valeo.ai

Hugging Face

SmolVLA: A vision-language-action model for affordable and efficient robotics

Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, Simon Alibert, Matthieu Cord, Thomas Wolf, Remi Cadene

2025

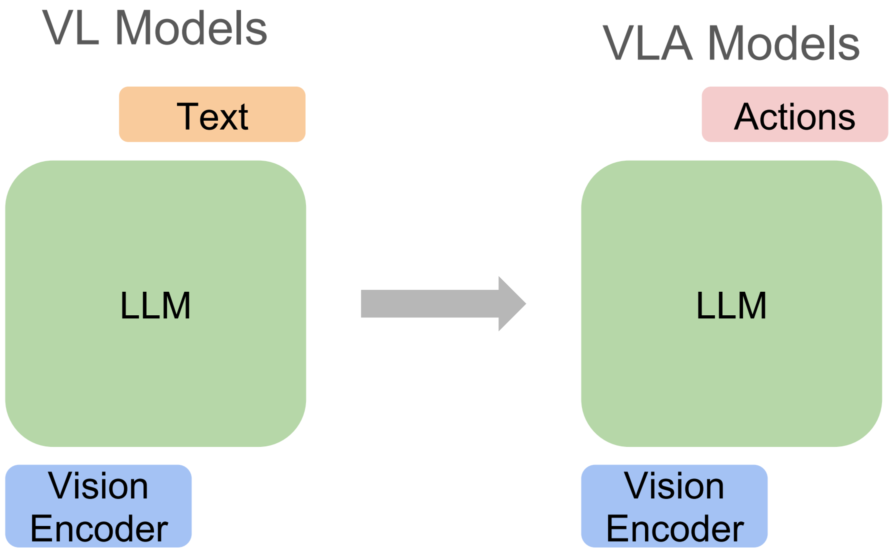
Hugging Face

SmolVLA: sorting cubes based on colors

Instruction:
Put the red cube in the right box and the blue cube in the left box

Hugging Face

VLAs: vision-language-action models



Hugging Face

VLAs: vision-language-action models

Open X-Embodiment: Robotic Learning Datasets and RT-X Models

Open X-Embodiment Collaboration[0]

NVIDIA

2025-3-28

GR00T N1: An Open Foundation Model for Generalist Humanoid Robots

Google DeepMind

https://robotics-transformer2.github.io

2023-8-1

RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control

DexVLA: Vision-Language Model with Plug-In Diffusion Expert for General Robot Control

π_0: A Vision-Language-Action Flow Model for General Robot Control

Physical Intelligence

Octo: An Open-Source Generalist Robot Policy

Octo Model Team

OpenVLA: An Open-Source Vision-Language-Action Model

SmolVLA: overview

Community datasets

→ **SmolVLA** →

Affordable robots

LeRobot

- Community datasets
- ~ 480 datasets
- Tabletop manipulation tasks
- VLM annotation

- Small (0.45B params)
- Efficient at training/inference
- Asynchronous inference

