVLA [2]	7B		AR	PrismaticVLM	84.7	88.4	79.2	53.7	76.5
SparseVLM [15]	7B	✓	AR	PrismaticVLM	79.8	67.0	72.6	39.4	64.7
FastV [14]	7B	✓	AR	PrismaticVLM	83.4	84.0	74.2	51.6	73.3
VLA-Cache [26]	7B	✓	AR	PrismaticVLM	83.8	85.8	76.4	52.8	74.7
FlashVLA [24]	7B	✓	AR	PrismaticVLM	84.2	86.4	75.4	51.4	74.4
SP-VLA [25]	7B	✓	AR	PrismaticVLM	75.4	85.6	84.4	54.2	74.9
WorldVLA [32]	7B		AR	Chameleon	85.6	89.0	82.6	59.0	79.1
WorldVLA* [32]	7B		AR	Chameleon	87.6	96.2	83.4	60.0	81.8
NORA [21]	3B		AR	Qwen-VL	85.6	87.8	77.0	45.0	73.9
SmolVLA [22]	2.25B		FM	SmolVLM	93.0	94.0	91.0	77.0	88.8
CogACT [6]	7B		FM	PrismaticVLM	97.2	98.0	90.2	88.8	93.6
π_0 [3]	3.3B		FM	PaliGemma	96.8	**98.8**	95.8	85.2	94.2
π_0-fast [4]	3.3B		FM	PaliGemma	96.4	96.8	88.6	60.2	85.5
NORA-Long [21]	3B		PD	Qwen-VL	92.2	95.4	89.4	74.6	87.9
OpenVLA-OFT* [7]	7B		PD	PrismaticVLM	97.6	94.2	95.2	92.0	94.8
LightVLA (Ours)	7B	✓	PD	PrismaticVLM	**98.4** (90±15 tokens)	98.4 (78±11 tokens)	**98.2** (64±10 tokens)	**94.6** (79±11 tokens)	**97.4**

CogACT are obtained from [44]. As the results indicate, LightVLA delivers the best performance across all tasks. Notably, LightVLA outperforms its baseline model, OpenVLA-OFT on all task suites by a large margin. Moreover, compared to OpenVLA-OFT which consumes 512 visual tokens, LightVLA only retains 78 visual tokens on average, indicating that most visual tokens are not contributing to overall performance. The results not only shed light on the sparsity of visual modality, but also proves that performance and efficiency are collateral goals that can be optimized simultaneously.

C. Analysis on Computation Efficiency

Here, we present the comparison of acceleration and performance of LightVLA and previous VLA acceleration approaches in Table II. Results show that LightVLA achieves the highest average success rate while substantially reducing latency and computational demands compared with other strong baselines. Compared to OpenVLA-OFT, LightVLA not only reduces FLOPs and latency by 59.1% and 38.2%, but also improves success rate by 2.6%. Remarkably, among all VLA acceleration approaches in Table II, LightVLA is the only one that boosts performance. Our findings prove that in the pursuit of optimal performance, the efficiency of VLA models can also be optimized due to the elimination of visual sparsity.

D. Ablation Study

Impact of noise factor schedule. In this study, we propose to gradually decrease the intensity level of sampling noise for more diverse token selection schemes. To validate this technique, we compare LightVLA with two variants: (1) LightVLA without sampling noise, and (2) LightVLA with constant sampling noise (i.e. without noise decay). The results are presented in Table III, which show that LightVLA outperforms both variants. A deeper analysis into the token pruning schemes of these variants reveals why noise factor schedule works. Compared to LightVLA, the variant without sampling noise retains fewer visual tokens, which oftentimes leads to loss of important semantic information, especially in semantically dense scenarios such as Object and Goal. The introduction of sampling noise alleviates this problem by encouraging more diverse token pruning choices. Besides, the second variant shows that with constant sampling noise, LightVLA finds it hard to learn to prune tokens, resulting in a significantly higher token number.

TABLE II: Comparison of acceleration and performance on the LIBERO benchmark between LightVLA and other methods. We report visual token counts, GPU types, computational cost (TFLOPs), end-to-end latency, and averaged task success rate.

