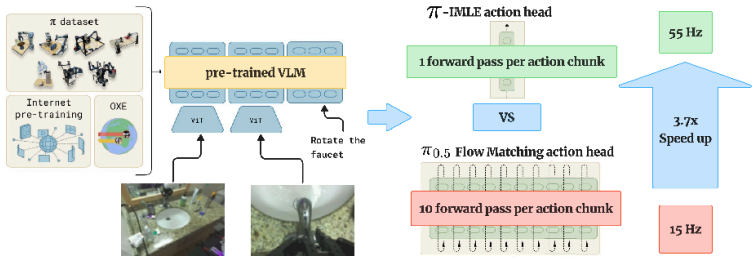
π-IMLE: Fast Single-Step Action Generation for Vision-Language-Action Models

Kian Hosseinkhani[1*], George Shramko[1], Mehran Aghabozorgi[1], Qinhe Peng[2], Jianing Qian[2], Tristan Engst[1], Alireza Moazeni[1], Dinesh Jayaraman[2], Ke Li[1]

[1]APEX Lab · School of Computing Science · Simon Fraser University [2]University of Pennsylvania

{kian_hosseinkhani, george_shramko, maa143, tme3, sam62, keli}@sfu.ca · {pengqh20, jianingq, dineshj}@seas.upenn.edu

❶ Motivation

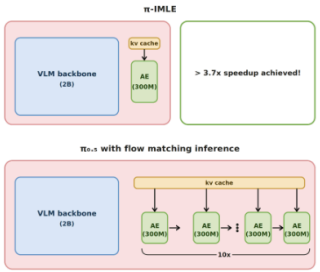


Why VLAs are slow
- VLAs → strong cross-task generalization → **slow** (built atop VLMs).
- **Stop-wait-execute** → robot idles each prediction → **jerky motion**.
- **Wide perception–action loop** → **policy acts on stale observations**.

Our fix: π-IMLE
- **SOTA speed** → single-step generation → **less jerky, higher reaction rate**.
- **Multimodal** → many valid ways to do a task, cIMLE learns *all* modes.

55 Hz · 3.67× faster | **98.3% LIBERO (best)** | **Robust on LIBERO-plus** | **Beats π0.5 on every real task**

❷ Our Method: π-IMLE



1 Frozen VLM backbone
Replace only the action head — training cost is minimal.

2 Single-step generator
G_θ maps (observation O_t, noise z) → action chunk $\hat{A}$ in one forward pass.

3 Trained via cIMLE
Mode-covering by construction, no iterative steps needed.

❸ Conditional IMLE: Formulation

- **Sample.** For each observation–action pair $(o_t^{(i)}, A_{gt}^{(i)})$, draw **m noise vectors** $z_{i,1..m} \sim \mathcal{N}(0, I)$ and produce a set of m candidates:

$$\hat{A}_{i,j} = G_\theta\left(o_t^{(i)}, z_{i,j}\right), \quad j = 1, \ldots, m$$

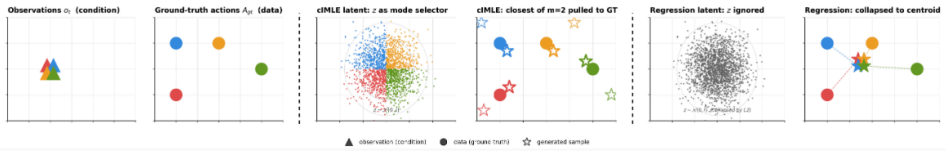
- **Assign.** Each $A_{gt}^{(i)}$ is assigned to its nearest candidate from $o_t^{(i)}$'s candidate set:

$$j^*(i) = \arg\min_j \|\hat{A}_{i,j} - A_{gt}^{(i)}\|_2^2$$

- **Update.** Average L2 loss over every ground-truth action:

$$\mathcal{L}_{\text{cIMLE}} = \frac{1}{n}\sum_{i=1}^{n} \|\hat{A}_{i,j^*(i)} - A_{gt}^{(i)}\|_2^2$$

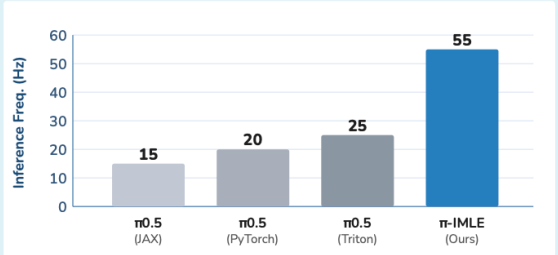
The loss penalizes every ground-truth action equally: any uncovered behavior mode raises the loss.



