Robotics

Austin, Oliver, Jeffrey

Table of Contents

Robotics in 2025

Eric Jang - “all roads lead to robotics”

Classical to Robot Learning

Data as a fossil fuel

Democratized Hardware

Rapid industry interest

Robosuite (ARISE)

2017, SVL, now monitored by NVIDIA Gear, UT Austin PRL, and SVL.

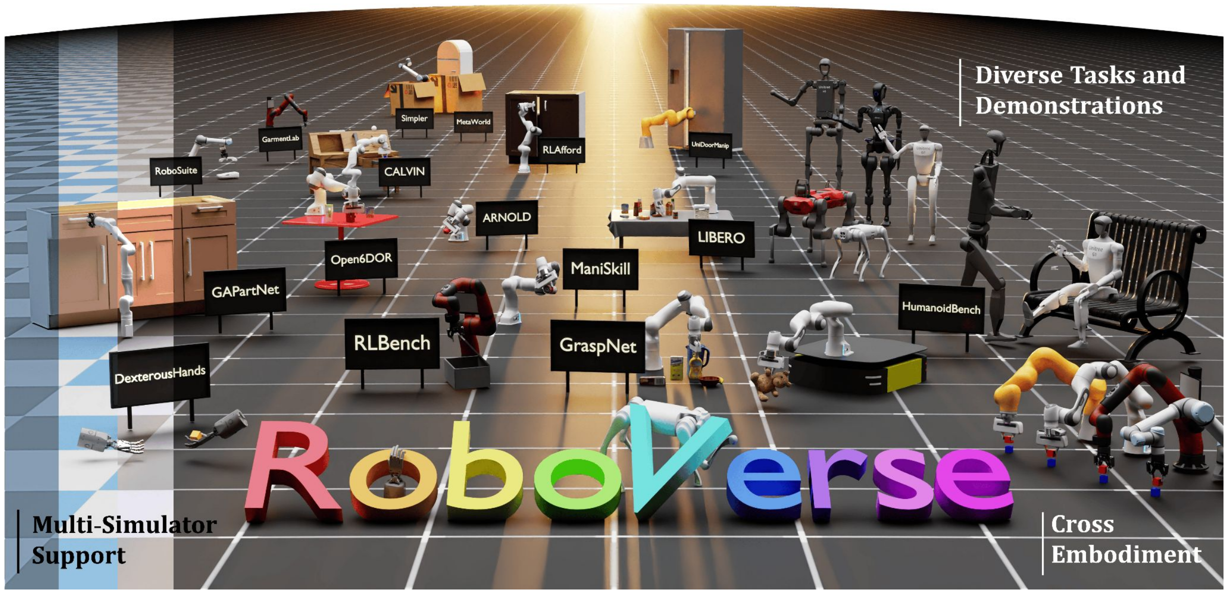
(GDM) MuJoCo-based simulator

Most popular choice in academia

Standardized benchmarks, policy training, imitation learning

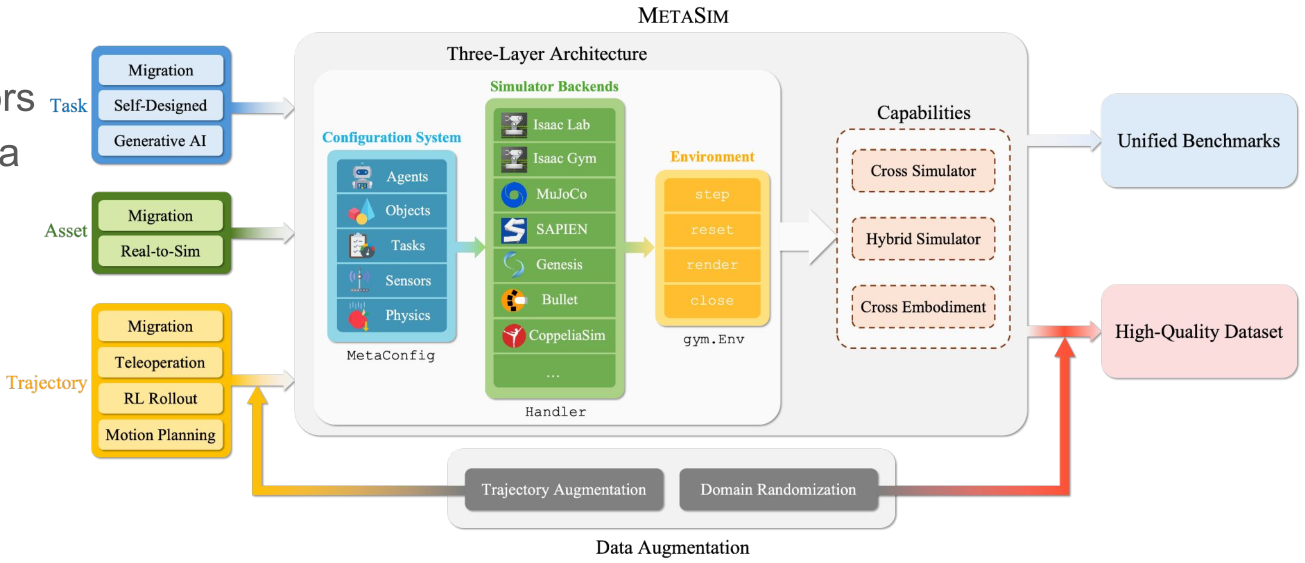
Roboverse (BAIR)

