上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

Spatial VLA

Exploring Spatial Representations for Visual-Language-Action Model

TeleAI Paper Discussion

2025-03-03

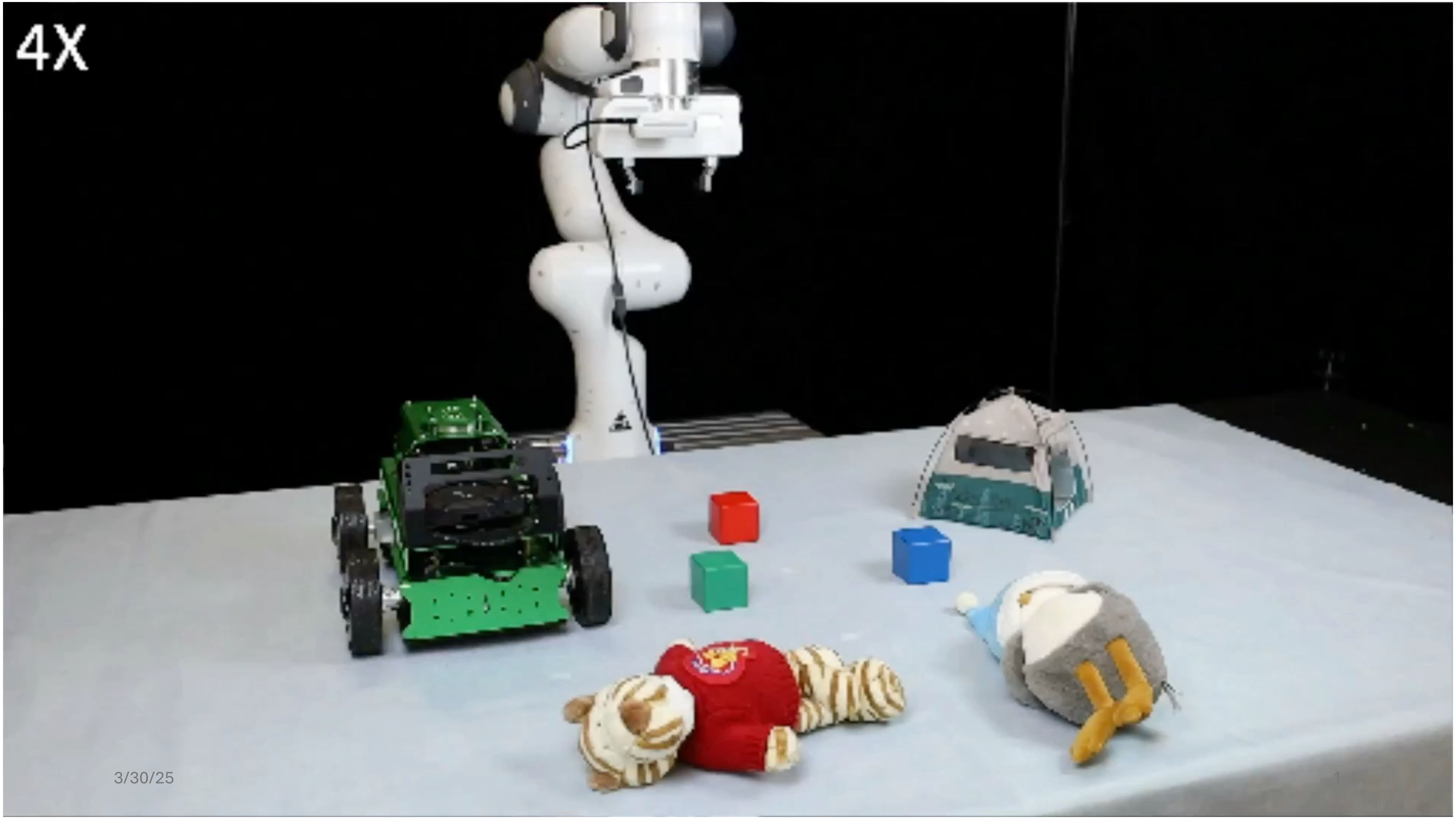
https://spatialvla.github.io


Qu Delin (Speaker)
Fudan University


Song Haoming
Shanghai Jiao Tong University

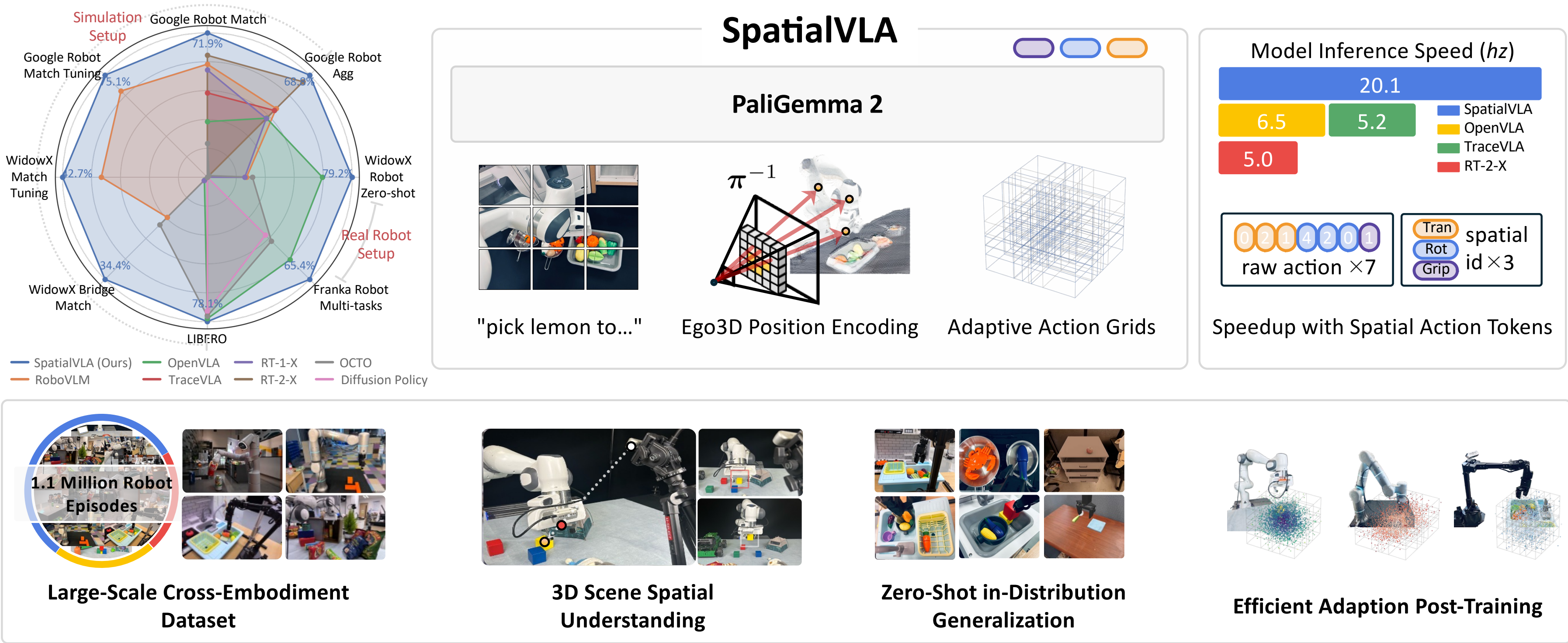

Chen Qizhi
Zhejiang University


Wang Dong
Shanghai AI Laboratory


4X

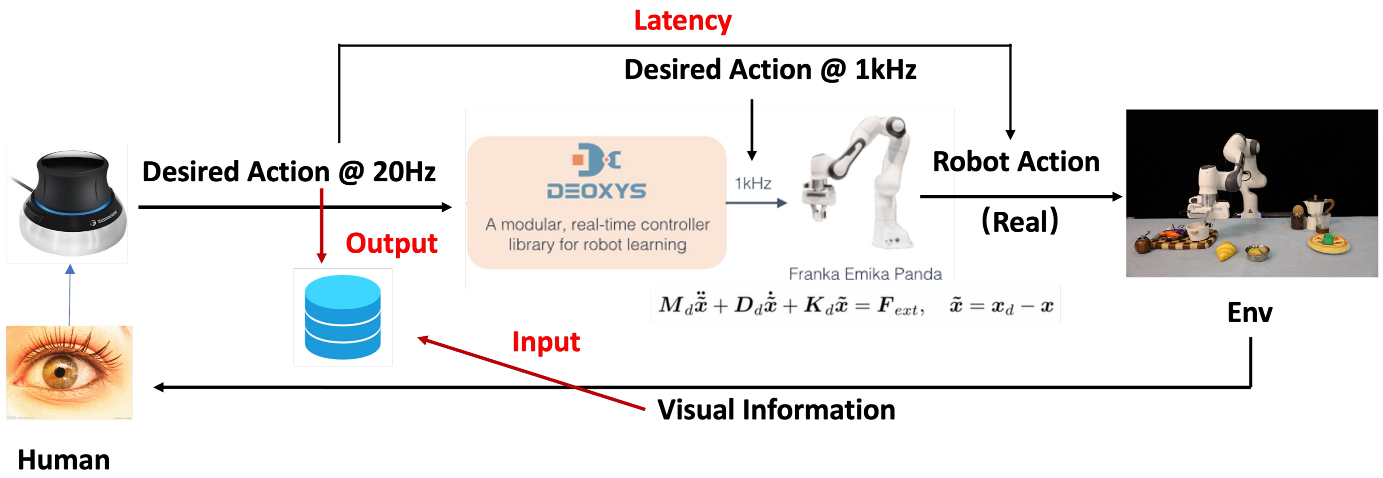
SpatialVLA

A spatial-enhanced vision-language-action model trained on 1.1 million real robot episodes



Problem Formulation

- ❑ Input
  - ❑ Language Instruction: L
  - ❑ Observations
    - ❑ RGB Image: $I_{t-\Delta}$, I_t
    - ❑ Robot Proprio: S_t
    - ❑ Depth: Z_t
- ❑ Output:
  - ❑ Action: a_t

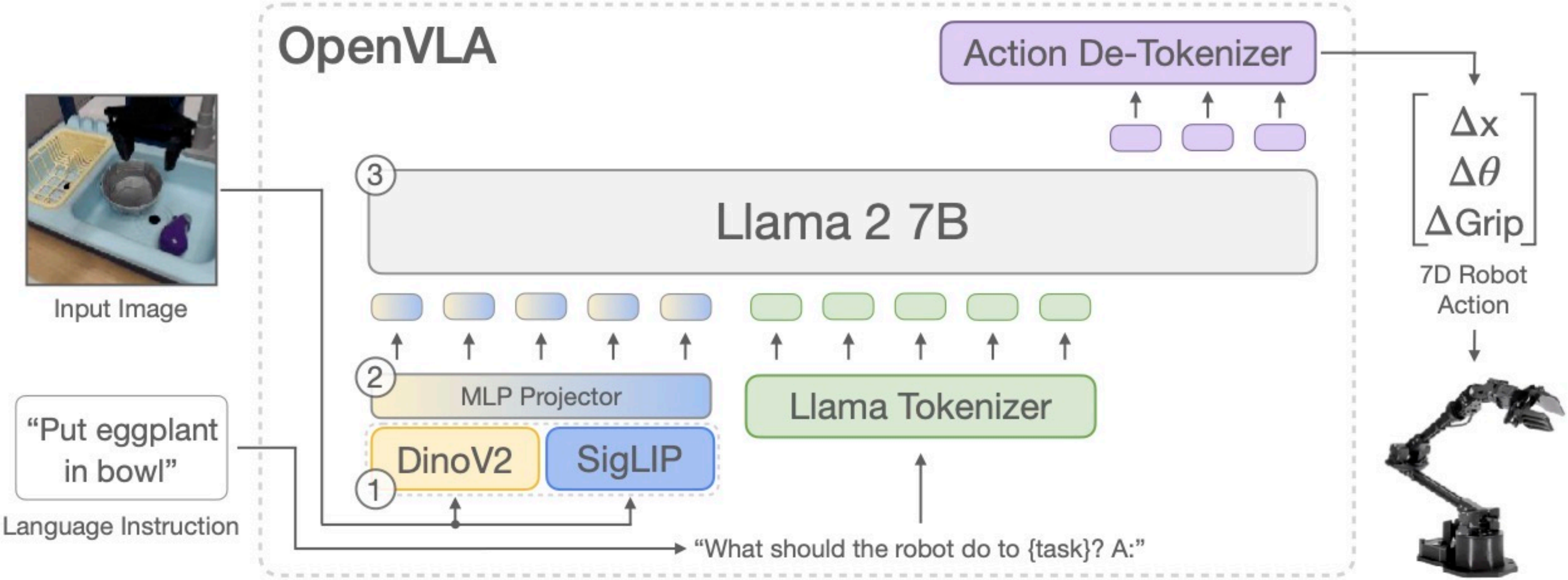


Delta eef pose 7d

State -> 14+1

Generalist Robot Policies

3D physical intelligence cross multiple robot environments and tasks



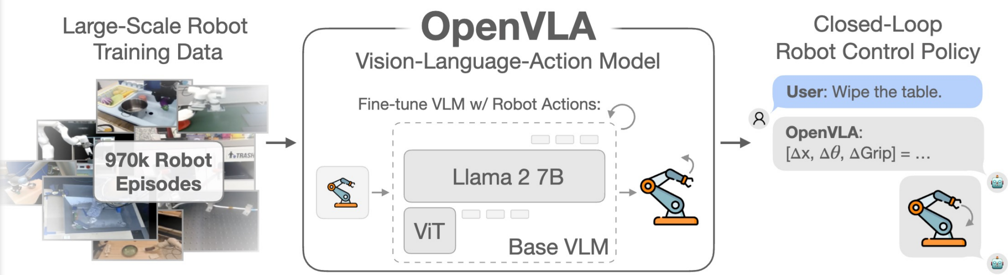
language into action[1]