- So100, So101
- 100s $
- 3D printed robots

Hugging Face

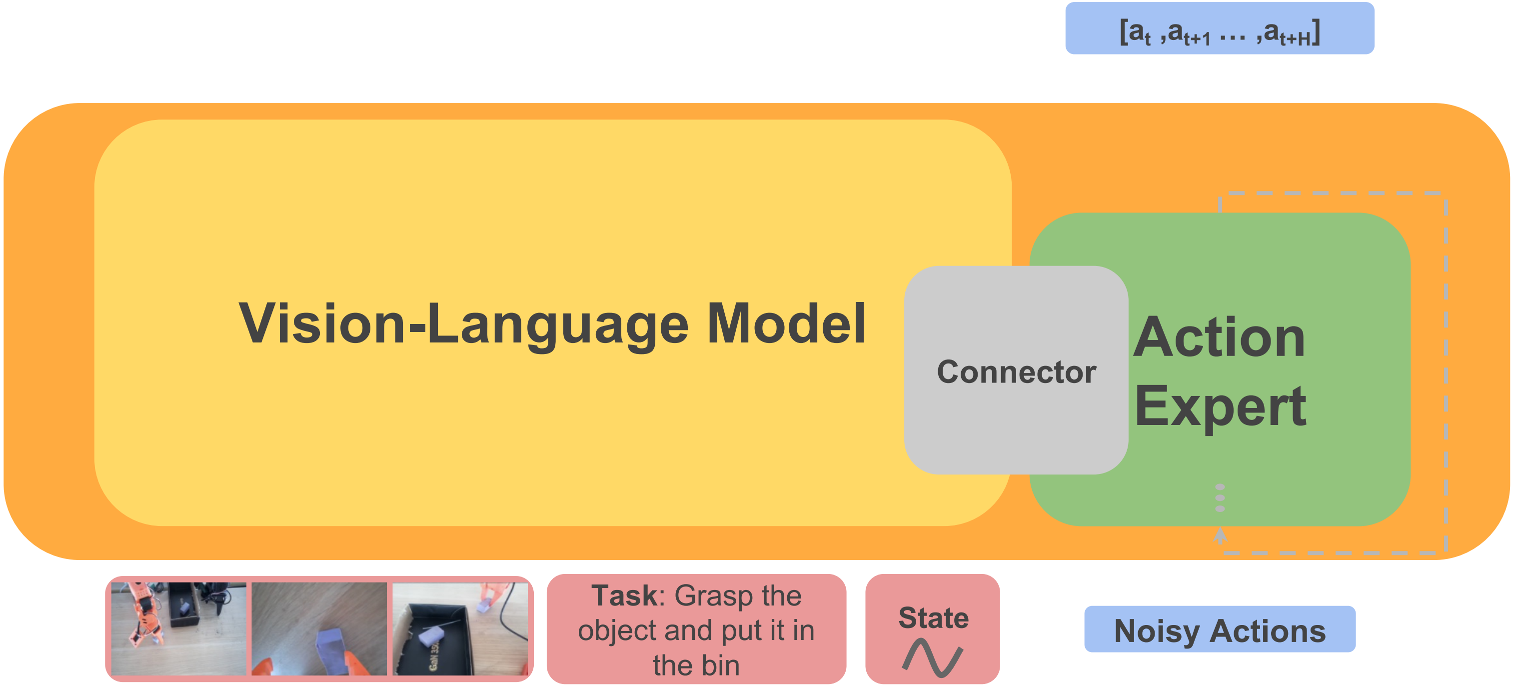
SmolVLA: model architecture

SmolVLA

Hugging Face

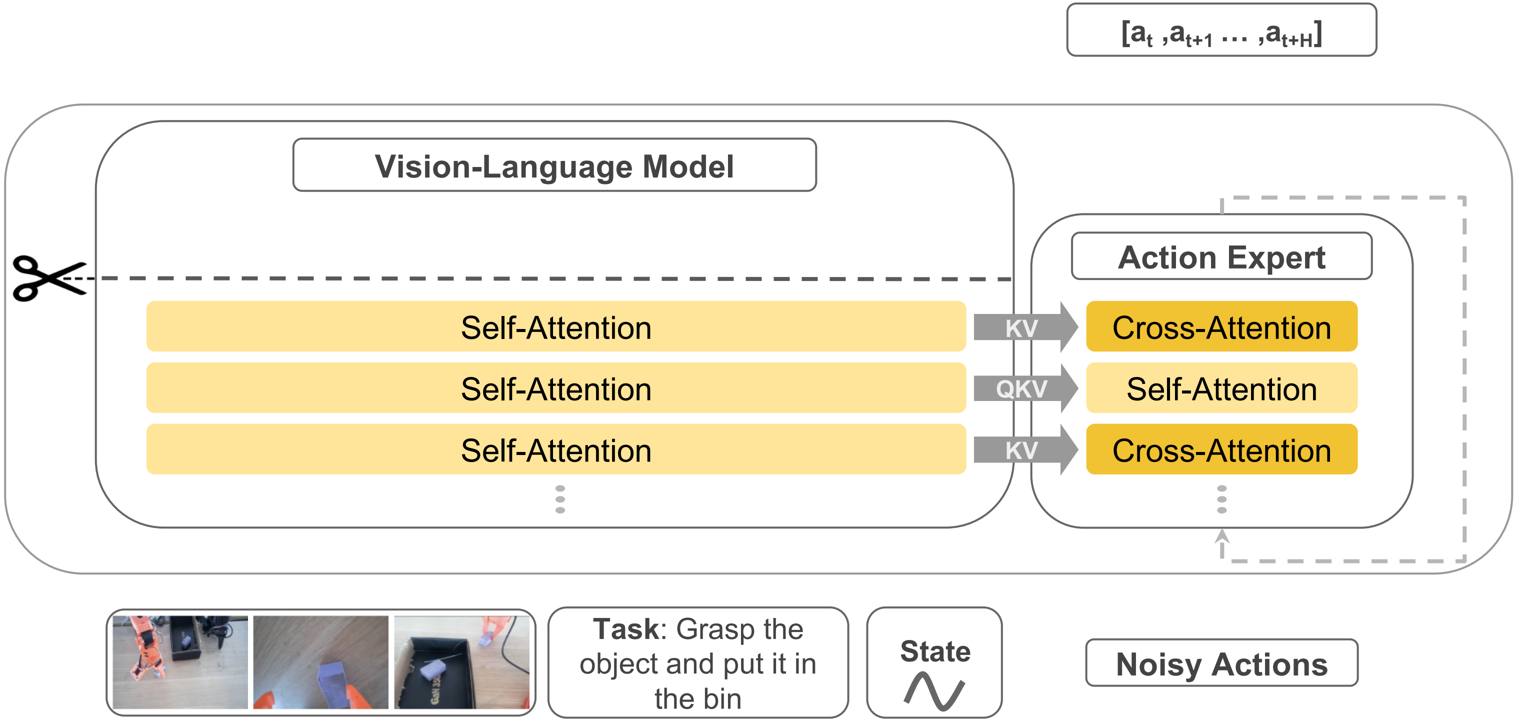
SmolVLA: model architecture

- VLM (SmolVLM-2)
- Action expert (transformer + flow matching)
- Linear connectors



SmolVLA: model architecture

- Skipping layers
- Interleaved CA/SA
- States to prefix
- Causal attention on actions
- Few visual tokens



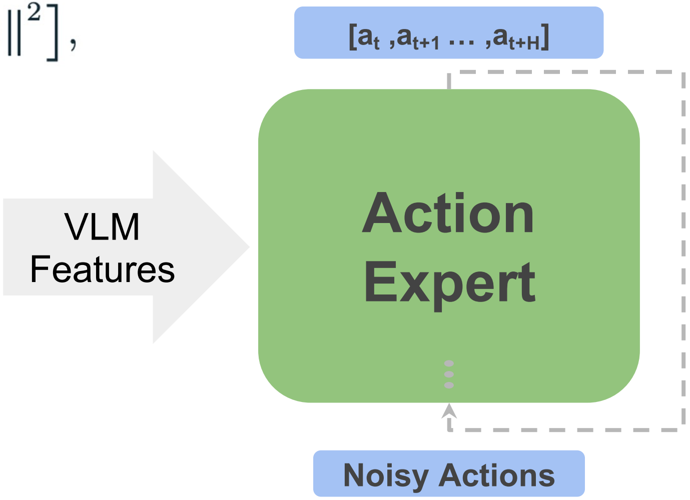
Hugging Face

SmolVLA: action expert with flow matching

$$\mathcal{L}^\tau(\theta) = \mathbb{E}_{p(\mathbf{A}_t|\mathbf{o}_t),q(\mathbf{A}_t^\tau|\mathbf{A}_t)}\left[\left\|\mathbf{v}_\theta(\mathbf{A}_t^\tau,\mathbf{o}_t) - \mathbf{u}(\mathbf{A}_t^\tau \mid \mathbf{A}_t)\right\|^2\right],$$

$$\mathbf{u}(\mathbf{A}_t^\tau \mid \mathbf{A}_t) = \epsilon - \mathbf{A}_t$$

$$\mathbf{A}_t^\tau = \tau\mathbf{A}_t + (1-\tau)\epsilon,$$



Training objective	Success Rate (%) — LIBERO				
	S	O	G	10	Avg
Flow matching	89	94	85	53	80.25
Regression	92	85	86	38	75.25

Lipman, Yaron, et al. "Flow matching for generative modeling." *arXiv preprint arXiv:2210.02747* (2022).

