Visual-Language-Action Models

Paper Reading

Jianglin Lu

https://jianglin954.github.io/
jianglinlu@outlook.com

Outline

1. Vision-Language-Action Models
   - RT-2
   - OpenVLA
   - π_0
   - UniVLA
2. 3D Vision-Language-Action Models
   - SpatialVLA
   - 3D-VLA
3. Dataset
   - CALVIN
   - LIBERO
   - SimplerEnv

Outline

1. Vision-Language-Action Models
2. 3D Vision-Language-Action Models
3. Dataset

Outline

1. Vision-Language-Action Models
   - RT-2
   - OpenVLA
   - π_0
   - UniVLA
2. 3D Vision-Language-Action Models
3. Dataset

Robotics Transformer 2 (RT-2)

Motivation:

- Existing methods generally address only the "higher level" aspects of robotic planning, essentially taking the role of a state machine that interprets commands and parses them into individual primitives, which are then executed by separate low-level controllers that themselves do not benefit from the rich semantic knowledge of Internet-scale models during training.
- *Can large pretrained vision-language models be integrated directly into low-level robotic control to boost generalization and enable emergent semantic reasoning?*

Brohan et al. Vision-Language-Action Models Transfer Web Knowledge to Robotic Control, CoRL, 2023

Robotics Transformer 2 (RT-2)

Approaches: `https://robotics-transformer2.github.io/`

Δ Directly train vision-language models designed for open-vocabulary visual question answering and visual dialogue to output low-level robot actions, along with solving other Internet-scale vision-language tasks.

Δ Tokenize the actions into text tokens and create multimodal sentences that respond to robotic instructions paired with camera observations by producing corresponding actions.

Δ If we augment the command with chain of thought prompting, the model is able to make even more complex semantic inferences.

Brohan et al. Vision-Language-Action Models Transfer Web Knowledge to Robotic Control, CoRL, 2023

Robotics Transformer 2 (RT-2)

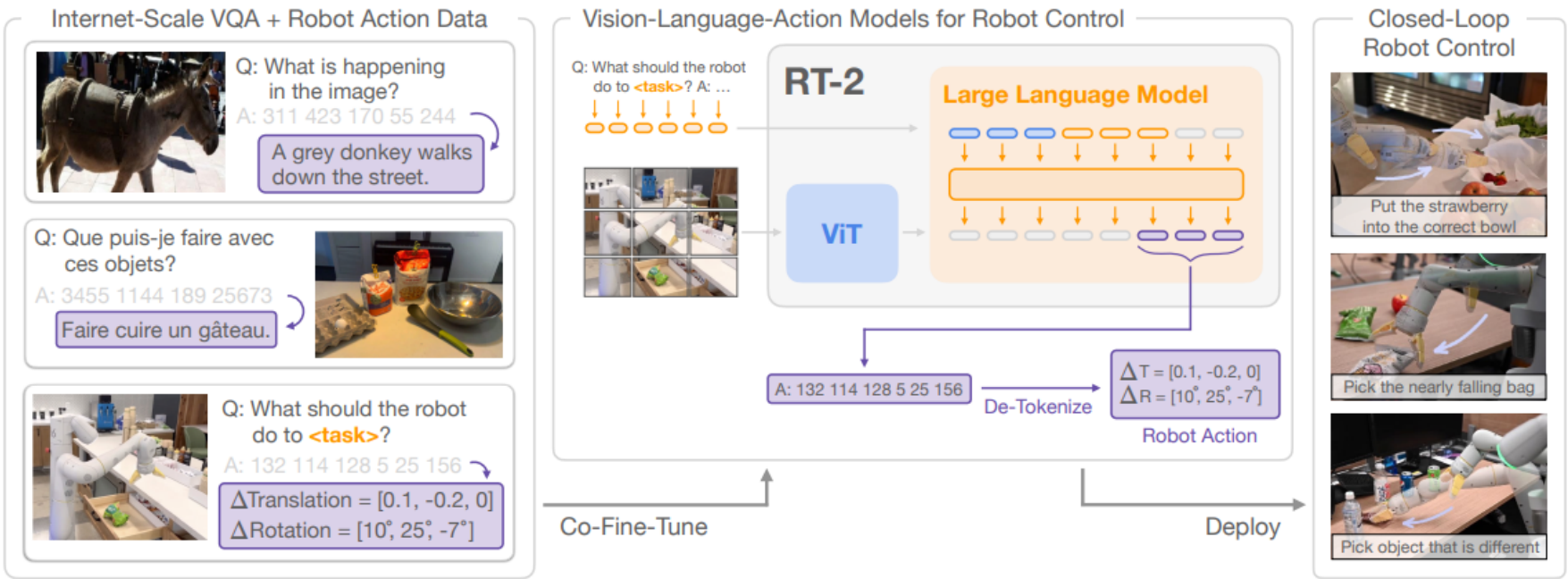


Figure 1 | RT-2 overview: we represent robot actions as another language, which can be cast into text tokens and trained together with Internet-scale vision-language datasets. During inference, the text tokens are de-tokenized into robot actions, enabling closed loop control. This allows us to leverage the backbone and pretraining of vision-language models in learning robotic policies, transferring some of their generalization, semantic understanding, and reasoning to robotic control. We demonstrate examples of RT-2 execution on the project website: robotics-transformer2.github.io.

Brohan et al. Vision-Language-Action Models Transfer Web Knowledge to Robotic Control, CoRL, 2023

Robotics Transformer 2 (RT-2)

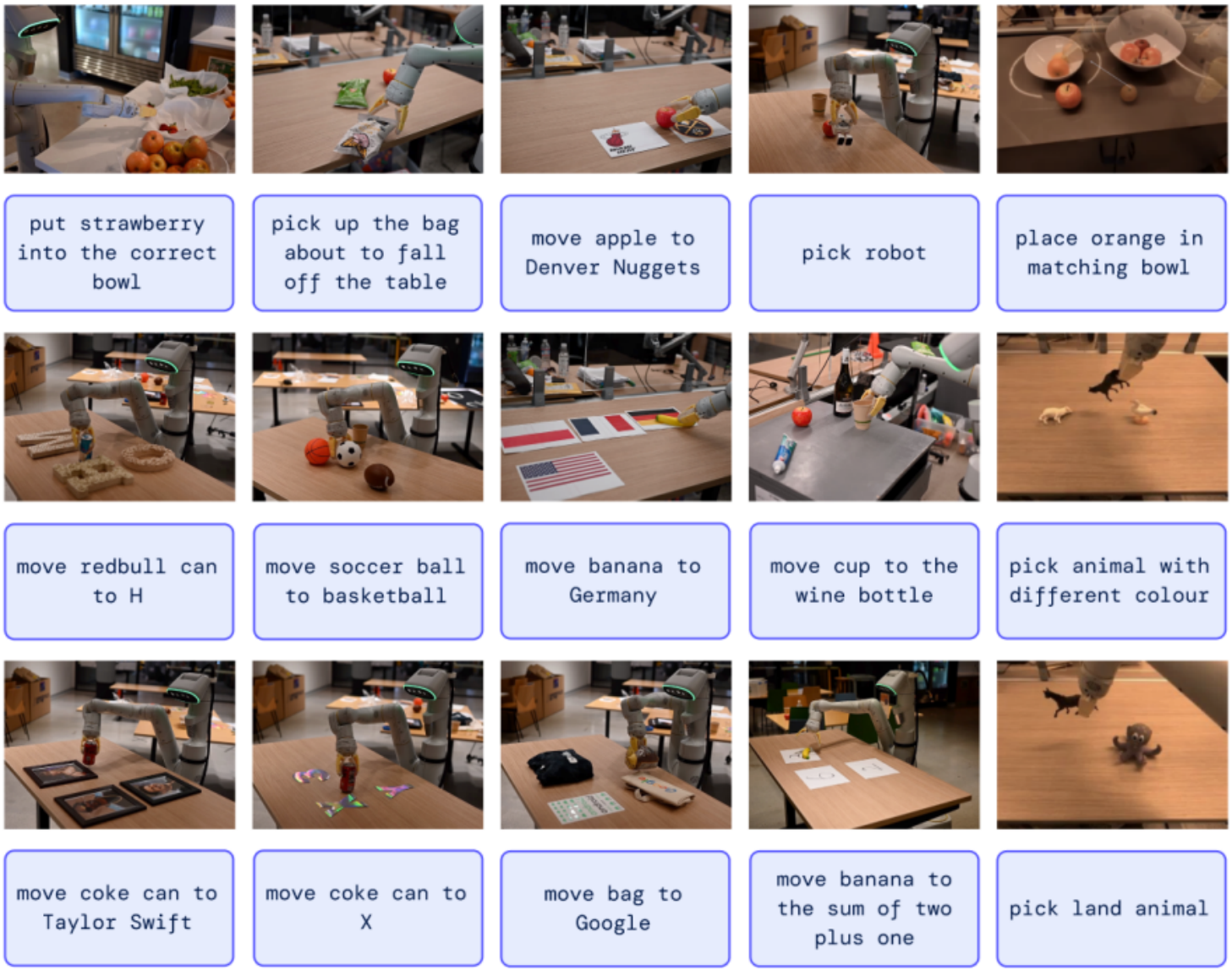


Figure 2 | RT-2 is able to generalize to a variety of real-world situations that require reasoning, symbol

Robotics Transformer 2 (RT-2)

Robot-Action Fine-tuning:

Δ In order to define a target for VLM fine-tuning we convert the action vector into a single string by simply concatenating action tokens for each dimension with a space character:
"terminate Δpos_x Δpos_y Δpos_z Δrot_x Δrot_y Δrot_z *gripper_extension*"
A possible instantiation could be: "1 128 91 241 5 101 127".

Δ A key technical detail of the training recipe that improves robot performance is co-fine-tuning robotics data with the original web data instead of naïve finetuning on robot data only.

Δ Constrain output vocabulary via only sampling valid action tokens when the model is prompted with a robot-action task, whereas the model is still allowed to output the full range of natural language tokens on standard vision-language tasks.

Brohan et al. Vision-Language-Action Models Transfer Web Knowledge to Robotic Control, CoRL, 2023

Robotics Transformer 2 (RT-2)

Limitation:

Δ The model's physical skills are still limited to the distribution of skills seen in the robot data, but it learns to deploy those skills in new ways. We believe this is a result of the dataset not being varied enough along the axes of skills.

Δ The computation cost of these models is high, and as these methods are applied to settings that demand high-frequency control, real-time inference may become a major bottleneck.

Brohan et al. Vision-Language-Action Models Transfer Web Knowledge to Robotic Control, CoRL, 2023

Outline

1. Vision-Language-Action Models
   - RT-2
   - OpenVLA
   - π_0
   - UniVLA
2. 3D Vision-Language-Action Models
3. Dataset

An Open-Source VLA Model (OpenVLA)

Motivation:

- Current VLAs are closed, with limited visibility into model architecture, training procedures, and data mixture. They focus on training and evaluating in single robot or simulated setups and thus lack generality.
- Existing works do not provide best practices for deploying and adapting VLAs to new robots, environments, and tasks, especially on commodity hardware (e.g., consumer-grade GPUs).

Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, CoRL, 2024

An Open-Source VLA Model (OpenVLA)

Approaches: `https://github.com/openvla/openvla`

△ Introduce OpenVLA, a 7B-parameter open-source VLA that establishes a new state of the art for generalist robot manipulation policies.

△ OpenVLA consists of a pretrained visually-conditioned language model backbone that captures visual features at multiple granularities, fine-tuned on a large, diverse dataset of 970k robot manipulation trajectories from the Open-X Embodiment dataset.

△ The first to demonstrate the effectiveness of compute-efficient fine-tuning methods leveraging low-rank adaptation and model quantization to facilitate adapting OpenVLA models on consumer-grade GPUs instead of large server nodes without compromising performance.

Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, CoRL, 2024

An Open-Source VLA Model (OpenVLA)

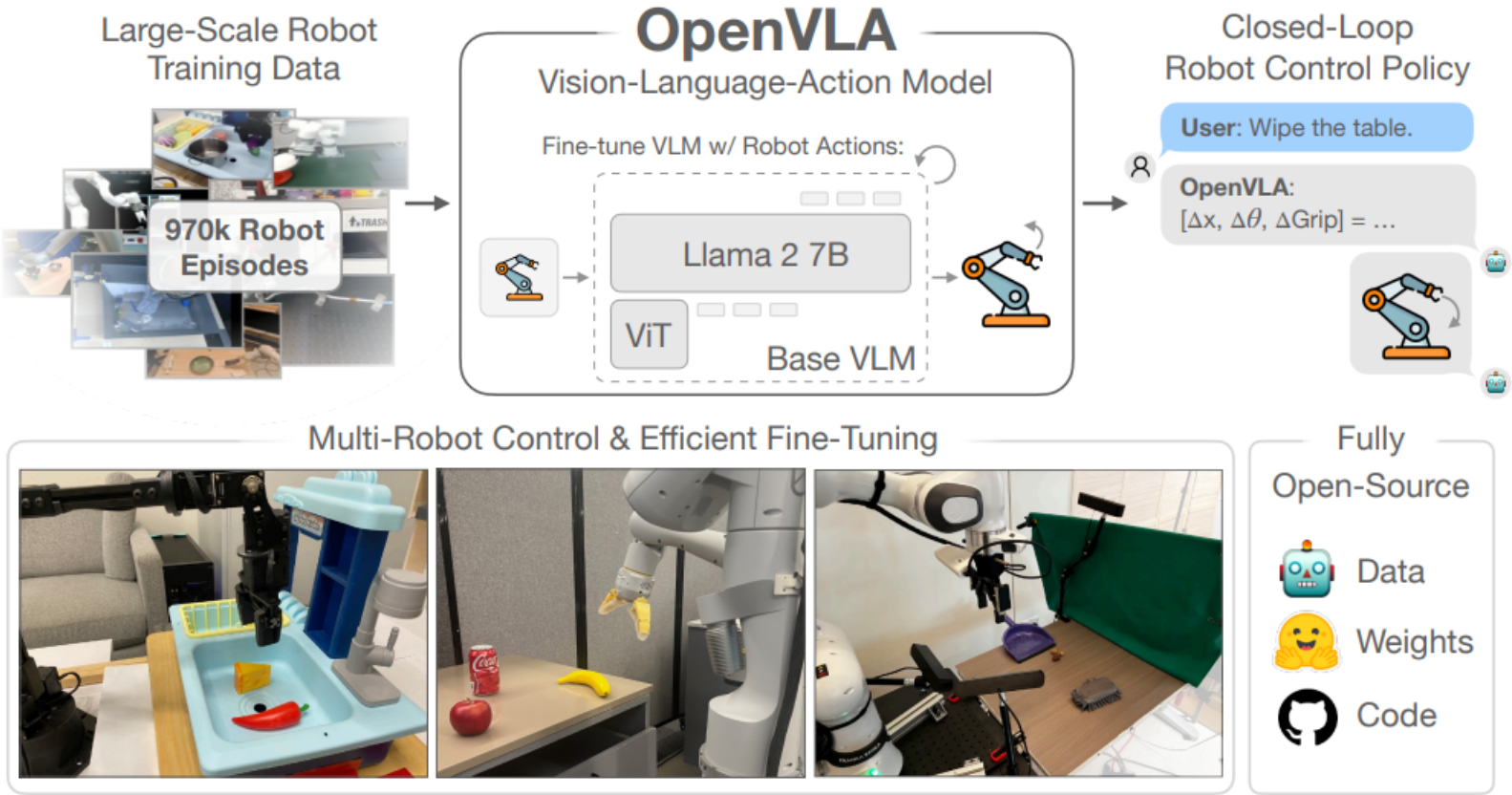


Figure 1: We present OpenVLA, a 7B-parameter open-source vision-language-action model (VLA), trained on 970k robot episodes from the Open X-Embodiment dataset [1]. OpenVLA sets a new state of the art for generalist robot manipulation policies. It supports controlling multiple robots out of the box and can be quickly adapted to new robot domains via parameter-efficient fine-tuning. The OpenVLA checkpoints and PyTorch training pipeline are fully open-source and models can be downloaded and fine-tuned from HuggingFace.

Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, CoRL, 2024

An Open-Source VLA Model (OpenVLA)

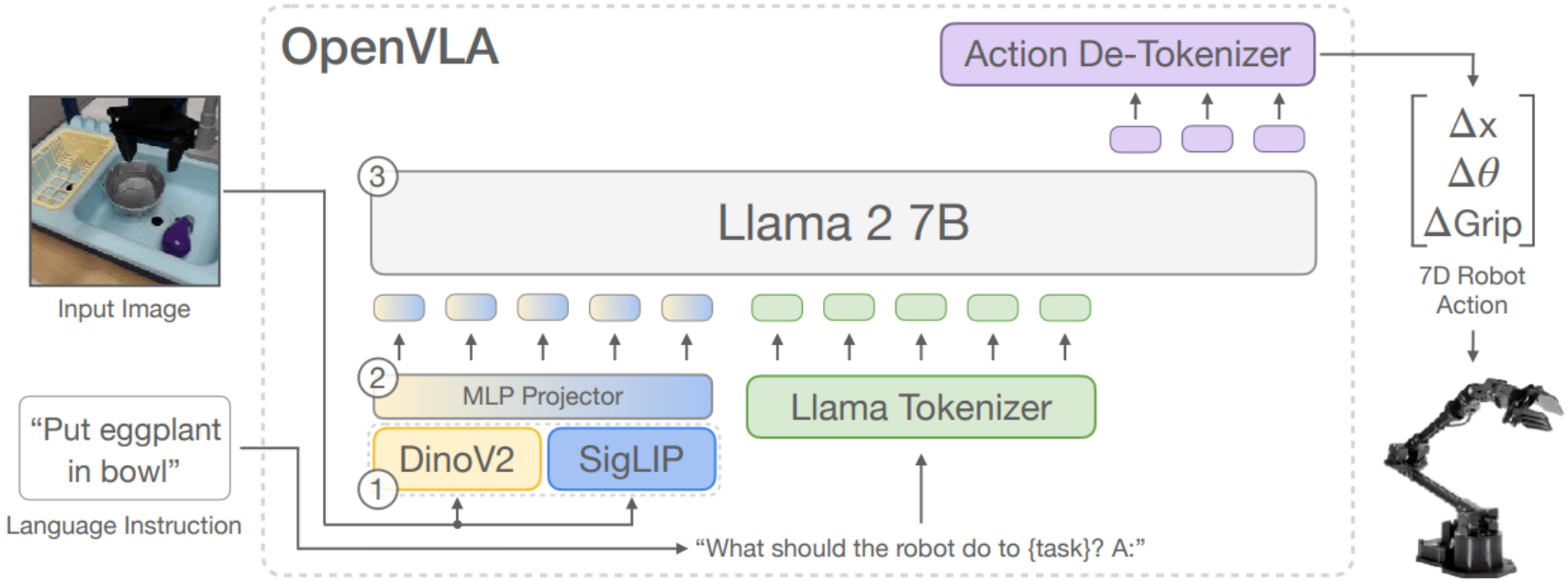


Figure 2: **OpenVLA model architecture.** Given an image observation and a language instruction, the model predicts 7-dimensional robot control actions. The architecture consists of three key components: (1) a **vision encoder** that concatenates Dino V2 [25] and SigLIP [79] features, (2) a **projector** that maps visual features to the language embedding space, and (3) the **LLM backbone**, a Llama 2 7B-parameter large language model [10].

Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, CoRL, 2024

An Open-Source VLA Model (OpenVLA)

Key learnings from explorations:

VLM Backbone. The Prismatic policy outperforms the LLaVA policy by roughly 10% in absolute success rate across both simple single-object tasks and multi-object, language grounding tasks, thanks to improved spatial reasoning capabilities afforded by the fused SigLIP-DinoV2 backbones.

Image Resolution. On many VLM benchmarks, increased resolution does improve performance, but we did not see this trend (yet) for VLAs.

Fine-Tuning Vision Encoder. Fine-tuning the vision encoder during VLA training to be crucial for good VLA performance, as the pretrained vision backbone may not capture sufficient fine-grained spatial details about important parts of the scene to enable precise robotic control.

Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, CoRL, 2024

An Open-Source VLA Model (OpenVLA)

Key learnings from explorations:

Training Epochs. It is important for VLA training to iterate through the training dataset significantly more times, with real robot performance continually increasing until training action token accuracy surpasses 95%. The final training run completes 27 epochs through its training dataset.

Learning Rate. Achieved the best results using a fixed learning rate of 2e-5 (the same learning rate used during VLM pretraining). Did not find learning rate warmup to provide benefits.

Training. The final model is trained on a cluster of 64 A100 GPUs for 14 days, or a total of 21,500 A100-hours, using a batch size of 2048.

Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, CoRL, 2024

An Open-Source VLA Model (OpenVLA)

We list our used data mixture in Table 3. The mixture mostly follows [5], with a few additional datasets.

OpenVLA Training Dataset Mixture	
Fractal [92]	12.7%
Kuka [45]	12.7%
Bridge[6, 47]	13.3%
Taco Play [93, 94]	3.0%
Jaco Play [95]	0.4%
Berkeley Cable Routing [96]	0.2%
Roboturk [97]	2.3%
Viola [98]	0.9%
Berkeley Autolab UR5 [99]	1.2%
Toto [100]	2.0%
Language Table [101]	4.4%
Stanford Hydra Dataset [102]	4.4%
Austin Buds Dataset [103]	0.2%
NYU Franka Play Dataset [104]	0.8%
Furniture Bench Dataset [105]	2.4%
UCSD Kitchen Dataset [106]	<0.1%
Austin Sailor Dataset [107]	2.2%
Austin Sirius Dataset [108]	1.7%
DLR EDAN Shared Control [109]	<0.1%
IAMLab CMU Pickup Insert [110]	0.9%
UTAustin Mutex [111]	2.2%
Berkeley Fanuc Manipulation [112]	0.7%
CMU Stretch [113]	0.2%
BC-Z [55]	7.5%
FMB Dataset [114]	7.1%
DobbE [115]	1.4%
DROID [11]	10.0%[6]

Table 3: OpenVLA training data mixture using datasets from the Open X-Embodiment dataset [1], following [5] with a few additions.

Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, CoRL, 2024

An Open-Source VLA Model (OpenVLA)

Limitation:

△ Real-world robot setups are heterogeneous, with a wide range of possible sensory inputs. OpenVLA currently only supports single-image observations.

△ Improving the inference throughput of OpenVLA is critical to enable VLA control for high-frequency control setups such as ALOHA, which runs at 50Hz.

△ Many VLA design questions remain underexplored: What effect does the size of the base VLM have on VLA performance? Does co-training on robot action prediction data and Internet-scale vision-language data substantially improve VLA performance? What visual features are best-suited for VLA models?

Kim et al. OpenVLA: An Open-Source Vision-Language-Action Model, CoRL, 2024

Outline

1. Vision-Language-Action Models
   - RT-2
   - OpenVLA
   - π_0
   - UniVLA
2. 3D Vision-Language-Action Models
3. Dataset

Vision-Language-Action Flow Model π_0

Motivation: Developing generalist robot policies (robot foundation models) involves a number of major challenges.

- Research must be done at a very large scale, because the full benefits of large-scale pre-training are often not present at smaller scales.
- It requires developing the right model architectures that can effectively make use of diverse data sources, while at the same time being able to represent the intricate and subtle behaviors necessary to interact with complex physical scenes.
- It requires the right training recipe. Much of the recent progress with large models in NLP and computer vision has relied heavily on delicate strategies for curating pre-training and post-training data.

Black et al. π_0: A Vision-Language-Action Flow Model for General Robot Control, arXiv, 2024

Vision-Language-Action Flow Model π_0

Approaches: `https://www.pi.website/blog/pi0`

Δ Utilize a pre-trained VLM to import Internet-scale experience, inheriting the general knowledge, semantic reasoning, and problem-solving abilities of language and vision-language models.

Δ Utilize a variety of diverse robot data sources, employing cross-embodiment training, where data from many robot types is combined into the same model.

Δ Use an action chunking architecture with flow matching to represent complex continuous action distributions, and use a novel action expert that augments the standard VLM with flow-based outputs.

Black et al. π_0: A Vision-Language-Action Flow Model for General Robot Control, arXiv, 2024

Vision-Language-Action Flow Model π_0

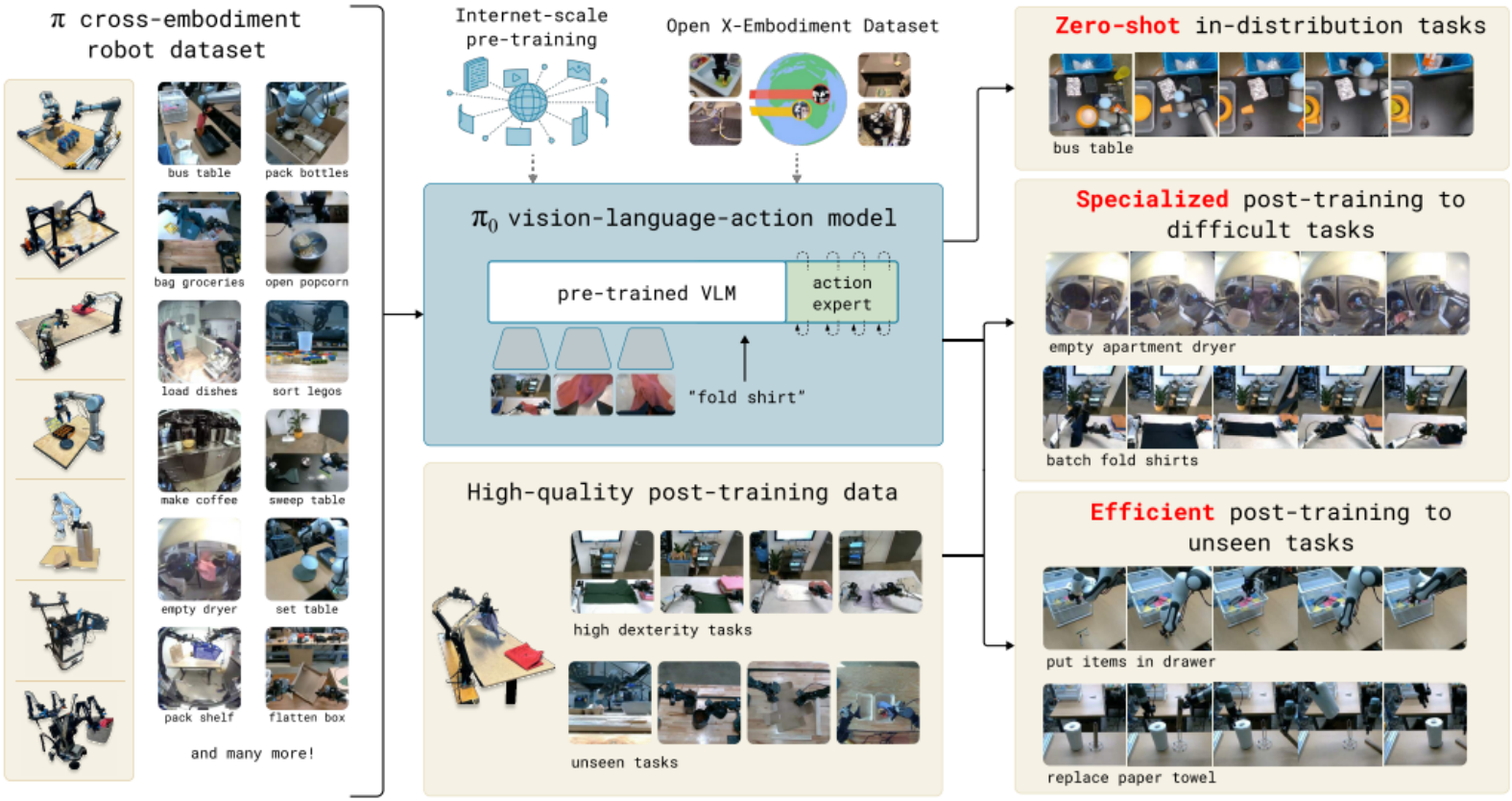


Fig. 1: Our generalist robot policy uses a pre-trained vision-language model (VLM) backbone, as well as a diverse cross-embodiment dataset with a variety of dexterous manipulation tasks. The model is adapted to robot control by adding a separate *action expert* that produces continuous actions via flow matching, enabling precise and fluent manipulation skills. The model can then be used directly to perform tasks based on a prompt, or fine-tuned on high-quality data to enable complex multi-stage tasks, such as folding multiple articles of laundry or assembling a box.

Black et al. π_0: A Vision-Language-Action Flow Model for General Robot Control, arXiv, 2024

Vision-Language-Action Flow Model π_0

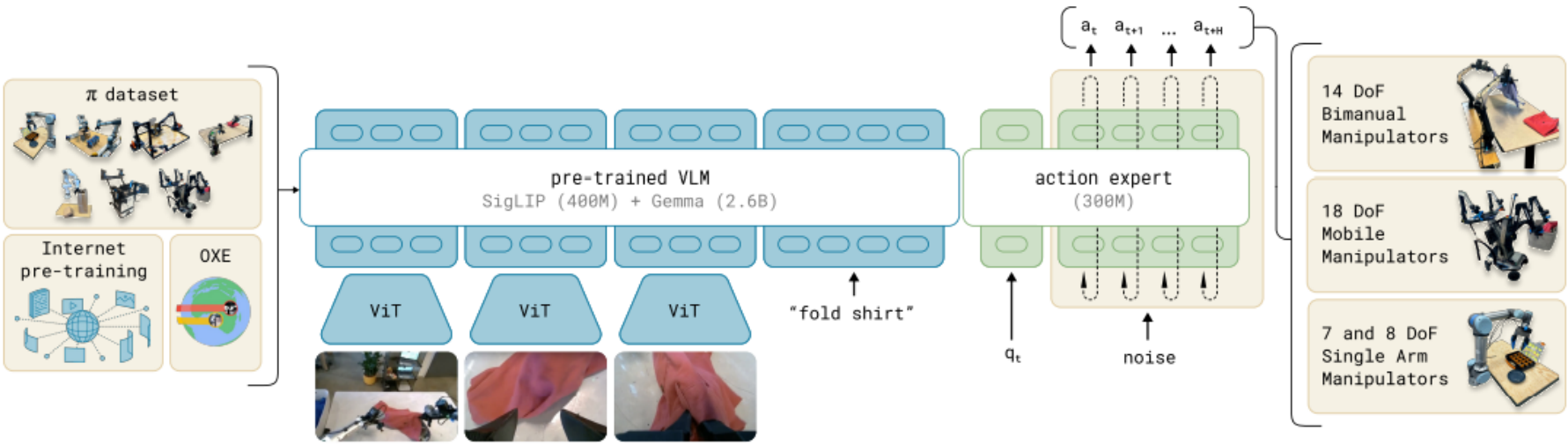


Fig. 3: **Overview of our framework.** We start with a pre-training mixture, which consists of both our own dexterous manipulation datasets and open-source data. We use this mixture to train our flow matching VLA model, which consists of a larger VLM backbone and a smaller *action expert* for processing robot states and actions. The VLM backbone weights are initialized from PaliGemma [5], providing representations learned from large-scale Internet pre-training. The resulting π_0 model can be used to control multiple robot embodiments with differing action spaces to accomplish a wide variety of tasks.

Black et al. π_0: A Vision-Language-Action Flow Model for General Robot Control, arXiv, 2024

Vision-Language-Action Flow Model π_0

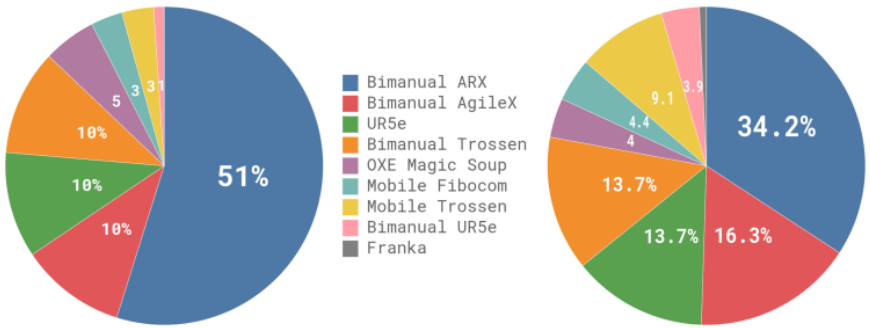


Fig. 4: **Overview of our dataset**: The pre-training mixture consists of a subset of OXE [10] and the π dataset. We use a subset of OXE, which we refer to as OXE Magic Soup [24]. The right figure illustrates the weight of the different datasets in the pre-training mixture. The left figure illustrates their relative sizes as measured by the number of steps.

Fig. 5: **The robots used in our experiments.** These include single and dual-arm manipulators with 6-DoF and 7-DoF arms, as well as holonomic and nonholonomic mobile manipulators. π_0 is trained jointly on all of these platforms.

Black et al. π_0: A Vision-Language-Action Flow Model for General Robot Control, arXiv, 2024

Vision-Language-Action Flow Model π_0

Limitation:

Δ Understanding what type of data is more helpful to add and how it should be weighted remains an open problem.