Language-Action Model[2]

General Robot Control[3]

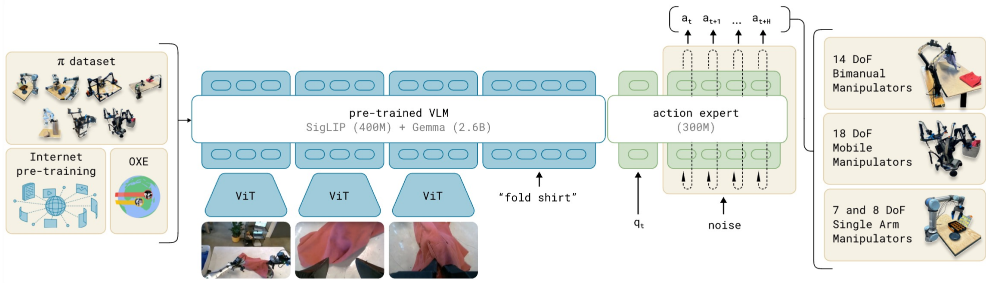
Foundation Vision-Language-Action Model Paradise

❑ VLM Auto-Regression[1][2]



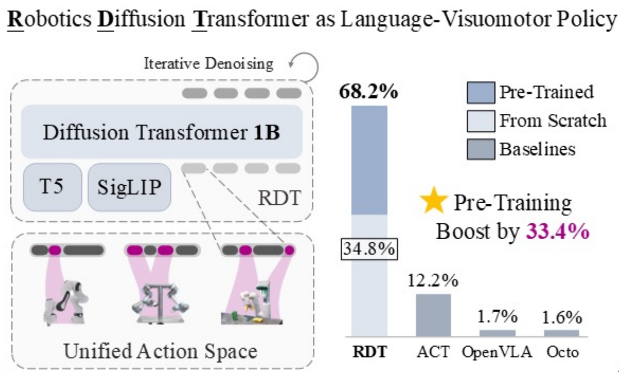
Large scaling training, instruction flowing

❑ VLM Denoise[3][4]



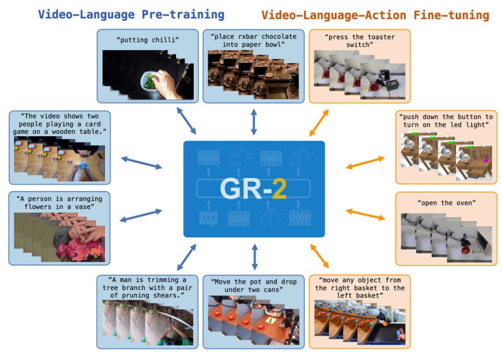
Fine gained controlling, high frequency inference

❑ Scaling Transformers / Diffusion [5][6]



Fine gained Controlling

❑ World Model[7][8] (VPP + inverse dynamic model)



Benefiting from **world model**, higher potential

[1] OpenVLA: An Open-Source Vision-Language-Action Model

[2] FAST: Efficient Action Tokenization for Vision-Language-Action Models

[3] π0: A Vision-Language-Action Flow Model for General Robot Control

[4] Octo: An Open-Source Generalist Robot Policy

[5] Diffusion Policy Visuomotor Policy Learning via Action Diffusion

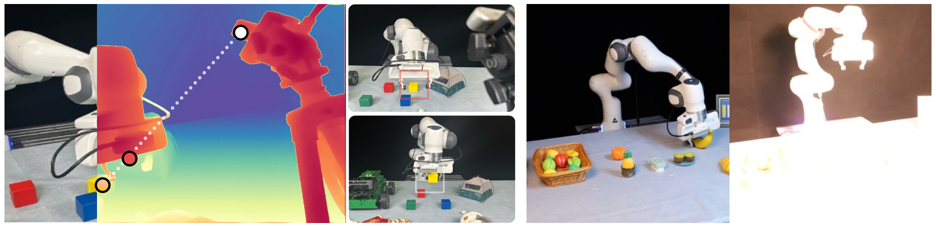
[6] RDT-1B: a Diffusion Foundation Model for Bimanual Manipulation

[7] GR-2: A Generative Video-Language-Action Model with Web-Scale Knowledge for Robot Manipulation

[8] Video Prediction Policy: A Generalist Robot Policy with Predictive Visual

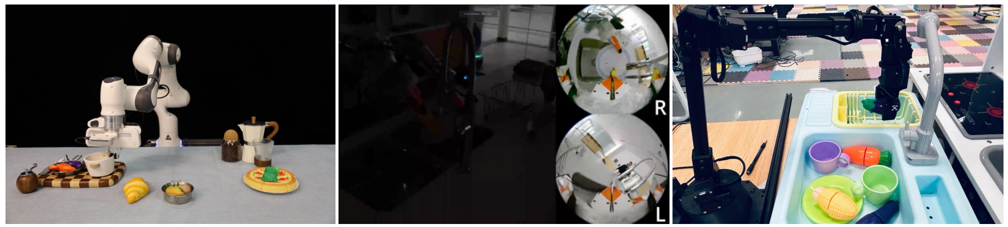
How to effectively equip VLA models with a profound spatial understanding of the 3D physical world?

- ❑ Visual appearance variation



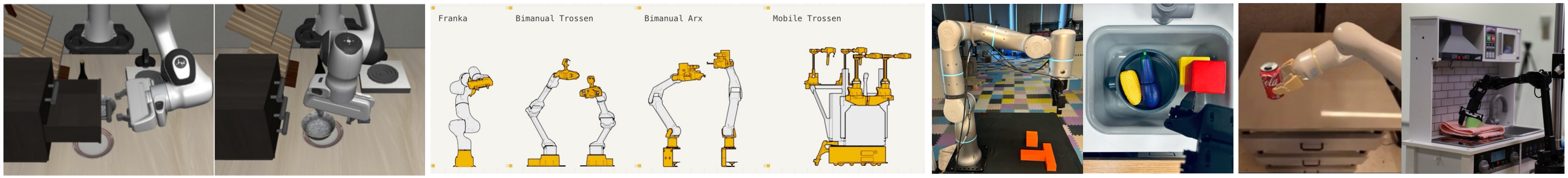
diverse scenarios, e.g., single-view cameras, varied lighting

- ❑ Robot observations are not 3D-aligned



wrist or third-person

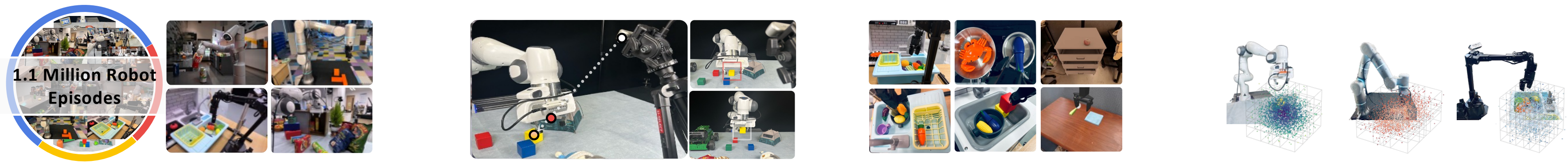
- ❑ Heterogeneous Action Movement Characteristics



different degrees of freedom | diverse motion controllers | workspace configuration | task complexity

Goals of Spatial VLA

- ❑ Foundation VLA model with a profound **spatial understanding of 3D physical world**
  - ❑ Robust and efficient **across scene variation**, e.g., visual appearance, object layouts
  - ❑ Purely Hugging Face-based, **concise code** with efficient performance 🤗
  - ❑ Achieves **SOTA performance** across a diverse range of evaluations

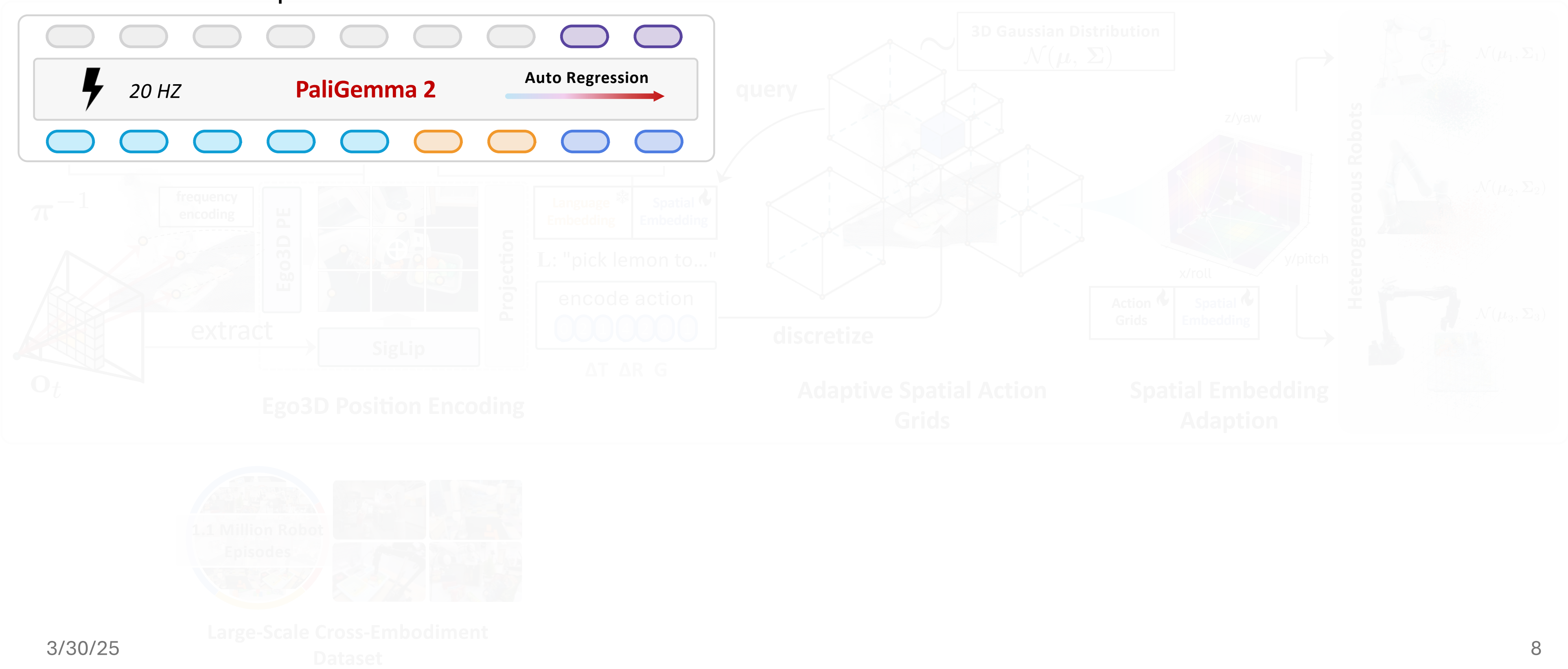


Large-Scale Cross-Embodiment Dataset | **3D Scene Spatial Understanding** | **Zero-Shot in-Distribution Generalization** | **Efficient Adaption Post-Training**