Method	Visual Token	GPU	TFLOPs	Latency (ms)	SR (%) Avg.
OpenVLA	256	A100	-	-	76.5
SparseVLM	100 (max.)	RTX 4090	1.4	83	64.7
FastV	100 (max.)	RTX 4090	1.9	53	73.3
VLA-Cache	100 (max.)	RTX 4090	1.4	32	74.7
FlashVLA	192	H100	0.7	55	73.7
SP-VLA	229 (avg.)	A100	3.1	-	74.9
OpenVLA-OFT	512	H20	8.8	34	94.8
LightVLA	78 (avg.)	H20	3.6	21	97.4

TABLE III: Impact of noise factor schedule on LightVLA.

Variant	Spatial SR(%)	Object SR(%)	Goal SR(%)	Long SR(%)	Avg.	# Tokens (Avg.)
LightVLA	98.4	**98.4**	**98.2**	94.6	**97.4**	78
- w/o noise	98.8	97.6	97.2	94.2	97.0	72
- w/o schedule	**99.4**	97.8	96.0	**94.8**	97.0	112

Impact of retained tokens on performance. One highlight of LightVLA is its ability to adaptively distinguish useful tokens. To validate this ability, we manipulate the token pruning scheme of LightVLA in the following ways. First, after LightVLA has retained k tokens, we supplement

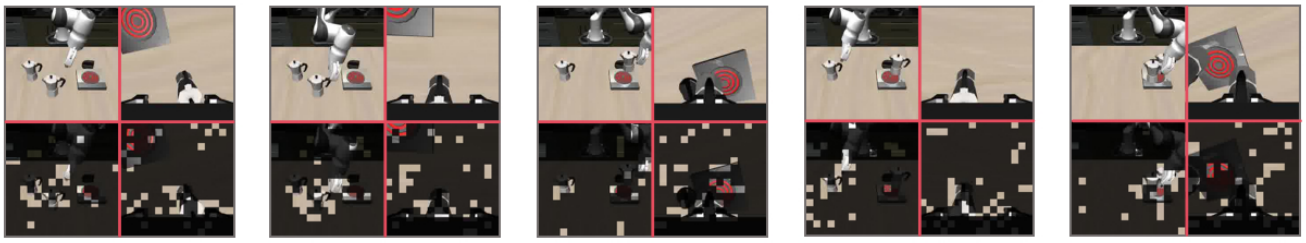

Fig. 3: An example of LIBERO-Long task: 'Put both moka pots on the stove'. Each frame consists of 4 images. Upper left: The 3rd person view camera. Upper right: The wrist camera. Lower left: The 3rd person view camera with pruned tokens masked. Lower right: The wrist camera with pruned tokens masked.

another k random tokens into the pruned set, resulting in $2k$ tokens sent into the LLM. This manipulation validates if any useful tokens have been omitted by LightVLA. Second, after LightVLA has retained k tokens, we randomly discard 10% of the tokens from the pruned set, resulting in $0.9k$ tokens sent into the LLM. This manipulation tests if any useless tokens have been retained by LightVLA. The results shown in Table IV demonstrate that any manipulation to the token pruning scheme will result in performance degradation, proving LightVLA's ability to retain only useful tokens and discard useless tokens.

TABLE IV: Impact of retained tokens on performance.

Model	# Tokens	Spatial SR(%)	Object SR(%)	Goal SR(%)	Long SR(%)	**Avg.**
LightVLA	k	**98.4**	**98.4**	**98.2**	**94.6**	**97.4**
LightVLA	$2k$	98.0	97.6	97.8	93.8	96.8
LightVLA	$0.9k$	98.2	97.8	97.2	93.0	96.6

E. Qualitative Visualization

To better illustrate the token pruning process, we take an episode as a demonstration to show the token selection dynamics when asking the VLA model to perform the manipulation task. The key frames are selected to represent critical phases of the manipulation task, including object interaction, and task completion. As shown in Fig. 3, the retained tokens concentrate on objects of interest, i.e. moka pots, stove, and the robotic arm itself, whereas most background tokens are pruned. Besides, it also shows that LightVLA learns to retain or prune more tokens when needed, as exemplified by the comparison between Frame$\#175$ and Frame$\#275$. The visualization results further demonstrate the effectiveness of LightVLA in adaptive token pruning.