April 26, 2025 by BAIR and Peking et. al



RoboVerse - MetaSIM

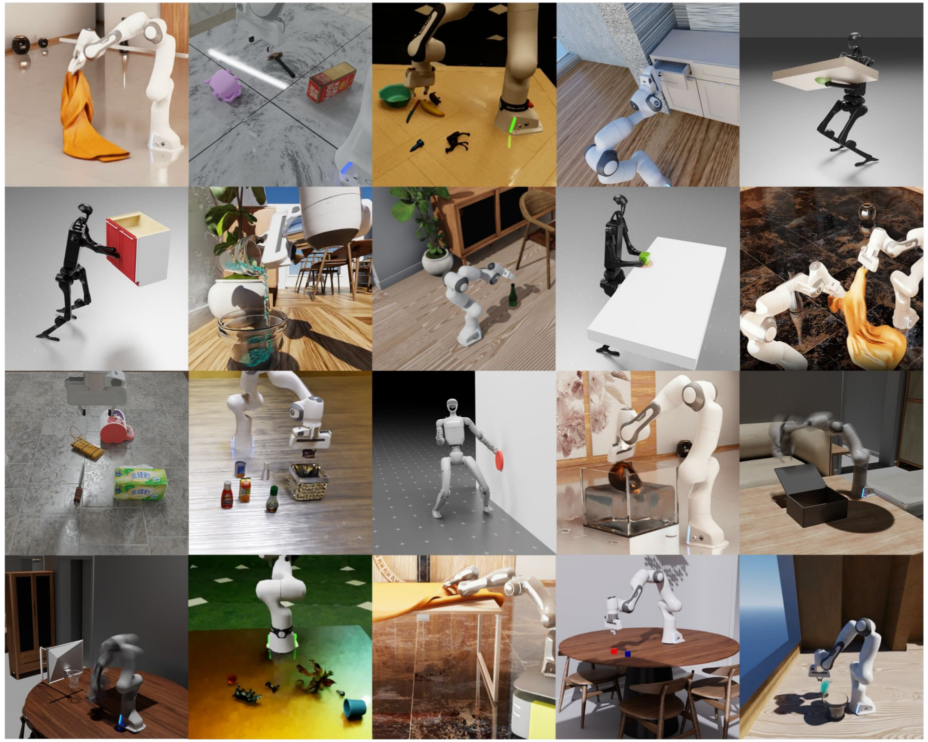
Unified simulation platform combining existing simulators and rendering engines into a single framework.



RoboVerse - Dataset

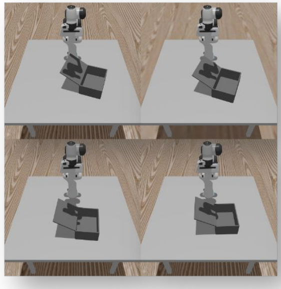
Large-scale, high-quality dataset.

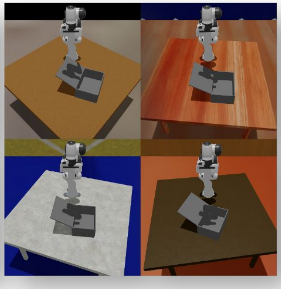
Upon training VLA, shown to generalized to unseen tasks and new environments.