Our Solution

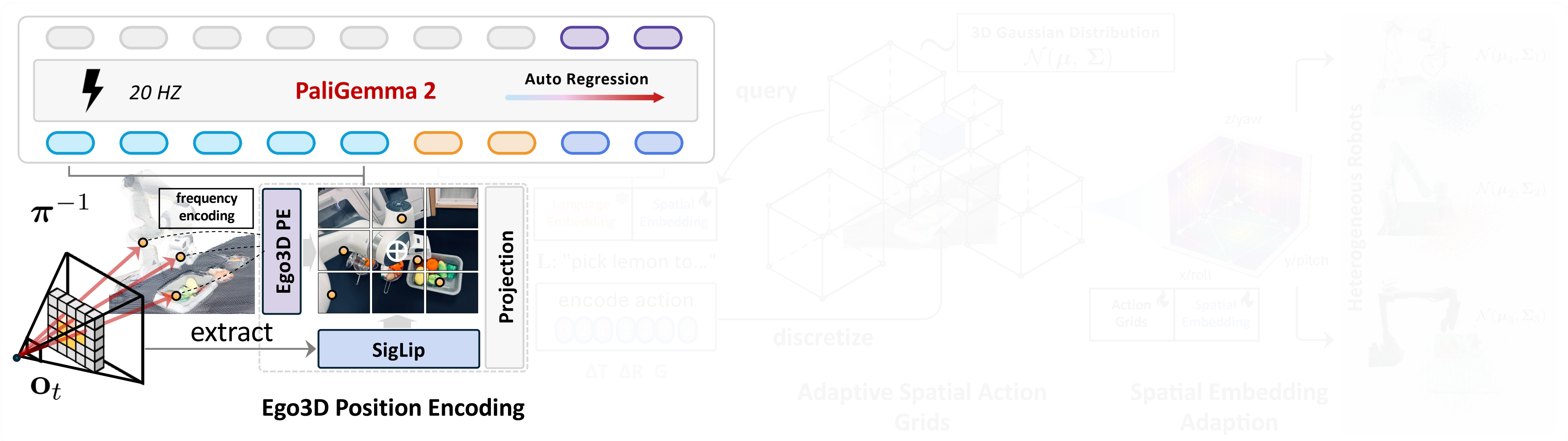
Spatial-aligned robot observation and action representations in universal 3D world

- LLM: Gemma2 3b
- Vision: SIGLIP patch16-224 203M



Our Solution

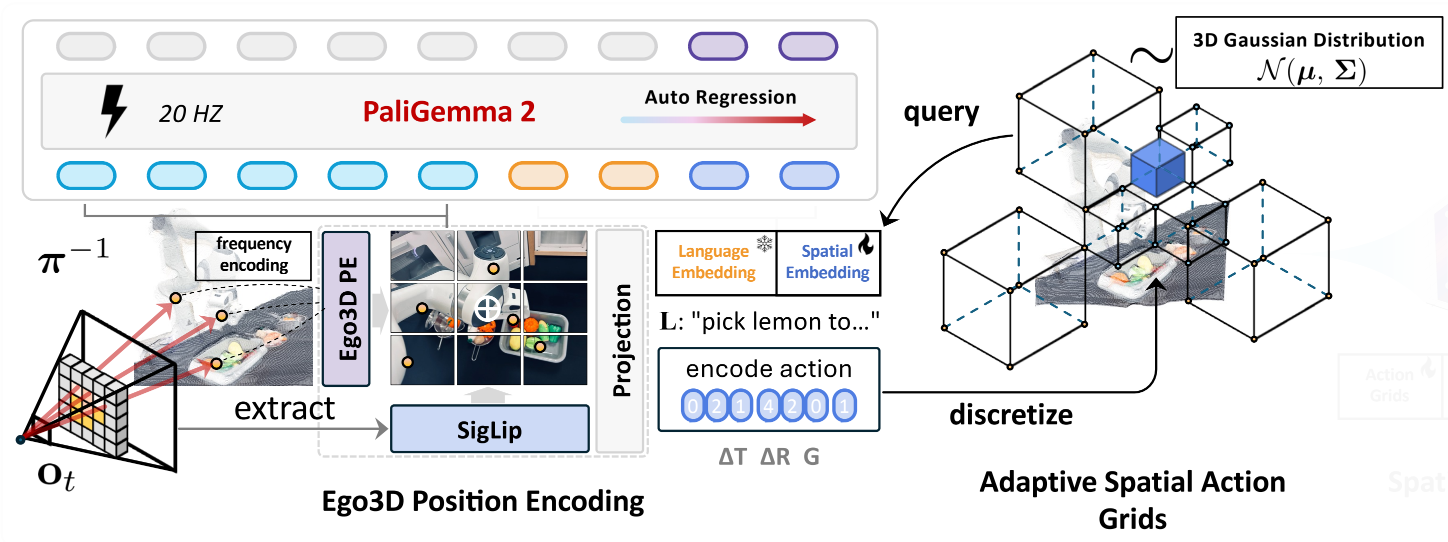
Spatial-aligned robot observation and action representations in universal 3D world



- Zoe-Depth 344M

Our Solution

Spatial-aligned robot observation and action representations in universal 3D world

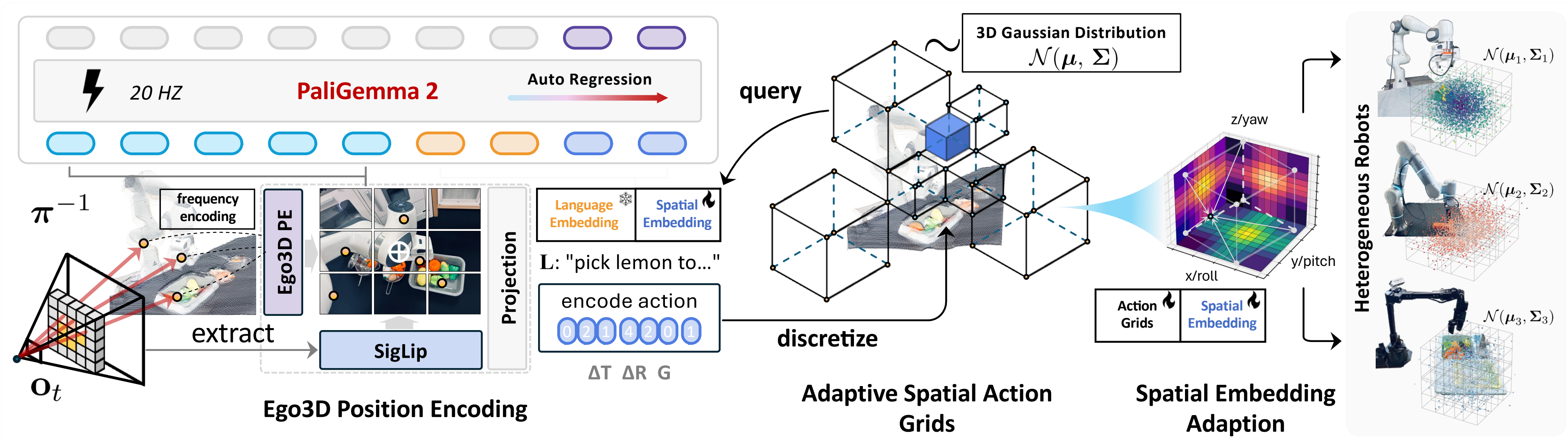


- 8196 Resolution
- Gaussian Distribution
- 18.9 M

<IMAGE> <IMAGE> <IMAGE> <IMAGE> ... <bos> What action should the robot take to pick the cup? \n
<ACTION00880> <ACTION05511> <ACTION08193> × 4
<eos>

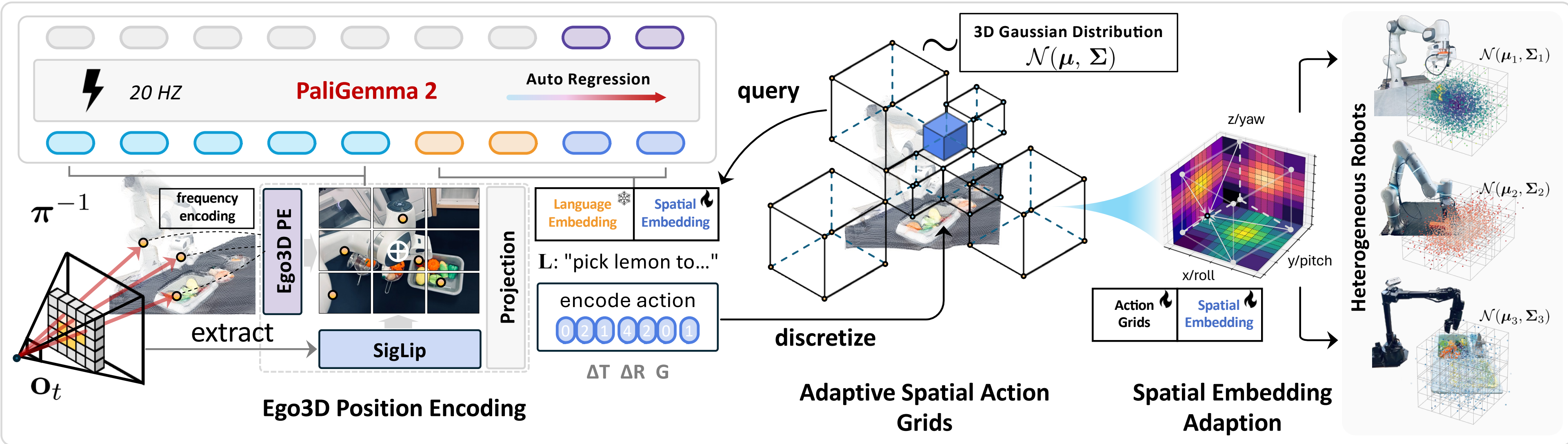
Our Solution

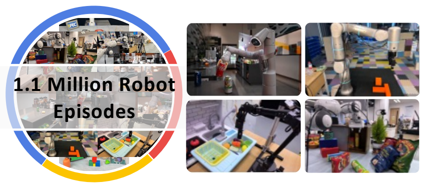
Spatial-aligned robot observation and action representations in universal 3D world



Our Solution

Spatial-aligned robot observation and action representations in universal 3D world





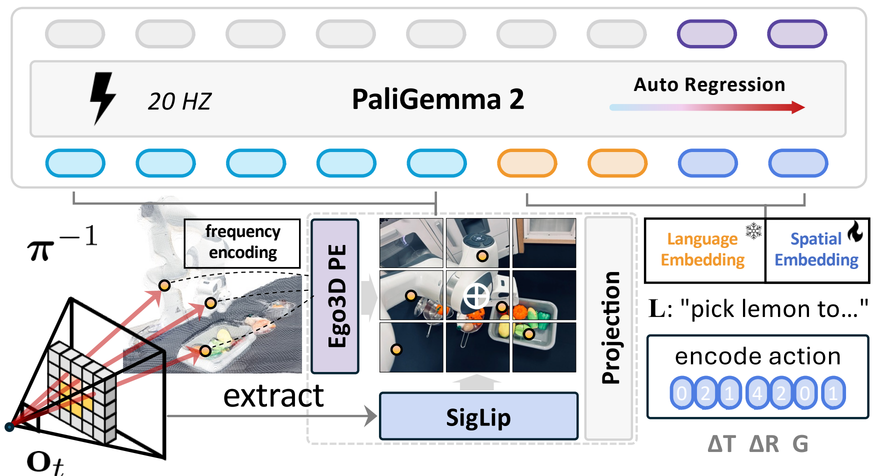
Large-Scale Cross-Embodiment Dataset

- 1.1 Million Robot Episodes
- 110 Million Samples
- 64 gpus * 10 days

<IMAGE> <IMAGE> <IMAGE> <IMAGE> <bos> What action should the robot take to pick the cup? \n
<ACTION00880> <ACTION05511> <ACTION08193> × 4
<eos>

Egocentric 3D Position Encoding

integrate 3D spatial context with semantic features



Ego3D Position Encoding

- ❑ How to obtain the depth observation and Camera Intrinsic?
- ❑ Why we use feature add instead of concatenate?
- ❑ Is accurate depth necessary for manipulation?
- ❑ How to maintain the stability of CLIP features?
- ❑ DINOv2 vs SigLIP?