Δ It remains unclear how to predict how much and what kind of data is needed to attain near-perfect performance.

Δ It remains to be seen how much positive transfer there is in combining highly diverse data, particularly from different tasks and different robots.

Black et al. π_0: A Vision-Language-Action Flow Model for General Robot Control, arXiv, 2024

Outline

1. Vision-Language-Action Models
   - RT-2
   - OpenVLA
   - π_0
   - UniVLA
2. 3D Vision-Language-Action Models
3. Dataset

Unified Vision-Language-Action Model (**UniVLA**)

Motivation:

- Most existing VLA approaches follow a language-centric paradigm: visual observations are first projected into a semantic space, and action policies are subsequently derived based on these representations. *This late-fusion strategy limits the formation of deeply coupled cross-modal representations and impedes the learning of temporal and causal dependencies across the perception-action loop*.
- *Can vision, language, and action be jointly modeled within a unified representation space* to facilitate tighter cross-modal integration and more effective policy learning?

Wang et al. Unified Vision-Language-Action Model, ICLR, 2026

Unified Vision-Language-Action Model (**UniVLA**)

Approaches: `https://github.com/baaivision/UniVLA`

Δ Unified token representation. At the modality level, vision, language, and action signals are all transformed into discrete tokens and modeled using a shared vocabulary.

Δ An autoregressive Markov chain-based sequence modeling approach, where observations and actions are interleaved.

Δ Integrating the world model paradigm during training and leveraging large-scale robotics videos for self-supervised learning, allowing the model to capture environment dynamics in a temporally consistent and causally grounded manner.

Wang et al. Unified Vision-Language-Action Model, ICLR, 2026

Unified Vision-Language-Action Model (**UniVLA**)

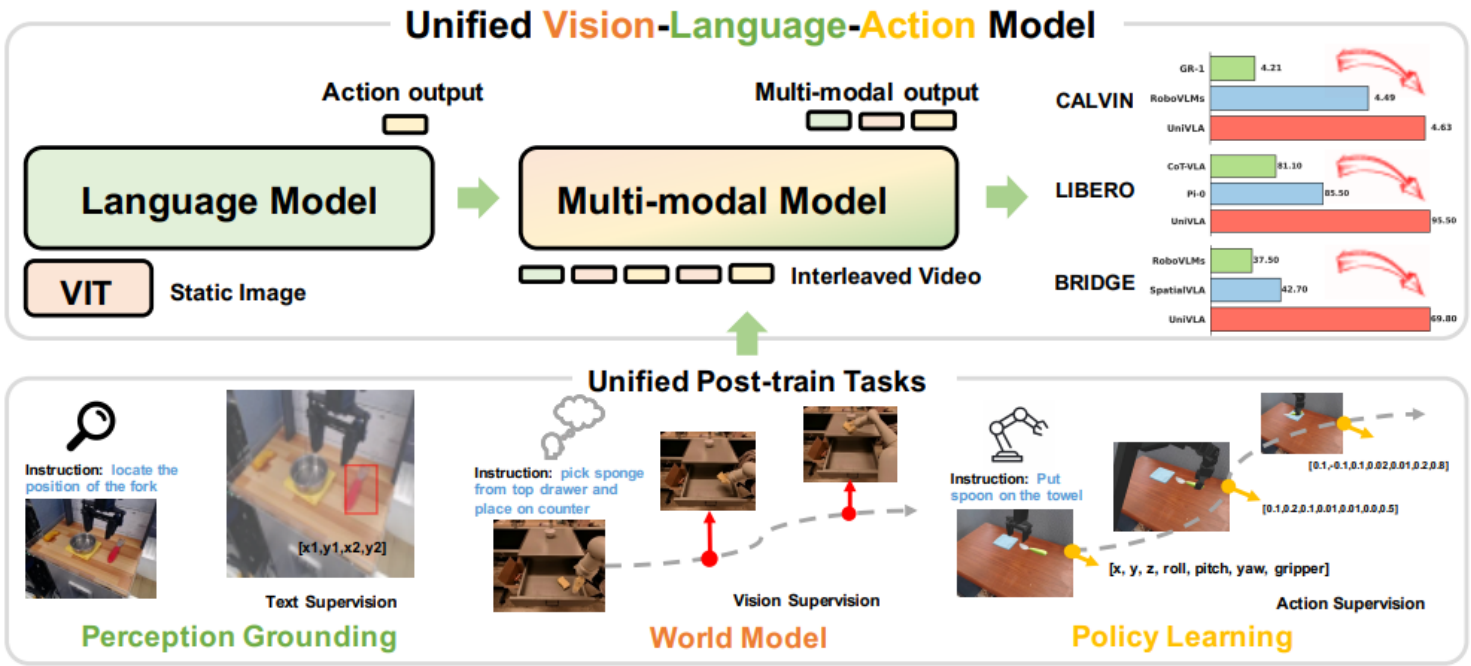


Figure 1: **We present UniVLA, a unified vision-language-action model.** Unlike prior VLA approaches that typically rely on an extra vision encoder to extract image features and generate only action outputs, UniVLA represents vision, language, and action as discrete tokens within a unified autoregressive framework. This unified modeling paradigm enables multi-modal outputs and supports a wide range of tasks—such as text-supervised perception grounding, vision-supervised world modeling, and action-supervised policy learning—within a single architecture. The unified token-based design further allows UniVLA to effectively leverage large-scale multimodal data, particularly video, for scalable and generalizable learning. UniVLA achieves new state-of-the-art results on CALVIN, LIBERO, and SimplerEnv-Bridge, significantly surpassing existing methods.

Wang et al. Unified Vision-Language-Action Model, ICLR, 2026

Unified Vision-Language-Action Model (**UniVLA**)

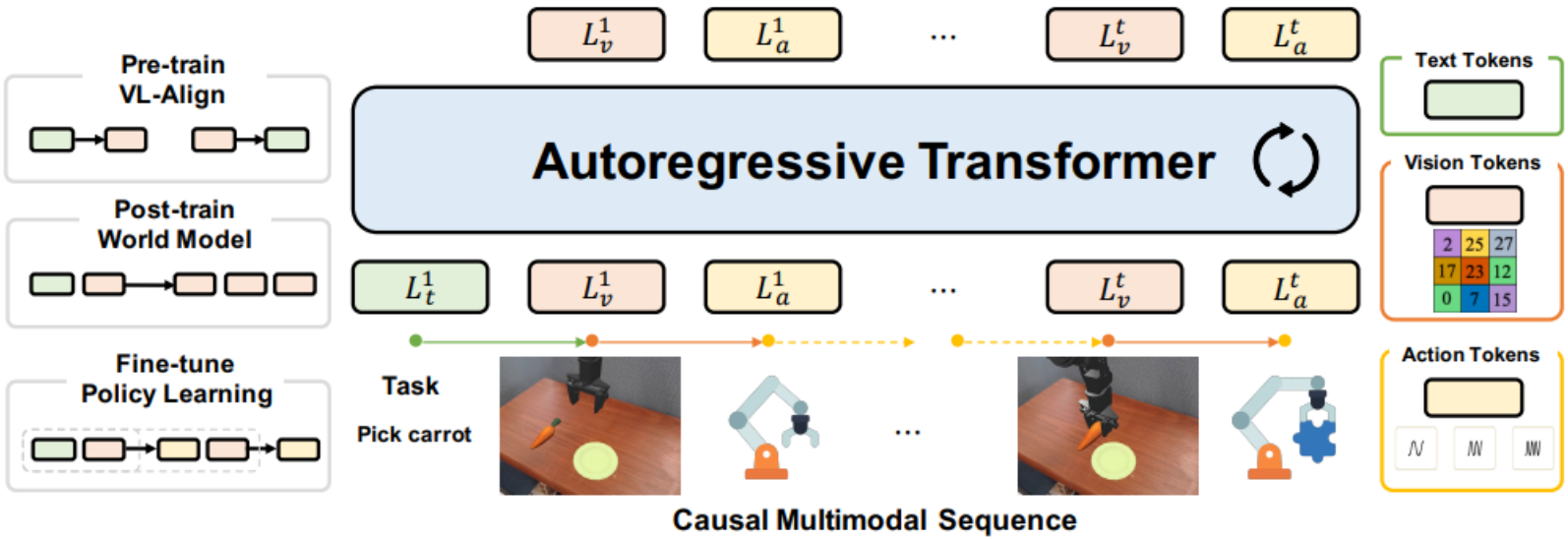


Figure 2: **Overview of the UniVLA framework.** Our model unifies information from different modalities into a discrete interleaved sequence, which is modeled using an autoregressive Transformer. To enable unified modeling, images are discretized using vector-quantized (VQ) encoders, while actions are transformed into the frequency domain and discretized via Discrete Cosine Transform (DCT) encoding. This causal multimodal sequence naturally preserves the temporal dynamics and causality required for real-world tasks. The model builds upon a pretrained vision-language model and follows a two-stage training strategy: (1) a post-training phase that adopts world-model training on large-scale datasets without requiring actions, and (2) a fine-tuning phase that interleaves actions into the sequence, enabling policy learning on downstream tasks.

Wang et al. Unified Vision-Language-Action Model, ICLR, 2026

Unified Vision-Language-Action Model (**UniVLA**)

Perform full-parameter training for 50k steps using 32 A100 GPUs (40GB), which takes approximately 4–5 days.

Table 8: **Post-training datasets.**

Dataset	Source	Data Type	Raw Videos	Used Videos	Interval
RT-1 [9]	Real	Text, Video, Action	87212	84084	3
BridgeV2 [65]	Real	Text, Video, Action	60064	28083	5
DROID [40]	Real	Text, Video, Action	275997	145641	15
Kuka [39]	Real	Text, Video, Action	580392	100000	3
TOTO [79]	Real	Text, Video, Action	902	899	20
Taco Play [60]	Real	Text, Video, Action	3242	3242	5
FMB [51]	Real	Text, Video, Action	8611	7876	5
Berkeley autolab ur5 [13]	Real	Text, Video, Action	896	896	5
VIOLA [81]	Real	Text, Video, Action	135	135	15
Cmu Play Fusion [14]	Real	Text, Video, Action	576	576	10
Utaustin Mutex [61]	Real	Text, Video, Action	1500	1500	10
CALVIN [54]	Sim	Text, Video, Action	22966	22966	5
LIBERO [49]	Sim	Text, Video, Action	3386	3386	10
ManiSkill2 [29]	Sim	Text, Video, Action	30213	193273	10
SSV2 [28]	Real	Text, Video	220847	220847	1

Wang et al. Unified Vision-Language-Action Model, ICLR, 2026

Unified Vision-Language-Action Model (**UniVLA**)

Limitations:

Δ While the unified multimodal framework exhibits strong capabilities in cross-modal learning, further research is needed to fully integrate it with reinforcement learning paradigms, enabling more robust and adaptive policy learning.