SmolVLA: pretraining on community datasets

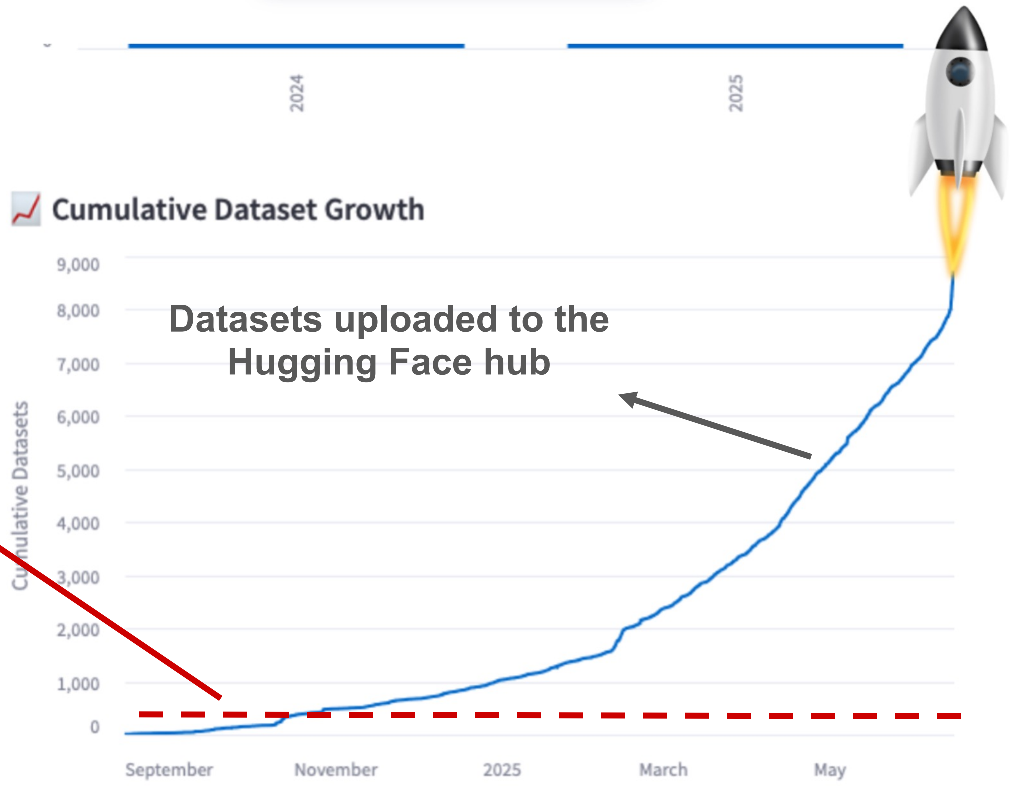
Community datasets

SmolVLA: pretraining on community datasets

# datasets	# episodes	# frames
481	22.9K	10.6M

SmolVLA pretraining dataset

(Other VLAs are trained on more than 1M episodes, e.g. OpenVLA/PI0)



SmolVLA: pretraining and multitask finetuning

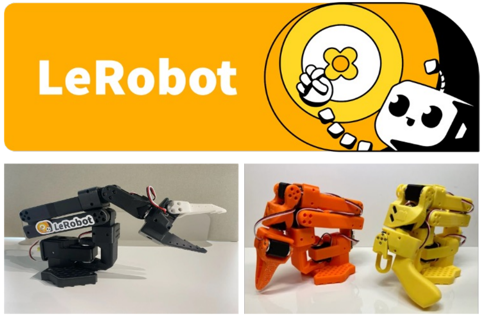
Policy	VLA pt.	Success Rate (%) — Real World			
		Pick-Place	Stacking	Sorting	Avg.
Single-task Training					
SmolVLA (0.45B)	No	55	45	20	40
Multi-task Training					
SmolVLA (0.45B)	No	80	40	35	51.7
SmolVLA (0.45B)	Yes	75	90	70	78.3