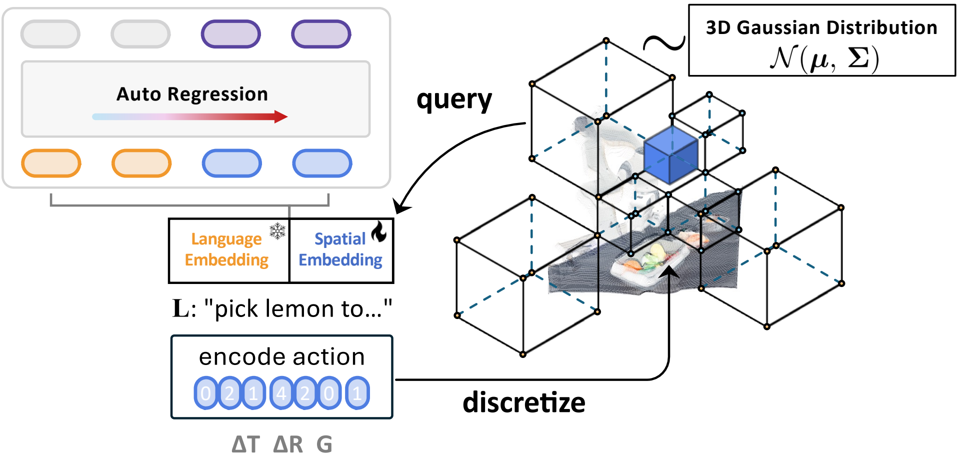
~~LLama3.2 3b + DINO v2~~

$$\mathbf{O_{3d}} = \mathbf{X} + \mathbf{P}' = \mathbf{X} + \mathrm{MLP}(\gamma(\mathbf{P})).$$

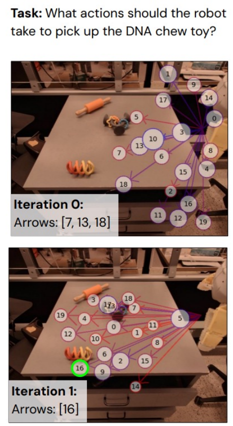
https://huggingface.co/docs/transformers/model_doc/zoedepth

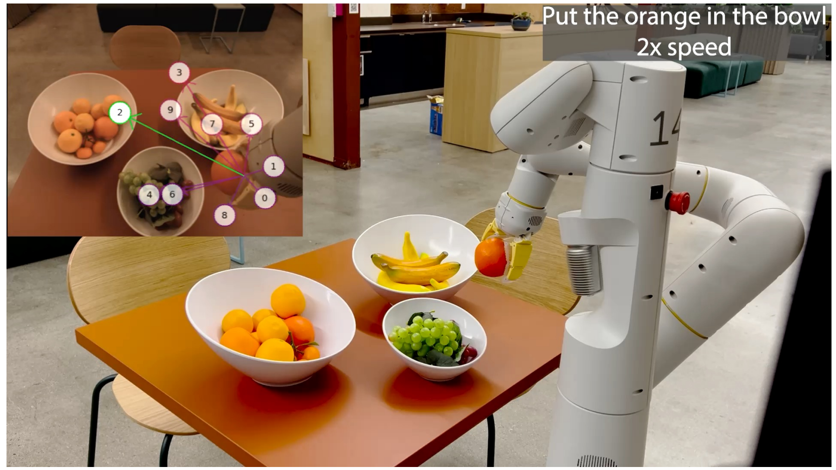
Adaptive Action Grids

Encode robot actions into adaptive spatial action grids and with the 3D physical world



Adaptive Spatial Action Grids

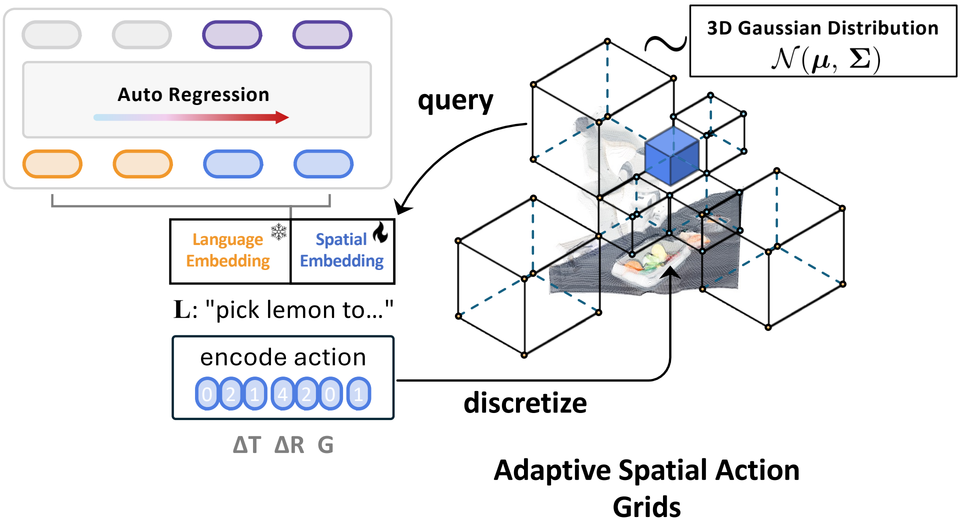




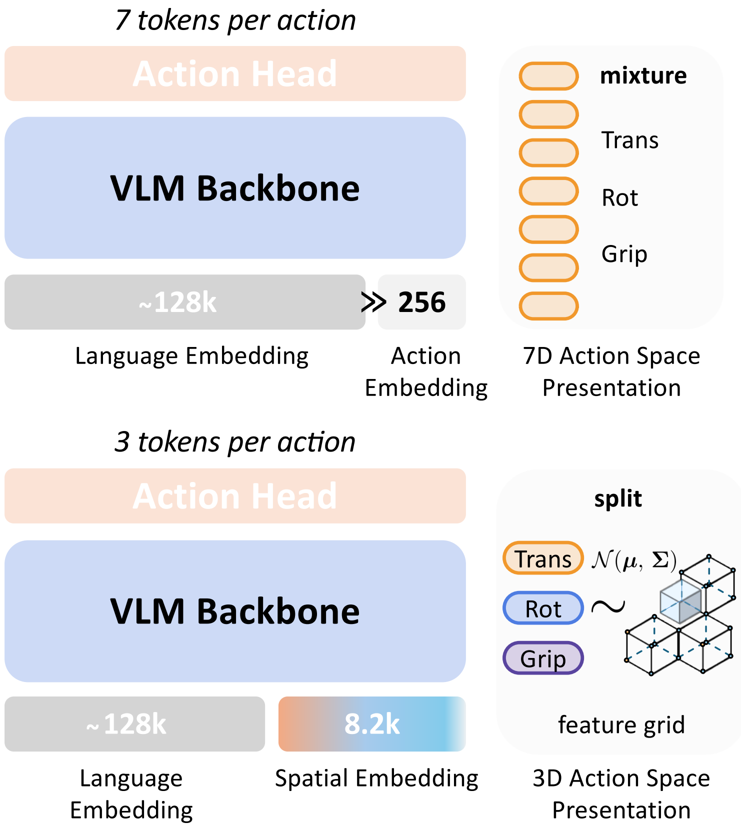
PIVOT: Iterative Visual Prompting Elicits Actionable Knowledge for VLMs

Adaptive Action Grids

Encode robot actions into adaptive spatial action grids and with the 3D physical world



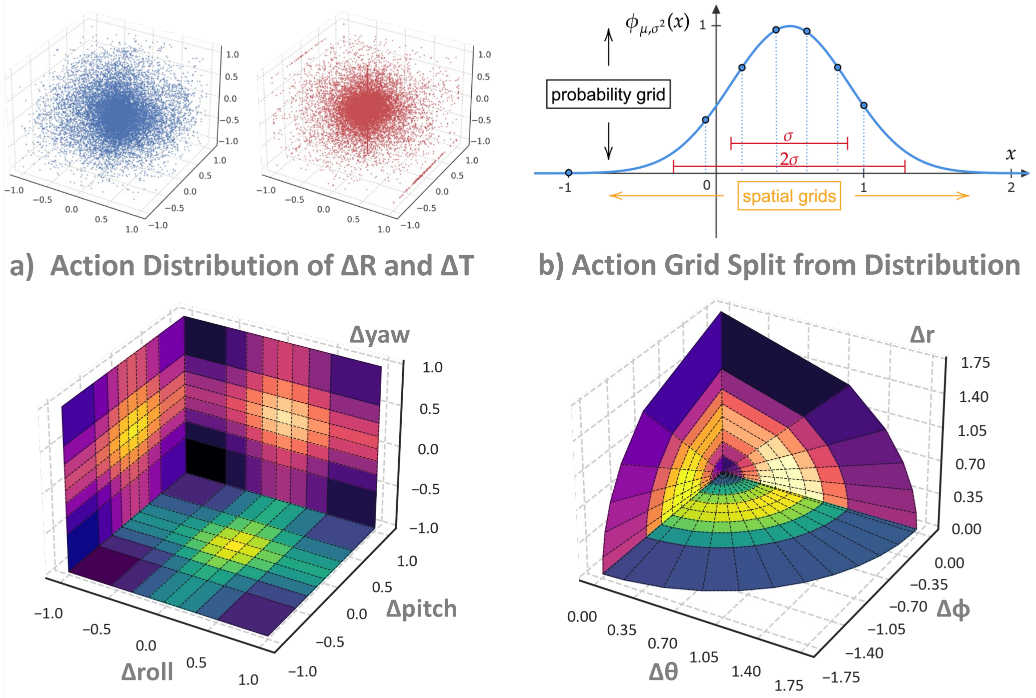
<IMAGE> <IMAGE> <IMAGE> <IMAGE> <bos> What action should the robot take to pick the cup? \n
<ACTION00880> <ACTION05511> <ACTION08193> × 4
<eos>



Examples

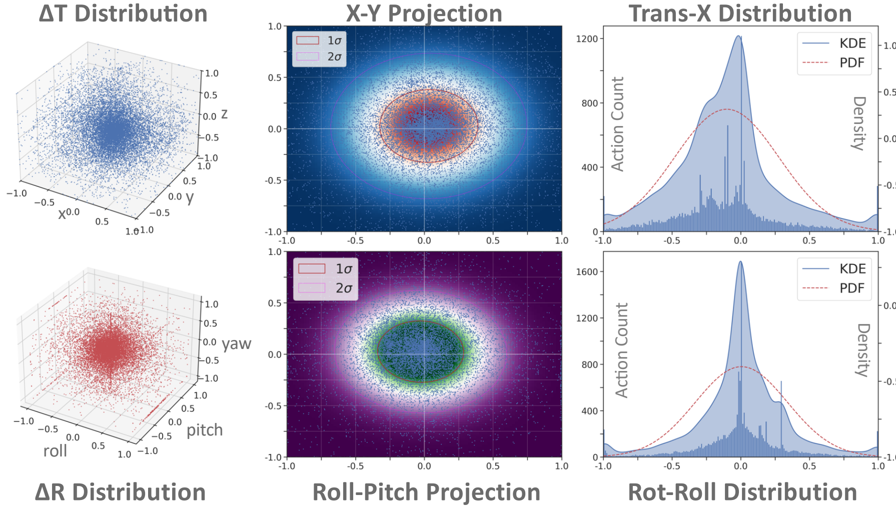
- Robot action space: 7 dimensions
  - 6-DoF delta end-effector pose: Δpos_x, Δpos_y, Δpos_z, Δrot_x, Δrot_y, Δrot_z
  - 1-DoF gripper control: Δgripper (binary: 0 = close, 1 = open)
- Each dimension is scaled to [-1, +1], then discretized into 8196 Spatial Grids
  - Δpos_x → -1 [1 | 2 | … | 254 | 255] +1
  - Δpos_y → -1 [1 | 2 | … | 254 | 255] +1
- …
- Therefore, each action $\hat{a}_t$ can be represented by a string of 3 tokens
- Example:
  - Raw action: [0.00 0.03 -0.82 0.00 -0.14 0.57 1.00]
  - Tokenized: <ACTION00880> <ACTION05511> <ACTION08193>

Adaptive Action Grids from Gaussian Distribution



a) Action Distribution of ΔR and ΔT
b) Action Grid Split from Distribution
c) Action Grids of ΔR and ΔT

- **Pre-training**: Construct adaptive action grids from dataset Gaussian distribution



$$a_2, \dots, a_{\mathrm{M}} = \arg\min_{a_2,\dots,a_{\mathrm{M}}} \int_{a_i}^{a_{i+1}} f(x)dx - 1/\mathrm{M},\ i = 1, \dots, \mathrm{M}$$

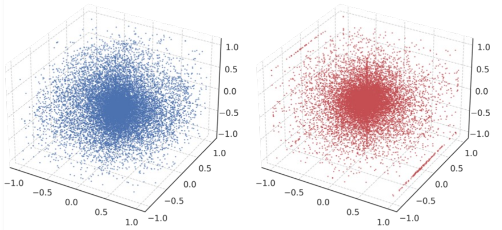
- How many Grids do we use during training? Does the resolution matter?
- Why we use the polar coordinate for translation?