Wang et al. Unified Vision-Language-Action Model, ICLR, 2026

Outline

1 Vision-Language-Action Models

2 3D Vision-Language-Action Models

3 Dataset

Outline

1. Vision-Language-Action Models
2. 3D Vision-Language-Action Models
   - SpatialVLA
   - 3D-VLA
3. Dataset

Spatial Representations for VLA (SpatialVLA)

Motivation:

- Existing VLA models are primarily confined to 2D observation inputs and lack precise perception and comprehension of the 3D physical world. How to effectively equip the VLA models with a profound spatial understanding of the 3D physical world?
- The observations from different robot embodiments are not 3D-aligned, because the camera sensors of different robots are various and mounted at different places, resulting in non-calibrated 3D observation spaces.
- Different robots have different action movement characteristics to accomplish diverse tasks, due to different degrees of freedom, motion controllers, workspace configurations, and task complexity, leading to significant difficulty in learning generalizable spatial actions.

Qu et al. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model, arXiv, 2025

Spatial Representations for VLA (SpatialVLA)

Approaches: `https://github.com/SpatialVLA/SpatialVLA`

Δ SpatialVLA perceives 3D world through Egocentric 3D (Ego3D) Position Encoding to integrate 3D spatial context with semantic features.

Δ SpatialVLA unifies the action space of various robots via Adaptive Action Grids, which discretizes the continuous robot actions into adaptive spatial grids according to statistical action distributions on the whole robot episodes and learns spatial action tokens on these grids to align cross-robot actions with the 3D spatial structure of the physical world.

Qu et al. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model, arXiv, 2025

Spatial Representations for VLA (SpatialVLA)

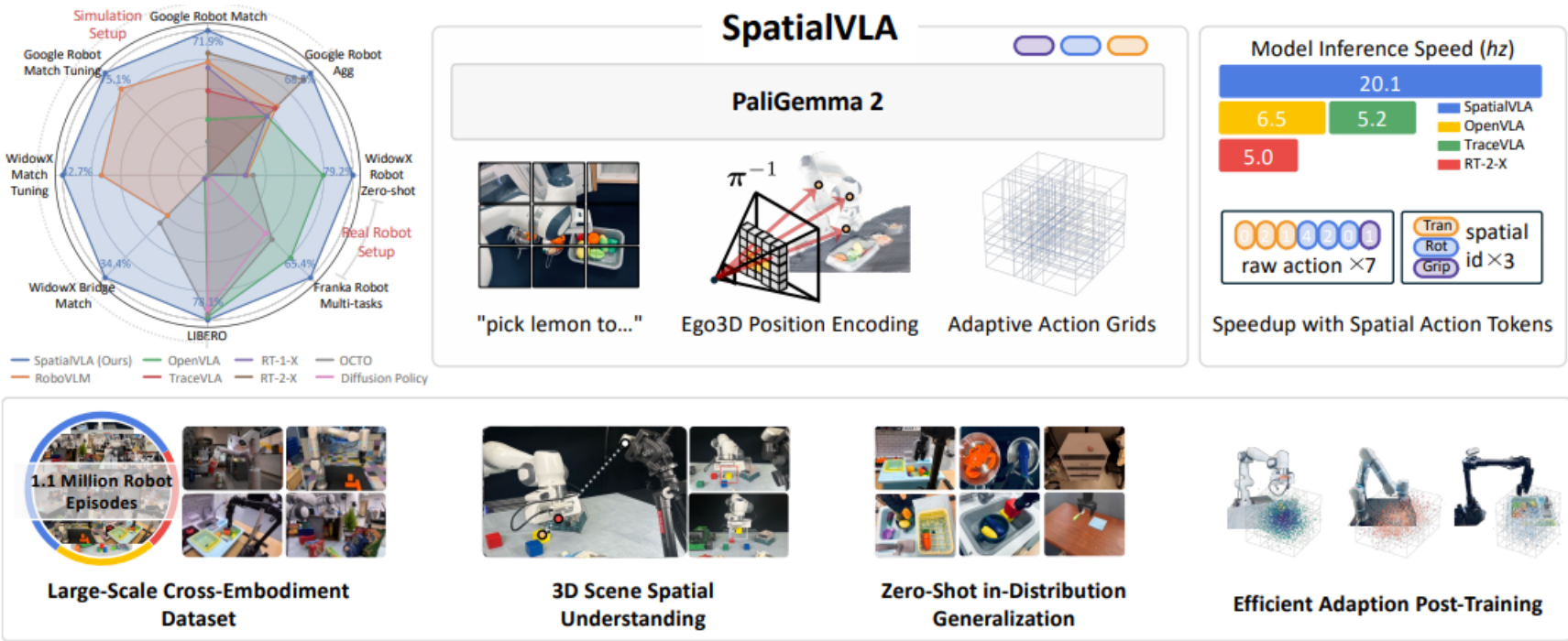


Fig. 1: We present SpatialVLA, a spatial-enhanced vision-language-action model that is trained on 1.1 Million real robot episodes. The model is equipped with Ego3D Position Encoding and Adaptive Action Grids to explore spatial representations for generalist robot policies, achieving superior 3D scene spatial understanding, zero-shot in-distribution generalization, and efficient adaption to new robot setups. The model achieves state-of-the-art performance across a diverse range of evaluations and shows significantly faster inference speed with fewer tokens per action.

Qu et al. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model, arXiv, 2025

Spatial Representations for VLA (SpatialVLA)

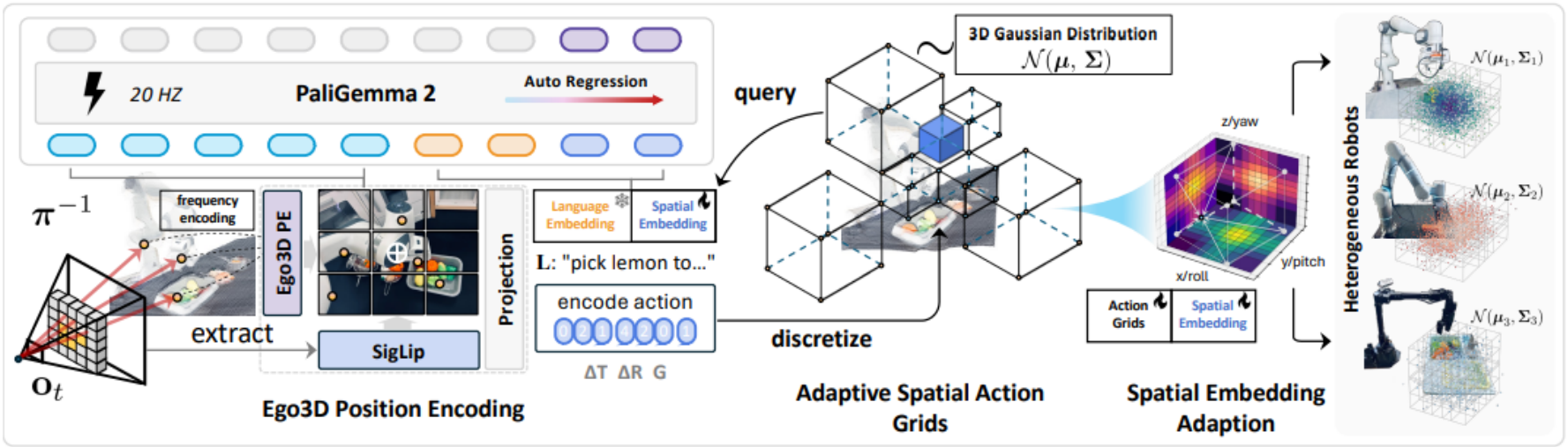


Fig. 2: **Overview of SpatialVLA**. Given an image observation $\mathbf{o}_t$ and a task instruction $\mathbf{L}$, the model processes the image using Ego3D Position Encoding and auto-regressively predicts spatial action tokens, which are then de-tokenized to generate continuous actions $\mathbf{A}_t$ for robot control. The model comprises three key components: (1) SigLIP vision encoder extracts 2D semantic features, which are then infused with 3D spatial context via Ego3D Position Encoding; (2) continuous 7D actions $\Delta\mathbf{T}, \Delta\mathbf{R}, \mathbf{G}$ are translated to 3 spatial action tokens by querying Adaptive Action Grids and auto-regressively predicted and de-tokenized for robot control; (3) in post-training, action grids and spatial embeddings are adapted from new Gaussian distributions to facilitate effective transfer to new robot setups.

Qu et al. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model, arXiv, 2025

Spatial Representations for VLA (SpatialVLA)

SpatialVLA is pretrained with 1.1 Million real-robot demonstrations from the OXE and RH20T dataset on a cluster of 64 A100 GPUs for 10 days, using a batch size of 2048.

TABLE I: **SimplerEnv evaluation across different policies on Google Robot tasks**. The zero-shot and fine-tuning results denote performance of OXE dataset [13] pre-trained models and Fractal dataset [6] fine-tuned models, respectively.

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 [65]	56.0%	60.0%	24.0%	46.0%	———	———	———	———
TraceVLA [71]	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 [48]	17.0%	4.2%	22.7%	16.8%	0.6%	3.1%	1.1%	1.1%
OpenVLA [30]	16.3%	46.2%	35.6%	27.7%	54.5%	47.7%	17.7%	39.8%
RoboVLM (zero-shot) [32]	72.7%	66.3%	26.8%	56.3%	68.3%	56.0%	8.5%	46.3%
RoboVLM (fine-tuning) [32]	77.3%	61.7%	43.5%	63.4%	75.6%	60.0%	10.6%	51.3%
π_0^* (BF16 uniform) [5]	88.0%	80.3%	56.0%	70.1%				
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%**

π_0^*: The results are referred from open-pi-zero.

TABLE II: **SimplerEnv evaluation across different policies on WidowX Robot tasks**. The zero-shot and fine-tuning results denote the performance of OXE dataset [13] pre-trained models and BridgeData V2 [64] fine-tuned models, respectively.

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 [48]	34.7%	12.5%	52.8%	8.3%	31.9%	0%	66.7%	43.1%	16.0%
Octo-Small [48]	77.8%	47.2%	27.8%	9.7%	40.3%	4.2%	87.5%	56.9%	30.0%
OpenVLA [30]	4.1%	0%	33.3%	0%	12.5%	0%	8.3%	4.1%	1.0%
RoboVLM (zero-shot) [32]	37.5%	20.8%	33.3%	25.0%	8.3%	8.3%	0.0%	0%	13.5%
RoboVLM (fine-tuning) [32]	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%**

Qu et al. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model, arXiv, 2025

Spatial Representations for VLA (SpatialVLA)

Limitation:

△ Gaussian modeling is suboptimal, as it can lead to grid clustering on specific coordinate axes in extreme robot operation scenarios, such as singleaxis motion, resulting in lost motion capabilities on other axes.

△ Although SpatialVLA achieves 21Hz inference speed, it is slower than diffusion decoding, which decodes tokens into multiple consecutive actions.

△ As the model relies solely on current frame observations and history tokens for action prediction, it faces challenges in long-horizon tasks.

△ SpatialVLA is pretrained on OXE and RH20T, but the variable quality of OXE data can hinder training.

Qu et al. SpatialVLA: Exploring Spatial Representations for Visual-Language-Action Model, arXiv, 2025

Outline

1. Vision-Language-Action Models
2. 3D Vision-Language-Action Models
   - SpatialVLA
   - 3D-VLA
3. Dataset

3D-VLA

Motivation:

- Human beings are situated within a far richer 3D physical world beyond 2D images - they reason, plan, and act based on their 3D understanding of the environment.
- Existing foundation models focus on language generation, unable to imagine modalities beyond language and simulate future states to facilitate action generation, which is a crucial aspect of world models.
- Existing embodied datasets mainly contain 2D images or videos, lacking 3D-related annotations for reasoning and planning in the 3D space.