V. Discussion

A. Learnable Query for Token Pruning

Since token pruning (especially the training-aware token pruning approach) is rarely investigated in the VLA research, besides the previously introduced parameter-free token pruning method LightVLA, we also implement LightVLA*, which exploits the *learnable query* with extra trainable parameters [13], [29] to select informative visual tokens. We consider applying this learnable query on two different positions which considers only the visual features or the joint visual-language features. Specifically, to filter out redundant visual tokens, we introduce N_q compression queries as token query head, designed to guide the model to learn to select visual tokens from all L_v visual tokens.

Learn to select at the vision encoder. The N_q compression queries interact with all visual tokens H_v, selectively extracting the important visual information to produce the selected visual tokens. Different from Eq. 1 of the parameter-free LightVLA, we introduce the learnable query $Q^* \in \mathbb{R}^{N_q \times D'}$ after the vision encoder, as visualized in Figure 4. Note that visual tokens are firstly concated at the channel level and then we perform our token pruning. Then we compute the token scoring as follows:

$$S^* = \frac{LN(Q^*) \cdot LN(H_v^T)}{\sqrt{D'}}, \tag{9}$$

where D' denotes the dimension of visual tokens before the projection layer, LN denotes the layer normalization layer to stablize the training process and $S^* \in \mathbb{R}^{N_q \times L_v}$ denotes the score matrix. We utilize the RMSNorm [45] to achieve the efficient layer normalization. According to S^*, each query selects the visual token with the highest score and the overall training process is differentiable via the Gumbel-softmax operation, which is consistent with LightVLA. LightVLA* introduces the learnable query Q^* and the mapping parameters of layer normalization as the extra parameters, and it can be supported by the efficient inference-oriented platforms like vLLM and SGLang.

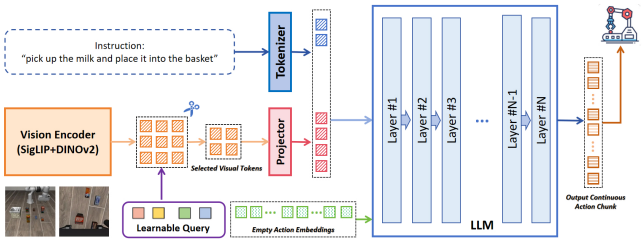

Fig. 4: Illustration of LightVLA* when pruning visual tokens at the vision encoder with the learnable query.

Learn to select at the early layer of LLM. Instead of operating at the image encoder, the compression queries Q_v

can interact with both visual tokens H_v and early-layer text tokens H_l in the LLM, as visualized in Figure 5. This allows the model leverages the semantic information from the task-level text to guide the token pruning process. The main idea is similar to the above part 'Learn to select at the vision encoder', but we introduce the attention scores (text tokens to visual tokens) as follows:

$$S^{\dagger} = \frac{LN(Q^{\dagger}) \cdot LN(H_v^T) + \zeta \cdot attn}{\sqrt{D}}, \tag{10}$$

where D denotes the projected vision token dimension, $attn$ indicates the attention scores of visual tokens to the text tokens, $Q^{\dagger} \in \mathbb{R}^{N_q \times D}$ denotes the learnable query and ζ denotes the learnable trade-off weight of the cross attention weights which is initialized with 1.0. The attention score is output by the corresponding decoder layer of LLM. We prune redundant visual tokens at early layers (layer#1 to layer#3) instead of deeper layers (layer#4 to layer#32), as the deep layers have already developed rich cross-modal representations where visual and textual features are deeply fused and semantically entangled. Another reason lies in pruning visual tokens at early layers of LLM facilitates the better computation decrease since each decoder layer requires extensive quadratic attention-based computation. Note that when pruning visual tokens at the decoder layer, LightVLA* requires output attention scores which can not be supported by above mentioned inference-oriented platforms.