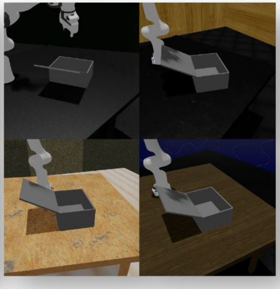


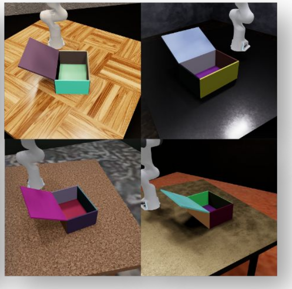
RoboVerse - Benchmark

Standardized benchmark for both imitation learning and reinforcement learning.


(a) Task Space


(b) Environment


(c) Camera


(d) Material & Lighting

LeRobot (Hugging Face)

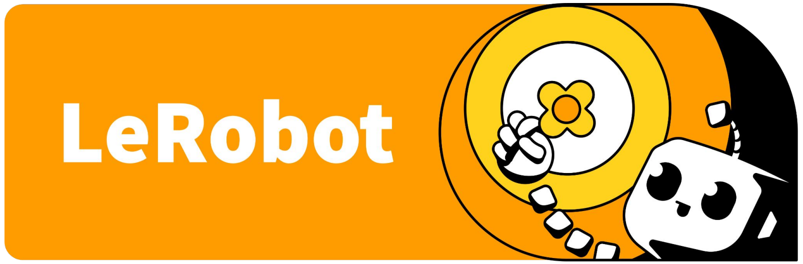
PyTorch based pipeline aimed to integrate the robotics stack into a single hub

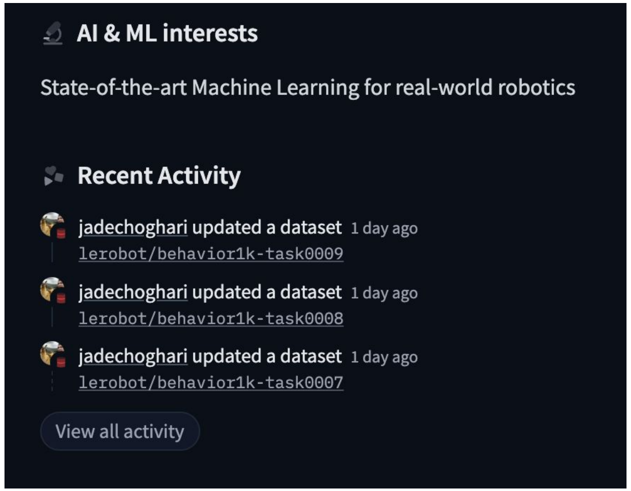
LeRobot Dataset

Very active open community via HF

Everything is pushed to a central hub

Open source for hardware as well




AI & ML interests

State-of-the-art Machine Learning for real-world robotics

Recent Activity

jadechoghari updated a dataset 1 day ago
lerobot/behavior1k-task0009

jadechoghari updated a dataset 1 day ago
lerobot/behavior1k-task0008

jadechoghari updated a dataset 1 day ago
lerobot/behavior1k-task0007

View all activity

Limitations and Use Case

	Robosuite	Roboverse	LeRobot
Simulator	MuJoCo	Meta-Sim (all)	All, but need to port yourself
Maturity	Stable	Developing	Developing Fast
Task Scope	Table-top manipulation	Broad manipulation across simulations	Real-world + whatever sims you attach
Benchmarks	Common baselines	Large synthetic dataset	Hub-native dataset
Policy types	IL / RL	IL / RL	IL / RL + VLA friendly

Hardware Bottlenecks

Settled on humanoid form factor

Problems:

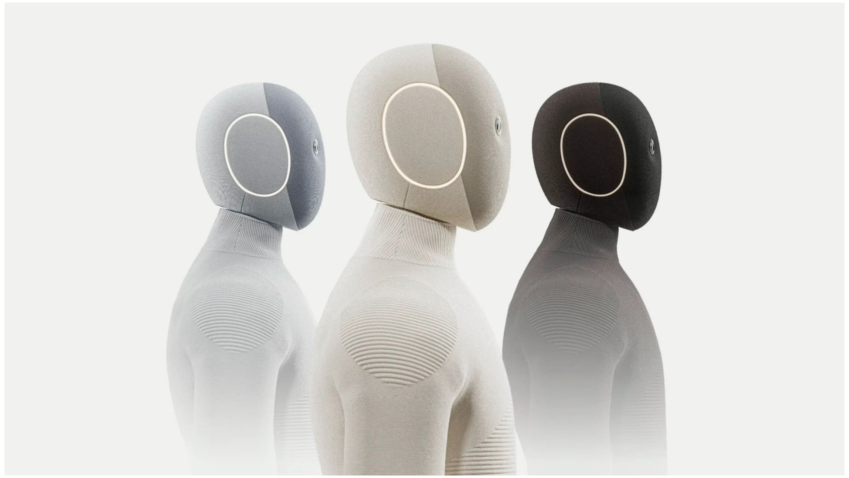
Dexterous Manipulation

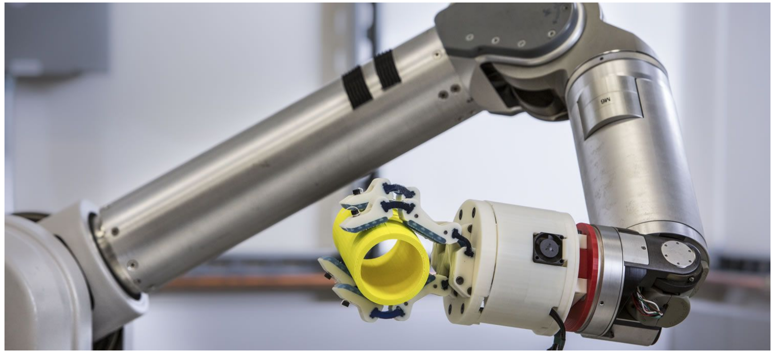
Battery Life and Thermal management

GPU and VRAM Limitations

Cost, but there's Unitree

Data Foundry





References

https://roboverseorg.github.io/

https://huggingface.co/spaces/lerobot/robot-learning-tutorial

https://github.com/huggingface/lerobot

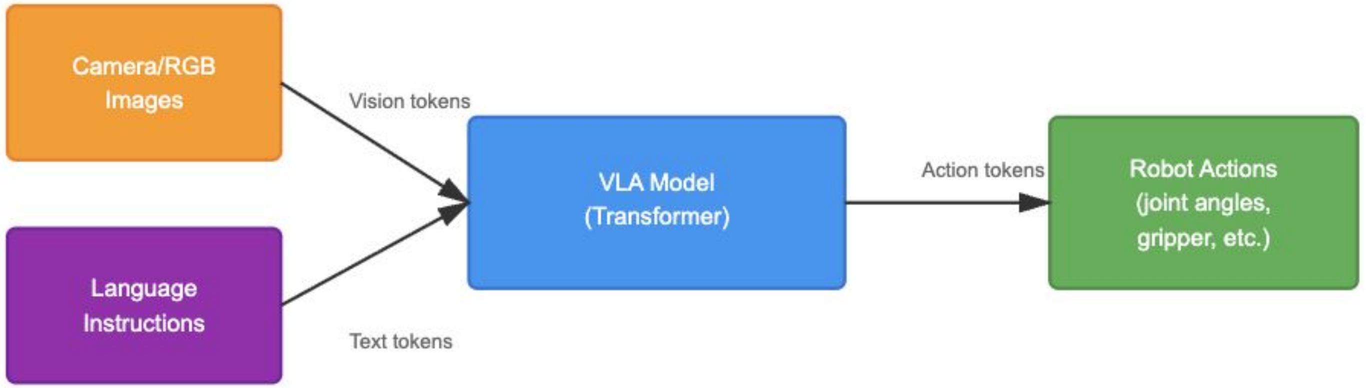
https://robosuite.ai/

https://evjang.com/2024/03/03/all-roads-robots.html

Generalist Policies

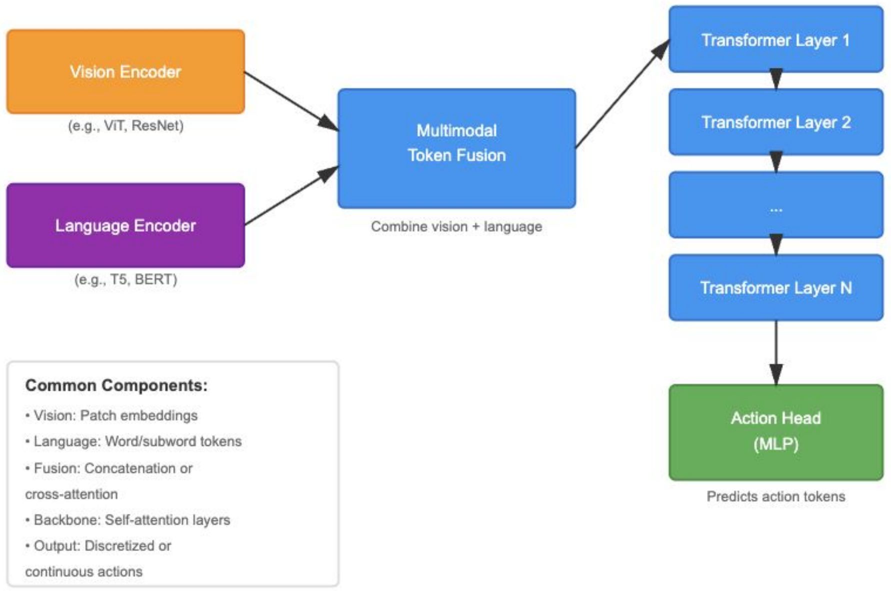
What is a VLA?

- Vision Language Action (VLA) Models build on VLMs by outputting action instead of text



VLA Architecture

- Combine image and text processing capabilities
