Embodied AI for Humanoid Robots

Presented by: Fukang Liu, Yilin Cai, Sizhe Wei

Nov 11, 2025

Georgia Tech

2025-3-28

GR00T N1: An Open Foundation Model for Generalist Humanoid Robots

NVIDIA[1]

Why humanoid robot?

A Timeline

"The Robot Dreams"

Rossum's Universal Robots
(1920)

"The Visionary Prototypes"

Honda Asimo
(2000)

Sony QRIO
(2003)

"The Reality Check"

DARPA Robotics Challenge
(2013-2015)

"The Initial Attempts"

WABOT-1
(1970-1973)

WABOT-2
(1980-1984)

"The Social Companions"

NAO
(2008)

Pepper
(2014)

"The Cambrian Explosion"

Recent Humanoid Wave
(circa 2022-Now)

[Source: https://www.youtube.com/watch?v=49LnlfM9DBU]

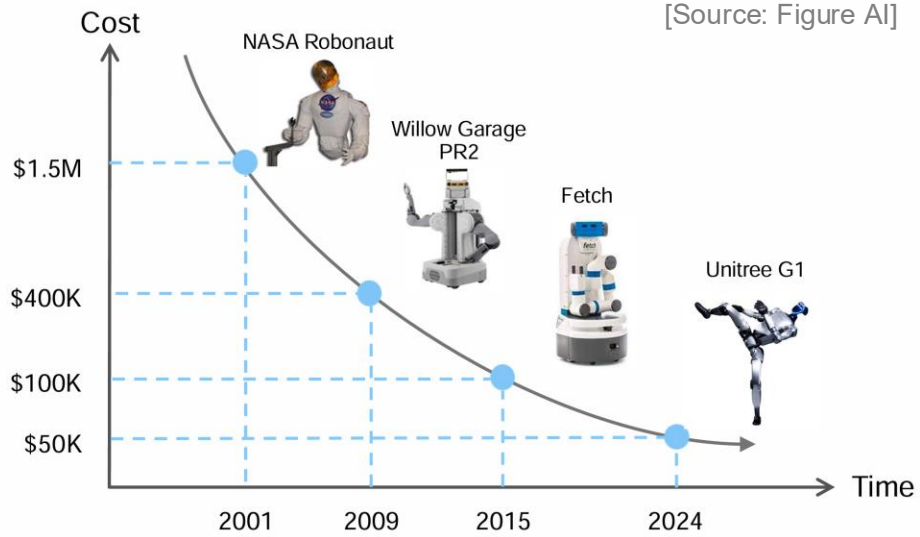
Georgia Tech

Why humanoid robot?

- **Versatility**: General-purpose robot autonomy needs a versatile body
- **Brownfield**: Human-like morphology. Humanoids can seamlessly integrate into human world infrastructure without modifying existing environment.
- **Hardware**: Robot hardware gets cheaper and more robust, democratizing transformative research
- **Impact**: Aging workforces, shrinking labor pools ….

Final goal:
March toward human-level physical intelligence

[Source: Figure AI]

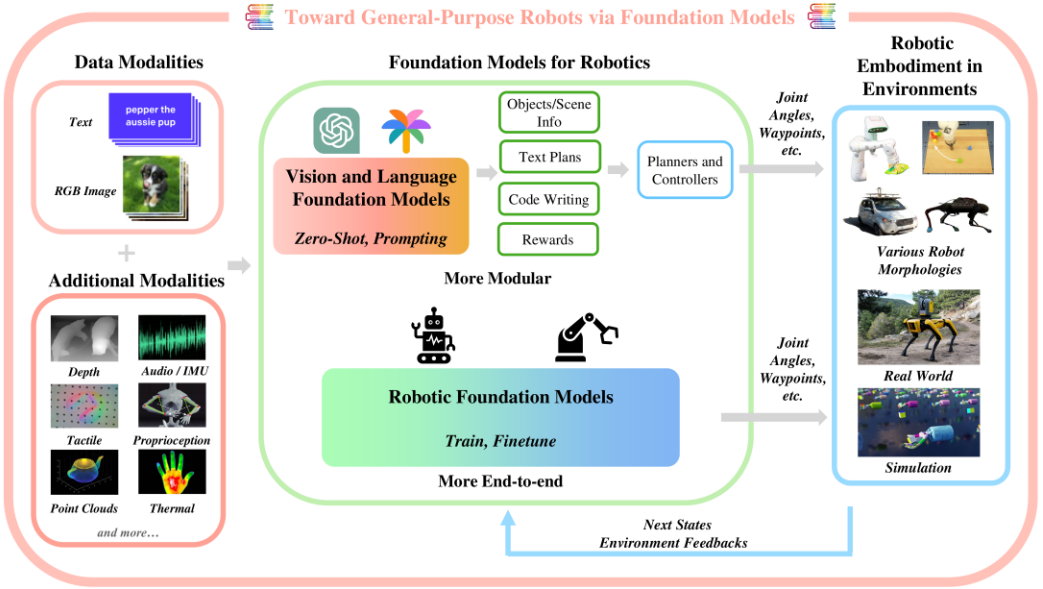


[Source: https://www.youtube.com/watch?v=49LnlfM9DBU]

Problem Statement

Goal of Foundation Models in Robotics

- **Generalist intelligence backbone** for robots, enabling them to ***understand, reason, and act*** across diverse ***tasks, environments, and robot embodiments*** using a single unified model



Hu, Yafei, et al. "Toward general-purpose robots via foundation models: A survey and meta-analysis." *arXiv preprint arXiv:2312.08782* (2023).

Challenges

- Unlike pixels or text, **robotic data isn't abundant or uniform** — every robot has unique:
  - Embodiment (morphology, kinematics)
  - Sensors (cameras, proprioception)
  - Control spaces (joints, tendons, torques)
- **No** Internet of humanoid robot **dataset exist** for large-scale pre-training, leading to **"data islands"**
- As a result, models trained on one robot often **fail to generalize** to others or to new physical tasks.

Georgia Tech

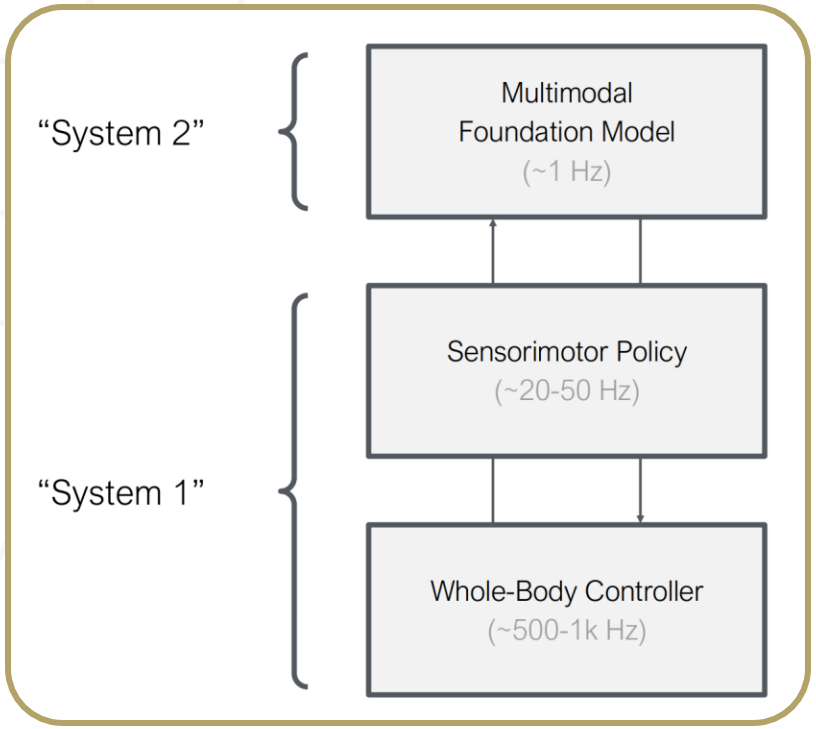
Problem Statement

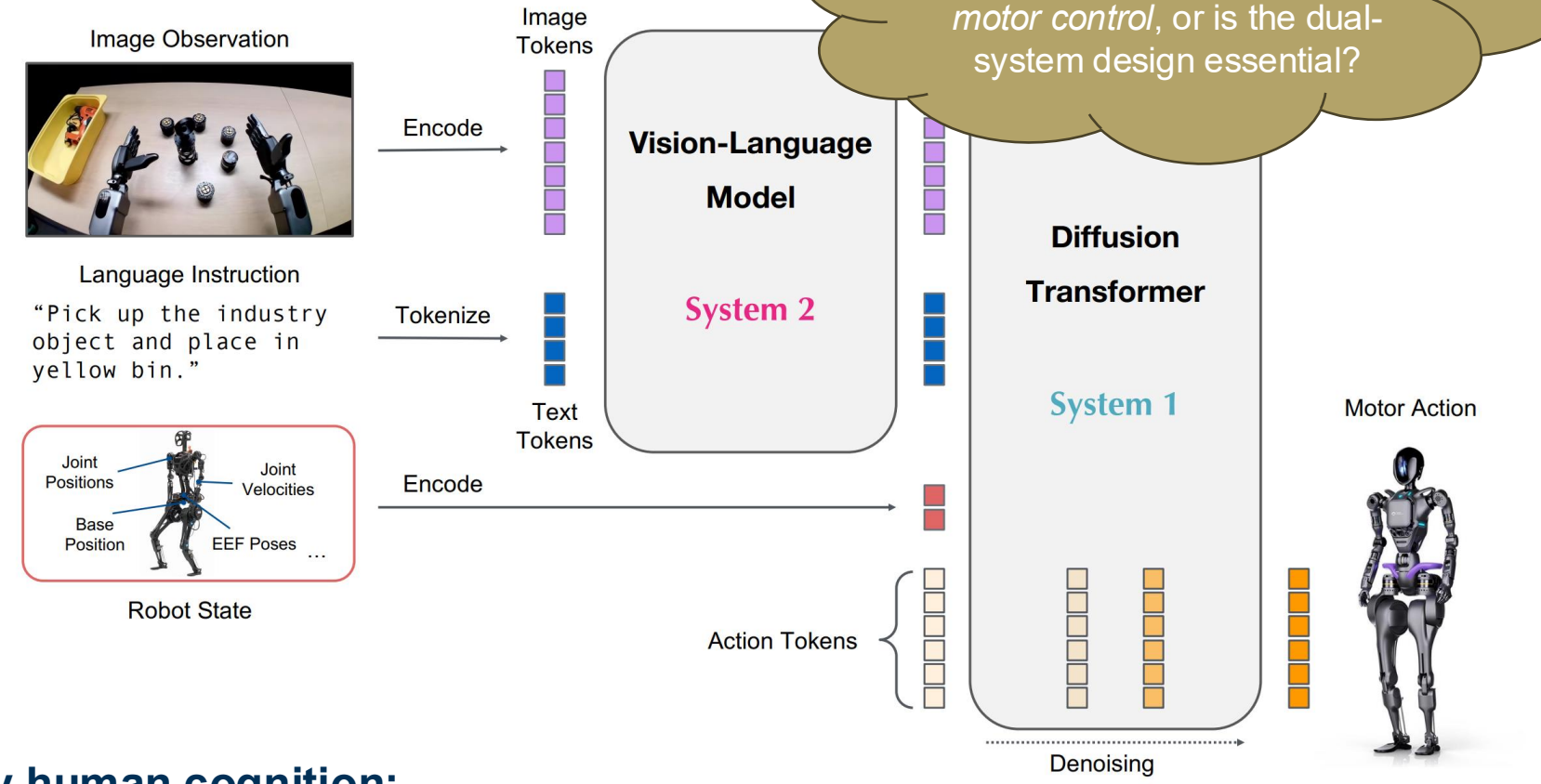
Challenges for humanoid robots specifically

- High degrees of freedom and bimanual coordination.
- Dynamic balance and whole-body motion couple perception, locomotion, and manipulation — hard to do whole-body teleoperation.
- Extensive cost and human effort in teleoperation-based data collection
- Embodiment variability (different limb proportions, motor torque limits, control modes, and joint ranges) breaks direct policy transfer between humanoid platforms.

GR00t: Method

Model Architecture





Dual-system architecture inspired by human cognition:

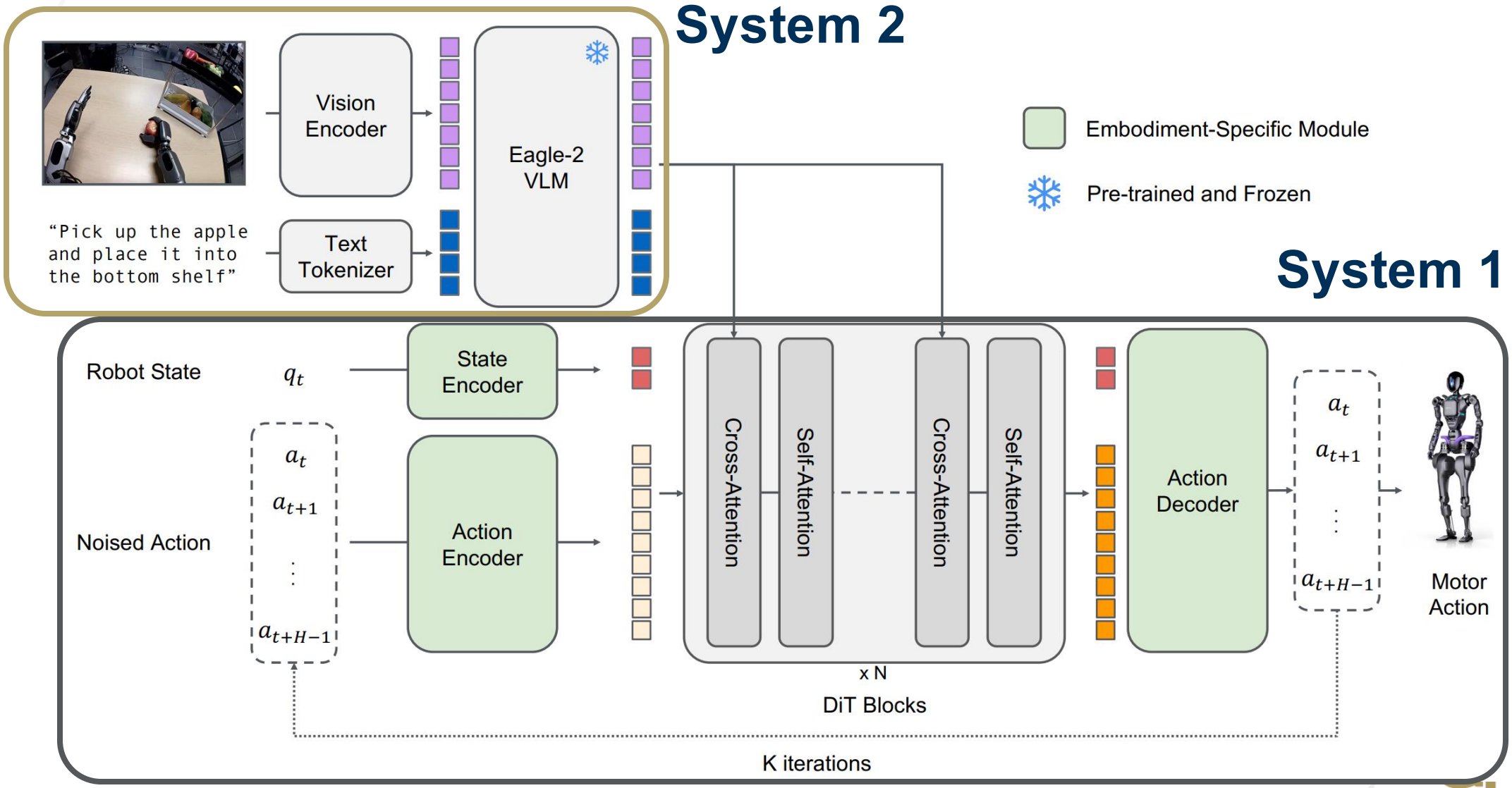
- System 2 (Reasoning): Vision-Language Model (VLM) interprets the environment and task.
- System 1 (Action): Diffusion Transformer generates continuous motor actions.