Pseudocode for SpatialVLA action encoding and decoding

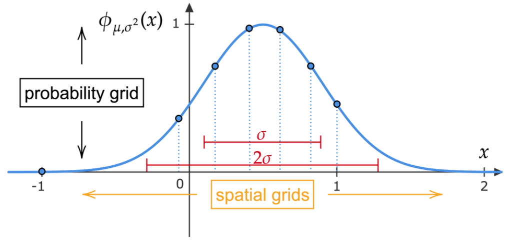
Algorithm 1 Python pseudocode for SpatialVLA action encoding and decoding.

```
# N: Number of Grid Intervals (e.g., 8)
# R: Range for Grids (e.g., [0, pi])
# P: Probability of Each Grids
# G: Adaptive Action Grids with Embedding of Size E
# cdf: Cumulative Distribution Function
# ppf: Percent Point Function

# create adptive action grids from gaussian distributions
for gaussians, grid_params in GS(theta, phi, r, roll, pitch, yaw):
   for (mu, sigma), (R, N)  in gaussians, grid_params:
         P = linspace(cdf(R, mu, sigma), cdf(R, mu, sigma), N + 1)
         G.x = ppf(P, mu, sigma) # coordinates
         G.fea = Embedding(N, E) # features

G.add_gripper() # add gripper 2 grids
# linearization 3d grids to share parameters with llm embedding
# trans: [N_theta * N_phi * N_r], rot: [N_roll * N_pitch * N_yaw]
# gripper: [N_gripper]
G.linearization()

# T: Number of Timesteps
# D: Dataset
for t in range(0, T):
   # if encode
   a = D(t) # normalized action [theta, phi, r, roll, pitch, yaw, gripper]

   # digitize continuous actions to 3d grids
   d_theta, d_phi, d_r = digitize(G, theta, phi, r) # trans
   d_roll, d_pitch, d_yaw = digitize(G, roll, pitch, yaw) # rot

   # linearization
   id_trans = linearize(d_theta, d_phi, d_r)
   id_rot = linearize(d_roll, d_pitch, d_yaw)
   id_gripper = 1 if gripper > 0.5 else 0 # gripper
   token_trans, token_rot, token_gripper = G.fea(id_trans, id_rot, id_gripper)

   # if decode
   (id_trans, id_rot, id_gripper) = SpatialVLA([image], prompt) # predict 3 action token id
   d_theta, d_phi, d_r = gridification(G, id_trans)
   d_roll, d_pitch, d_yaw = gridification(G, id_rot)
   gripper = id_gripper
   a = unnomalize(d_theta, d_phi, d_r, d_roll, d_pitch, d_yaw, gripper)
```

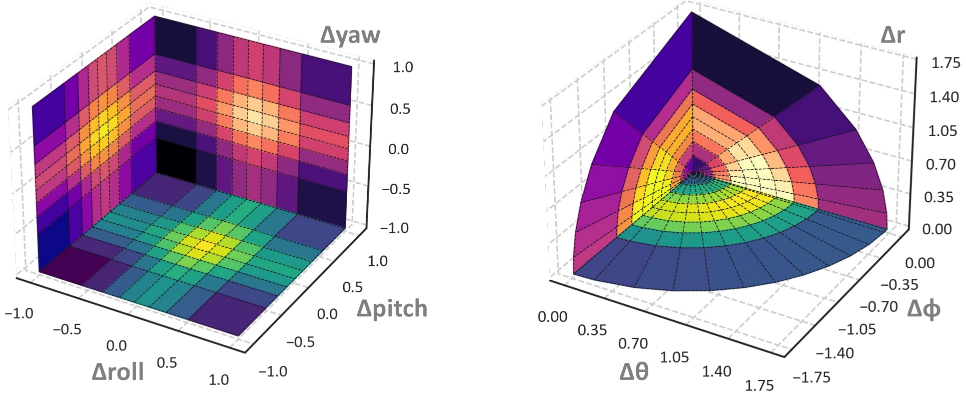
Adaptive Action Grids in pre-training and post-training



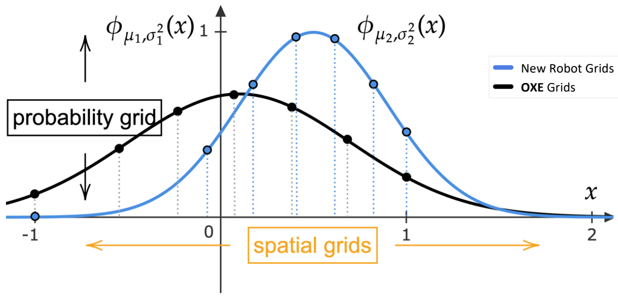
a) Action Distribution of ΔR and ΔT



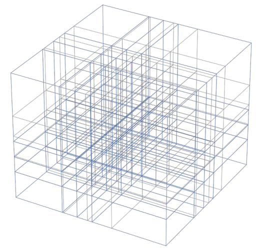
b) Action Grid Split from Distribution



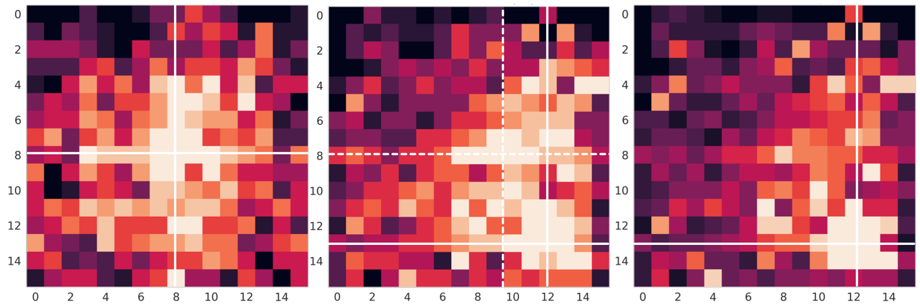
c) Action Grids of ΔR and ΔT



a) Action Grids Transfer



b) Action Grids Split



Embedding Pretrained on OXE

Embedding with Feature Adaption

Embedding Fine-tuned on LIBERO-Spatial

c) Embedding Feature Adaption

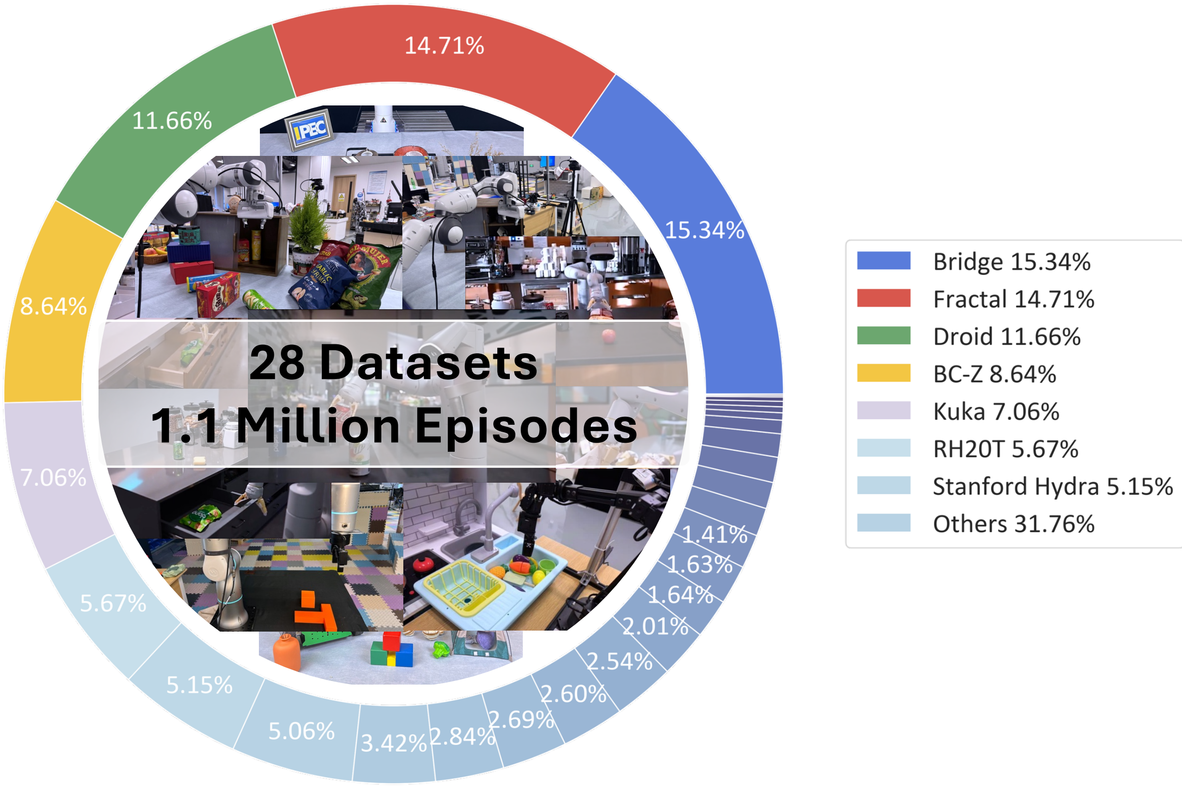
- ❑ **Pre-training**: Construct adaptive action grids from dataset Gaussian distribution
- ❑ **Post-training**: Adjust action grids and spatial feature from robot-specific dataset

#D Ablations on Design Decisions

#setting	Pick Coke Can		Move Near		Put Carrot on Plate		Put Eggplant in Yellow Basket	
	variant aggregation	visual matching	variant aggregation	visual matching	grasp carrot	success	grasp eggplant	success
[1]. **SpatialVLA**	81.6%	70.7%	79.2%	85.4%	41.7%	33.3%	91.7%	87.5%
[2]. ~ linear 256 bins	40.7%	19.0%	47.1%	52.9%	41.7%	33.3%	87.5%	70.8%
[3]. ~ uniform distribution	77.9%	28.0%	64.2%	55.0%	45.8%	12.5%	79.2%	54.2%
[4]. ~ resolution 1026	74.4%	67.3%	59.1%	54.2%	45.8%	25.0%	66.7%	54.2%
[5]. – ego3d encoding	68.9%	70.3%	66.7%	62.0%	54.2%	12.5%	75.0%	37.5%
[6]. – freeze llm embedding	70.2%	50.7%	63.1%	62.5%	33.3%	20.8%	95.8%	79.2%

Pre-training Ablations on the Mixture Dataset of Google Fractal and BridgeData V2

SpatialVLA Dataset Mixtures Details

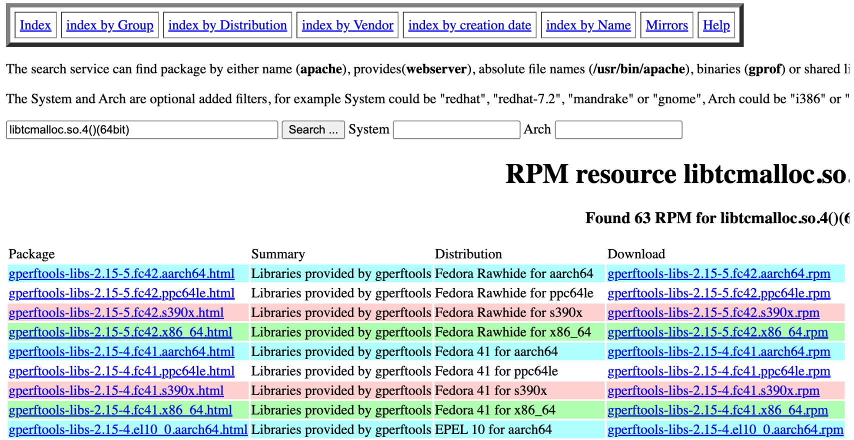


Dataset	Weight	trajectory	sample
Bridge [59, 17]	15.34%	60064	2135463
Fractal [6]	14.71%	87212	3786400
Droid [28]	11.66%	92233	27044326
BC-Z [25]	8.64%	43264	6015535
Kuka [26]	7.06%	209880	2455879
RH20T [18]	5.67%	104392	52644433
Stanford Hydra [2]	5.15%	570	358234
Language Table [41]	5.06%	442226	7045476
Taco Play [52, 43]	3.42%	3603	237798
Furniture Bench [22]	2.84%	5100	3948057
Roboturk [42]	2.69%	1995	187507
Utaustin Mutex [56]	2.60%	1500	361883
Austin Sailor [45]	2.54%	240	353094
Austin Sirius [37]	2.01%	559	279939
DobbE [55]	1.64%	5208	1139911
FMB Dataset [40]	1.63%	8612	1137459
Berkeley Autolab UR5 [9]	1.41%	1000	97939
Toto [66]	1.17%	1003	325699
Viola [71]	1.10%	150	76324
IAMLab CMU Pickup Insert [54]	1.05%	631	146241
NYU Franka [14]	0.97%	456	44875
Jaco Play [15]	0.56%	1085	77965
Berkeley Cable Routing [39]	0.30%	1647	42328
Austin Buds [69]	0.25%	50	34112
Berkeley Fanuc Manipulation [68]	0.22%	415	62613
CMU Stretch [44]	0.18%	135	25016
DLR EDAN Shared Control [50]	0.06%	104	8928
UCSD Kitchen [61]	0.06%	150	3970

Table.3 SpatialVLA Dataset Mixtures Details.