- Example: Combine vision transformer and BERT tokenizer, fuse the tokens, pass to a transformer architecture.
- Transformer is trained to output contextualized representations
- MLP is trained to convert those into robot actions



Common Components:
- Vision: Patch embeddings
- Language: Word/subword tokens
- Fusion: Concatenation or cross-attention
- Backbone: Self-attention layers
- Output: Discretized or continuous actions

Examples

Popular VLA Models:

RT-2 (Robotic Transformer 2): Uses vision-language model (PaLI-X) backbone, outputs discretized actions

OpenVLA: Open-source VLA based on LLaVA architecture, 7B parameters

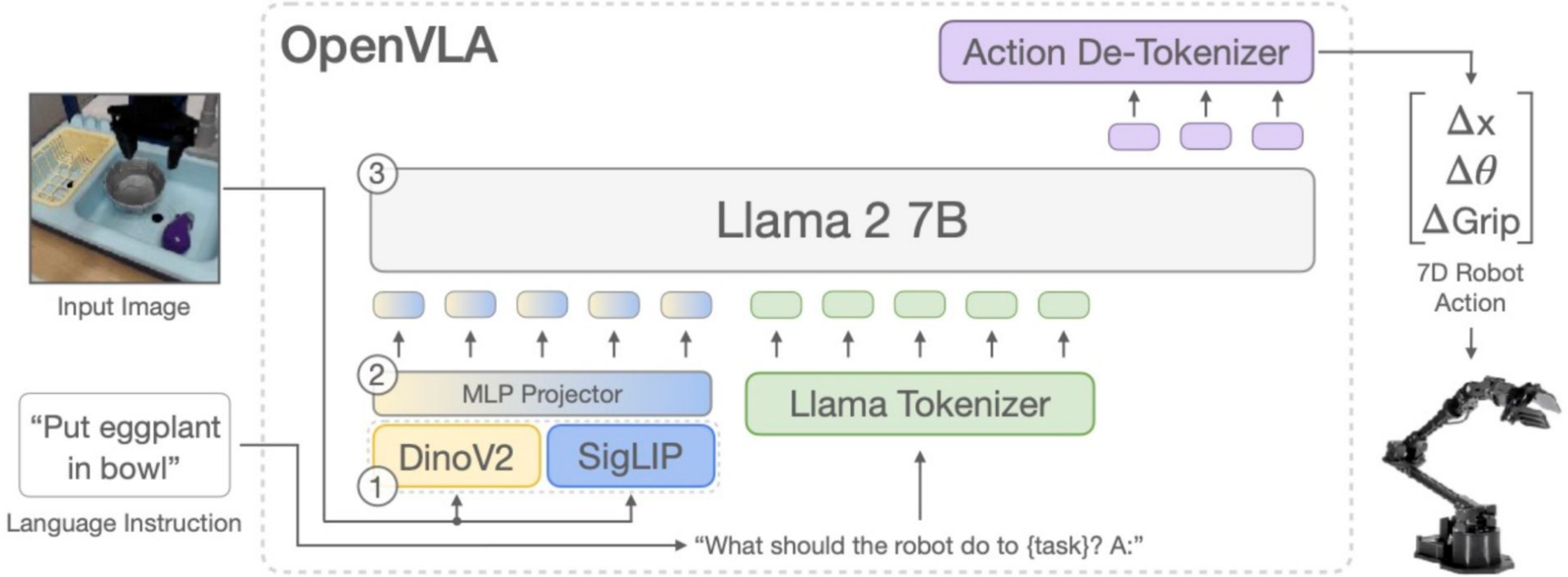
Octo: Generalist policy trained on 800k robot trajectories

