VLA-Cache: Efficient Vision-Language-Action Manipulation via Adaptive Token Caching

NVIDIA GTC 2026

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS 2025

THE UNIVERSITY OF SYDNEY

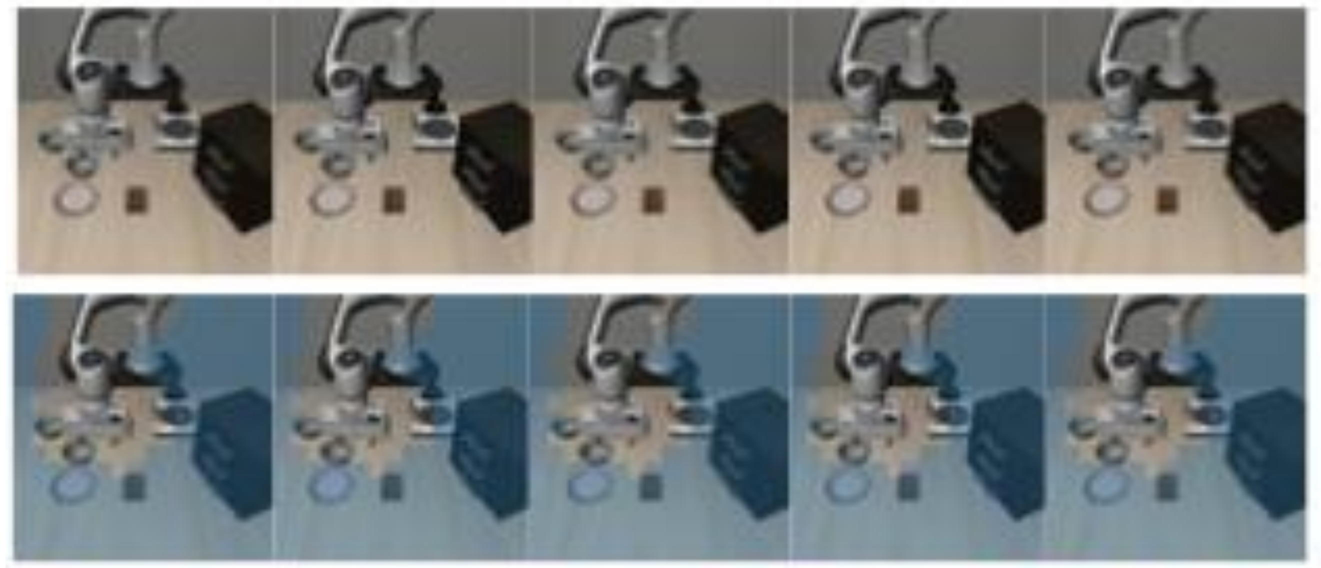
Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, Chang Xu

Email: s.xu@sydney.edu.au
Homepage: https://vla-cache.github.io

A training-free method for real-time acceleration in Vision-Language-Action models

Motivation

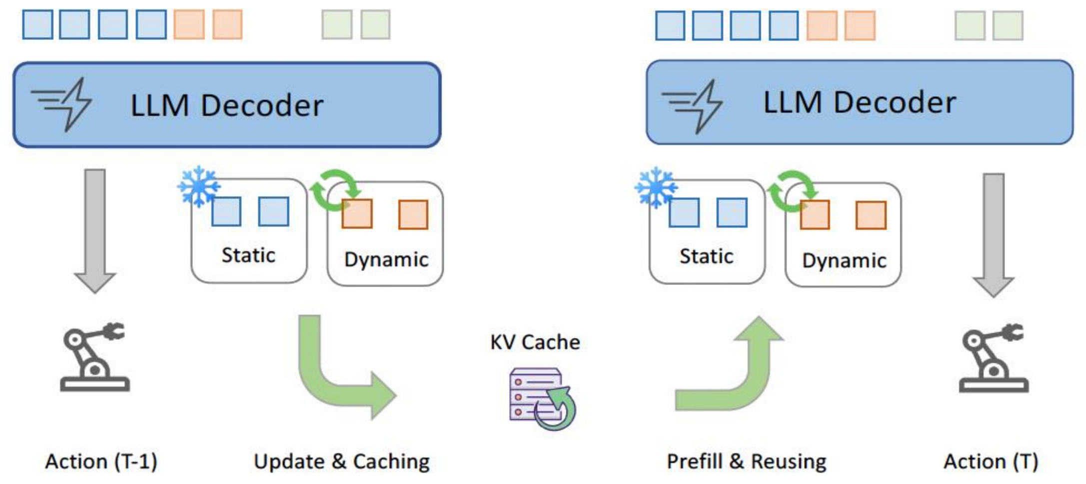
- **V**ision-**L**anguage-**A**ction models enable generalizable robotic manipulation.
- **High computational cost** in robotic VLA inference limits real-time control.
- Many visual tokens remain static across frames yet are redundantly processed.