How to pre-train or post-train Spatial VLA?

- ❑ **Dataset and Parallel**
  - ❑ RLDS with 65536 shuffle buffer (recommend LeRobot)
  - ❑ individually shuffle with random seed
  - ❑ Data augmentation matters: color jitter, crop and rotation
- ❑ **Computational Source**
  - ❑ **pre-train**: 64 A100 GPUS for 10 days
  - ❑ **post-train**: 4~8 A100 GPUS for 6 hours
- ❑ **Traning prams**
  - ❑ **lr**: 2e-5 for pre-train or full params post-train
    1e-4 for LoRA tuning, linear scheduler, **bs**: 2048
  - ❑ **params**: **full params training** except llm embeds
  - ❑ **mixed precision**: bf16, **DeepSpeed** zero1, 1 epoch (457M)
  - ❑ **Spatial Grids** 8194, **Ego3d** reso 2, freqs 8
  - ❑ **Template** <IMAGE><IMAGE><IMAGE><IMAGE><bos> What action should the robot take to pick the cup? \n <ACTION00880><ACTION05511><ACTION08193> × 4 <eos>
- ❑ **Deployment**
  - ❑ Action chunking helps, CogACT[1]
  - ❑ WidowX, Franka, FastUmi XArm



Index	index by Group	index by Distribution	index by Vendor	index by creation date	index by Name	Mirrors	Help

The search service can find package by either name (**apache**), provides(**webserver**), absolute file names (**/usr/bin/apache**), binaries (**gprof**) or shared li

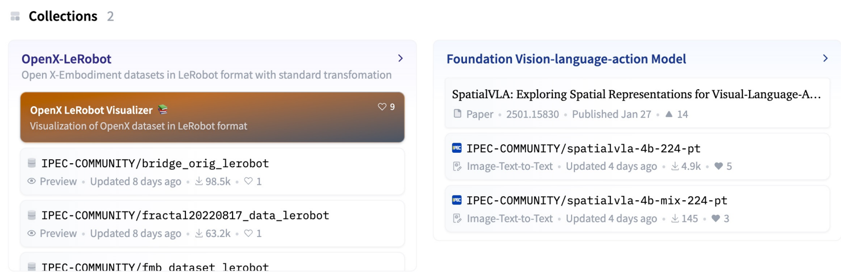
The System and Arch are optional added filters, for example System could be "redhat", "redhat-7.2", "mandrake" or "gnome", Arch could be "i386" or "

libtcmalloc.so.4()(64bit) Search ... System Arch

RPM resource libtcmalloc.so.

Found 63 RPM for libtcmalloc.so.4()(6

Package	Summary	Distribution	Download
gperftools-libs-2.15-5.fc42.aarch64.html	Libraries provided by gperftools	Fedora Rawhide for aarch64	gperftools-libs-2.15-5.fc42.aarch64.rpm
gperftools-libs-2.15-5.fc42.ppc64le.html	Libraries provided by gperftools	Fedora Rawhide for ppc64le	gperftools-libs-2.15-5.fc42.ppc64le.rpm
gperftools-libs-2.15-5.fc42.s390x.html	Libraries provided by gperftools	Fedora Rawhide for s390x	gperftools-libs-2.15-5.fc42.s390x.rpm
gperftools-libs-2.15-5.fc42.x86_64.html	Libraries provided by gperftools	Fedora Rawhide for x86_64	gperftools-libs-2.15-5.fc42.x86_64.rpm
gperftools-libs-2.15-4.fc41.aarch64.html	Libraries provided by gperftools	Fedora 41 for aarch64	gperftools-libs-2.15-4.fc41.aarch64.rpm
gperftools-libs-2.15-4.fc41.ppc64le.html	Libraries provided by gperftools	Fedora 41 for ppc64le	gperftools-libs-2.15-4.fc41.ppc64le.rpm
gperftools-libs-2.15-4.fc41.s390x.html	Libraries provided by gperftools	Fedora 41 for s390x	gperftools-libs-2.15-4.fc41.s390x.rpm
gperftools-libs-2.15-4.fc41.x86_64.html	Libraries provided by gperftools	Fedora 41 for x86_64	gperftools-libs-2.15-4.fc41.x86_64.rpm
gperftools-libs-2.15-4.el10_0.aarch64.html	Libraries provided by gperftools	EPEL 10 for aarch64	gperftools-libs-2.15-4.el10_0.aarch64.rpm



https://huggingface.co/IPEC-COMMUNITY

[1] CogACT: A Foundational Vision-Language-Action Model for Synergizing Cognition and Action in Robotic Manipulation

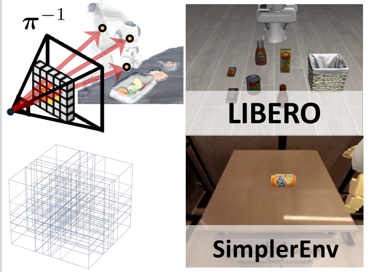
Experiments to serve as a generalist robot policy

❑ **Experiment Setup**: 7 robot learning scenarios, 16 real-robot tasks, and 48 simulation setups

#A Performing Zero-shot Robot Control

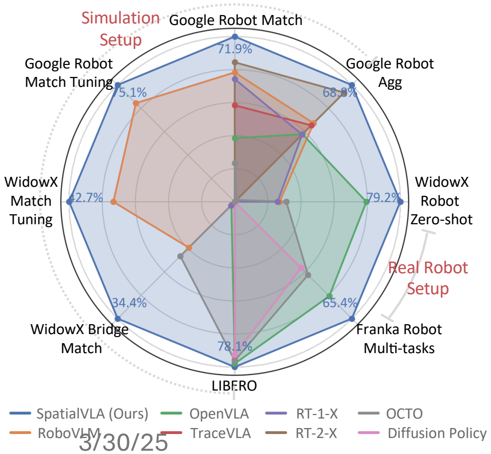
#B Effectively Adapting to New Robot Setups

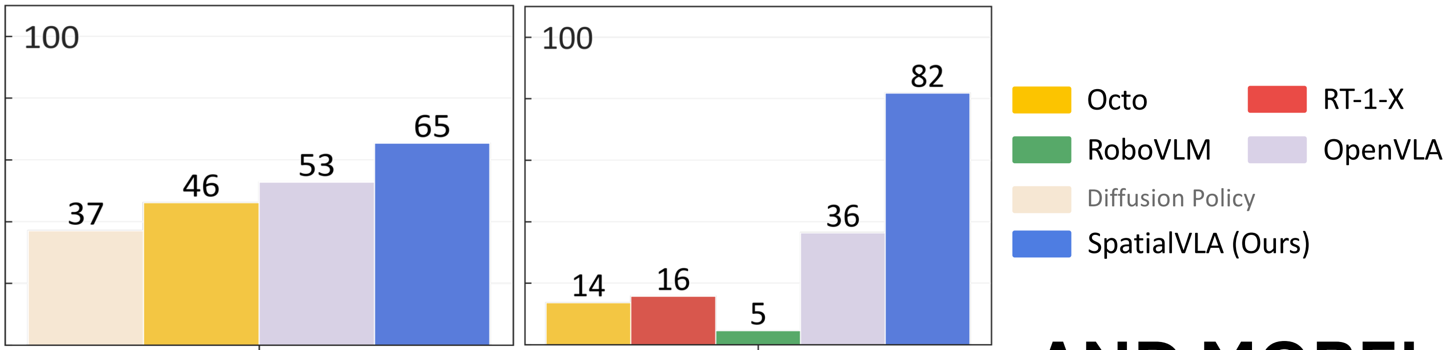
#C Evaluating Spatial Understanding Capability



#D Ablations on Design Decisions

❑ **Our model achieves SOTA performance across a diverse range of evaluations**





Adapting to New Robot Setup

Spatial Understanding

AND MORE!

#A Performing Zero-shot Robot Control (simpler env simulator)

- ❑ **How well does SpatialVLA directly perform on a variety of in-distribution tasks after pre-training on large-scale robotic data mixture?**

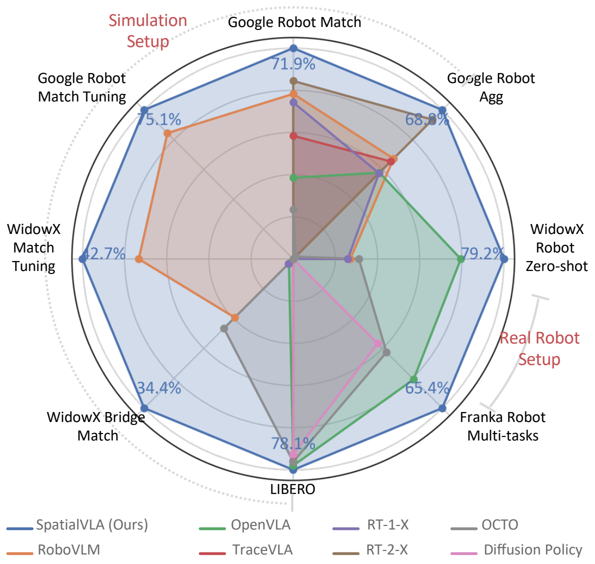


Table.1 SimplerEnv evaluation across different policies on **Google Robot** tasks.