\+ 11.7 pt

\+ 27.6 pt

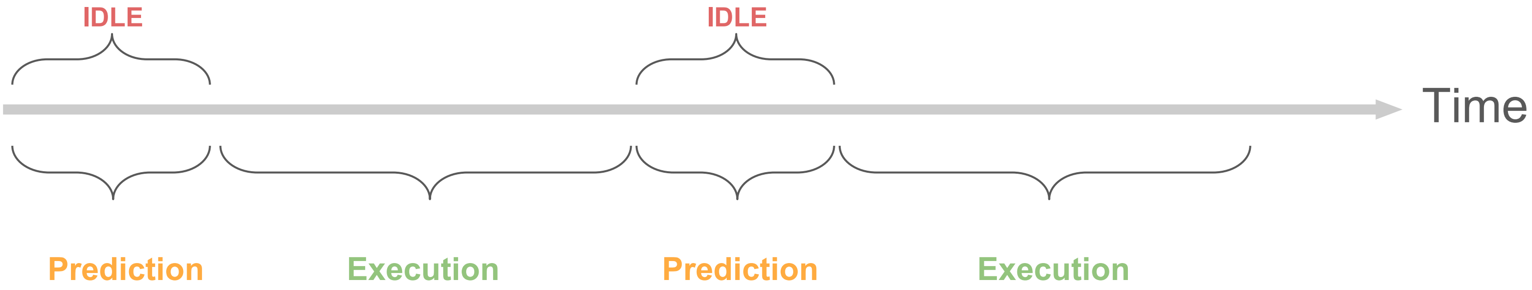
SmolVLA: asynchronous inference

Affordable robots

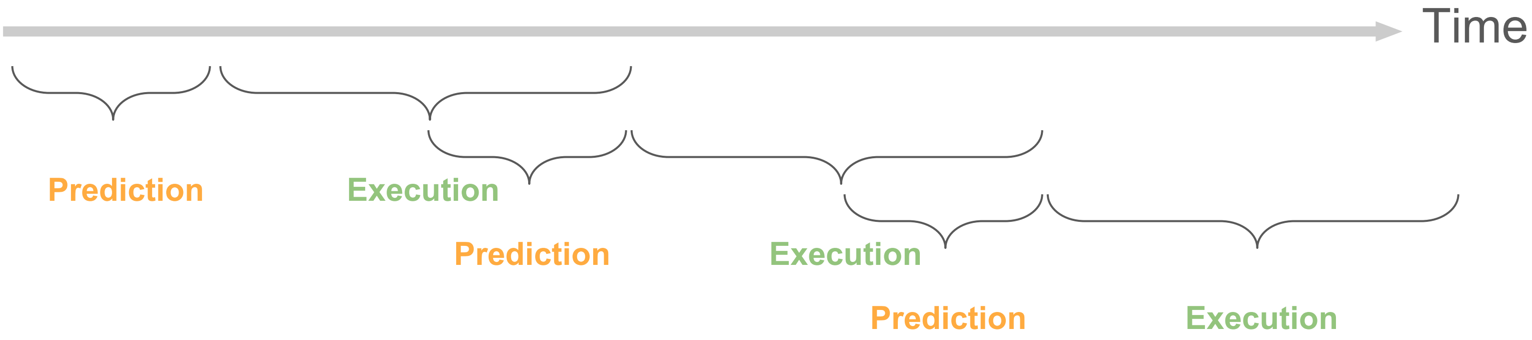


SmolVLA: asynchronous inference

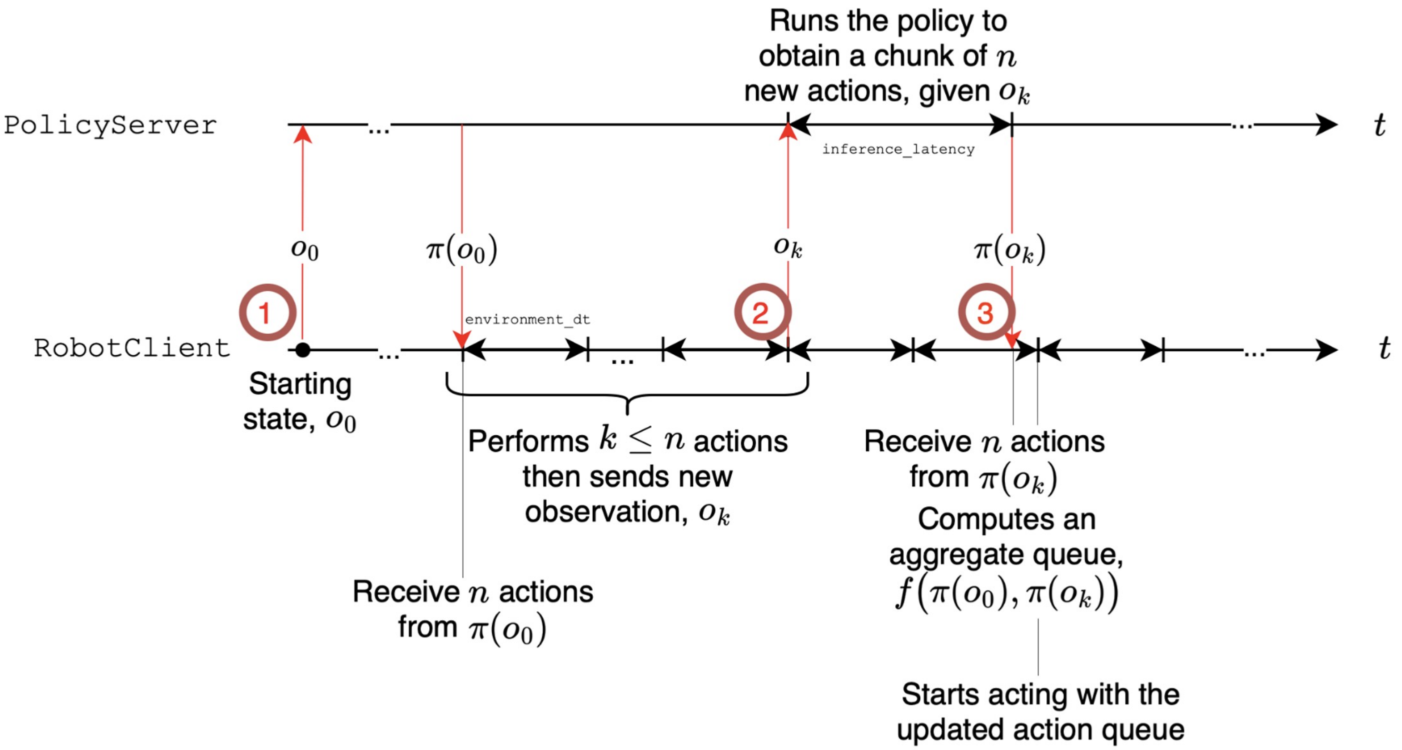
Current approach: synchronous inference



Our approach: asynchronous inference



SmolVLA: asynchronous inference



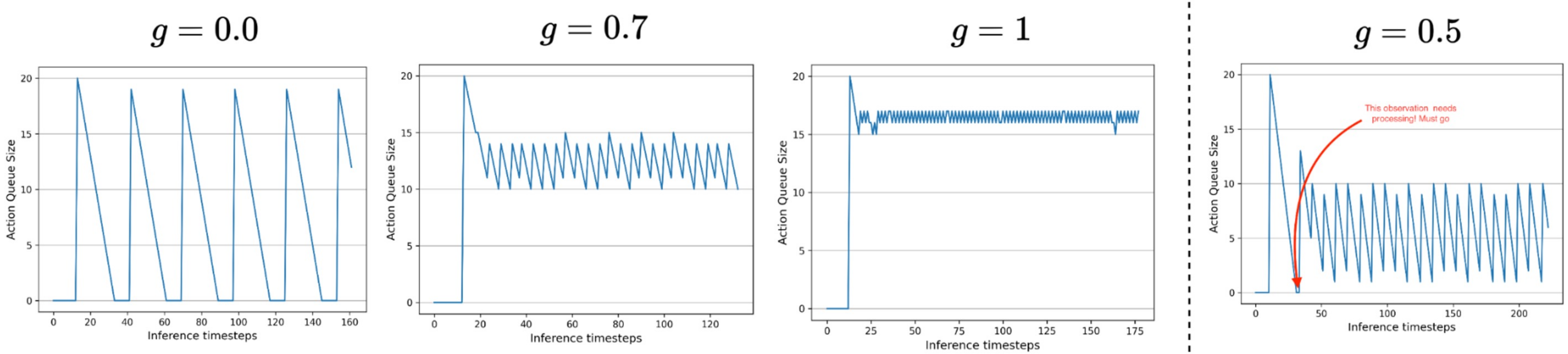
Hugging Face

SmolVLA: asynchronous inference

Algorithm 1 Asynchronous inference control-loop

Input: horizon T, chunk size n, threshold $g \in [0,1]$
Init: capture o_0; send o_0 to $\textsc{PolicyServer}$; receive $\mathbf{A}_0 \leftarrow \pi(o_0)$
for t **to** T **do**
  $a_t \leftarrow \textsc{PopFront}(\mathbf{A}_t)$
  $\textsc{Execute}(a_t)$ ▷ execute action at step t
  if $\frac{|\mathbf{A}_t|}{n} < g$ **then** ▷ queue below threshold
    capture new observation, o_{t+1}