Both are jointly trained end-to-end for coordinated perception and control.

Input: Image + Language + Robot State → Output: Motor Actions.

Georgia Tech

Model Architecture

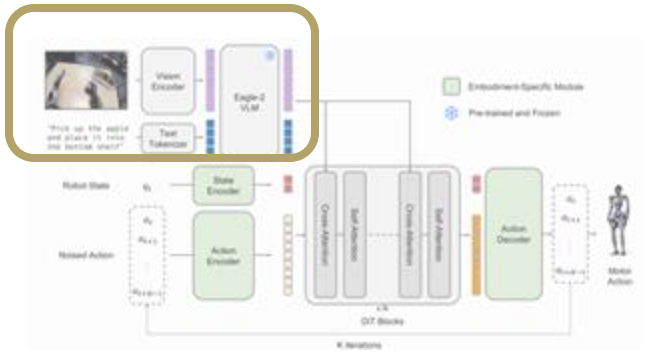


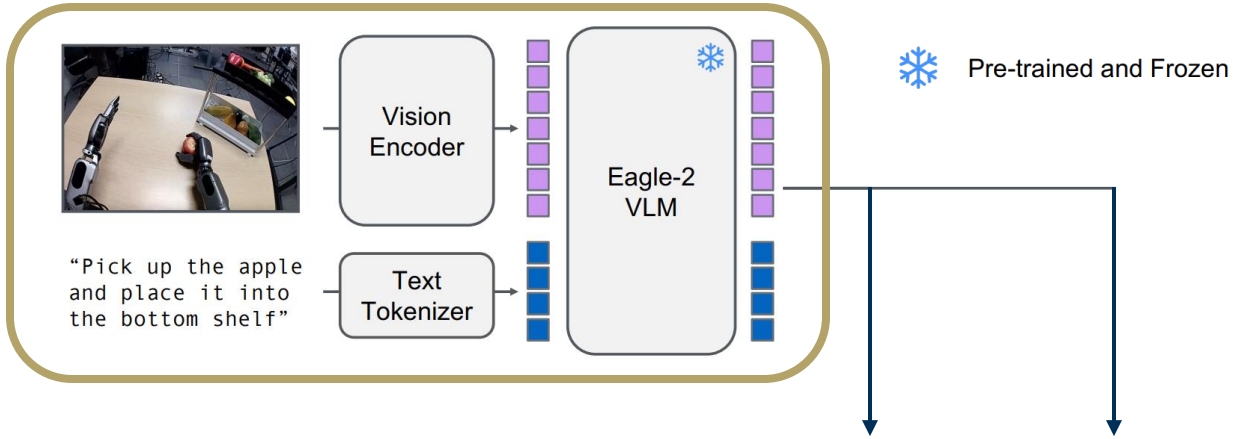
Georgia Tech

Model Architecture

System 2 – Vision-Language Module

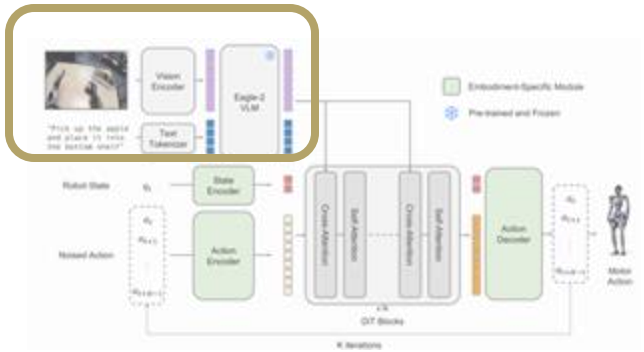
- Based on **Eagle-2 VLM** (a fusion of SigLIP-2 image encoder + SmolLM2 LLM).
- Encodes **images and task text** into shared feature tokens.
- Produces **vision-language embeddings** (middle-layer representations used for efficiency and accuracy).
- Operates at **10 Hz** for task understanding and reasoning.
- Output: High-level task/context representation → passed to System 1.





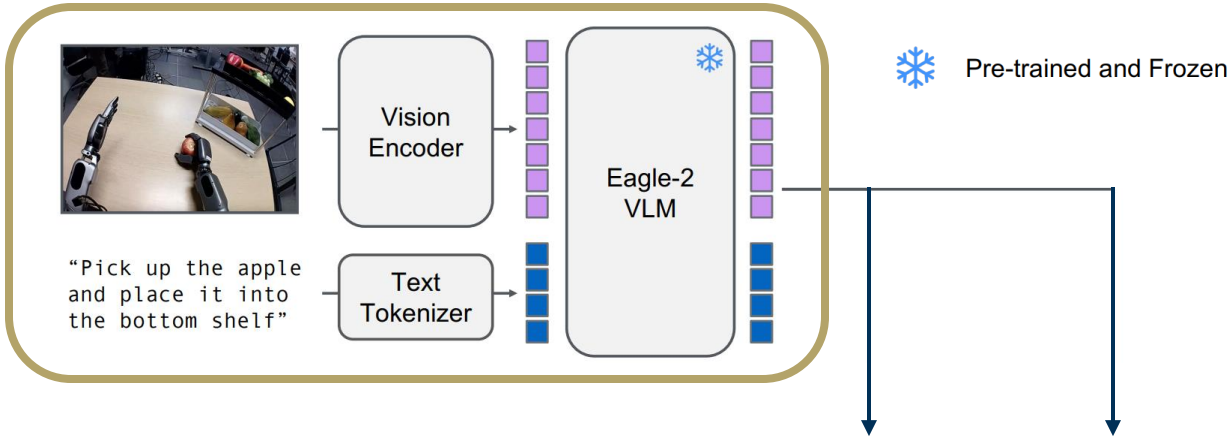
Model Architecture

System 2 – Vision-Language Module



Processing Pipeline:

- **Input:** Task text + one or more images.
- **Image Encoding:** Each 224×224 image → pixel-shuffle → **64 image tokens** per frame.
- **Language Encoding:** Text instruction tokenized in chat format (same as VLM training).
- **Fusion:** Image + text tokens jointly processed by the LLM → fused multimodal embeddings.
- **Feature Extraction:** Use **12th layer embeddings** (middle-layer) for balance of speed + performance.
- **Output:** Vision-language feature tensor (Batch × Seq × Hidden Dim) → fed into System 1 (Diffusion Transformer) via cross-attention.



Key Insights:

- **Pixel-shuffle compression** maintains spatial information with fewer tokens (8×8 grid).
- **Middle-layer embeddings** preserve grounded visual semantics while remaining efficient.
- Provides real-time (10 Hz) environment understanding and instruction reasoning.
- Serves as dynamic context for motor control generation in System 1.

Georgia Tech

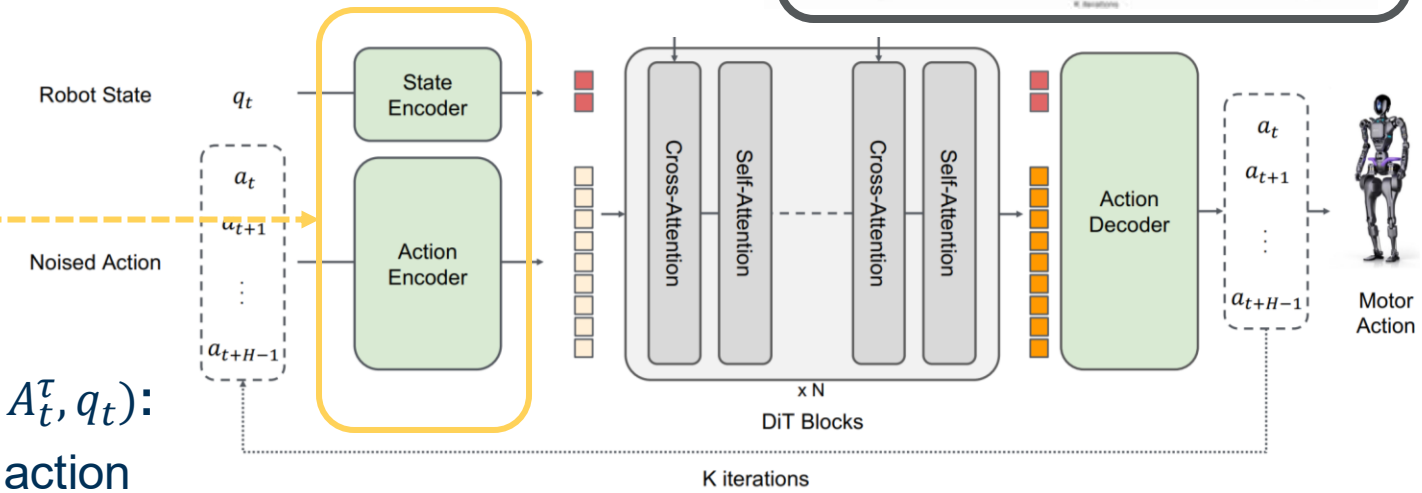
Model Architecture

System 1 – Vision-Language Module

Each robot embodiment has its own:

- State encoder
- Action decoder

Enables **cross-embodiment generalization** — same model handles multiple robots.



Diffusion Transformer (DiT) module $V_\theta(\phi_t, A_t^\tau, q_t)$:

- **Input**: robot state embeddings q_t, noised action tokens A_t^τ, and VLM embeddings ϕ_t.
- **Output**: Smooth, continuous motor actions across different embodiments.
- Uses **flow-matching** (diffusion-like denoising) to predict denoised motor actions.
- Alternating cross-attention (to link vision-language features) and self-attention (to model temporal action dependencies).

Georgia Tech

Model Architecture

System 1 – Vision-Language Module

Diffusion Transformer (DiT) with flow matching:

Training phase

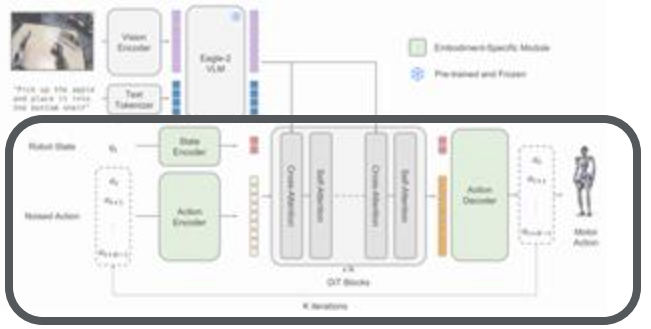
1. Take a ground-truth action sequence A_t.
2. Add noise using a random scalar, a flow-matching timestep $\tau \in [0,1]$:
$$A_t^\tau = \tau A_t + (1-\tau)\,\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$$
3. The model $V_\theta(\phi_t, A_t^\tau, q_t)$ predicts the vector field $\epsilon - A_t$ that drives A_t^τ back to A_t
4. Loss function: $\mathcal{L}_{\text{fm}} = \mathbb{E}_\tau[\|V_\theta(\phi_t, A_t^\tau, q_t) - (\epsilon - A_t)\|^2]$

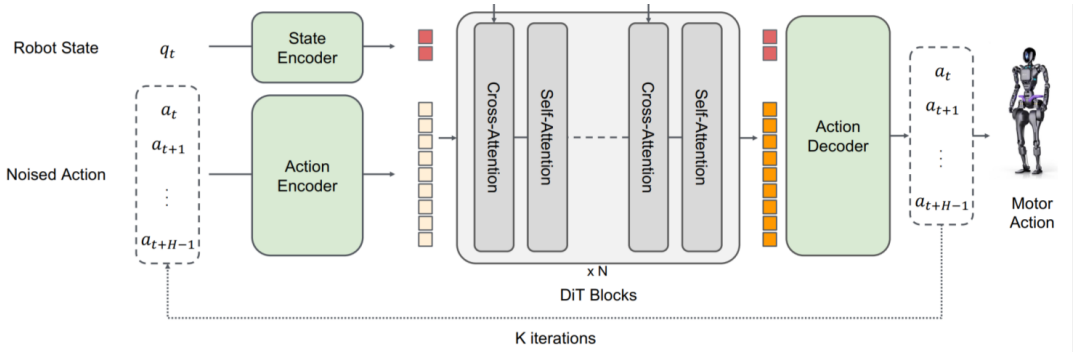
Inference phase

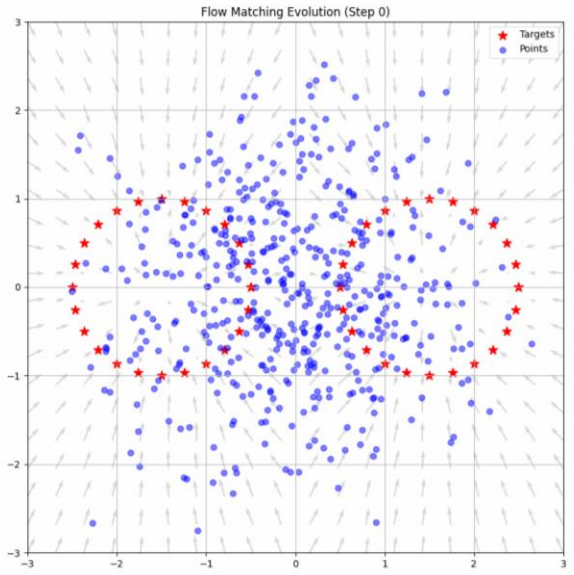
1. Start from random noise $A_t^0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$.
2. Iteratively update: $A_t^{\tau+1/K} = A_t^\tau + \frac{1}{K} V_\theta(\phi_t, A_t^\tau, q_t)$
3. Typically only **4 denoising steps (K = 4)** are needed to recover smooth action sequences.

(Each chunk contains 16 actions → efficient 120 Hz control.)

Action chunks:
At any given time t,
$A_t = [a_t, a_{t+1}, \ldots, a_{t+H-1}]$
the action vectors of timesteps t through $t + H - 1$.