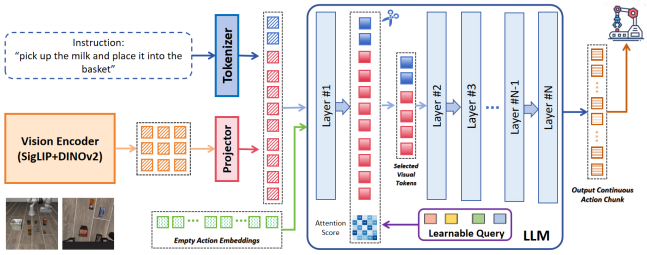

Fig. 5: Illustration of LightVLA* when pruning visual tokens at the first decoder layer of LLM with the learnable query.

Analysis. In LightVLA* experiments, we initialize $N_q = 128$ queries, which firstly retains 25% of visual tokens since our LightVLA indicates adaptively maintaining 10% to 20% visual tokens is sufficient. Experimental results are listed in Table V. We find both LightVLA* series and LightVLA achieves better performance against their counterparts in Table I. LightVLA* at the first decoder layer can achieve the best performance for the complex LIBERO-long task suite. We can also observe a slight performance decrease when pruning visual tokens in the deeper decoder layer.

B. Comparison with MoE

It is obvious that LightVLA and the mixture of expert (MoE) technique [46] share the similar intuition: Both techniques select a dense subset of elements from the whole collection to optimize the forward efficiency. The main dissimilarity between LightVLA and MoE is their different goals, which in turn lead to different behaviors. LightVLA aims to maximize performance while improving efficiency. Therefore, LightVLA focuses its token selection to only informative tokens. In contrast, MoE is proposed to divide specialized tasks into subtasks handled by experts. To balance knowledge and workload among experts, MoE distributes its selection evenly among experts without a particular focus. In conclusion, LightVLA and MoE are fundamentally different in both goals and behaviors, making them distinct techniques.

TABLE V: Experimental results with learnable token query. SR denotes the success rate (SR).

Method	Pruning Position	Spatial SR(%)	Object SR(%)	Goal SR(%)	Long SR(%)	Avg.
OpenVLA-OFT	-	97.6	94.2	95.2	92.0	94.8
LightVLA*	Vision Encoder	98.2	96.2	96.2	94.2	96.2
LightVLA*	LLM (layer#1)	98.0	**98.0**	97.2	**94.8**	97.0
LightVLA*	LLM (layer#2)	97.8	**98.0**	96.6	94.2	96.7
LightVLA*	LLM (layer#3)	97.6	**98.0**	97.0	93.8	96.6
LightVLA	Vision Encoder	**98.4**	97.6	**98.2**	94.6	**97.4**

VI. Conclusion

In this work, we investigate the inherent visual redundancy of vision-language-action (VLA) models, and propose the extra parameter-free visual token pruning framework LightVLA. With differentiable query-based token pruning process, it adaptively selects informative visual tokens. It achieves the state-of-the-art performance on the LIBERO benchmark with the significant computational optimization. We also propose another framework LightVLA* with learnable query as additional trainable parameters, which also outperforms against its counterparts.

Regarding future works, our research schedule is two-fold. Firstly, we aim to further investigate the visual redundancy in end-to-end VLMs or VLA models oriented for autonomous driving to optimize the overall efficiency and latency which facilitates the real-world deployment. Meanwhile, we plan to explore the efficient token pruning to VLMs or VLA models designed for complicated spatial intelligence tasks which facilitates the wider application of consumer-level devices like household robots.

ACKNOWLEDGMENT

We thank Pengxiang Li from LiAuto for discussion and Fang Yang from Tsinghua University for writing advice.

References

[1] C. Cheang, S. Chen, Z. Cui, Y. Hu, L. Huang, T. Kong, H. Li, Y. Li, Y. Liu, X. Ma, *et al.*, "Gr-3 technical report," *arXiv preprint arXiv:2507.15493*, 2025.