Zhen et al. 3D-VLA: A 3D Vision-Language-Action Generative World Model, ICML, 2024

3D-VLA

Approaches: `https://github.com/UMass-Embodied-AGI/3D-VLA`

△ Build 3D-VLA on top of a 3D large language model to equip the model with 3D understanding ability.

△ First pretrain a set of embodied diffusion models for RGBD-to-RGBD and point-to-point generation respectively. To efficiently bridge between the diffusion decoders of various modalities and the LLM embedding space, employ a projector that aligns multi-modal goal generation in 3D-VLA.

△ Curate a large-scale 3D embodied instruction tuning dataset, collecting 2M 3D-language-action data pairs, covering various tasks such as task captioning, action prediction, localization, multimodal goal generation, etc.

Zhen et al. 3D-VLA: A 3D Vision-Language-Action Generative World Model, ICML, 2024

3D-VLA

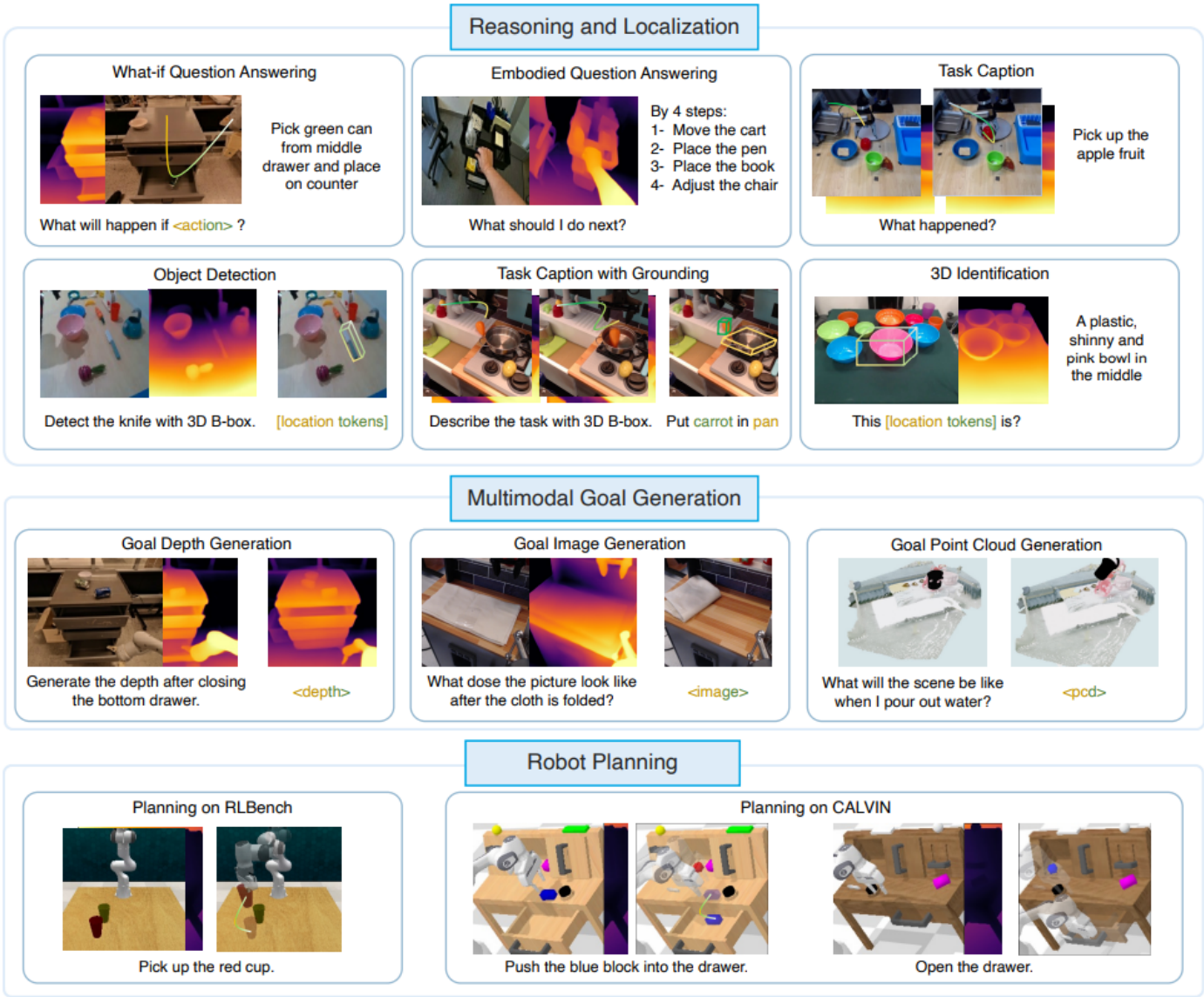


Figure 1. Examples from our 3D Embodied Instruction Tuning Dataset.

Zhen et al. 3D-VLA: A 3D Vision-Language-Action Generative World Model, ICML, 2024

3D-VLA

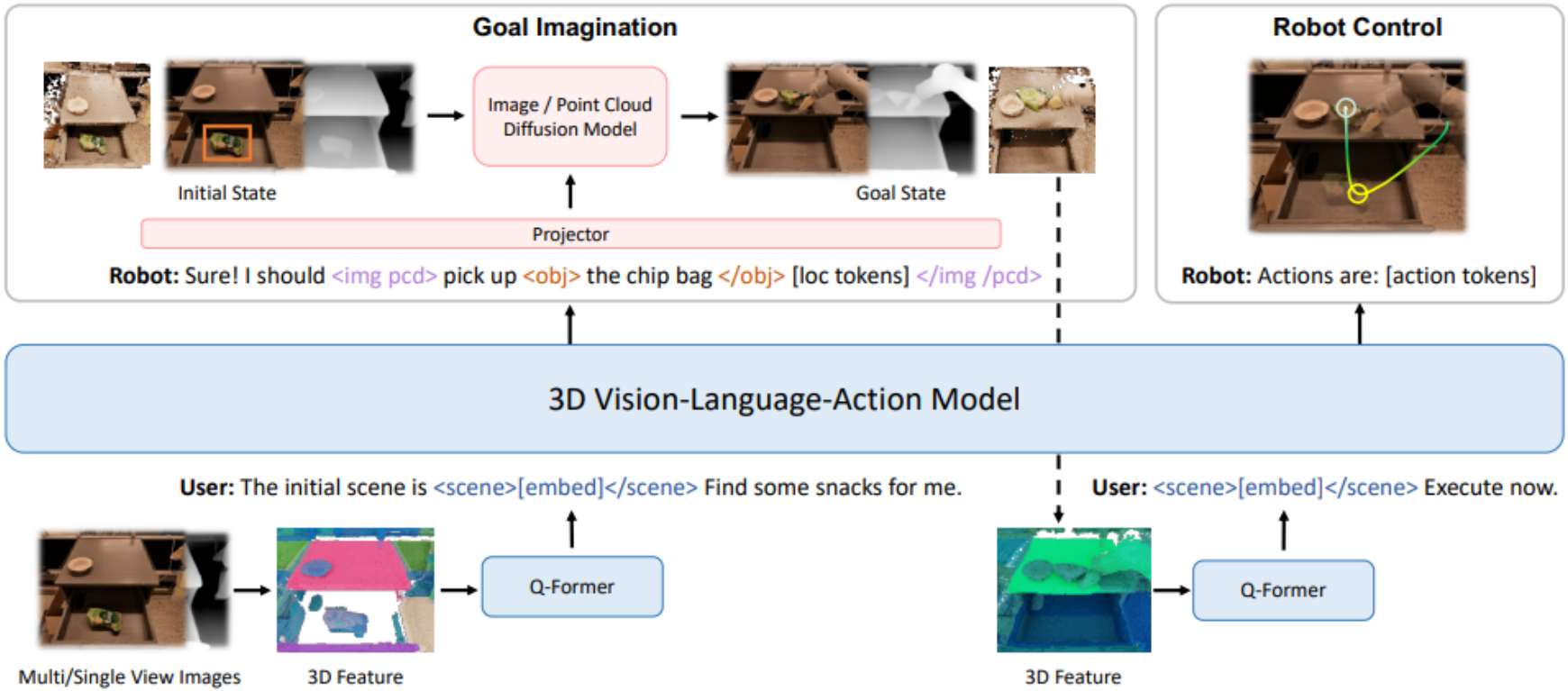


Figure 2. Overview of our 3D-VLA pipeline. The left part shows our goal-generation capability. Our model can imagine the final state image and point cloud based on the user's input. This generated goal state can then be fed back to our model to guide the robot control.

Zhen et al. 3D-VLA: A 3D Vision-Language-Action Generative World Model, ICML, 2024

3D-VLA

Pretraining: 30 epochs on 6 $\times$ 32 V100s, batch size 4 on each node. Alignment stage: a maximum of epochs of 20 on 6 $\times$ 64 V100s, batch size 2 on each node.

Tasks	Models	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGH-L	EM@1
Embodied QA	3D-LLM*	1.05	0.38	0.15	0.02	12.96	0.91	0.00
	BLIP2 $OPT_{2.7B}$*	7.39	3.17	0.03	0.02	3.87	7.40	3.03
	BLIP2 $FlanT5_{XL}$*	22.84	16.17	12.50	10.11	11.41	32.01	10.31
	$OpenFlamingo_{4B}$*	9.50	6.51	5.14	4.29	6.84	10.40	1.21
	$LLaVA_{7B}$*	11.66	8.06	6.01	4.58	12.59	14.17	5.67
	BLIP2 $FlanT5_{XL}$	37.31	27.20	20.32	15.48	17.80	38.92	15.35
	3D-VLA	**48.34**	**38.55**	**31.72**	**26.80**	**23.72**	**49.33**	**24.53**
Task Caption	3D-LLM*	0.78	0.16	0.07	0.05	0.57	1.33	0.00
	BLIP2 $FlanT5_{XL}$*	8.50	2.07	0.35	0.00	3.40	8.45	0.00
	$OpenFlamingo_{4B}$*	7.61	1.64	0.37	0.00	4.74	9.36	0.00
	$LLaVA_{7B}$*	2.63	0.69	0.16	0.00	2.63	4.65	0.00
	BLIP2 $FlanT5_{XL}$	22.05	11.40	5.72	3.16	8.72	26.12	7.75
	3D-VLA	**55.69**	**45.88**	**39.39**	**34.88**	**27.57**	**62.01**	**29.34**
What-if QA	BLIP2 $FlanT5_{XL}$	28.23	11.47	4.49	0.06	8.27	28.41	5.85
	3D-VLA	**53.09**	**40.94**	**34.34**	**29.38**	**26.83**	**52.82**	**14.7**
Dense Caption	3D-LLM*	0.52	0.22	0.16	0.13	0.34	0.64	0.00
	BLIP2 $FlanT5_{XL}$	36.17	24.72	18.06	13.96	17.83	40.56	13.10
	3D-VLA	**51.90**	**42.83**	**38.11**	**34.62**	**25.25**	**55.91**	**39.49**

Table 1. Evaluation on reasoning ability using held-in data. $*$ denotes zero-shot transfer results without training on our pre-train datasets.