Model	Visual Matching				Variant Aggregation			
	Pick Coke Can	Move Near	Open/Close Drawer	#Average	Pick Coke Can	Move Near	Open/Close Drawer	#Average
RT-1 [6] (Begin)	2.7%	5.0%	13.9%	6.8%	2.2%	4.0%	6.9%	4.2%
RT-1 [6] (15%)	71.0%	35.4%	56.5%	60.2%	81.3%	44.6%	26.7%	56.2%
RT-1 [6] (Converged)	85.7%	44.2%	73.0%	74.6%	89.8%	50.0%	32.3%	63.3%
HPT [60]	56.0%	60.0%	24.0%	46.0%	—	—	—	—
TraceVLA [65]	28.0%	53.7%	57.0%	42.0%	60.0%	56.4%	31.0%	45.0%
RT-1-X [13]	56.7%	31.7%	59.7%	53.4%	49.0%	32.3%	29.4%	39.6%
RT-2-X [13]	78.7%	77.9%	25.0%	60.7%	82.3%	79.2%	35.3%	64.3%
Octo-Base [46]	17.0%	4.2%	22.7%	16.8%	0.6%	3.1%	1.1%	1.1%
OpenVLA [29]	16.3%	46.2%	35.6%	27.7%	54.5%	47.7%	17.7%	39.8%
RoboVLM (zero-shot) [31]	72.7%	66.3%	26.8%	56.3%	68.3%	56.0%	8.5%	46.3%
RoboVLM (fine-tuning) [31]	77.3%	61.7%	43.5%	63.4%	75.6%	60.0%	10.6%	51.3%
SpatialVLA (zero-shot)	81.0%	69.6%	59.3%	**71.9%**	89.5%	71.7%	36.2%	**68.8%**
SpatialVLA (fine-tuning)	86.0%	77.9%	57.4%	**75.1%**	88.0%	72.7%	41.8%	**70.7%**

Model	Put Spoon on Towel		Put Carrot on Plate		Stack Green Block on Yellow Block		Put Eggplant in Yellow Basket		#Overall
	Grasp Spoon	Success	Grasp Carrot	Success	Grasp Green Block	Success	Grasp Eggplant	Success	Average
RT-1-X [13]	16.7%	0%	20.8%	4.2%	8.3%	0%	0.0%	0%	1.1%
Octo-Base [46]	34.7%	12.5%	52.8%	8.3%	31.9%	0%	66.7%	43.1%	16.0%
Octo-Small [46]	77.8%	47.2%	27.8%	9.7%	40.3%	4.2%	87.5%	56.9%	30.0%
OpenVLA [29]	4.1%	0%	33.3%	0%	12.5%	0%	8.3%	4.1%	1.0%
RoboVLM (zero-shot) [31]	37.5%	20.8%	33.3%	25.0%	8.3%	8.3%	0.0%	0%	13.5%
RoboVLM (fine-tuning) [31]	54.2%	29.2%	25.0%	25.0%	45.8%	12.5%	58.3%	58.3%	31.3%
SpatialVLA (zero-shot)	25.0%	20.8%	41.7%	20.8%	58.3%	25.0%	79.2%	70.8%	**34.4%**
SpatialVLA (fine-tuning)	20.8%	16.7%	29.2%	25.0%	62.5%	29.2%	100.0%	100.0%	**42.7%**

Table.2 SimplerEnv evaluation across different policies on **WidowX Robot** tasks.

https://github.com/DelinQu/SimplerEnv-OpenVLA

Performing Zero-shot Robot Control Evaluation on SimplerEnv

Vertical Laying

Standing

Horizontal Laying

Pick Coke Can

In terms of Zero-shot Robot Control,

#A Performing Zero-shot Robot Control (WidowX Robot Setup)

- ❑ **How well does SpatialVLA directly perform on a variety of in-distribution tasks after pre-training on large-scale robotic data mixture?**

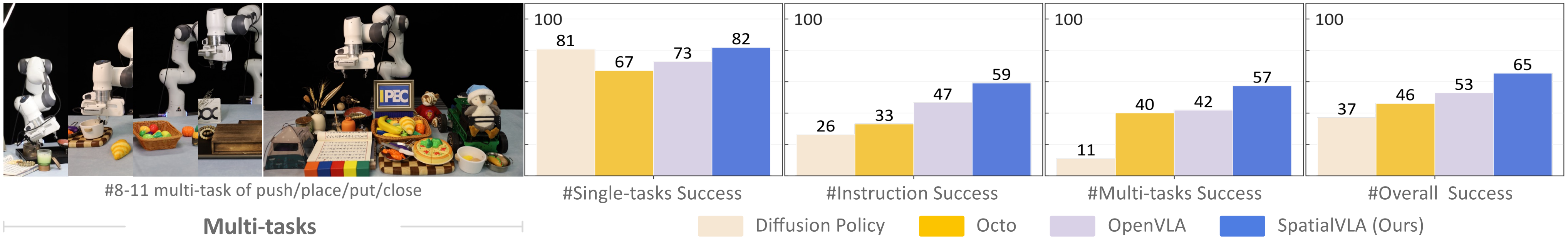
#close microwave
#lift red chill pepper
#put the carrot in the plate

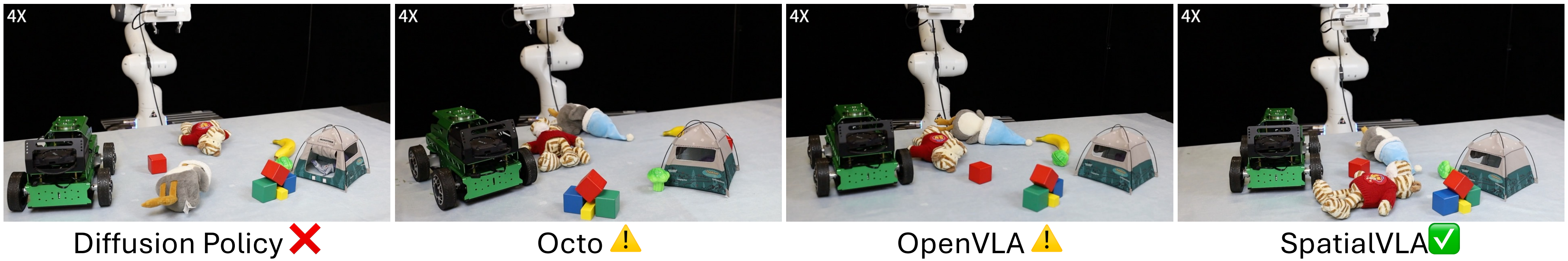
#put the eggplant in the basket
#put purple cup on the white plate
#put green cup on the pink cloth
#put purple cup on the pink cloth

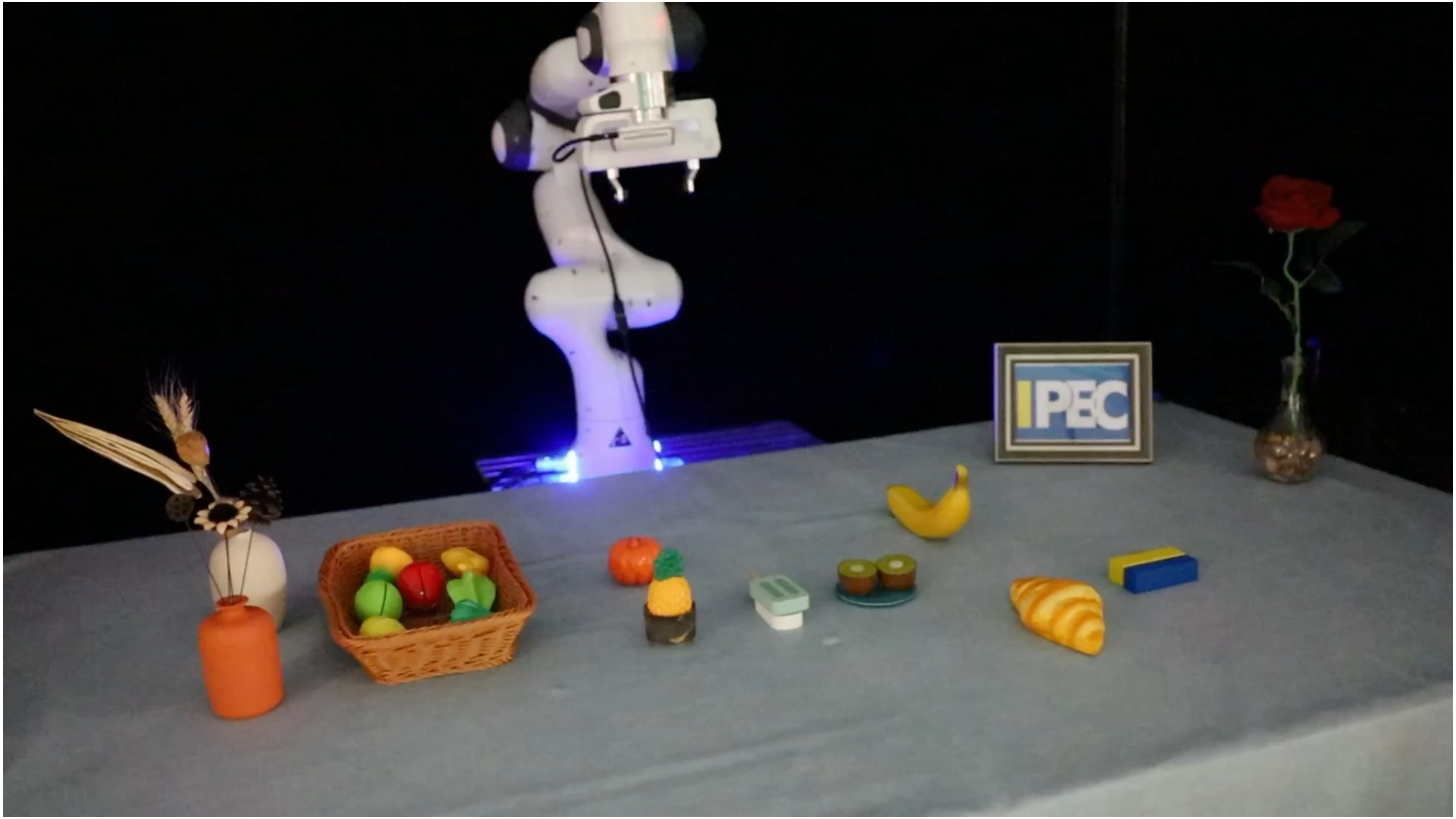
unseen background · unseen object pose · motion distractors · language grounding

#B Effectively Adapting to New Robot Setups

❑ **Experiment Setup**: 7 robot learning scenarios, 16 real-robot tasks, and 48 simulation setups






IPEC

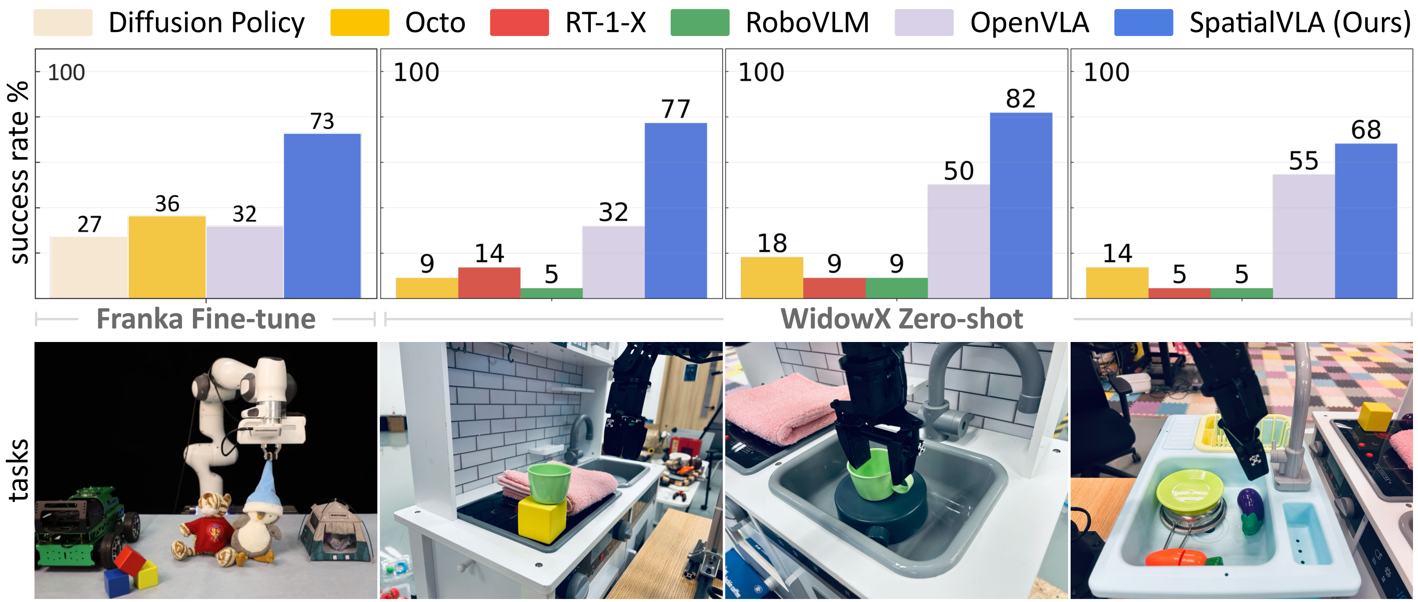
Adapting to New Robot Setups on Libero

Libero-Spatial

In terms of testing the model's ability to adapt to new robot setups,

#C Evaluating Spatial Understanding Capability

❑ **Experiment Setup**: 7 robot learning scenarios, 16 real-robot tasks, and 48 simulation setups