Georgia Tech

Datasets

- Small scale and expensive to collect
- Ease of use for imitation learning, direct transfer

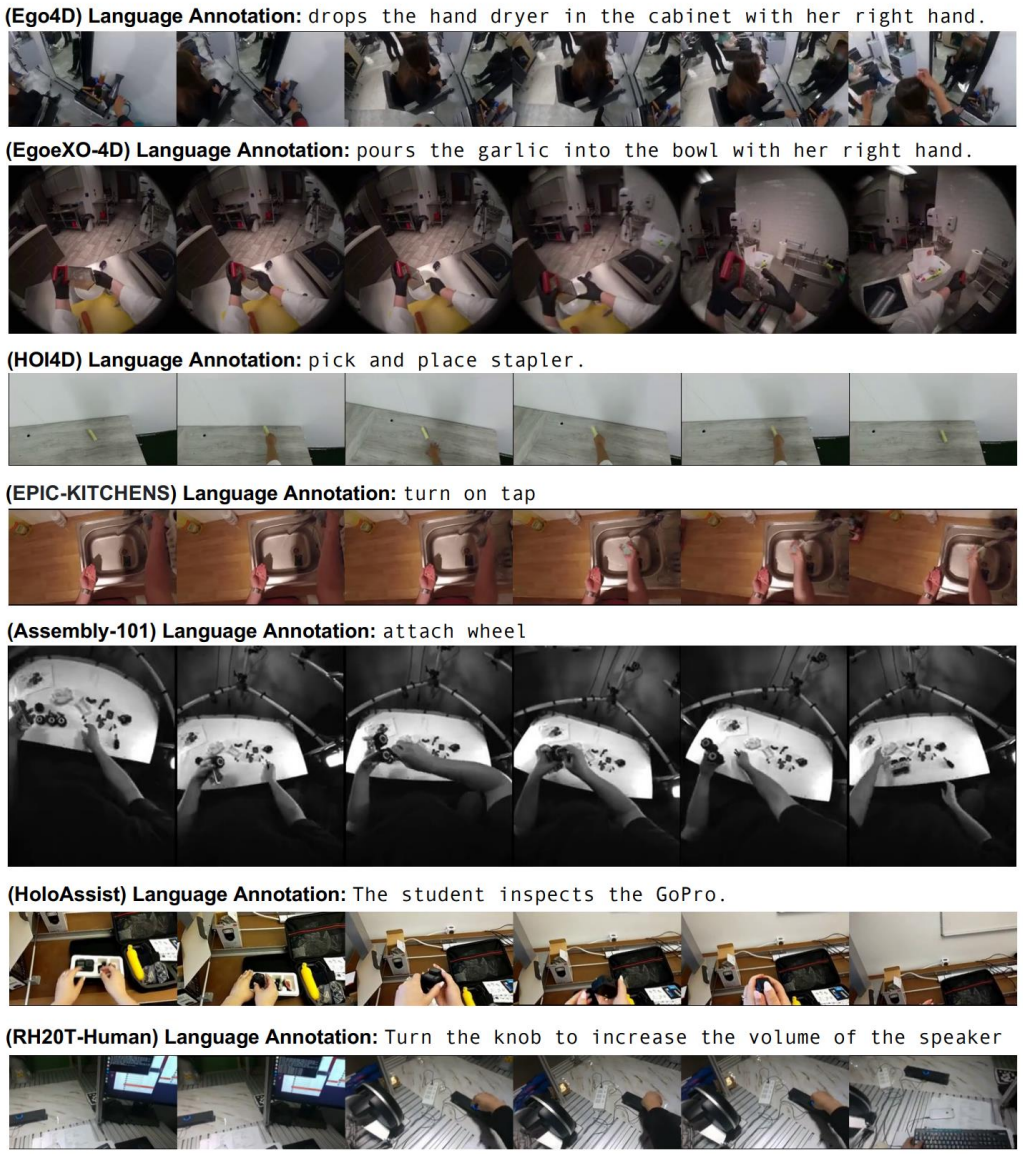
- Unlimited simulated data (in theory)
- Content creation challenge, reality gap, computational burden

- Massive scale and ever-growing
- Multimodal and unstructured
- Human-centered data

Georgia Tech

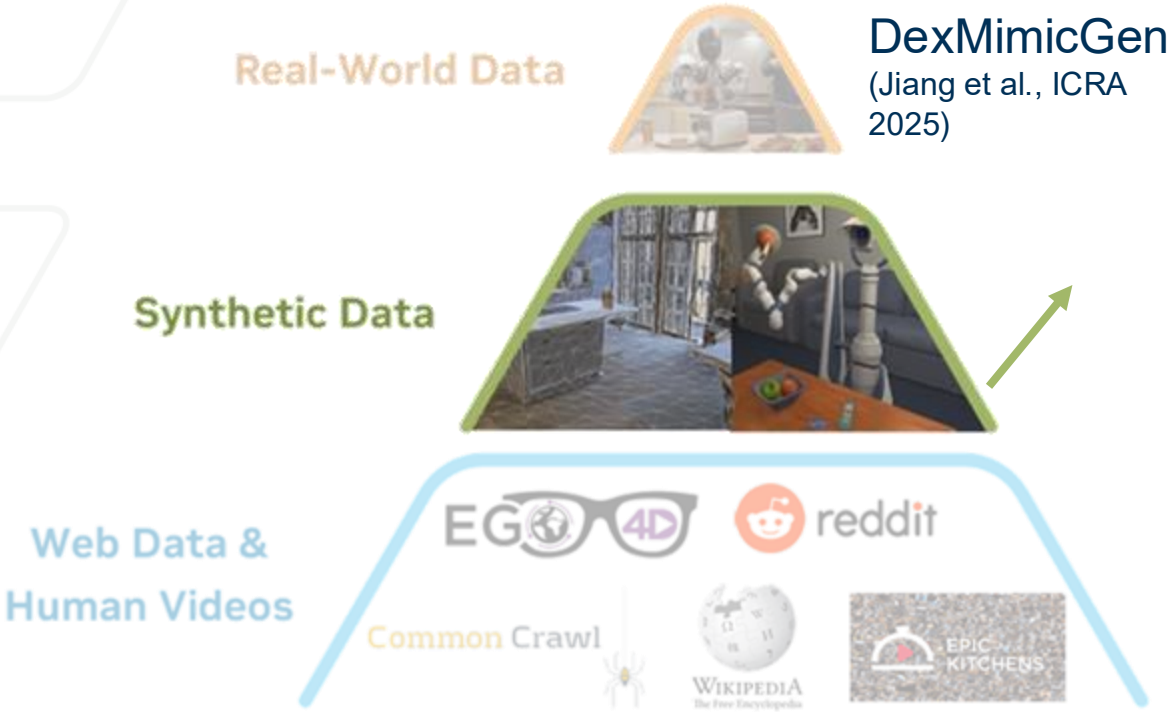
Datasets

Real-world human behaviors:
grasping, tool use, cooking, assembly, and other task-oriented activities performed in natural environments)



Georgia Tech

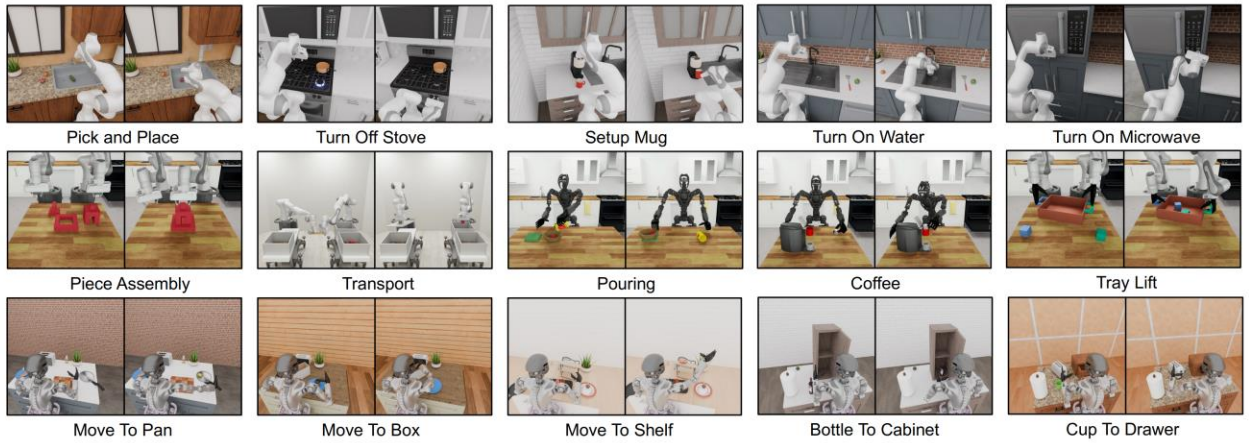
Datasets



Simulation trajectories

RoboCasa
(Nasiriany et al., RSS 2024)

DexMimicGen
(Jiang et al., ICRA 2025)



- Tasks follow the behavior “rearrange A from B to C”
- 54 unique combinations of source and target receptacle categories
- Objects and receptacles in randomized locations
- 10,000 new demonstrations for each (source, target)

1) **Simulation trajectories** automatically multiplied from a small number of human demonstrations within physics simulators
2) **Neural trajectories** derived from videos produced by off-the-shelf neural generation models

Georgia Tech

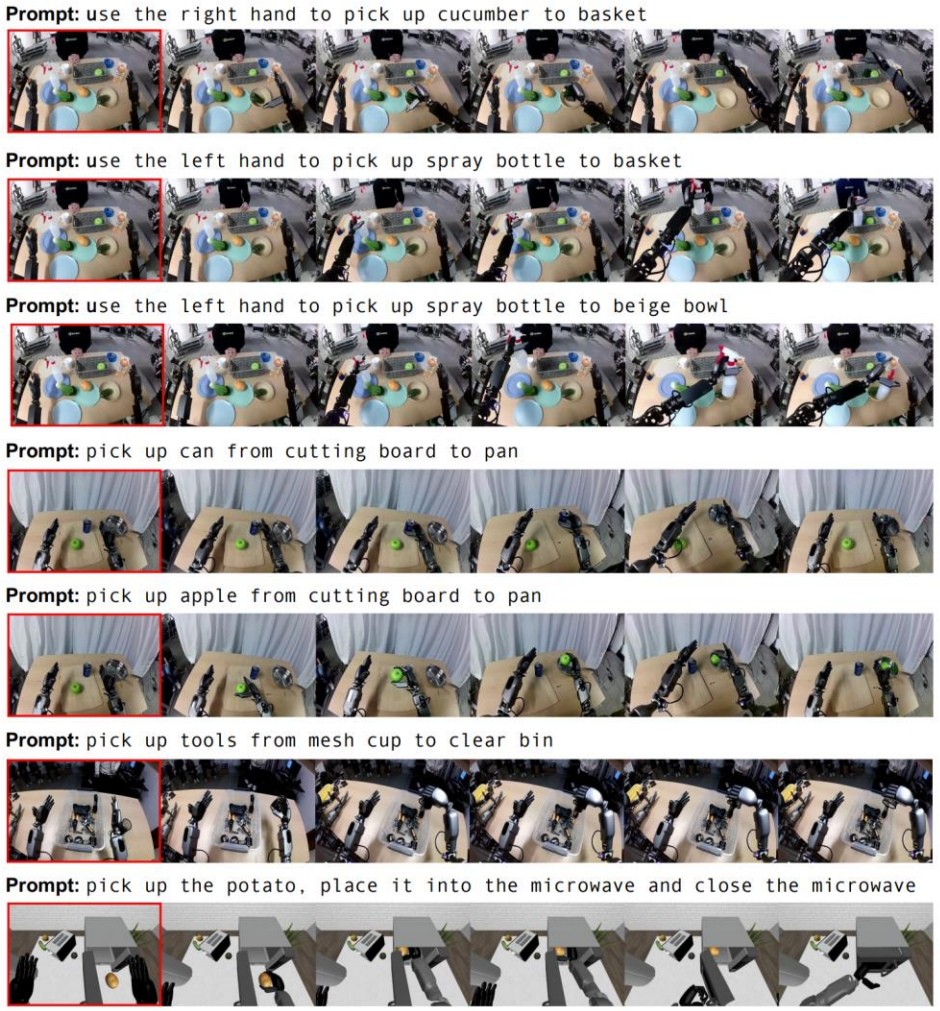
Datasets

1) **Simulation trajectories** automatically multiplied from a small number of human demonstrations within physics simulators
2) **Neural trajectories** derived from videos produced by off-the-shelf neural generation models

Neural trajectories

- Image-to-video model finetuned from WAN2.1-I2V-14B (Wan Team, 2025)
- Given existing initial frames with novel language prompts

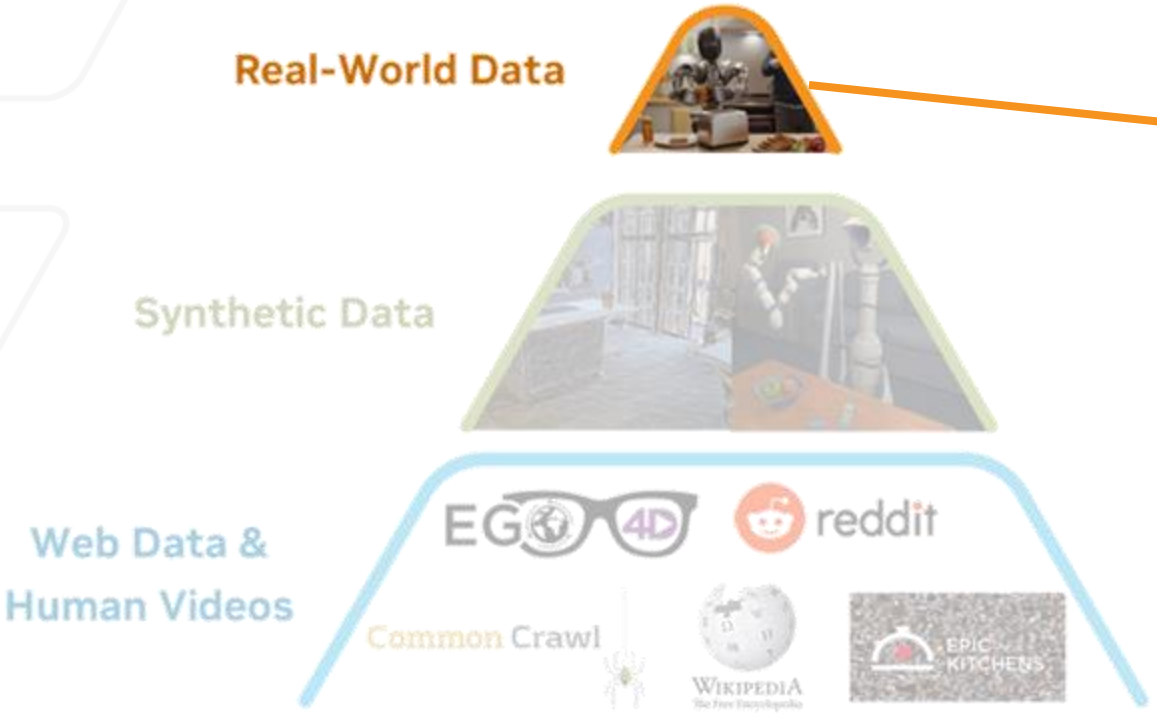
"pick up {object} from {location A} to {location B}"

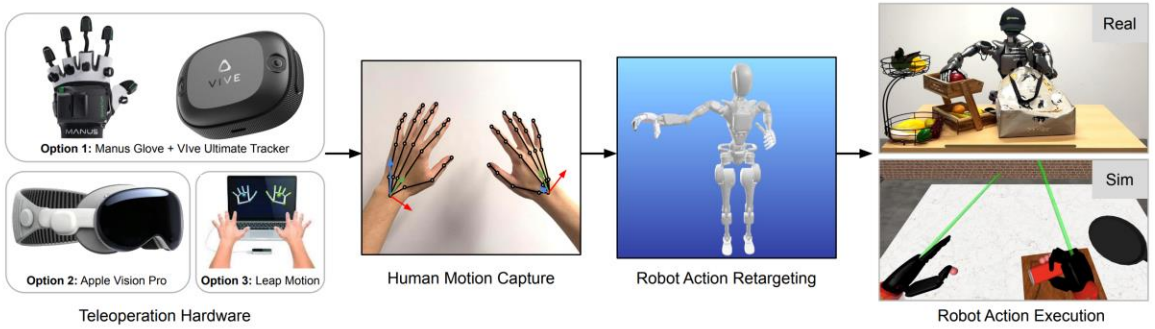


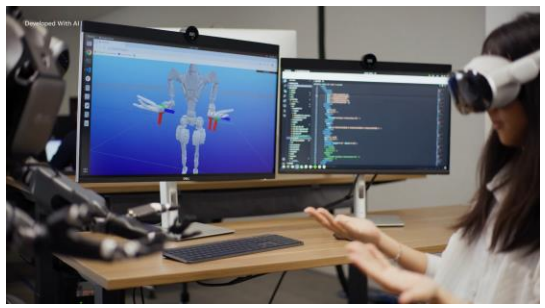
- 88h real teleop → 827h neural-generated data (~10× expansion).
- Enhanced by multimodal LLM filtering and captioning to ensure instruction compliance.