❹ Single-Step Generation Lifts the Inference Ceiling

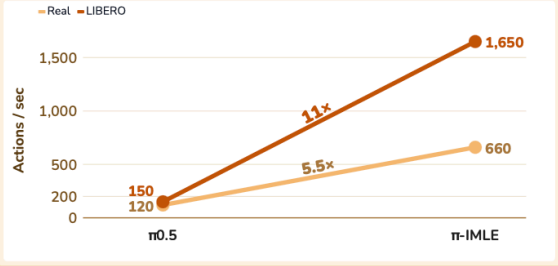
Inference Frequency (Hz)

π-IMLE attains **55 Hz**, surpassing every system-level $\pi_{0.5}$ optimization capped by its 10-step flow-matching head.



Action Throughput (actions / sec)

Faster forward passes combined with a longer execution horizon compound to **up to 11×** higher throughput.



❺ LIBERO Benchmark

40 tabletop tasks across 4 suites × 50 ep/task. **π-IMLE is the only method that simultaneously leads in success rate AND inference frequency.**

Model	H	Spat.	Obj.	Goal	Long	Avg ↑	Inf. ↑
$\pi_{0.5}$	10	97.2	99.0	97.8	**96.0**	97.5	1.0×
$\pi_{0.5}$	30	95.0	98.0	96.2	95.2	96.1	1.0×
OpenVLA-OFT	—	95.2	94.2	95.2	93.2	94.5	0.69×
Shallow-$\pi_{0.5}$-L9	—	**99.0**	98.0	97.0	93.0	97.0	1.7×
Shallow-$\pi_{0.5}$-L6	—	98.0	96.0	94.0	90.0	95.0	2.3×
π-IMLE	**10**	98.0	**100.0**	**99.0**	**96.0**	**98.3**	**3.67×**
π-IMLE	**30**	97.2	99.0	96.2	95.8	97.1	**3.67×**

VLA Wall-Clock per Episode (Simulation)

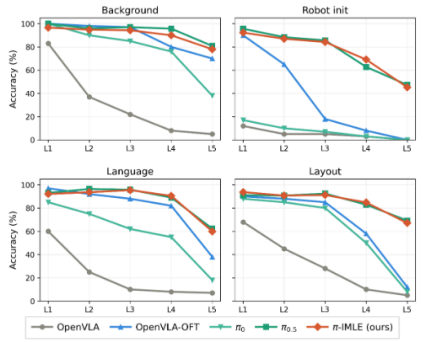
Def. Wall-clock on policy inference per episode (excluding robot execution/env overhead).

System	Fwd ↓	VLA-WC (s) ↓	Faster ↑
$\pi_{0.5}$ (JAX, H=10)	15.4	1.03	1.0×
π-IMLE (H=10)	15.5	**0.28**	**3.6×**
π-IMLE (H=30)	**5.4**	**0.10**	**10.5×**

Result. Longer horizon × faster forward pass → **10.5× less inference time.**

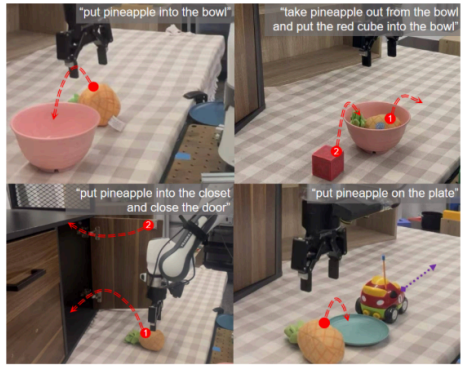
❻ LIBERO-plus: Robustness Under Distribution Shift

Robustness under distribution shift. Systematic test-time perturbations across **4 axes** (background, robot initial state, language, layout) × **5 severity levels (L1–L5)**.



π-IMLE retains $\pi_{0.5}$'s robustness across all axes and severities; baselines' success rates drop sharply as severity increases. OpenVLA-OFT's L1 regression collapses to the median mode under shift, while cIMLE enforces **mode coverage by construction**.

❼ Real-World Experiments — Franka Panda



Zero-shot from DROID with two camera views (wrist + scene). $\pi_{0.5}$ at 15 Hz vs. π-IMLE at 55 Hz — same VLM, different head.

Task	Type	π-IMLE ↑	$\pi_{0.5}$ ↑	WC × ↓
Pineapple in bowl	single	**10/10**	9/10	**6.6×**
Swap pineapple & cube	multi	**9/10**	8/10	**5.7×**
Pineapple in cabinet	multi	**8/10**	7/10	**3.9×**
Moving plate	reactive	**9/10**	6/10	**5.4×**

9 / 10 vs 6 / 10 on the **dynamic moving-plate task** — faster inference closes the perception–action loop, $\pi_{0.5}$ chases stale targets.