Zhen et al. 3D-VLA: A 3D Vision-Language-Action Generative World Model, ICML, 2024

Outline

1 Vision-Language-Action Models

2 3D Vision-Language-Action Models

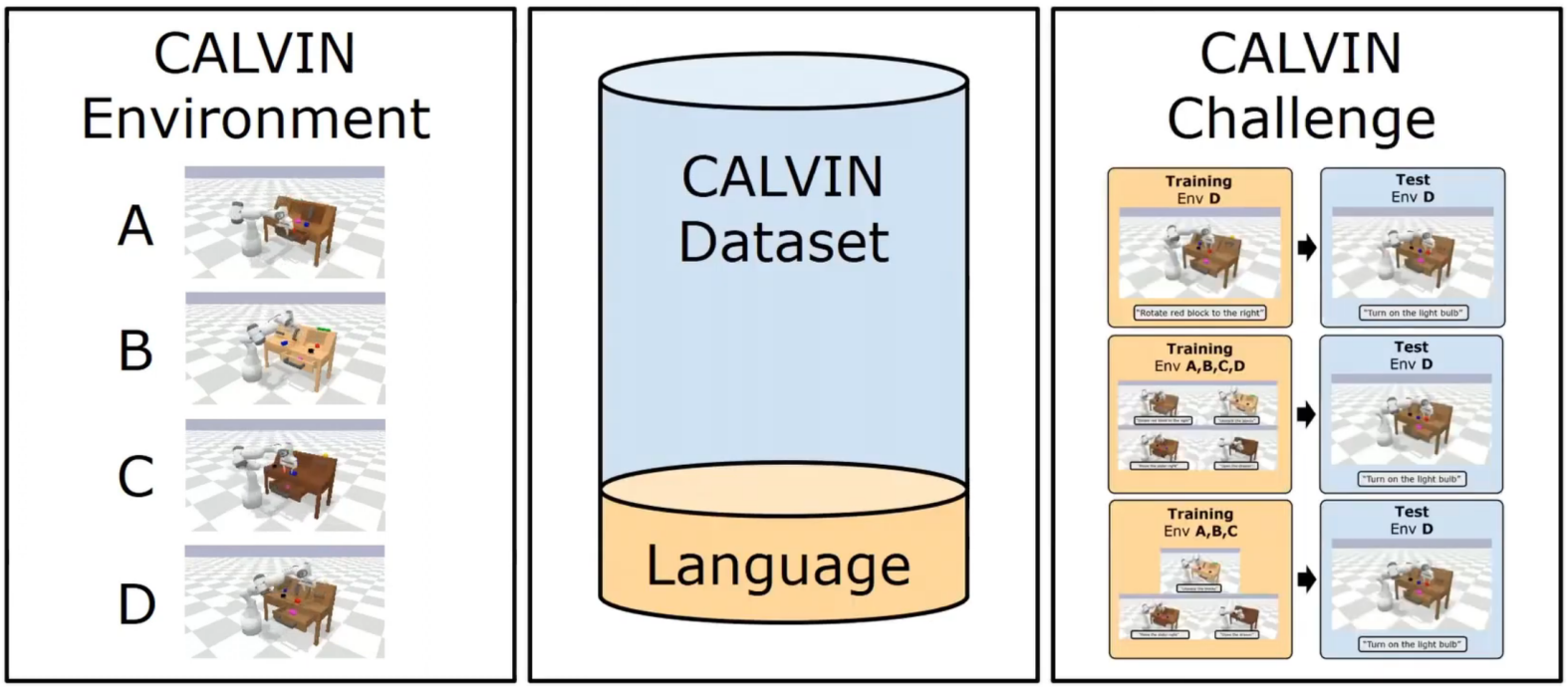
3 Dataset

Outline

1. Vision-Language-Action Models
2. 3D Vision-Language-Action Models
3. Dataset
   - CALVIN
   - LIBERO
   - SimplerEnv

CALVIN https://github.com/mees/calvin

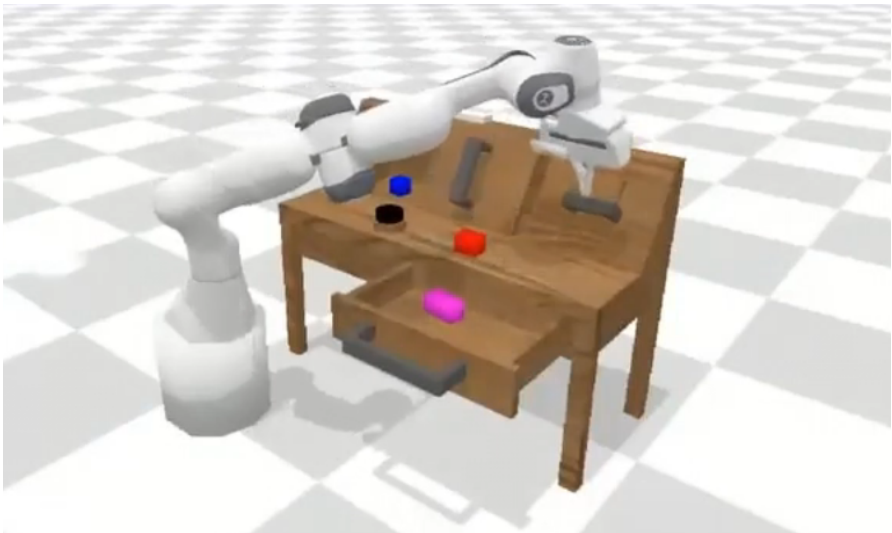
CALVIN is a simulated benchmark tailored for evaluating long-horizon, language conditioned robotic manipulation. It comprises four simulated environments (A, B, C, and D), each containing demonstration trajectories collected via human teleoperation. The benchmark encompasses 34 distinct manipulation tasks with a total of 1,000 unique language instructions.

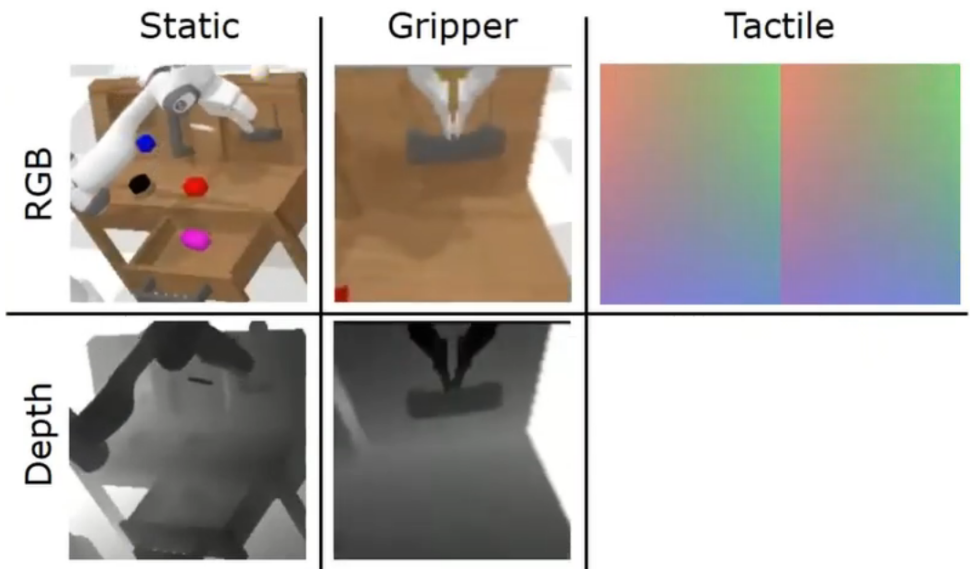


Mees et al. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks, arXiv, 2022

CALVIN https://github.com/mees/calvin

Multimodal observations with a wide range of sensors, including RGB and depth images from both the fixed and the gripper camera, and a vision-based tactile sensor.





Mees et al. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks, arXiv, 2022

CALVIN https://github.com/mees/calvin

Observation Space	
RGB static camera	$200 \times 200 \times 3$
Depth static camera	200×200
RGB gripper camera	$84 \times 84 \times 3$
Depth gripper camera	84×84
Tactile image	$120 \times 160 \times 2$
Proprioceptive state	EE position (3) EE orientation (3) Gripper width (1) Joint positions (7) Gripper action (1)
Action Space	
Absolute cartesian pose (w.r.t. world frame)	EE position (3) EE orientation (3) Gripper action (1)
Relative cartesian displacement (w.r.t. gripper frame)	EE position (3) EE orientation (3) Gripper action (1)
Joint action	Joint positions (7) Gripper action (1)

Task	Natural language instructions
rotate red block right	"rotate the red block 90 degrees to the right" "turn the red block right"
push blue block left	"go slide the blue block to the left" "push left the blue block"
move slider left	"grasp the door handle, then slide the door to the left" "slide the door to the left"
open drawer	"grasp the handle of the drawer and open it" "go open the drawer"
lift red block	"lift the red block from the table" "pick up the red block"
pick pink block from drawer	"pick up the pink block lying in the drawer"
place in slider	"put the grasped object in the slider"
stack blocks	"stack blocks on top of each other"
unstack blocks	"collapse the stacked blocks" "go to the tower of blocks and take off the top one"
turn on light bulb	"toggle the light switch to turn on the light bulb"
turn off green light	"push the button to turn off the green light"

Outline

1. Vision-Language-Action Models
2. 3D Vision-Language-Action Models
3. Dataset
   - CALVIN
   - LIBERO
   - SimplerEnv

LIBERO https://github.com/Lifelong-Robot-Learning/LIBERO

The LIBERO benchmark is a comprehensive suite for lifelong robotic manipulation, comprising four task suites with 10 tasks and 50 human demonstrations each. These suites are designed to evaluate different generalization abilities: LIBERO-Spatial tests spatial reasoning by varying layouts with fixed objects; LIBERO-Object assesses object-level generalization with varying objects in a fixed scene; LIBERO-Goal targets goal-conditioned behavior by varying task goals; and LIBERO-Long (LIBERO-10) features long-horizon, compositional tasks with diverse objects, layouts, and goals, challenging temporal and compositional reasoning.