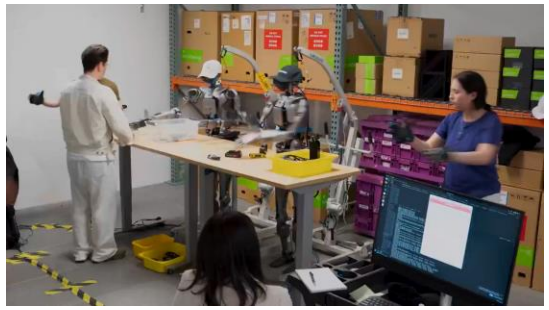
Georgia Tech

Datasets

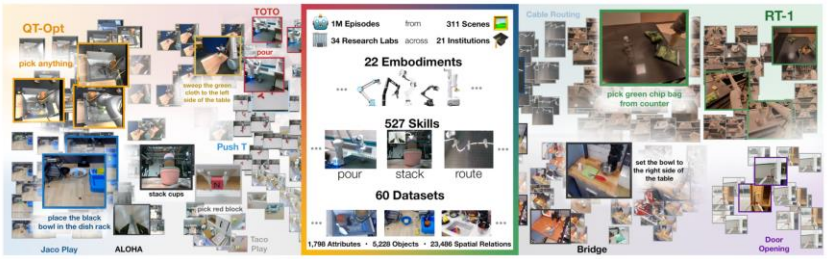








Open X-Embodiment



- **GR00T N1 Humanoid Dataset:** Teleoperated GR-1 humanoid tasks (grasp, move, place).
- **Open X-Embodiment:** Cross-robot manipulation datasets (RT-1, Bridge-v2, DROID, etc.).
- **AgiBot-Alpha:** Large-scale multi-robot trajectories with tool use and collaboration.

Georgia Tech

Datasets

Latent Action Learning using VQ-VAE

To handle any dataset (human egocentric videos and neural trajectories) that lacks explicit robot action labels

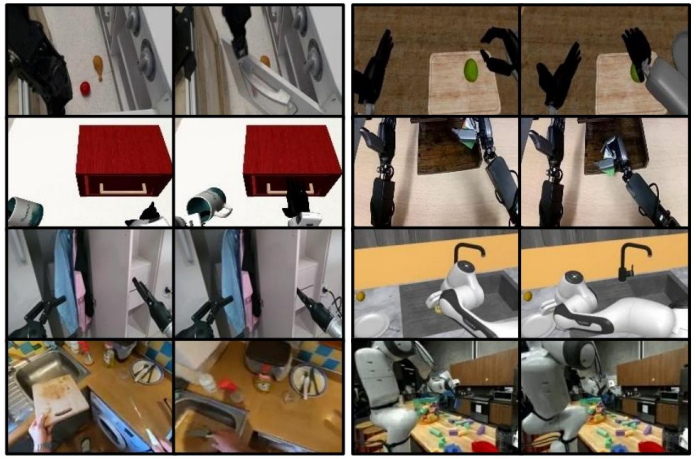
- Train a **VQ-VAE** model on consecutive video frames to learn *latent actions* representing motion between frames.
- **Encoder:** takes current frame x_t and future frame x_{t+H} →outputs latent action z_t.
- **Decoder:** reconstructs x_{t+H} from x_t and z_t.
- **Objective:** VQ-VAE loss aligns continuous embeddings to the nearest *codebook vector*, ensuring a discrete, shared action representation.

What it represents:

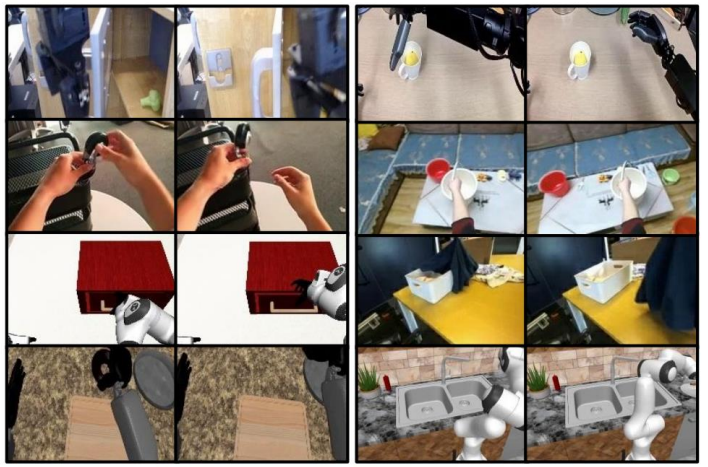
- A *learned embedding* of motion — not real torque, velocity, or joint angles.
- It captures "what kind of motion happened," in a **robot-agnostic latent space**.
- Treated as a **separate embodiment (LAPA)** during pretraining so that the model learns to interpret this space consistently across all data sources.

When used:

- **Pre-training phase** → used to align human, synthetic, and robot data under one unified representation.



Latent action 1: Move the right arm to the left



Latent action 2: Move the right arm to the right

Datasets

IDM (Inverse Dynamics Model)

Goal: Predict *realistic robot actions* given visual state transitions, trained per embodiment.

How it works:

- Train a model (Diffusion Transformer, System 1) to map from **two images** $(x_t, x_{t+H}) \rightarrow$**robot action sequence** a_t.
- *conditioned* on specific robot dynamics, learns the mapping between observed motion and *the actual control commands*.
- Uses a **flow-matching loss** (like the policy model).

What it represents:

- A *robot-specific* action label (e.g., joint velocities, torques, tendon lengths).
- Pseudo-labels neural trajectories or videos with plausible actions that the robot could have taken.

When used:

- **Post-training phase** → for fine-tuning with neural-generated trajectories when real actions are missing.

- **LAPA** is like learning “verbs” in a universal language of motion (“move left,” “reach up”).
- **IDM** translates those verbs into each robot’s **motor commands** (“bend joint 3 by 10°”).

Datasets

Table 7: Pre-training Dataset Statistics

Dataset	Length (Frames)	Duration (hr)	FPS	Camera View	Category
GR-1 Teleop Pre-Training	6.4M	88.4	20	Egocentric	Real robot
DROID (OXE)	23.1M	428.3	15	Left, Right, Wrist	Real robot
RT-1 (OXE)	3.7M	338.4	3	Egocentric	Real robot
Language Table (OXE)	7.0M	195.7	10	Front-facing	Real robot
Bridge-v2 (OXE)	2.0M	111.1	5	Shoulder, left, right, wrist	Real robot
MUTEX (OXE)	362K	5.0	20	Wrist	Real robot
Plex (OXE)	77K	1.1	20	Wrist	Real robot
RoboSet (OXE)	1.4M	78.9	5	Left, Right, Wrist	Real robot
Agibot-Alpha	213.8M	1,979.4	30	Egocentric, left, right	Real robot
RH20T-Robot	4.5M	62.5	20	Wrist	Real robot
Ego4D	154.4M	2,144.7	20	Egocentric	Human
Ego-Exo4D	8.9M	123.0	30	Egocentric	Human
Assembly-101	1.4M	19.3	20	Egocentric	Human
HOI4D	892K	12.4	20	Egocentric	Human
HoloAssist	12.2M	169.6	20	Egocentric	Human
RH20T-Human	1.2M	16.3	20	Egocentric	Human
EPIC-KITCHENS	2.3M	31.7	20	Egocentric	Human
GR-1 Simulation Pre-Training	125.5M	1,742.6	20	Egocentric	Simulation
GR-1 Neural Videos	23.8M	827.3	8	Egocentric	Neural-generated
Total robot data	262.3M	3,288.8	–	–	–
Total human data	181.3M	2,517.0	–	–	–
Total simulation data	125.5M	1,742.6	–	–	–
Total neural data	23.8M	827.3	–	–	–
Total	592.9M	8,375.7	–	–	–

Training

1. Pre-training Phase

- **Objective:** Train with **flow-matching loss** on mixed data:
- **Data usage:**
  - **Human videos:** use latent actions (from VQ-VAE).
  - **Real Robot:** use real actions + latent actions.
  - **Neural trajectories:** use latent + IDM-predicted actions.
- **Goal:** Learn a *generalizable cross-embodiment policy* that unifies all data under a single latent action space.

2. Post-training Phase

- Fine-tune on each single robot (embodiment-specific tasks).
- Keep language model frozen, tune action and perception modules.
- Use neural-generated data to augment limited real data (1:1 mix).
- Label synthetic data with latent or IDM pseudo-actions.
- **Goal** → achieve robust adaptation with minimal real-world data.

Stage	Purpose	Data Type	Label Source
Pre-training	Generalization across embodiments	Human, synthetic, real robot	Latent actions (LAPA)
Post-training	Embodiment-specific fine-tuning	Real robot data	LAPA or IDM pseudo-labels
Post-training w/ Neural Trajectories	Low-data augmentation	Synthetic neural videos	IDM-predicted pseudo-actions

GR00t: Experiments and Evaluation

Georgia Tech

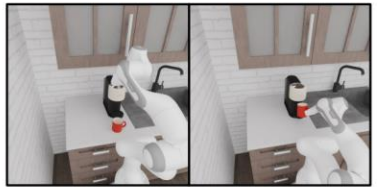
Experiments and Results

Simulation Benchmarks

- RoboCasa Kitchen (24 tasks, Franka Emika Panda arm)
  - pick-and-place, door opening and closing, pressing buttons, turning faucets, and more


Pick and Place

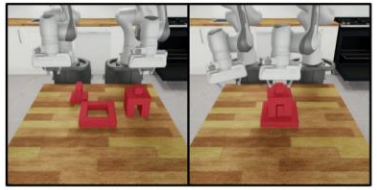

Turn Off Stove

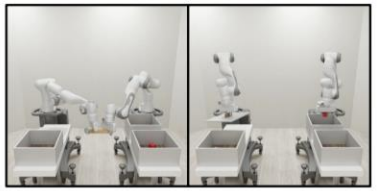

Setup Mug

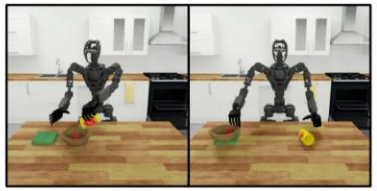

Turn On Water

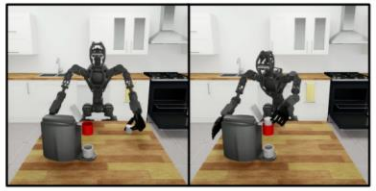

Turn On Microwave

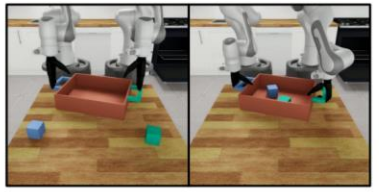
- DexMimicGen Cross-Embodiment Suite (9 tasks)
  - Bimanual Panda Arms with Parallel-Jaw Grippers: threading, piece assembly, and transport
  - Bimanual Panda Arms with Dexterous Hands: box cleanup, drawer cleanup, and tray lifting
  - GR-1 Humanoid with Dexterous Hands: pouring, coffee preparation, and can sorting


Piece Assembly


Transport

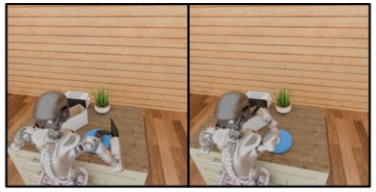

Pouring

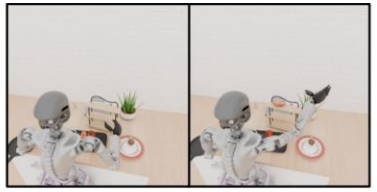

Coffee

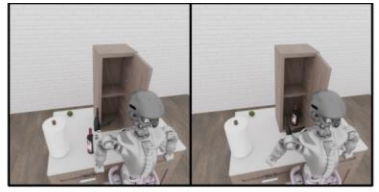

Tray Lift

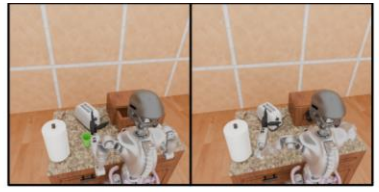
- GR-1 Tabletop Tasks (24 tasks, GR-1)
  - rearranging objects


Move To Pan


Move To Box


Move To Shelf


Bottle To Cabinet


Cup To Drawer

Georgia Tech

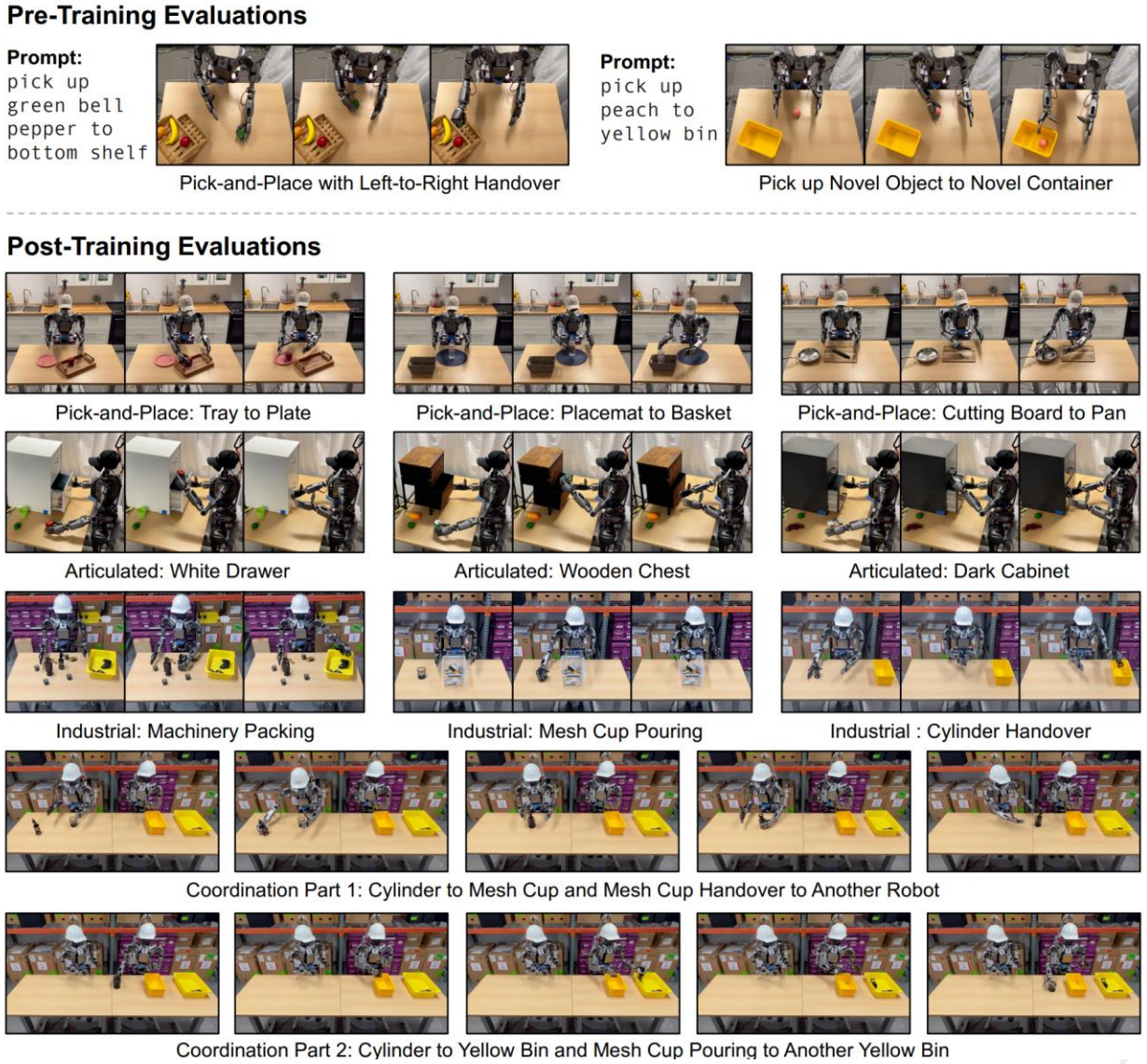
Experiments and Results

Real-World Benchmarks

(left) left-to-right handover

(right)placement of novel objects into an unseen target container

- Object-to-Container Pick-and-Place
  - 5 tasks, Pick-and-Place
- Articulated Object Manipulation
  - 3 tasks, Articulated
- Industrial Object Manipulation
  - 3 tasks, Industrial
- Multi-Agent Coordination
  - 2 tasks, Coordination



Georgia Tech

Experiments and Results

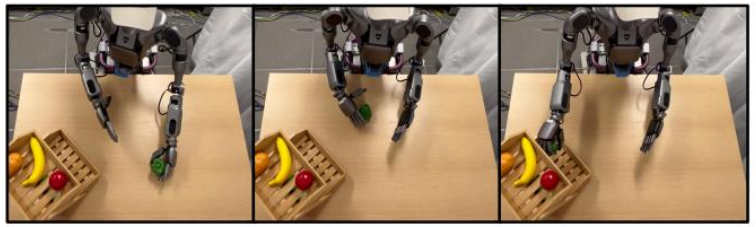
Pre-training Evaluation (Real GR-1 Robot)

- **Task 1:** Place object on bottom shelf (requires bimanual transfer).
- **Task 2:** Place *novel object* into unseen container.
- **Results:**
  - Task 1 → **76.6 %** success (11.5 / 15 trials)
  - Task 2 → **73.3 %** success (11 / 15 trials)

→ Shows **strong generalization** and **effective coordination** from large-scale pre-training.