PIVOT: Uses iterative planning with VLAs for complex tasks

OpenVLA

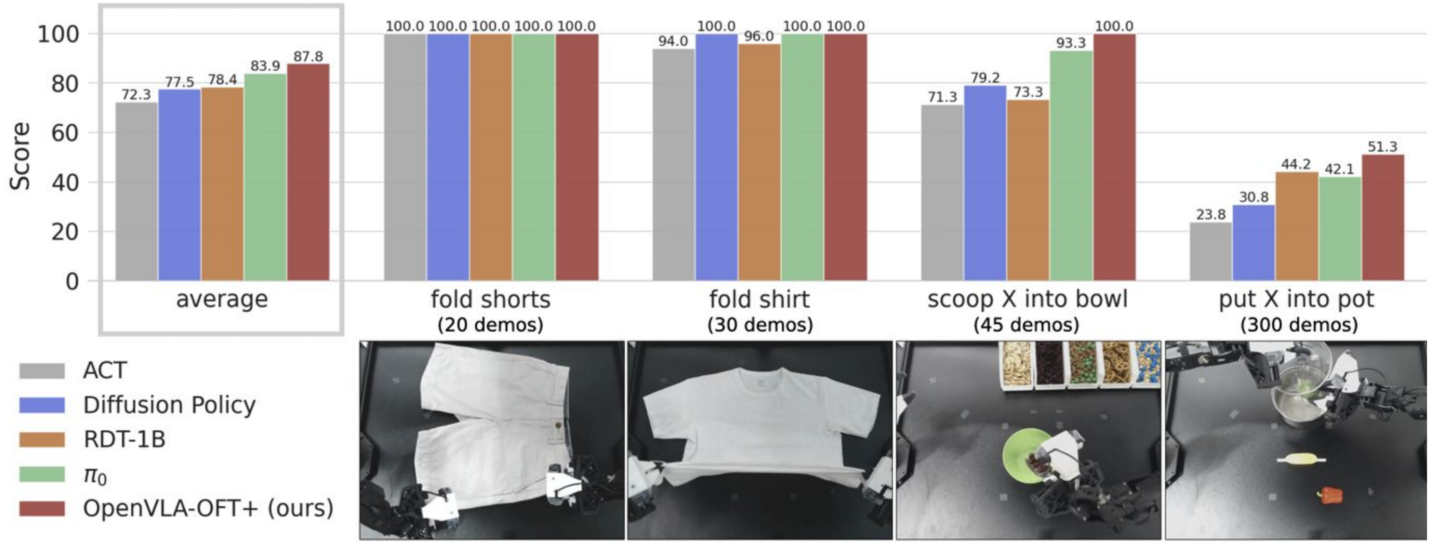
- First open-source VLA model.
- OpenVLA consists of four key components:
    - Fused visual encoder combining SigLIP and DINOv2 backbones that processes image inputs into patch embeddings
    - Built in projector maps these embeddings into the language model's input space
    - Llama-2-7B language model backbone that predicts tokenized output actions.
    - Discrete actions processed through softmax-based token prediction
- 256 discrete action tokens from the Llama vocabulary to represent robot control values with 8-bit resolution, covering a 7-DoF action space.