    if $\textsc{NeedsProcessing}(o_{t+1})$ **then** ▷ similarity filter, or triggers direct processing
      `async_handle` $\leftarrow \textsc{AsyncInfer}(o_{t+1})$ ▷ Trigger new chunk prediction (non blocking)
      $\tilde{\mathbf{A}}_{t+1} \leftarrow \pi(o_{t+1})$ ▷ New queue is predicted with the policy
      $\mathbf{A}_{t+1} \leftarrow f(\mathbf{A}_t, \tilde{\mathbf{A}}_{t+1})$ ▷ aggregate overlaps (if any)
    end if
  end if
  if $\textsc{NotCompleted}$(`async_handle`) **then**
    $\mathbf{A}_{t+1} \leftarrow \mathbf{A}_t$ ▷ No update on queue (inference is not over just yet)
  end if
end for

SmolVLA: asynchronous inference



(A) **Without** observation filtering

(B) **With** observation filtering

SmolVLA: asynchronous inference

SmolVLA: asynchronous inference

Inference	Success Rate (%) — Real World			
	Pick-Place	Stacking	Sorting	Avg
Sync	75	90	70	78.3
Async	80	90	50	73.3

(a) | Performance (success rates).

Inference	Time (s) — Real World		
	Total	Avg	Std
Sync	137.5	13.75	2.42
Async	97.0	9.70	2.95

(b) | Task completion time.

Inference	# of Cubes — Real World		
	Total	Avg	Std
Sync	9	1.8	0.45
Async	19	3.8	1.3

(c) | Performance in fixed time.

- Performance on par with sync inference in typical evaluation setups
- Faster to complete tasks
- Better reactivity and adaptability to environment changes

SmolVLA: sync vs async inference

- Async inference is faster
- Complete more tasks in fixed time frame

Hugging Face

SmolVLA: main results (real world)

Policy	Success Rate (%) — Real World			
	Pick-Place	Stacking	Sorting	Avg.
Single-task Training				
ACT	70	50	25	48.3
Multi-task Training				
π_0 (3.5B)	100	40	45	61.7
SmolVLA (0.45B)	75	90	70	78.3

Table 3 | Real-world benchmarks (SO100). Success rate (%) across three tasks using policies trained in multi-task and single-task settings.