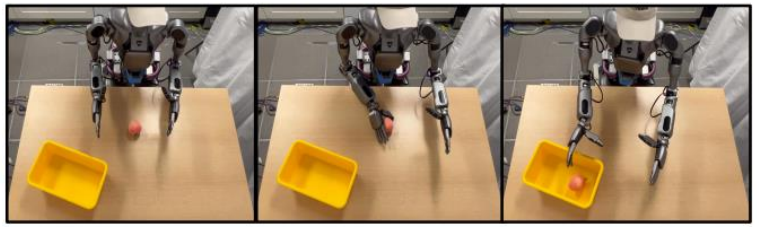
Pre-Training Evaluations

Prompt:
`pick up green bell pepper to bottom shelf`



Pick-and-Place with Left-to-Right Handover

Prompt:
`pick up peach to yellow bin`



Pick up Novel Object to Novel Container

Experiments and Results

Post-training Evaluation

Simulation

- Benchmarks: **RoboCasa**, **DexMimicGen**, **GR-1 Simulation**
- **Data Regimes:** 30 / 100 / 300 demos per task.
- GR00T N1 → **Consistently outperforms** from-scratch baselines in all benchmarks.

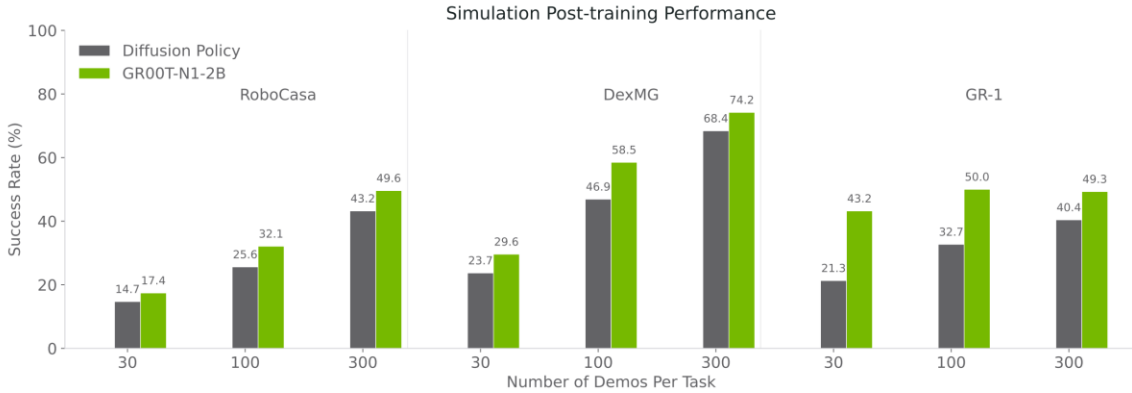
Real Robot:

- Compared with *Diffusion Policy*.
- Trained on only 10 % of data → just 3.8 % lower than Diffusion Policy (full data).
- **+32.4 % gain** (10 % data) and **+30.4 % gain** (full data) overall.

→ Demonstrates **data efficiency** and **strong embodiment transfer**.

Simulation Results: 100 demonstrations per task

	RoboCasa	DexMG	GR-1	Average
BC Transformer	26.3%	53.9%	16.1%	26.4%
Diffusion Policy	25.6%	56.1%	32.7%	33.4%
GR00T-N1-2B	**32.1%**	**66.5%**	**50.0%**	**45.0%**



Real-World Results

	Pick-and-Place	Articulated	Industrial	Coordination	Average
Diffusion Policy (10% Data)	3.0%	14.3%	6.7%	27.5%	10.2%
Diffusion Policy (Full Data)	36.0%	38.6%	61.0%	62.5%	46.4%
GR00T-N1-2B (10% Data)	35.0%	62.0%	31.0%	50.0%	42.6%
GR00T-N1-2B (Full Data)	**82.0%**	**70.9%**	**70.0%**	**82.5%**	**76.8%**

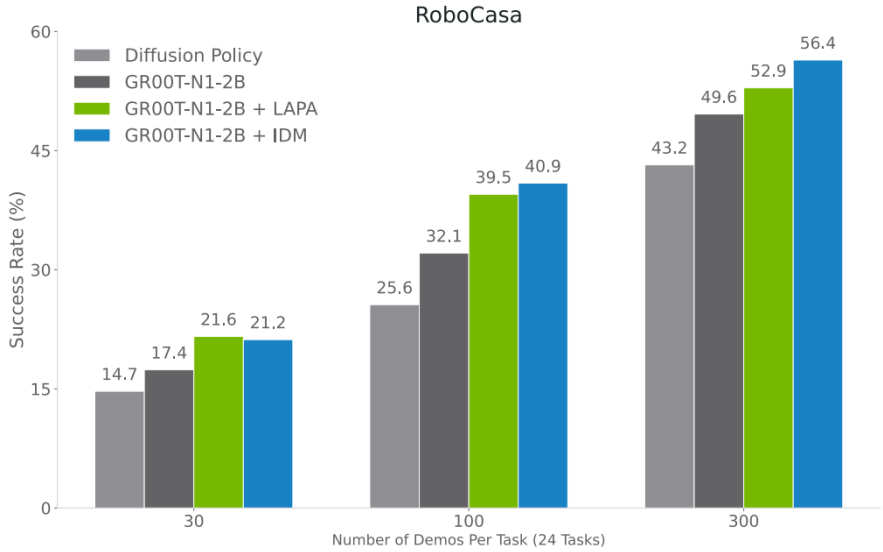
Georgia Tech

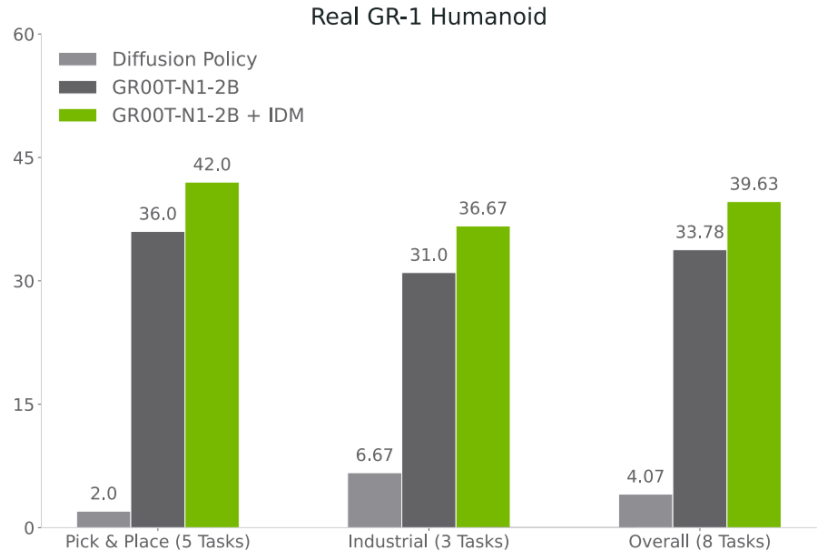
Experiments and Results

Post-training + Neural Trajectories

- **Simulation (RoboCasa):**
  +4.2 %, +8.8 %, +6.8 % improvements for 30 / 100 / 300 data regimes.
- **Real GR-1 Humanoid:**
  +5.8 % average improvement across 8 tasks.
- **Label comparison:**
  - *LAPA* > *IDM* in low-data regime (30 demos).
  - *IDM* > *LAPA* with more data (100 – 300 demos).

→ Neural trajectories + pseudo-labels enhance learning under data scarcity.





An **extension experiment** showing how synthetic data can further boost the model's generalization and efficiency, even when there is only limited percentage of real robot data.

Georgia Tech

Experiments and Results

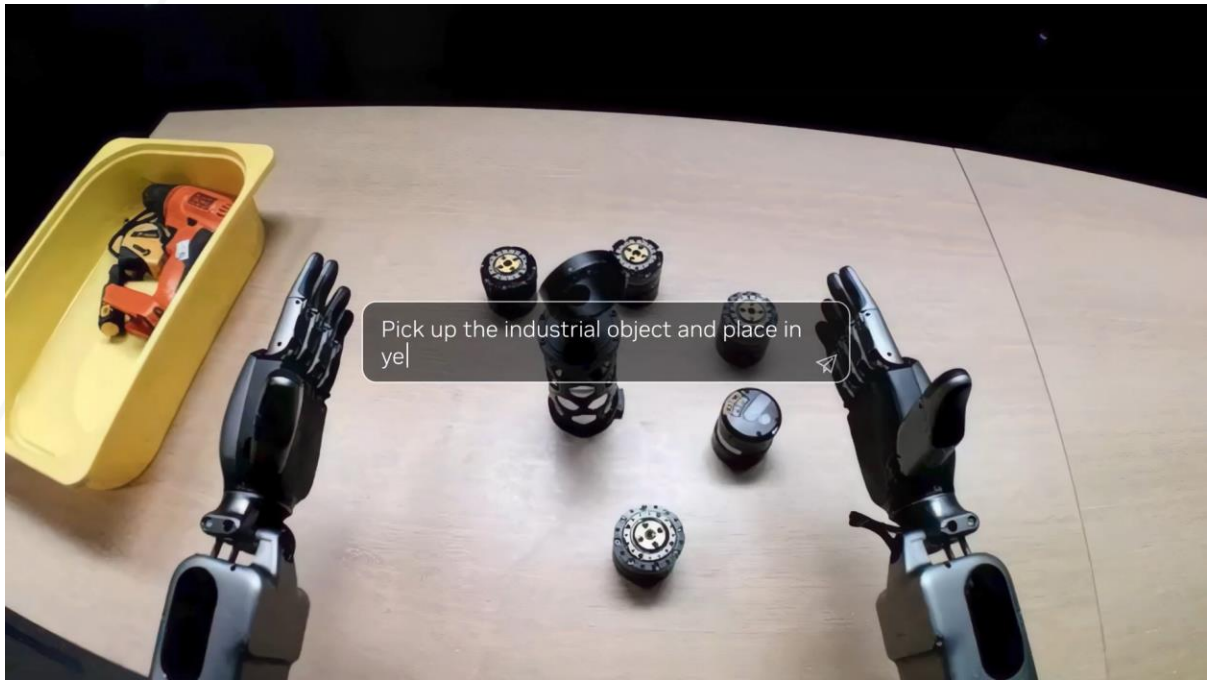
Table 5: Success rate on real-world tasks with the GR-1 humanoid robot.

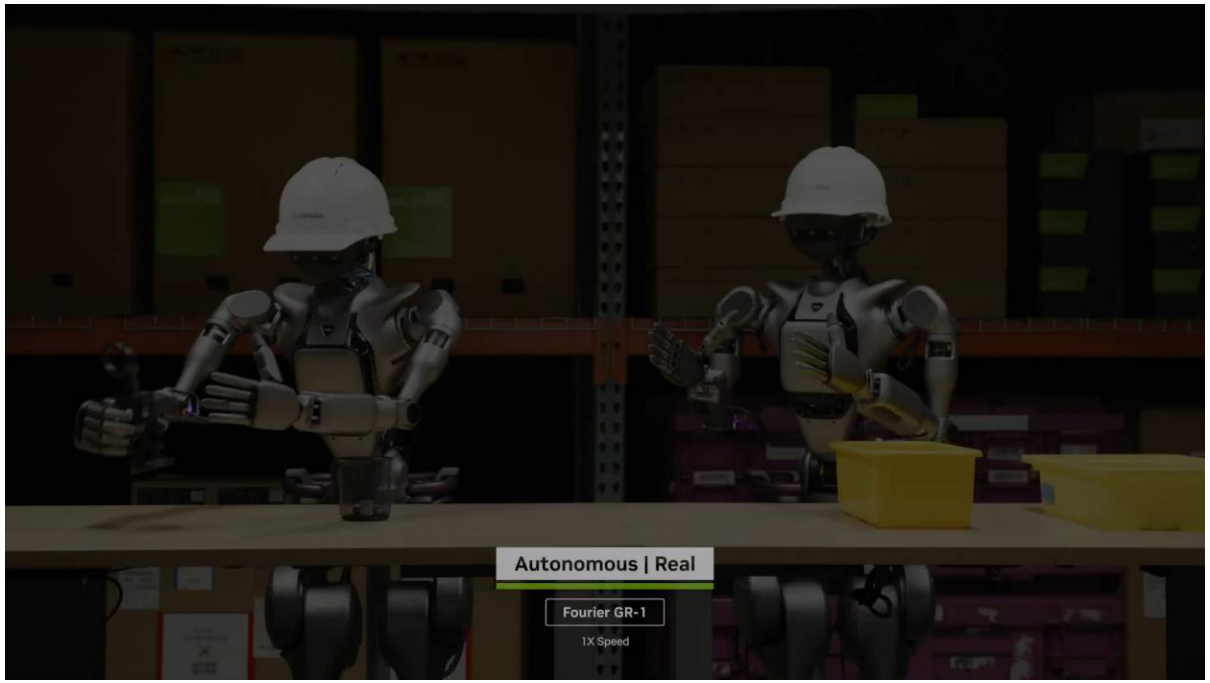
Task	Diffusion Policy		GR00T-N1-2B	
	10% Data	Full Data	10% Data	Full Data
Tray to Plate	0.0%	20.0%	40.0%	100.0%
Cutting Board to Basket	0.0%	30.0%	10.0%	100.0%
Cutting Board to Pan	0.0%	60.0%	60.0%	80.0%
Plate to Bowl	0.0%	40.0%	30.0%	100.0%
Placemat to Basket	10.0%	60.0%	40.0%	80.0%
Pick-and-Place Seen Object Average	2.0%	42.0%	36.0%	92.0%
Tray to Plate	0.0%	20.0%	30.0%	80.0%
Cutting Board to Basket	10.0%	20.0%	60.0%	60.0%
Cutting Board to Pan	0.0%	40.0%	40.0%	80.0%
Plate to Bowl	0.0%	20.0%	10.0%	40.0%
Placemat to Basket	10.0%	50.0%	30.0%	100.0%
Pick-and-Place Unseen Object Average	4.0%	30.0%	34.0%	72.0%
Pick-and-Place Average	**3.0%**	**36.0%**	**35.0%**	**82.0%**
White Drawer	6.6%	36.4%	26.4%	79.9%
Dark Cabinet	0.0%	46.2%	86.6%	69.7%
Wooden Chest	36.4%	33.2%	72.9%	63.2%
Articulated Average	**14.3%**	**38.6%**	**62.0%**	**70.9%**
Machinery Packing	20.0%	44.0%	8.0%	56.0%
Mesh Cup Pouring	0.0%	62.5%	65.0%	67.5%
Cylinder Handover	0.0%	76.5%	20.0%	86.6%
Industrial Average	**6.7%**	**61.0%**	**31.0%**	**70.0%**
Coordination Part 1	45.0%	65.0%	70.0%	80.0%
Coordination Part 2	10.0%	60.0%	30.0%	85.0%
Coordination Average	**27.5%**	**62.5%**	**50.0%**	**82.5%**
Average	**10.2%**	**46.4%**	**42.6%**	**76.8%**

- Some tasks plateau even with full data → **data scaling alone may not solve embodiment complexity.**
- Failures often relate to **contact dynamics, occlusions, and fine manipulation** → areas where *physics priors* or *model-based reasoning* could help.