[2] M. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, D. Sadigh, S. Levine, and C. Finn, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[3] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter, *et al.*, "pi0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[4] 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.

[5] Physical Intelligence, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," 2025.

[6] Q. Li, Y. Liang, Z. Wang, L. Luo, X. Chen, Z. Liao, X. Zhang, Y. Wang, G. Wu, L. Wang, *et al.*, "Cogact: A foundational vision-language-action model for synergizing cognition and action in robotic manipulation," *arXiv preprint arXiv:2411.19650*, 2024.

[7] M. J. Kim, C. Finn, and P. Liang, "Fine-tuning vision-language-action models: Optimizing speed and success," *arXiv preprint arXiv:2502.19645*, 2025.

[8] J. Lee, J. Duan, H. Fang, Y. Deng, S. Liu, B. Li, B. Fang, J. Zhang, Y. R. Wang, S. Lee, *et al.*, "Molmoact: Action reasoning models that can reason in space," *arXiv preprint arXiv:2508.07917*, 2025.

[9] Figure AI, "Helix: A vision-language-action model for generalist humanoid control," https://www.figure.ai/news/helix, February 2025.

[10] T. Jiang, Q. Dong, Y. Ma, X. Ji, and Y. Liu, "Customizable multimodal trajectory prediction via nodes of interest selection for autonomous vehicles," *Expert Systems with Applications*, vol. 288, p. 128222, 2025.

[11] X. Jiang, Y. Ma, P. Li, L. Xu, X. Wen, K. Zhan, Z. Xia, P. Jia, X. Lang, and S. Sun, "Transdiffuser: End-to-end trajectory generation with decorrelated multi-modal representation for autonomous driving," *arXiv preprint arXiv:2505.09315*, 2025.

[12] Y. Yang, L. Zhang, Z. Chen, H. Wang, Y. Gao, Z. Wang, C.-L. Zhang, and K.-W. Chang, "Efficientvla: Training-free acceleration and compression for vision-language-action models," 2025.

[13] S. Zhang, C.-L. Zhang, Z. Wang, and K.-W. Chang, "Llava-mini: Efficient image and video large multimodal models with one vision token," 2025.

[14] C. Liang, S. Zhang, L. Zhang, Z. Wang, H. Wang, C.-L. Zhang, K.-W. Chang, T. Wang, and Y. You, "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 (ECCV)*, 2024.

[15] Y. Zhang, C.-K. Fan, J. Ma, W. Zheng, T. Huang, K. Cheng, D. Gudovskiy, T. Okuno, Y. Nakata, K. Keutzer, *et al.*, "Sparsevlm: Visual token sparsification for efficient vision-language model inference," *arXiv preprint arXiv:2410.04417*, 2024.

[16] 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.

[17] L. Xing, Q. Huang, X. Dong, J. Lu, P. Zhang, Y. Zang, Y. Cao, C. He, J. Wang, F. Wu, *et al.*, "Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction," *arXiv preprint arXiv:2410.17247*, 2024.

[18] Y. Sun, Y. Xin, H. Li, J. Sun, C. Lin, and R. Batista-Navarro, "Lvpruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models," *arXiv preprint arXiv:2501.13652*, 2025.

[19] Q. Zhang, A. Cheng, M. Lu, R. Zhang, Z. Zhuo, J. Cao, S. Guo, Q. She, and S. Zhang, "Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms," *arXiv preprint arXiv:2412.01818*, 2024.

[20] Q. Zhang, M. Liu, L. Li, M. Lu, Y. Zhang, J. Pan, Q. She, and S. Zhang, "Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms," *arXiv preprint arXiv:2506.10967*, 2025.

[21] C.-Y. Hung, Q. Sun, P. Hong, A. Zadeh, C. Li, U. Tan, N. Majumder, S. Poria, *et al.*, "Nora: A small open-sourced generalist vision language action model for embodied tasks," *arXiv preprint arXiv:2504.19854*, 2025.