Mees et al. CALVIN: A Benchmark for Language-Conditioned Policy Learning for Long-Horizon Robot Manipulation Tasks, NeurIPS, 2023

LIBERO https://github.com/Lifelong-Robot-Learning/LIBERO

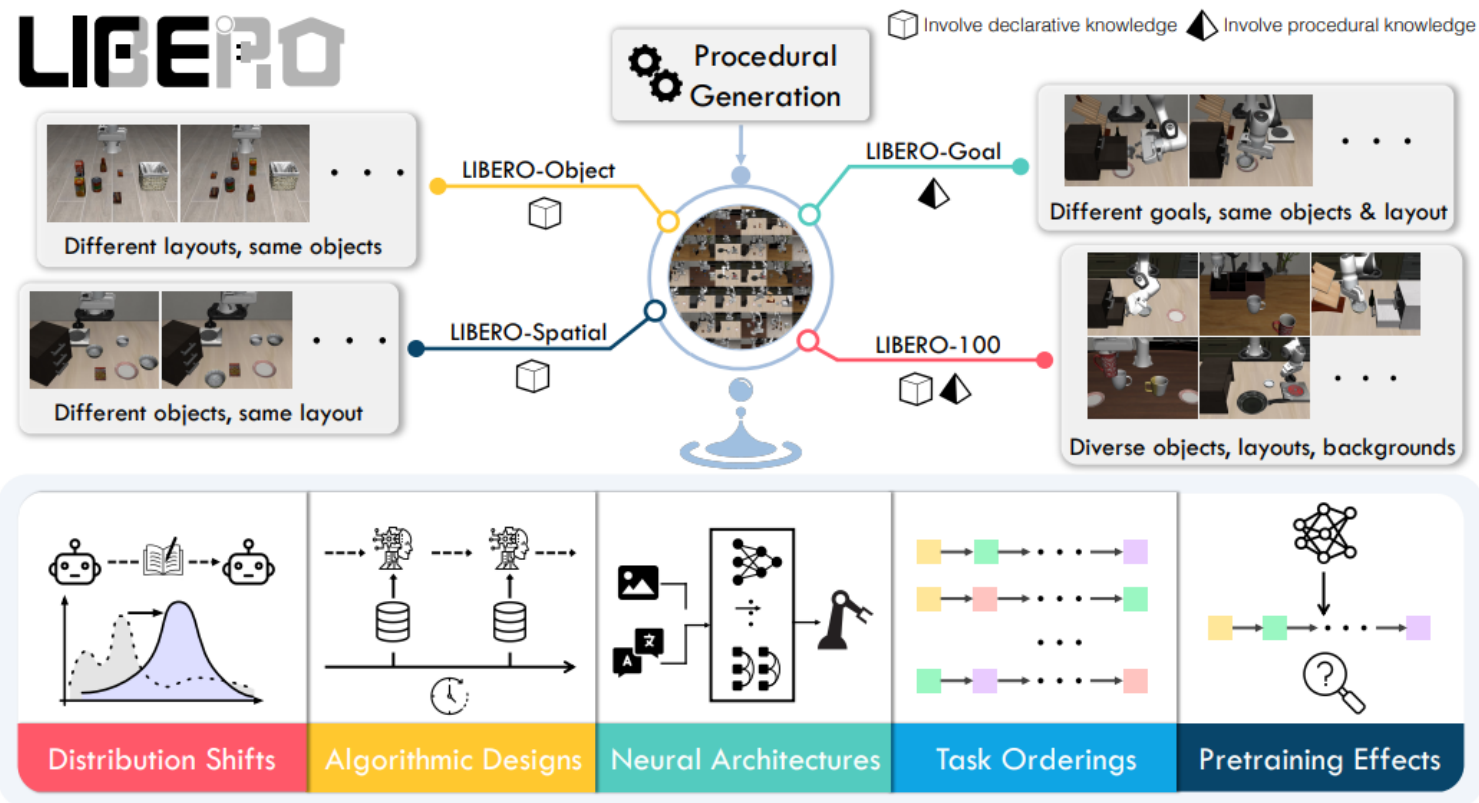


Figure 1: **Top**: LIBERO has four procedurally-generated task suites: LIBERO-SPATIAL, LIBERO-OBJECT, and LIBERO-GOAL have 10 tasks each and require transferring knowledge about spatial relationships, objects, and task goals; LIBERO-100 has 100 tasks and requires the transfer of entangled knowledge. **Bottom**: we investigate five key research topics in LLDM on LIBERO.

Liu et al. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning, NeurIPS, 2023

LIBERO https://github.com/Lifelong-Robot-Learning/LIBERO

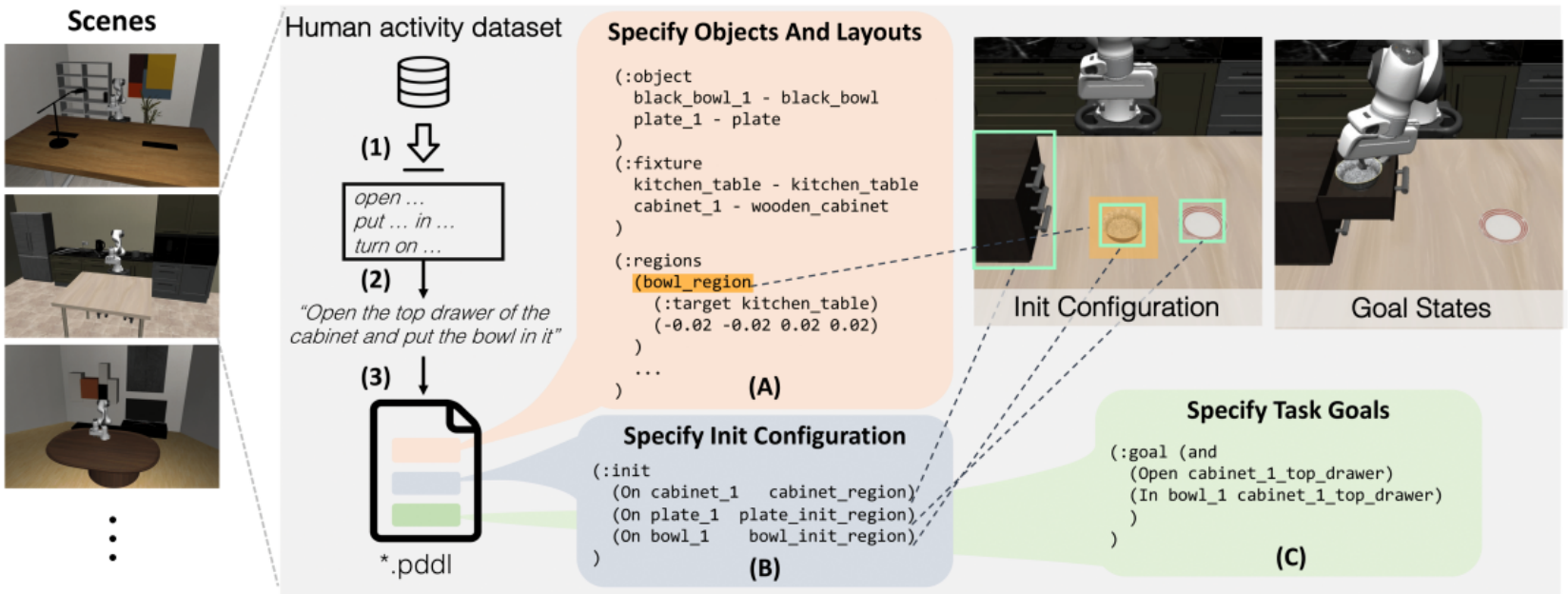


Figure 2: LIBERO's procedural generation pipeline: Extracting behavioral templates from a large-scale human activity dataset **(1)**, Ego4D, for generating task instructions **(2)**; Based on the task description, selecting the scene and generating the PDDL description file **(3)** that specifies the objects and layouts **(A)**, the initial object configurations **(B)**, and the task goal **(C)**.

Liu et al. LIBERO: Benchmarking Knowledge Transfer for Lifelong Robot Learning, NeurIPS, 2023

Outline

1. Vision-Language-Action Models
2. 3D Vision-Language-Action Models
3. Dataset
   - CALVIN
   - LIBERO
   - SimplerEnv

SimplerEnv

https://github.com/simpler-env/SimplerEnv

SimplerEnv serves as a simulation benchmark designed to evaluate the transferability and generalization capabilities of models trained on real-world video data. It incorporates diverse manipulation setups across both the WidowX and Google Robot platforms, encompassing variations in lighting conditions, object textures, color distributions, and camera viewpoints.

Li et al. Evaluating Real-World Robot Manipulation Policies in Simulation, CoRL, 2024

SimplerEnv

https://github.com/simpler-env/SimplerEnv

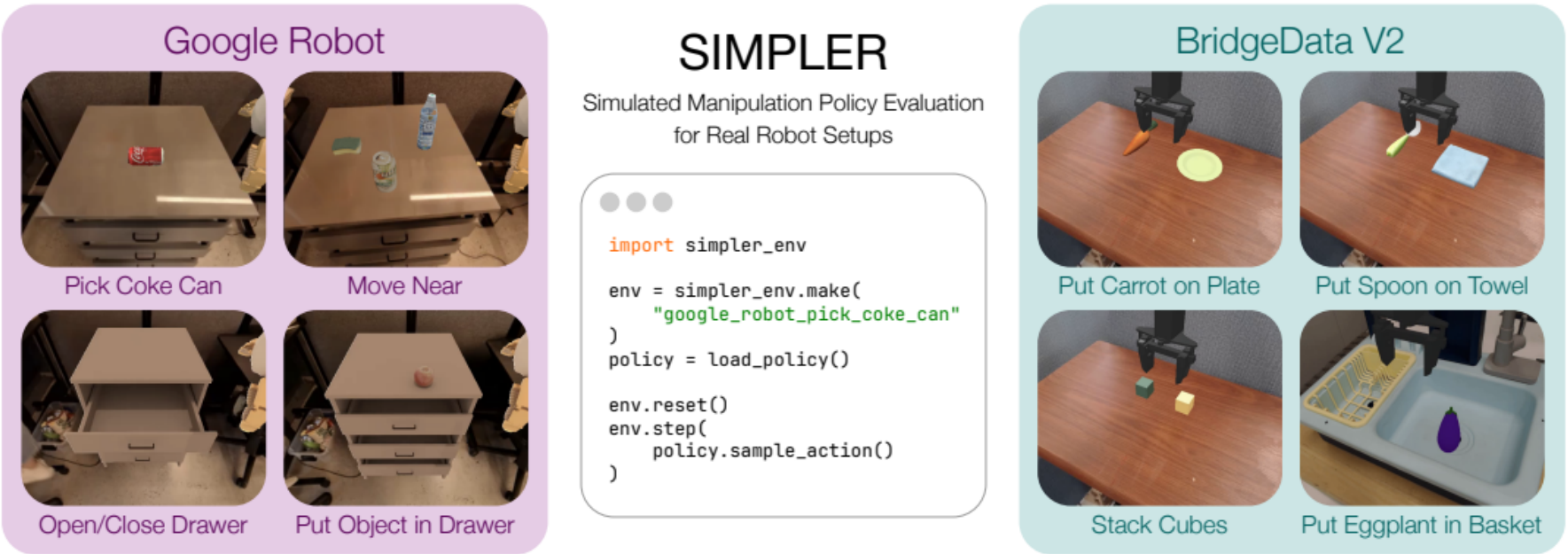


Fig. 2: We introduce SIMPLER, a suite of open-source simulated evaluation environments for common real robot manipulation setups, namely the Google Robot evaluations from the RT-series of works [6, 5, 11], and environments from the BridgeData V2 dataset [66]. All environments can be imported with a single line of code and can be interacted with through a standard Gym interface. Additionally, we open-source policy inference code for real-to-sim evaluation of common generalist robot policies (RT-1 [6], RT-1-X [11], and Octo [50]), and we provide detailed guides for evaluating new policies and creating new evaluation environments. All materials are available at https://simpler-env.github.io.

Li et al. Evaluating Real-World Robot Manipulation Policies in Simulation, CoRL, 2024