Policy	Success Rate (%) — Real World	
	In Distribution	Out of Distribution
Single-task Training		
ACT	70	40
SmolVLA (0.45B)	90	50

Table 4 | Real-world benchmark (SO101). Success rate (%) for the Pick-Place-Lego task using policies trained in single-task setting.

SmolVLA: main results (simulation)

Benchmark	Policy (# Params)	VLA Pt.	Success Rate (%) — Simulation				
LIBERO			**Spatial**	**Object**	**Goal**	**Long**	**Avg.**
	Diffusion Policy (Khazatsky et al., 2024)	No	78.3	92.5	68.3	50.5	72.4
	Octo (0.09B) (Team et al., 2024)	Yes	78.9	85.7	84.6	51.1	75.1
	OpenVLA (7B) (Kim et al., 2024)	Yes	84.7	88.4	79.2	53.7	76.5
	π_0 (Paligemma-3B)	No	87	63	89	48	71.8
	π_0 (3.3B)	Yes	90	86	95	73	86.0
	SmolVLA (0.24B)	No	87	93	88	63	82.75
	SmolVLA (0.45B)	No	90	96	92	71	87.3
	SmolVLA (2.25B)	No	93	94	91	77	**88.75**
Meta-World			**Easy**	**Medium**	**Hard**	**Very Hard**	**Avg.**
	Diffusion Policy (Chi et al., 2023)	No	23.1	10.7	1.9	6.1	10.5
	TinyVLA (Zhou et al., 2024)	No	77.6	21.5	11.4	15.8	31.6
	π_0 (3.5B-Paligemma)	No	80.4	40.9	36.7	44.0	50.5
	π_0 (3.5B)	Yes	71.8	48.2	41.7	30.0	47.9
	SmolVLA (0.24B)	No	86.43	46.36	35	60	56.95
	SmolVLA (0.45B)	No	82.5	41.8	45.0	60.0	57.3
	SmolVLA (2.25B)	No	87.14	51.82	70	64	**68.24**

Hugging Face

SmolVLA: ablation study (skipping layers)

	N	Success Rate (%) — LIBERO				
		S	O	G	10	Avg
VLM-500M	8	77	88	86	49	75.0
	16	88	91	91	44	78.5
	24	86	97	86	49	79.5
	32	89	94	85	53	80.3
	Skip %2	84	90	83	45	75.5
	VLM-256M	86	83	75	59	75.8

SmolVLA: ablation study (action expert size)

Expert width (w.r.t. VLM)	Success Rate (%) — LIBERO				
	S	O	G	10	Avg
×1.00	87	96	90	56	82.3
×0.75	82	89	84	55	77.5
×0.50	89	94	85	53	80.3
×0.25	76	97	83	39	73.8

SmolVLA: ablation study (attention mask)

Attention mask	Success Rate (%) — LIBERO				
	S	O	G	10	Avg
Bidir	79	86	82	23	67.5
Causal	80	94	84	40	74.5

SmolVLA: ablation study

Action Steps	Success Rate (%) — LIBERO				
	S	O	G	10	Avg
1	89	94	85	53	80.3
10	89	94	91	57	82.8
30	76	91	74	42	70.8
50	54	70	58	25	51.8

Sampling more observations leads to better scores

Chunk Size	Success Rate (%) — LIBERO				
	S	O	G	10	Avg
1	45	77	54	24	50.0
10	90	94	94	58	84.0
30	85	94	87	48	78.5
50	89	94	85	53	80.3
100	83	88	85	42	74.5

Training to predict chunk of actions is better than predicting single action

Hugging Face

SmolVLA: new robot (So101)

Hugging Face

SmolVLA: conclusion

Future work:

- Pretraining on more community and academic datasets
- Cross-embodiment training
- Scaling model size
- Better VLMs for robotics

Resources: 30K GPUhs + 100s euros for the hardware + 1-2 people

Code and assets in LeRobot: https://github.com/huggingface/lerobot

LeRobot