[22] M. Shukor, D. Aubakirova, F. Capuano, P. Kooijmans, S. Palma, A. Zouitine, M. Aractingi, C. Pascal, M. Russi, A. Marafioti, *et al.*, "Smolvla: A vision-language-action model for affordable and efficient robotics," *arXiv preprint arXiv:2506.01844*, 2025.

[23] J. Wen, Y. Zhu, J. Li, M. Zhu, Z. Tang, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen, *et al.*, "Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation," *IEEE Robotics and Automation Letters*, 2025.

[24] X. Tan, Y. Yang, P. Ye, J. Zheng, B. Bai, X. Wang, J. Hao, and T. Chen, "Think twice, act once: Token-aware compression and action reuse for efficient inference in vision-language-action models," *arXiv preprint arXiv:2505.21200*, 2025.

[25] Y. Li, Y. Meng, Z. Sun, K. Ji, C. Tang, J. Fan, X. Ma, S. Xia, Z. Wang, and W. Zhu, "Sp-vla: A joint model scheduling and token pruning approach for vla model acceleration," *arXiv preprint arXiv:2506.12723*, 2025.

[26] S. Xu, Y. Wang, C. Xia, D. Zhu, T. Huang, and C. Xu, "Vla-cache: Towards efficient vision-language-action model via adaptive token caching in robotic manipulation," *arXiv preprint arXiv:2502.02175*, 2025.

[27] A. Potapczynski, G. Loaiza-Ganem, and J. P. Cunningham, "Invertible gaussian reparameterization: Revisiting the gumbel-softmax," *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 311–12 321, 2020.

[28] H. Liu, C. Li, Y. Li, and Y. J. Lee, "Improved baselines with visual instruction tuning," in *IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 26 296–26 306.

[29] 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 *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[30] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, *et al.*, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," in *IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 24 185–24 198.

[31] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, *et al.*, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *arXiv preprint arXiv:2409.12191*, 2024.

[32] J. Cen, C. Yu, H. Yuan, Y. Jiang, S. Huang, J. Guo, X. Li, Y. Song, H. Luo, F. Wang, *et al.*, "Worldvla: Towards autoregressive action world model," *arXiv preprint arXiv:2506.21539*, 2025.

[33] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman, "Token merging: Your vit but faster," *arXiv preprint arXiv:2210.09461*, 2022.

[34] S. Kim, S. Shen, D. Thorsley, A. Gholami, W. Kwon, J. Hassoun, and K. Keutzer, "Learned token pruning for transformers," in *ACM SIGKDD conference on knowledge discovery and data mining*, 2022, pp. 784–794.

[35] L. Cao, Z. Zhang, Y. Qu, and Y. Shen, "Fastvggt: Training-free acceleration of visual geometry transformer," 2025.

[36] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Symposium on operating systems principles*, 2023, pp. 611–626.

[37] L. Zheng, L. Yin, Z. Xie, C. L. Sun, J. Huang, C. H. Yu, S. Cao, C. Kozyrakis, I. Stoica, J. E. Gonzalez, *et al.*, "Sglang: Efficient execution of structured language model programs," *Advances in neural information processing systems*, vol. 37, pp. 62 557–62 583, 2024.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[39] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 44 776–44 791, 2023.

[40] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.

[41] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986.

[42] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[43] E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, *et al.*, "Lora: Low-rank adaptation of large language models," in *International Conference on Learning Representations*, 2022.

[44] L. Sun, B. Xie, Y. Liu, H. Shi, T. Wang, and J. Cao, "Geovla: Empowering 3d representations in vision-language-action models," *arXiv preprint arXiv:2508.09071*, 2025.

[45] B. Zhang and R. Sennrich, "Root mean square layer normalization," *Advances in neural information processing systems*, vol. 32, 2019.

[46] D. Dai, C. Deng, C. Zhao, R. Xu, H. Gao, D. Chen, J. Li, W. Zeng, X. Yu, Y. Wu, *et al.*, "Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models," *arXiv preprint arXiv:2401.06066*, 2024.