OpenVLA Architecture



OpenVLA-OFT

- Optimized for fine tuning
  - Parallelized instead of autoregressive decoding
  - Continuous outputs instead of discrete outputs
- Compares L1 Regression and Diffusion approaches
  - L1 is easier to calculate (requires less compute)
  - L1 converges faster
  - Diffusion can do really poorly when some training examples are incorrect, L1 does a better job smoothing out the noise



OpenVLA, a 7B-parameter model, outperforms the 55B-parameter RT-2-X model by 16.5% absolute success rate across 29 evaluation tasks on the WidowX and Google Robot embodiments

Fine-tuning on LIBERO

Checkpoints and guide: https://github.com/moojink/openvla-oft/blob/main/LIBERO.md

4 datasets (LIBERO-Spatial, LIBERO-Object, LIBERO-Goal, and LIBERO-10)

- Clone the OpenVLA and LIBERO repos.
- Run `./vla-scripts/finetune.py`
- Then run `./experiments/robot/libero/run_libero_eval.py`

The paper used 8 A100 or H100 GPUs with 80 GB memory. Takes about 1-2 days per dataset

Misha has fewer GPUs, so it’s estimated to take about 7-8 days per dataset.

Qualitative Examples

ACT: Scoop raisins into bowl

Diffusion: Scoop pretzels into bowl

Qualitative Examples

After OFT training on OpenVLA

- L1 Regression is less noisy than diffusion, allows for more errors in training data

Autoregressive vs Diffusion

Aspect	**Diffusion**	**Autoregressive**
Examples	RDT-1B (1.2B), Diffusion Policy, π0	RT-2, OpenVLA, π0-FAST
Prediction Method	Full trajectories via denoising	Token-by-token or action chunks
Action Distribution	Multimodal (multiple solutions)	Unimodal (single solution)
Inference Speed	Slow (10-100 steps, ~100-200ms)	Fast (single pass, < 10ms)
Best For	Dexterous tasks and contact rich control	Language-based taste, fast deployment, general-purpose
Use when	Data-constrained	Compute-constrained

References

https://openvla-oft.github.io/

https://blog.ml.cmu.edu/2025/09/22/diffusion-beats-autoregressive-in-data-constrained-settings/

https://github.com/moojink/openvla-oft/blob/main/LIBERO.md

Pi0

- Target generalist robotics policy
  - Data from many robot types is combined into the same model
- Key Components:
  - VLM backbone: PaliGemma (SigLIP + Gemma) + robotics inputs
  - Flow-matching action head allows for high frequency, continuous action
  - Two “experts”: robotics vs vision + language
- Emphasis on multimodal data labeling on whole trajectories and sub-trajectories for modality alignment