Key insight: Instead of re-encoding all visual tokens per frame, we can reuse information that remains unchanged.

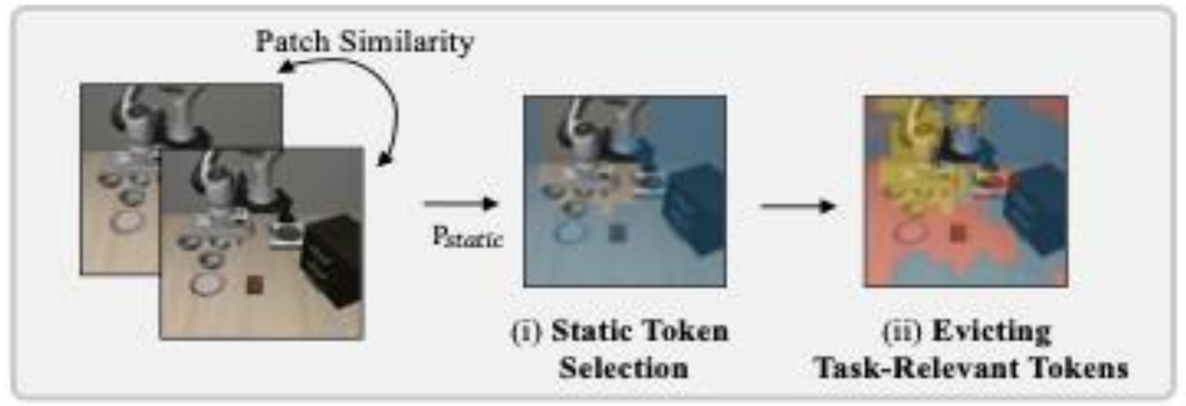
Contributions

- **VLA-Cache** provides a training-free caching method for real-time robotic control.
- Reuses static tokens for faster, accurate inference.
- Experiments show **1.7× faster latency** and **15% higher control frequency** with no performance loss.



Dynamic Token Selection

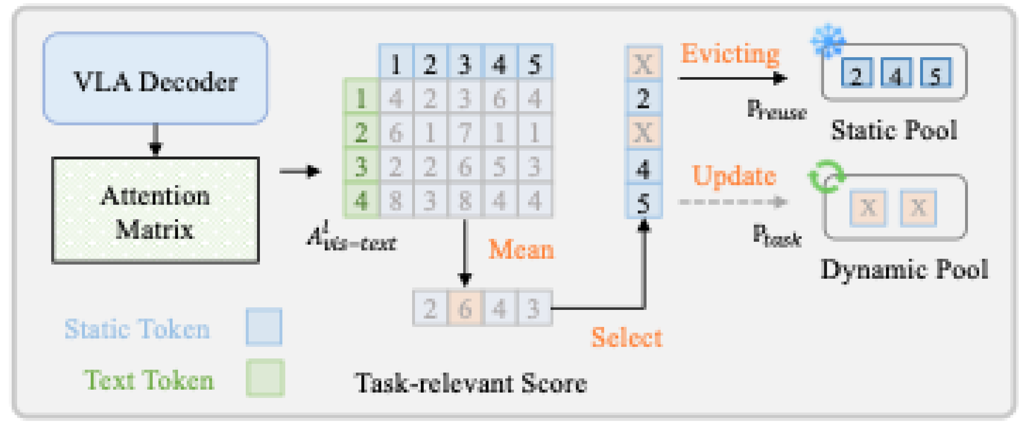
- **Static-token caching:** identifies visually unchanged patches cross frames and caches them for reuse.
- **Task-relevant filtering:** uses cross-attention scores from the language decoder to remove semantically critical regions.