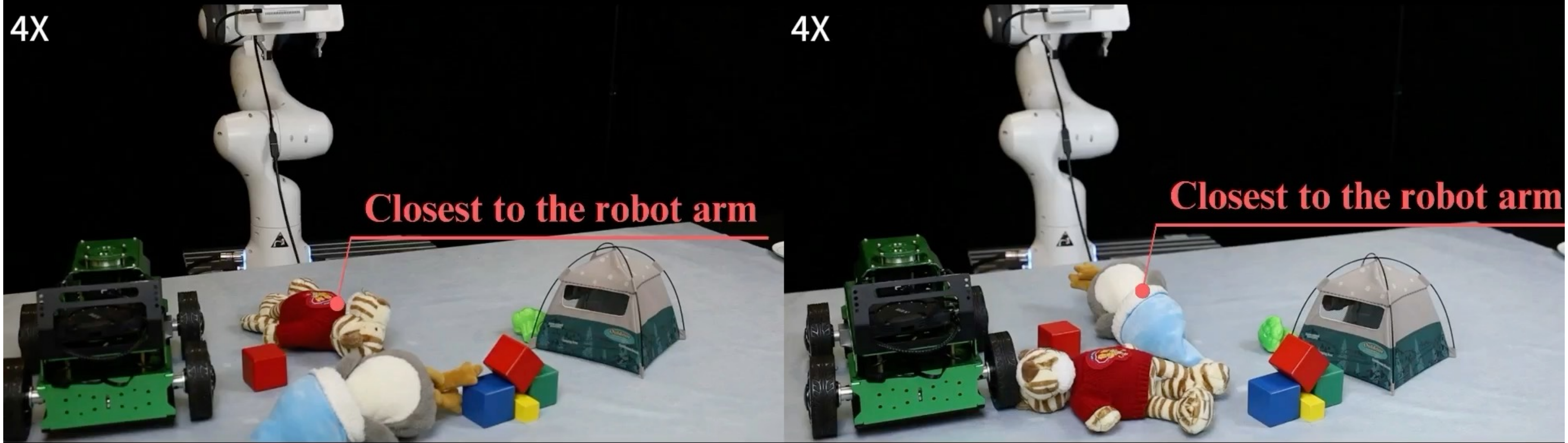
#1 place plush toy **closest** to robot on car

#2 put green cup on the pink cloth **(cup height change)**

#3 put green cup on the pink cloth **(cup height change)**

#4 put the carrot in the plate **(pot height change)**

Evaluating Spatial Understanding Capability (fine-tuning tasks)



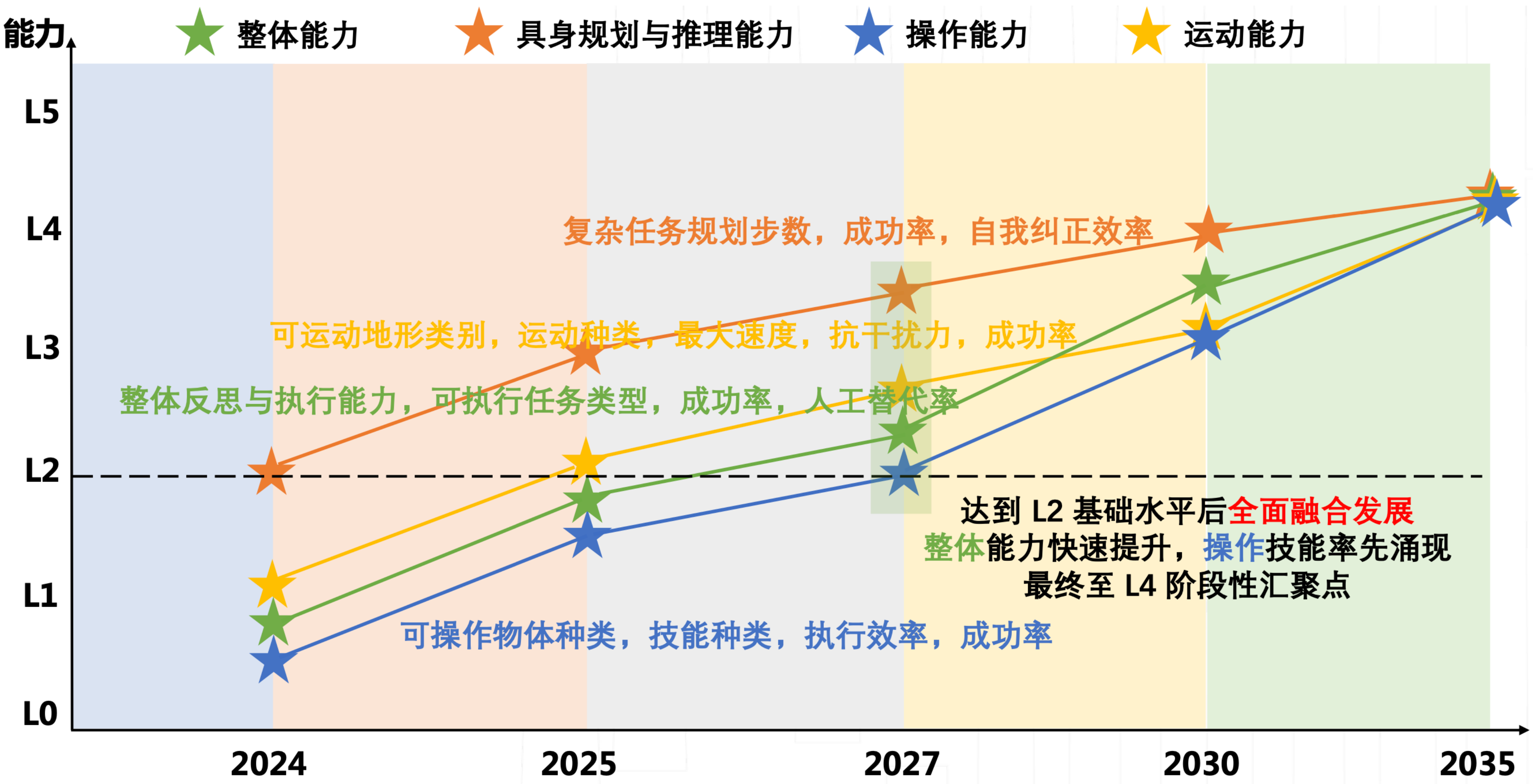
place plush toy closest to robot on car

In addition, to further verify the spatial understanding ability of the model,

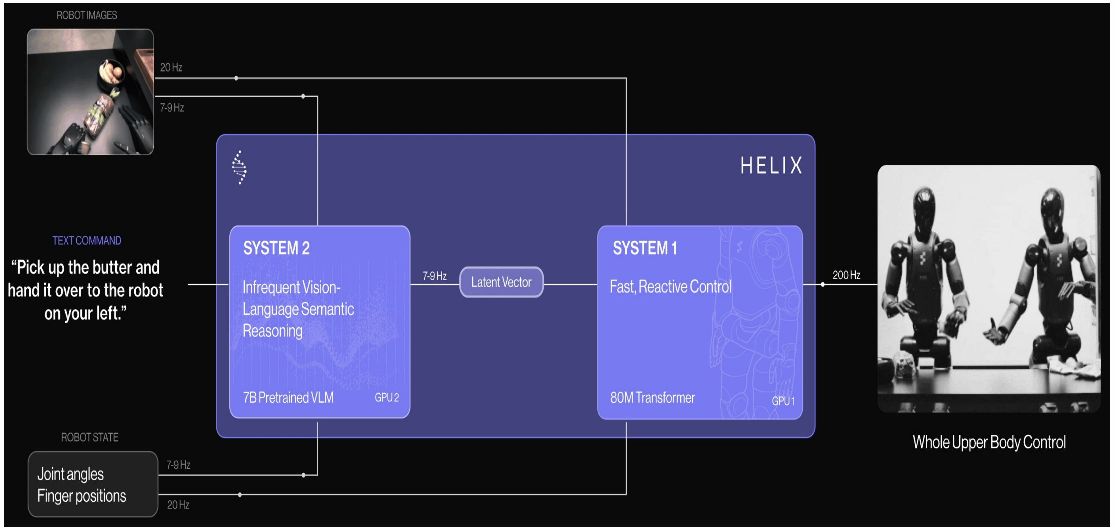
Limitations of Spatial VLA

- **Training only on robot action data, w/o QA**
  - Insufficient instruction-following ability
  - language model collapse
- **More Generalizable Distribution Fitting**
  - Is modeling data distributions as Gaussian optimal? Single axis motion?
  - Dataset noises can further distort the spatial grid distribution
- **More Flexible VLA architectures beyond AR**
  - 21Hz inference speed is slower than diffusion decoding
  - Integrating diffusion decoding with spatial grid and exploring dynamic token numbers for action mapping will be valuable
- **Not well-suited for Long-Horizon Reasoning**
- **Higher-Quality Diverse Data**
  - Pre-trained on OXE and RH20T, but the variable quality of OXE data can hinder training
  - Exploring optimal data composition and distilling for boosting model efficiency and generalizability

Roadmap of Embodied AI



Toward the More Generalist Agents System



Helix: A Vision-Language-Action Model for Generalist Humanoid Control, Figure AI

Hi Robot: Open-Ended Instruction Following with Hierarchical Vision-Language-Action Models, Physical Intelligence

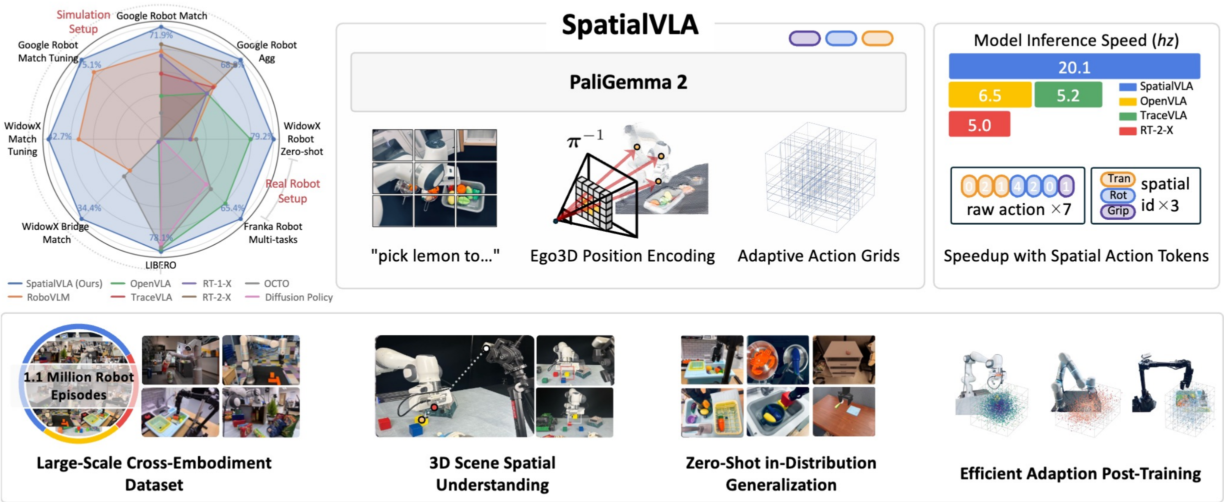
Toward the More Generalist Agents System

- ❑ **Hierarchical and Flexible VLA architectures**
  - ❑ Hierarchical Autonomy Stack: System1-System2
  - ❑ Cross-embodied Learning
  - ❑ More powerful foundation vision-language-action model
    - ❑ Long-Horizon Reasoning
    - ❑ Spatial awareness in physical word, e.g., 3d, bounding box, April Tags
    - ❑ Instruction-following ability
- ❑ **Higher-Quality Diverse Data**
  - ❑ Combine simulation data and real-world data
  - ❑ Web Data
  - ❑ Reward Data
- ❑ **Reinforcement Learning with Large Datasets [1][2]**
- ❑ **Test-time-scaling and reasoning**
- ❑ **Robotic lifelong reinforcement learning**
- ❑ **Large-Scale Benchmark**
  - ❑ **LIBERO, RL Bench, CALVIN, Simpler Env**
  - ❑ **VLA Bench**

Thanks for all the collaborators

SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model

Delin Qu*[1], Haoming Song*[1], Qizhi Chen*[1], Yuanqi Yao[1], Xinyi Ye[1], Yan Ding[1], Zhigang Wang[1]
Jiayuan Gu[2], Bin Zhao[1], Dong Wang[1], Xuelong Li[13]
[1]Shanghai AI Laboratory, [2]ShanghaiTech, [3]TeleAI
https://spatialvla.github.io



Roadmap of Embodied AI

Simulator
Real:Sim

Test-time-scaling

BenchMark

Large-scale Dataset

Foundation Vision-language-action Model