Experiments and Results

Qualitative Results





Source: https://developer.nvidia.com/blog/accelerate-generalist-humanoid-robot-development-with-nvidia-isaac-gr00t-n1/

Experiments and Results

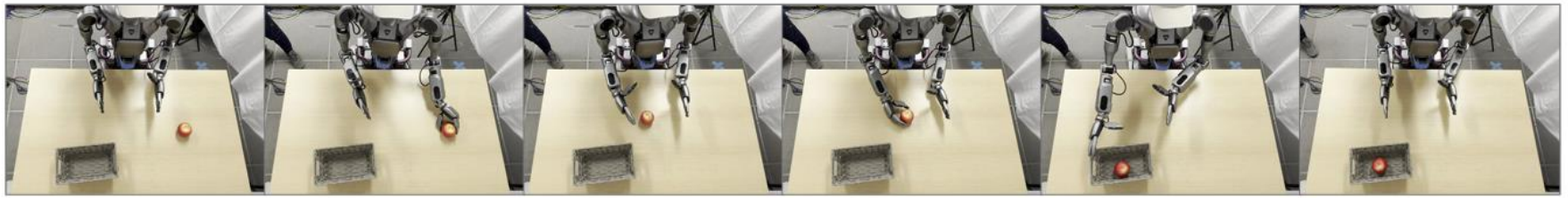
Qualitative Results

Real Humanoid Behavior ("Pick up red apple and place in basket")

- Apple placed left of the hand → tests bimanual coordination.
- **Pre-trained model:** grasps with left hand → hands to right → places in basket ✅
- **Post-trained model:** fails (learned only single-hand behavior).

→ **Pre-training preserves general coordination skills** that fine-tuning can over-specialize away.

Task: `Pick up red apple and place it into the basket`



Motion Quality Comparison (Post-training Real Robot)

- **GR00T N1:** smooth motions, accurate grasps.
- **Diffusion Policy:** jerky motion, slow start, frequent mis-grasps.

→ GR00T N1 achieves **smoother and more reliable real-world control**.

Summary

GR00T N1 is the **first large-scale generalist humanoid foundation model** that unifies reasoning, perception, and control across heterogeneous data sources and robot embodiments.

Limitations

- Focused mainly on **short-horizon tabletop manipulation** — lacks long-horizon loco-manipulation skills.
- Requires **stronger vision-language backbone** for better spatial reasoning and language understanding.
- Needs improvements in **humanoid hardware** and **model architecture** to support more complex motions.
- Synthetic data generation still limited by:
  - Low diversity and realism.
  - Difficulty maintaining **physical consistency** in generated trajectories.
- Lacking important sensing modalities (torque, tactile, …)

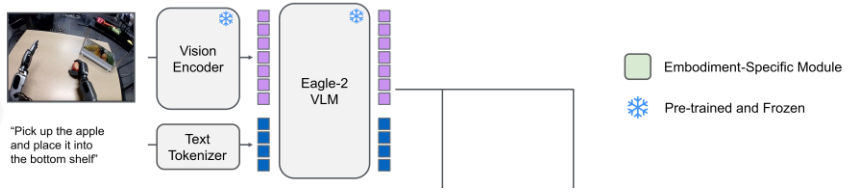
Follow up

GR00T N1.5

An Improved Open Foundation Model for Generalist Humanoid Robots

11 June 2025

Improved VLM Grounding Capabilities



Model	Size	GR-1 grounding IoU (↑)	RefCOCOg-val IoU (↑)
Qwen2.5VL	3B	35.5	85.2
GR00T N1.5 VLM	2.1B	**40.4**	**89.6**

- Keeps the **language–vision expert intact** during both pre- and post-training.
- Preserves strong linguistic and visual reasoning → **better instruction following** and **generalization**.

FLARE Objective – Learning by Watching

- Adds **Future Latent Representation Alignment** → learns from *human egocentric videos*.
- Robot learns new tasks just by *watching humans*, even without labeled robot data.

DreamGen Integration – Learning by Imagining

- Uses **synthetic neural trajectories** from video world models to generate new robot data.
- Expands task diversity → **better zero-shot and few-shot generalization**.

❖ Higher success rate, more diverse data sources, significantly improved language following capabilities.

❖ Significantly better **performance in low-data, zero-shot, and novel-verb tasks**

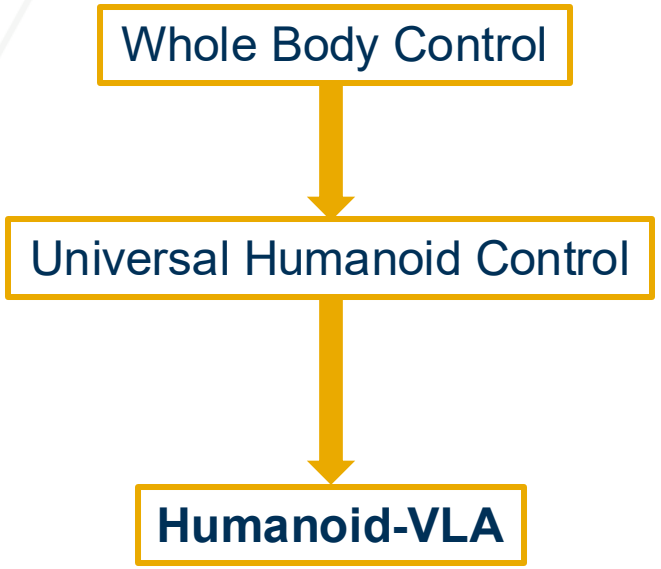
Georgia Tech

Humanoid-VLA: Towards universal humanoid control with visual integration

Pengxiang Ding*[12] Jianfei Ma*[3] Xinyang Tong*[1] Binghong Zou[3] Xinxin Luo[3] Yiguo Fan1 Ting Wang[1]
Hongchao Lu[1] Panzhong Mo[3] Jinxin Liu[3] Yuefan Wang[12] Huaicheng Zhou[3] Wenshuo Feng[3] Jiacheng Liu[12]
Siteng Huang[1] Donglin Wang[13]

Georgia Tech

Motivation



- Pros: High-fidelity motion control
- Cons: Reactive mechanisms -- dynamically adjusting motions in response to external inputs

- Pros: Ego-centric visual integration
- Cons: Data scarcity
  - Lack of synchronized first-person view (FPV) data
  - Teleoperation is expensive

- Language Understanding
- Egocentric Scene Perception
- Motion Control

Motivation

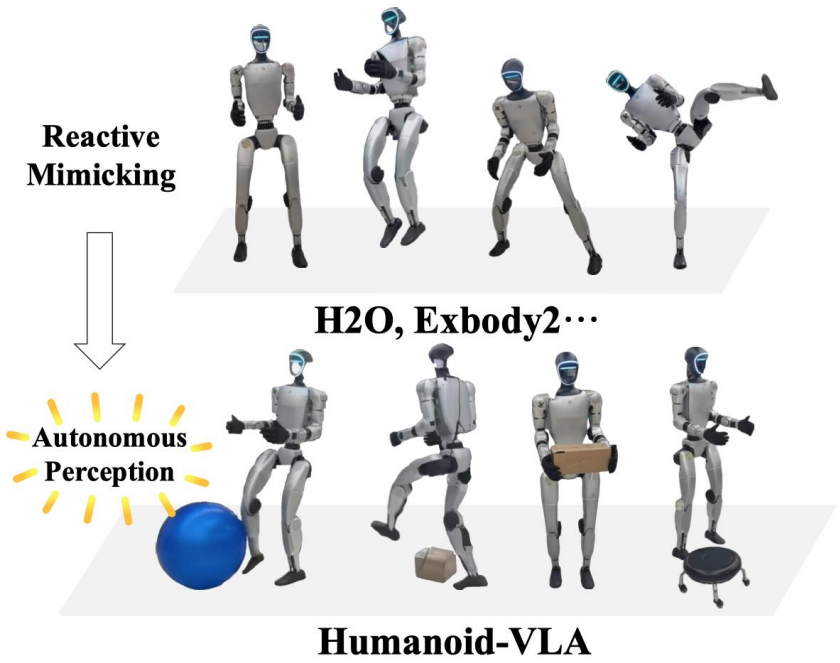


Figure 1: **Comparison between previous works and our approach.** With the capability of autonomous perception, Humanoid-VLA can perform tasks to interact with objects, significantly advancing beyond previous methods that rely on mimicking human demonstrations for motion execution.

Humanoid-VLA

- Language Understanding
- Egocentric Scene Perception
- Motion Control

Feasible & Cost-effective paradigm:

A. Language-motion pre-alignment:
   - Nonegocentric motion dataset with textual description
   - Learn universal motion patterns and action semantics

B. Video-conditioned Fine-tuning:
   - Egocentric visual context
   - Enable contextual motion generation

On More Thing

Self-supervised data augmentation strategy:
   - Auto generate pseudo-annotation derived from motion data
   - Converts raw motion sequence into informative QA pair

Data Acquisition Challenges

Prior datasets:
Small, well-curated, motion-language pairs → good quality but low diversity

Large online video datasets:
Rich motion diversity but lack language annotations

Problem:
Scarcity of paired motion-language data hinders pre-alignment training

Existing methods:
- Manual label: expensive
- LLM label: noisy & incomplete

Category	Text	Motion	Clips	Frames	Hours
Motion capture	✓	✓	29K	0.3M	4.1
Online Video	✗	✓	0.8M	541M	7515.7
Synthetic Data	✓	✓	100K	16M	227.7
Total			**0.929M**	**557.3M**	**7790.2**

Table 1: **Datasets Statistics**

Self-Supervised Data Augmentation

Existing Annotation Approaches

- **Manual Annotation**: accurate but costly and slow
- **Video LLMs (VLLMs)**: scalable but
  - Often noisy/incomplete/imprecise
  - Fail to describe fine-grained motions or complex actions
- Both are suboptimal for motion-language alignment

Self-Supervised Data Augmentation

- Avoids explicit manual annotations
- Key idea: derive self-supervised tasks directly from motion data
- Example:
  - Mask body joints temporarily (e.g., left arm)
  - Model reconstructs missing movement
  - Instructional prompt: “Missing left arm <Occlusion> motion data — please complete the motion.”
- Enables automatic, scalable, and accurate pseudo-annotations

Data Acquisition Pipeline

Cost-effective annotation method

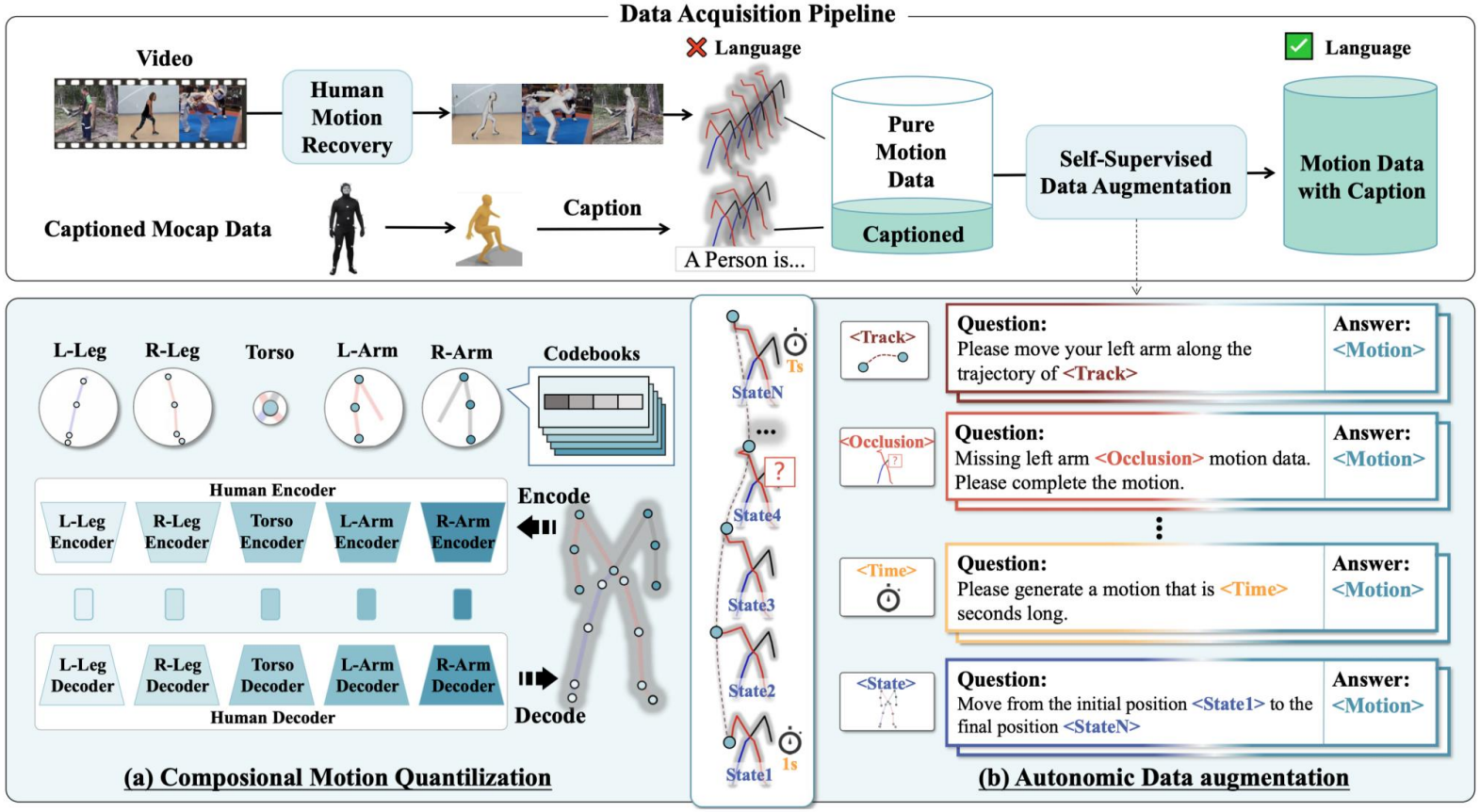
- Designing various self-supervised tasks directly derived from motion data

- Instructional prompts:
  - *"missing left arm <Occlusion> motion data. Please complete the motion"*
- Target outputs:
  - Ground motion

Two key modules:

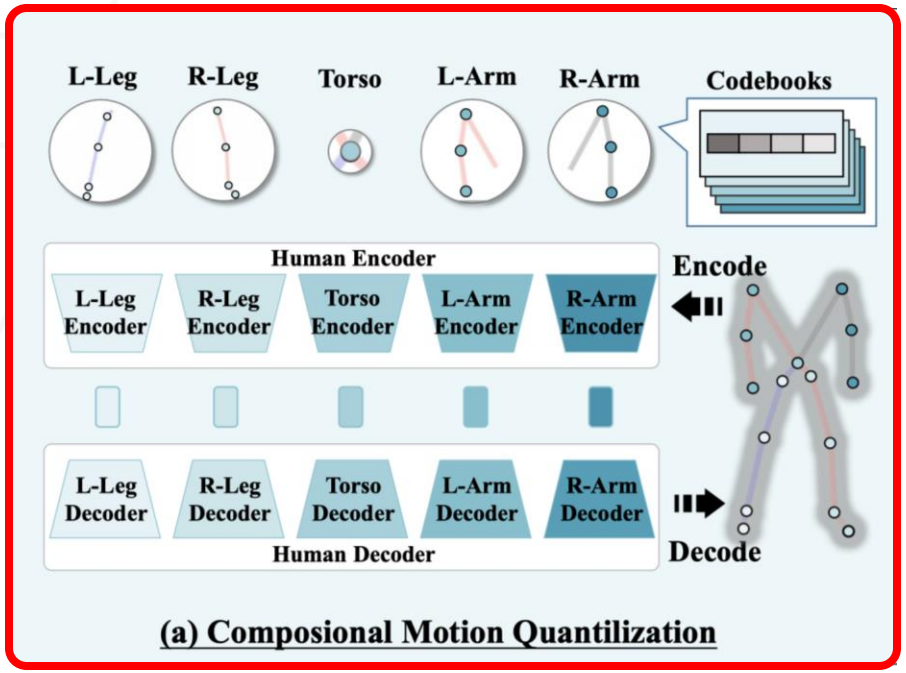
✓ Compositional Motion Quantitation

✓ Autonomic Data Augmentation



Data Acquisition Pipeline

Key module 1: Compositional Motion Quantitation



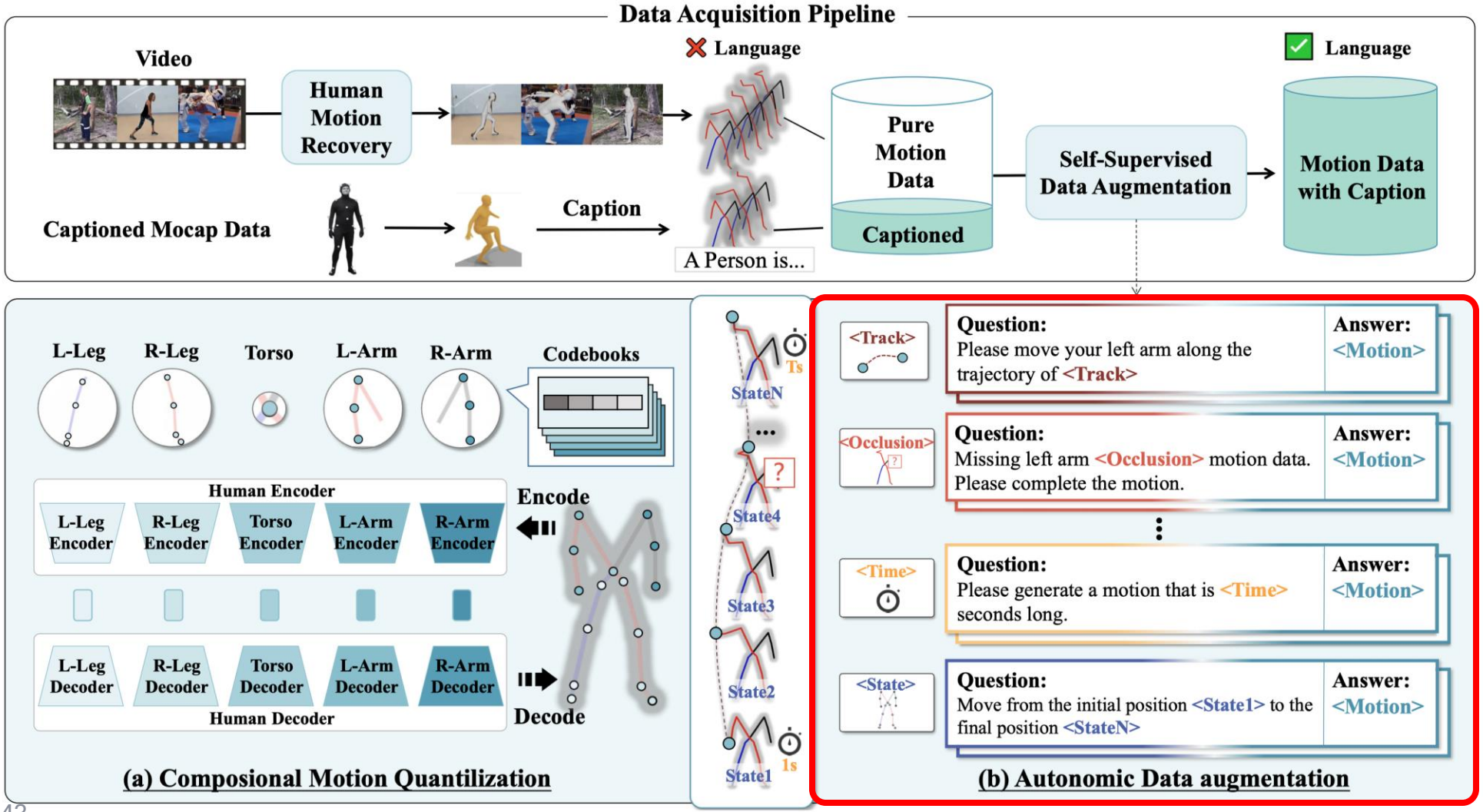
- Decomposition of motion into five body parts:
  - left leg, right leg, torso, left arm, right arm
- Each part encoded independently into token
- Motion Encoder: encodes body part data
- Motion Decoder: reconstructs full pose

$$\hat{z}_t = \mathcal{E}_m(c_t), \quad \hat{c}_t = \mathcal{D}_m(\hat{z}_t)$$

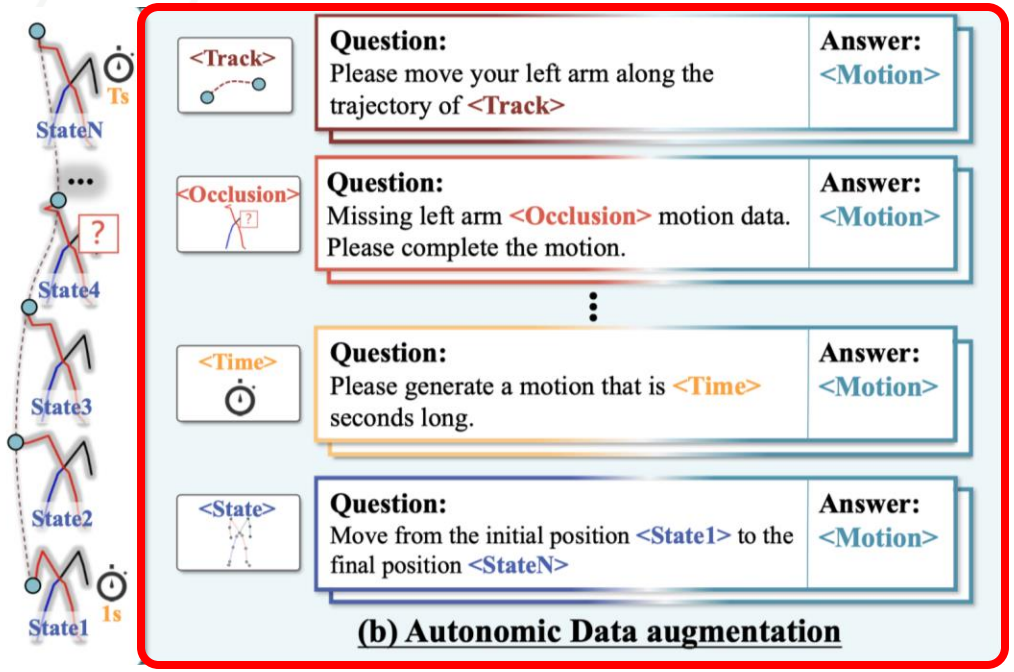
$$\mathcal{L}_{hvq} = \underbrace{\|c_t - \hat{c}_t\|_2}_{\mathcal{L}_{\text{rec}}} + \underbrace{\|\text{sg}(z_t) - \hat{z}_t\|_2}_{\mathcal{L}_{\text{emb}}} + \underbrace{\|z_t - \text{sg}(\hat{z}_t)\|_2}_{\mathcal{L}_{\text{com}}}$$

✓ Form flexible operations on the motion sequence at the token level.
E.g. replace, perturb, or rearrange the tokens corresponding to specific body parts to generate new motion patterns.

Data Acquisition Pipeline



Data Acquisition Pipeline



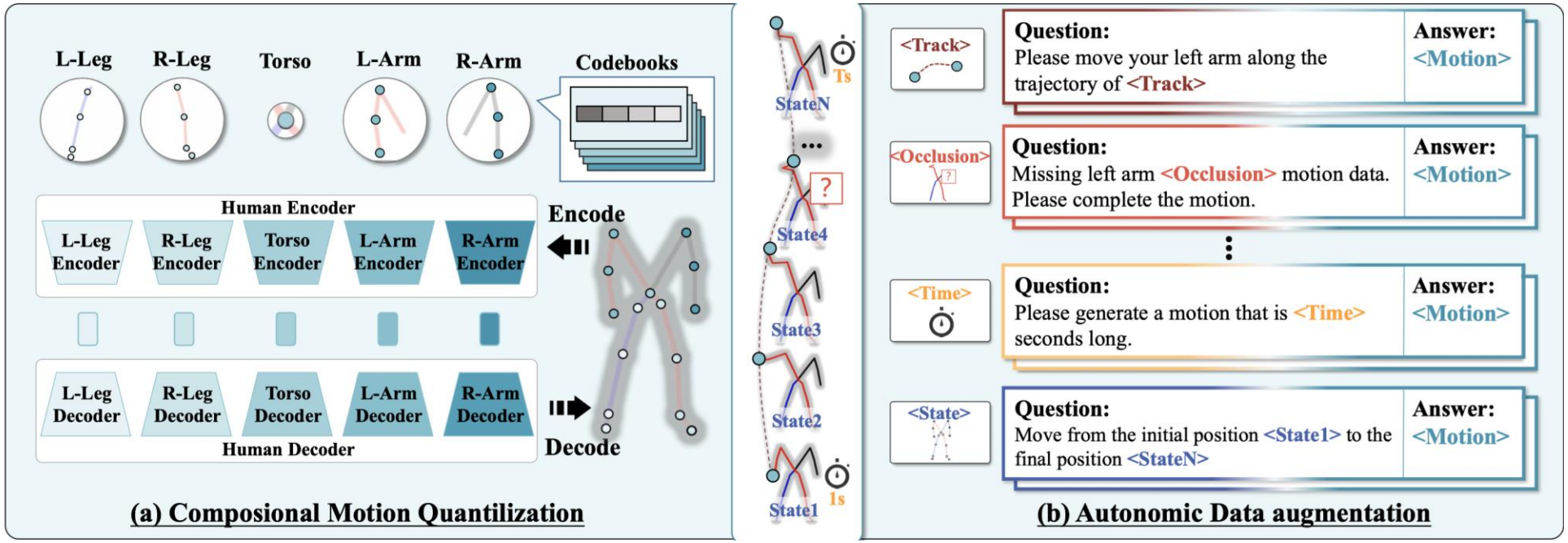
- Four augmentation types:
  - <Track>, <Time>, <Occlusion>, <State>
- Example:
  - Isolate root joint's trajectory (<Track>)
  - Generate instruction: "Please move your center position along the trajectory of <Track>."

✓ Creates new instruction–motion pairs from unlabeled data: effectively augments datasets that initially lacked linguistic annotations, enabling their use in tasks requiring text-motion alignment.

Georgia Tech

Data Acquisition Pipeline

1) Highly flexible and extensible
2) Leverages motion data's inherent temporal and spatial dynamics, allowing models to learn richer and more robust motion-language relationships
3) Interleaved datasets enhances cross-modal alignment by incorporating both motion and text in inputs and outputs.



Georgia Tech

Framework: two-stage training

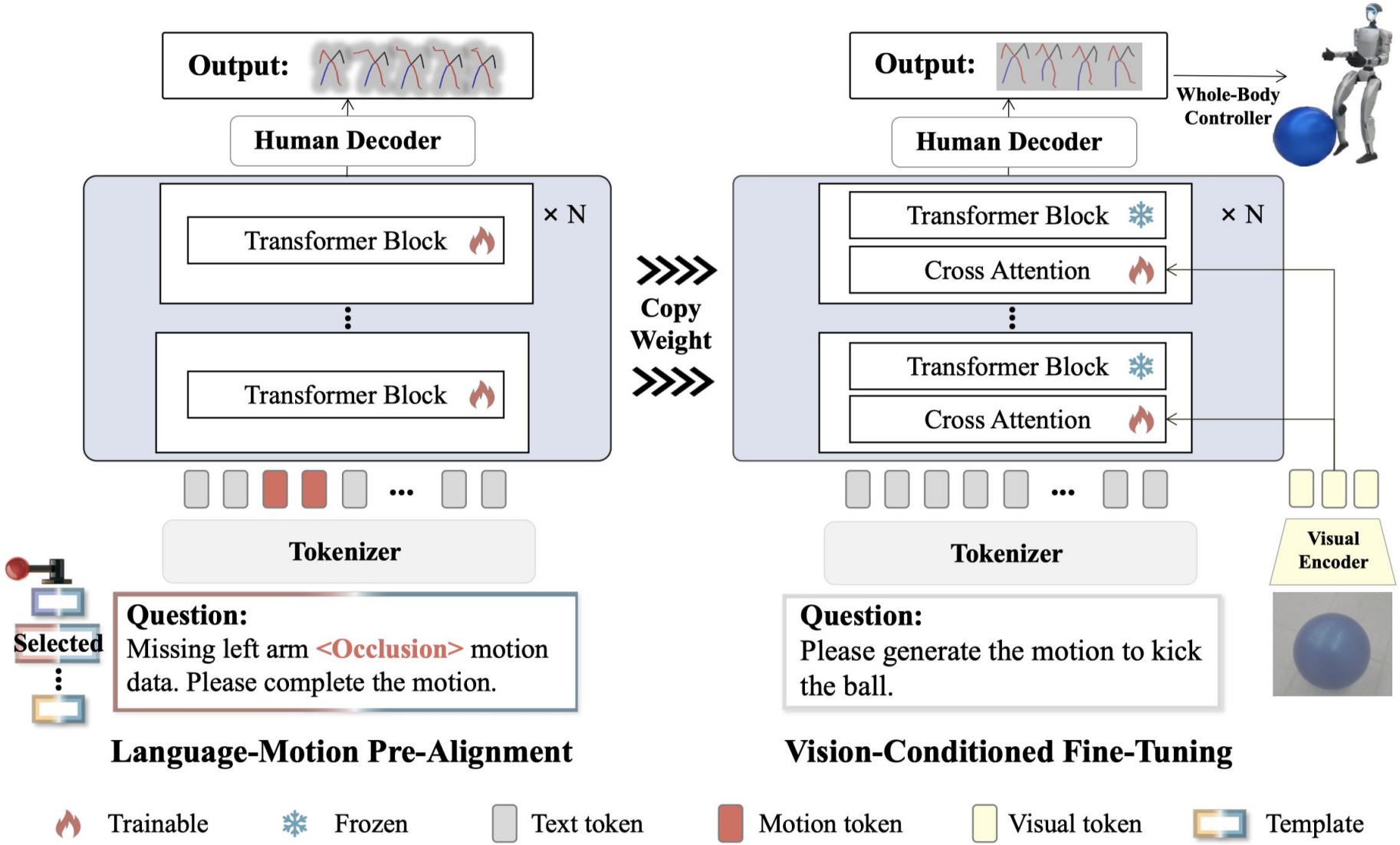


Figure 2: **Overview of Humanoid-VLA**. Humanoid-VLA includes two main parts: language-Motion Pre-alignment and vision-conditioned fine-tuning.