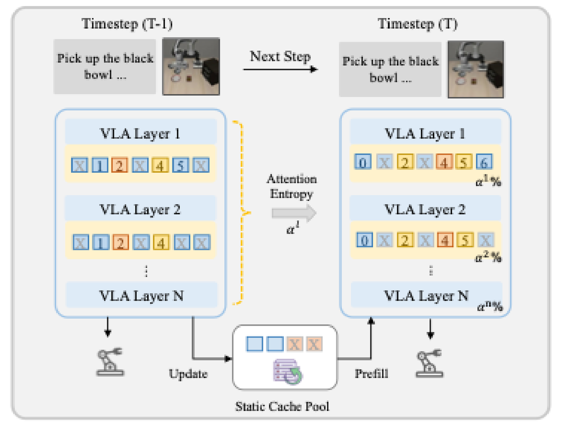


"Reuse what's redundant, recompute what matters."

Adaptive Token Caching

- Attention patterns differ across decoder layers.
- Layers with more concentrated attention reuse more tokens.
- **Layer-adaptive reuse** balances efficiency and fidelity.





Module-wise Performance Gains

- Static-token caching reduces latency but harms success rate.
- Task-relevant filtering restores accuracy.
- Layer-adaptive reuse achieving the best overall performance.

Method	SR (%) ↑	Latency (ms) ↓
OpenVLA	84.4	51.56
+ Static Token	74.2	31.03
+ Evict Task-Relevant	82.6	31.03
+ Layer Adaptive	83.8	32.22

Takeaway: static caching accelerates, filtering safeguards accuracy, and layer-adaptive reuse achieves optimal balance.

Simulation Results

Evaluations on the **LIBERO** and **SIMPLER** benchmarks show that VLA-Cache consistently accelerates VLA inference while maintaining comparable or higher success rates.

LIBERO

Method	Success Rate ↑					FLOPs (T)↓	Latency (ms)↓	Control Freq. (Hz)↑
	Spatial	Object	Goal	Long	Average			
OpenVLA	84.4%	86.6%	75.6%	53.2%	75.0%	1.864	51.91	4.23
+ SparseVLM	79.8%	67.0%	72.6%	39.4%	64.7%	1.407	83.39	3.72
+ FastV	83.4%	84.0%	74.2%	51.6%	73.3%	1.864	53.28	4.19
+ **VLA-Cache**	**83.8%**	**85.8%**	**76.4%**	**52.8%**	**74.7%**	**1.355**	**31.83**	**4.59**
OpenVLA-OFT	97.8%	97.6%	97.6%	94.2%	96.8%	4.013	79.05	65.10
+ **VLA-Cache**	**98.3%**	97.5%	**98.3%**	**95.4%**	**97.4%**	**3.097**	**62.59**	**78.98**

SIMPLER

SIMPLER	Method	Success Rate ↑					FLOPs (T)↓	Latency (ms) ↓	Control Freq. (Hz) ↑
		PickCan	MoveNear	Drawer	DrawerApple	Average			
Matching	CogACT	91.3%	**85.0%**	**71.8%**	50.9%	**74.8%**	1.847	54.29	12.42
	+ **VLA-Cache**	**92.0%**	83.3%	70.5%	**51.6%**	74.4%	**1.496**	**39.63**	**14.66**
Aggregation	CogACT	89.6%	80.8%	28.3%	**46.6%**	61.3%	1.807	53.54	12.36
	+ **VLA-Cache**	**91.7%**	**79.3%**	**32.5%**	45.8%	**62.3%**	**1.493**	**39.11**	**14.48**

VLA-Cache achieves 1.7× lower latency and improved control frequency without retraining.

Real-world Results

On real-robot manipulation tasks with OpenVLA, VLA-Cache transfers effectively from simulation to reality.

Method	Success Rate ↑					FLOPs (T)↓	Latency (ms) ↓	Control Freq. (Hz) ↑
	PickPot	PlaceCube	PutSausage	WipeTable	Average			
OpenVLA	**95.0%**	83.3%	80.0%	70.0%	82.1%	1.814	64.16	4.02
+ **VLA-Cache**	90.0%	**90.0%**	**85.0%**	**73.3%**	**84.6%**	**1.303**	**51.85**	**4.21**

Dynamic Viewpoint & Scene

VLA-Cache remains robust under **camera motion** and **dynamic scene changes**, enabling reliable inference from wrist-mounted views and complex real-time environments.

Dynamic Scene

Wrist Camera