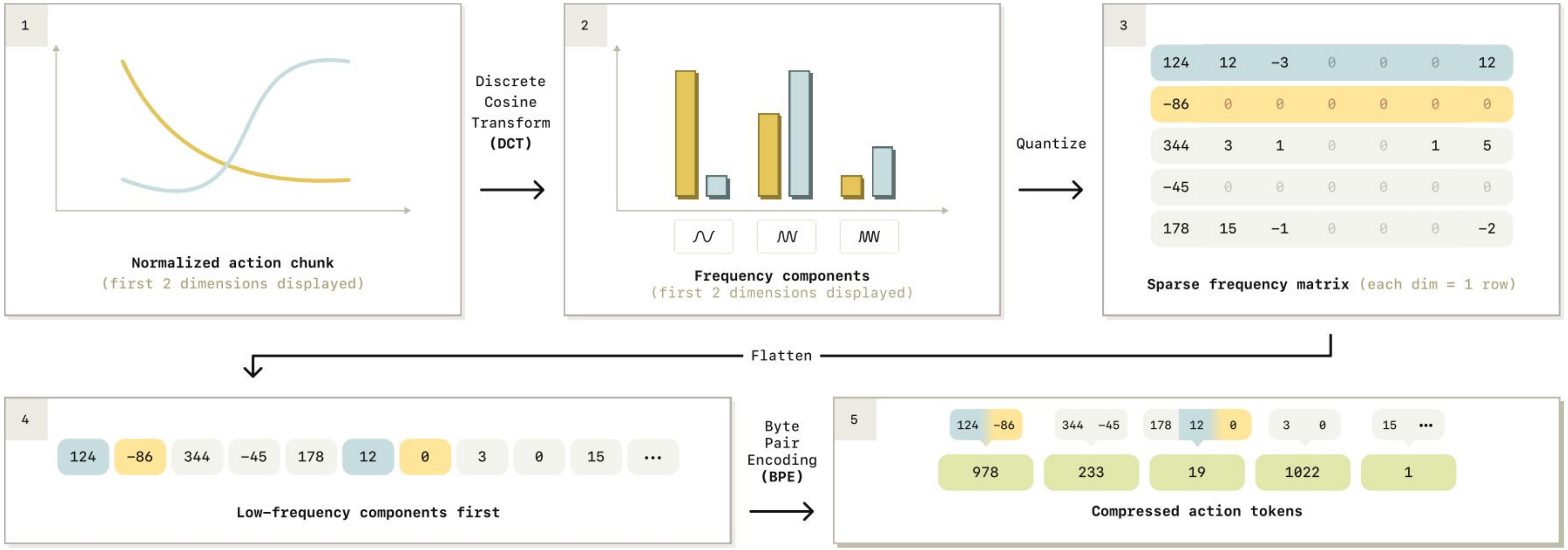
Pi0-FAST

- Diffusion / Flow matching training is very slow compared to autoregressive
- Autoregressive VLAs output discrete tokens that map to continuous actions
- Hard to capture high frequency behaviors
- Tackle with Discrete Cosine Transform (DCT) signal compression method
    - Variant of discrete Fourier Transform but only real (cosine) components, leading to better representation
- Leads to 5x faster training

$$C(u) = a(u) \sum_{x=0}^{N-1} f(x) \cos\left[\frac{(2x+1)u\pi}{2N} \right]$$

$$u = 0, 1, \ldots, N-1$$

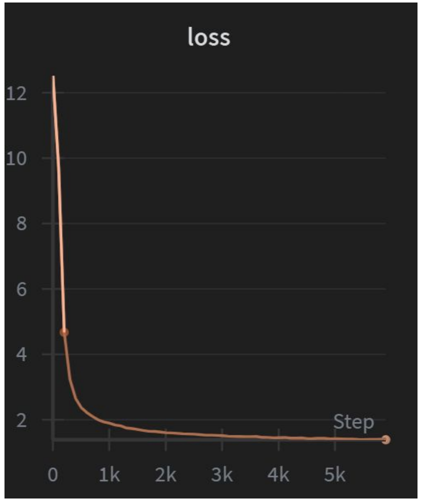
$$a(u) = \begin{cases} \sqrt{\frac{1}{N}} & u = 0 \\ \sqrt{\frac{2}{N}} & u = 1, \ldots, N-1 \end{cases}$$



1
Normalized action chunk
(first 2 dimensions displayed)
Discrete Cosine Transform (DCT)
2
Frequency components
(first 2 dimensions displayed)
Quantize
3
124 12 -3 0 0 0 12
-86 0 0 0 0 0 0
344 3 1 0 0 1 5
-45 0 0 0 0 0 0
178 15 -1 0 0 0 -2
Sparse frequency matrix (each dim = 1 row)
Flatten
4
124 -86 344 -45 178 12 0 3 0 15 ...
Low-frequency components first
Byte Pair Encoding (BPE)
5
124 -86
344 -45
178 12 0
3 0
15 ...
978
233
19
1022
1
Compressed action tokens

Fine-tuning Pi0

- Physical Intelligence contains many checkpoints for Pi0, Pi0-Fast, Pi0.5 and finetuned versions on LIBERO, ALOHA, DROID, etc.
- Finetuning is very simple: convert dataset to LeRobot and create a finetuning config



Continuing from Pi0

- After Pi0, VLA robotic foundation models have built on their approach
- Nvidia GR00T N1
  - Diffusion Transformer Action Head
  - Auxiliary object detection loss to align vision modality
  - Action labels for human and synthetic video data generated by VQVAE
- Gemini Robotics
  - Similar to Pi0, VLA model with generalist capabilities (Zero-shot Transfer)
  - Uses a reasoning VLM to orchestrate and break down complex tasks

References

https://arxiv.org/pdf/2410.24164v1

https://arxiv.org/pdf/2501.09747

https://tuul.ai/research/groot-n1-robotic-foundation-model