Georgia Tech

Language-Motion Pre-Alignment

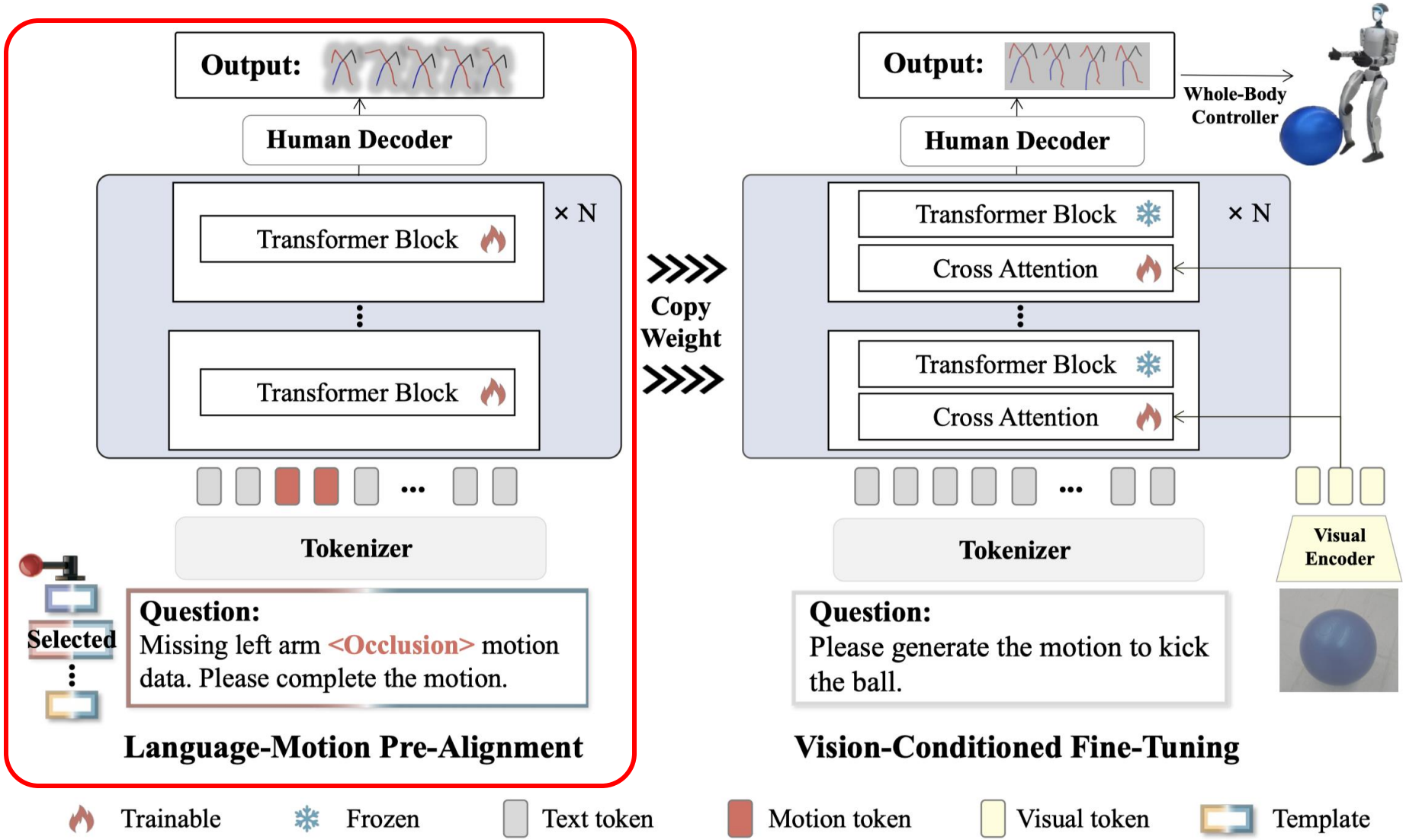
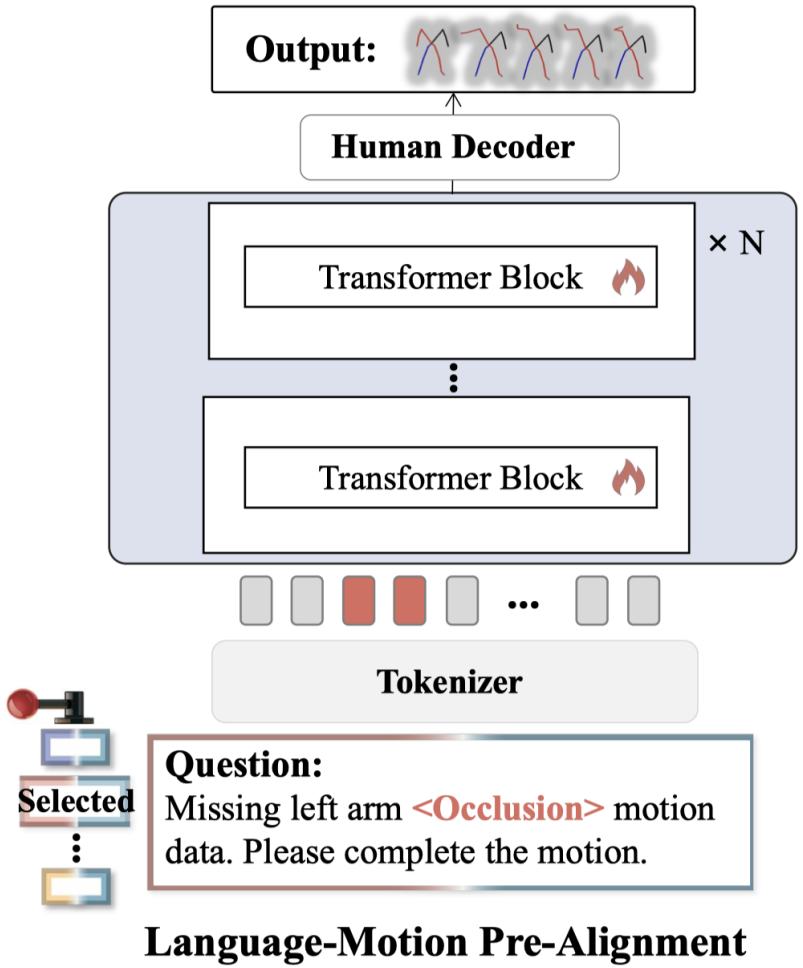


Figure 2: **Overview of Humanoid-VLA**. Humanoid-VLA includes two main parts: language-Motion Pre-alignment and vision-conditioned fine-tuning.

Georgia Tech

Methodology: Language-Motion Pre-Alignment



Language-Motion Pre-Alignment

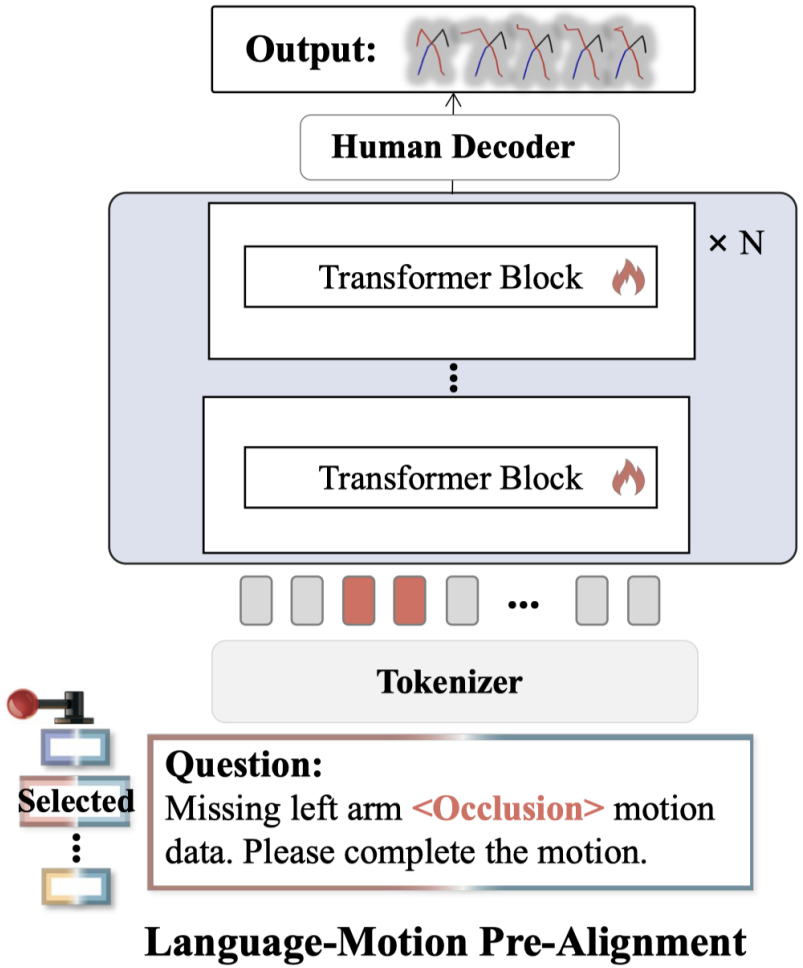
Goal:
Align non-egocentric human motion data with language descriptions

Purpose:
- ✓ Learn motion patterns & action semantics from large-scale motion data
- ✓ Enable motion-language learning without requiring egocentric visual input

Outcome:
Provides a foundation for motion generation and understanding

Georgia Tech

Methodology: Language-Motion Pre-Alignment



Language-Motion Pre-Alignment

- ✓ LLM maps input conditions to motion sequences
- ✓ Self-supervised augmentations & compositional encoding: enable seamless embedding of motion + text

  - Example instruction:
  - "Plan a sequence of actions ending with <State> over <Time> seconds."
  - <State> = discrete motion token, <Time> = temporal motion duration

1. Combine motion and language into shared codebook:

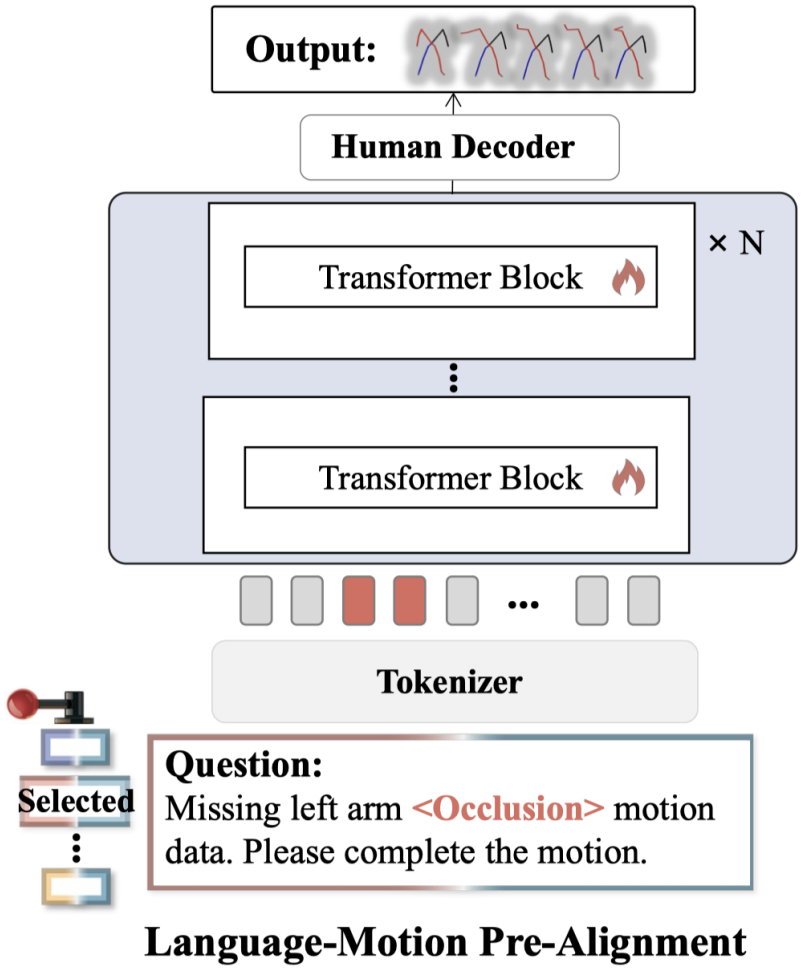
$$V = \{V_m, V_l\}$$

2. Encode both motion z_t and temporal representations d_t into token sequence $X_d = \{x_i^d\}_{i=1}^N$

- ✓ Enables LLMs to process mixed motion-language inputs

Georgia Tech

Methodology: Language-Motion Pre-Alignment



Language-Motion Pre-Alignment

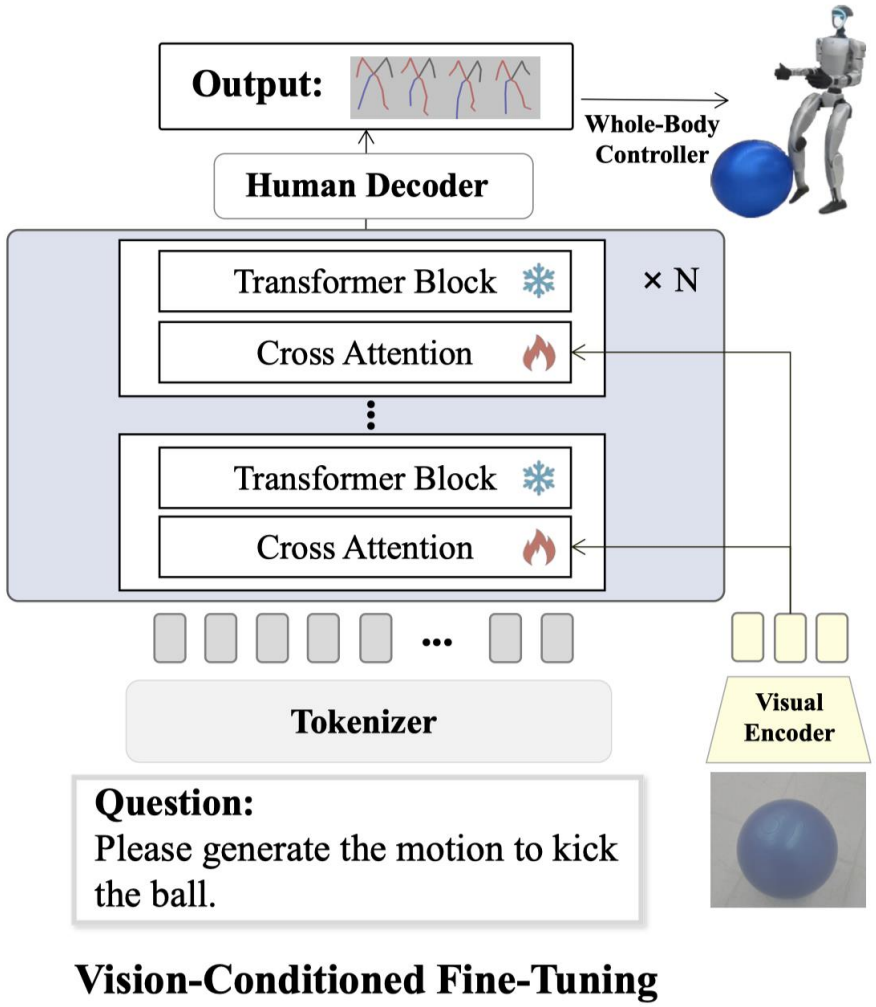
1. Model predicts next motion token x_o^i given prior context (similar to language modeling)
2. Training objective:

$$\mathcal{L}_{LLM} = -\sum_i \log p(x_o^i | x_o^{<i}, x_d)$$

3. Generated output sequence → reconstructs discrete motion

$$S = \{s_t\}_{t=1}^{T}$$

Methodology: Vision-Conditioned Fine-Tuning

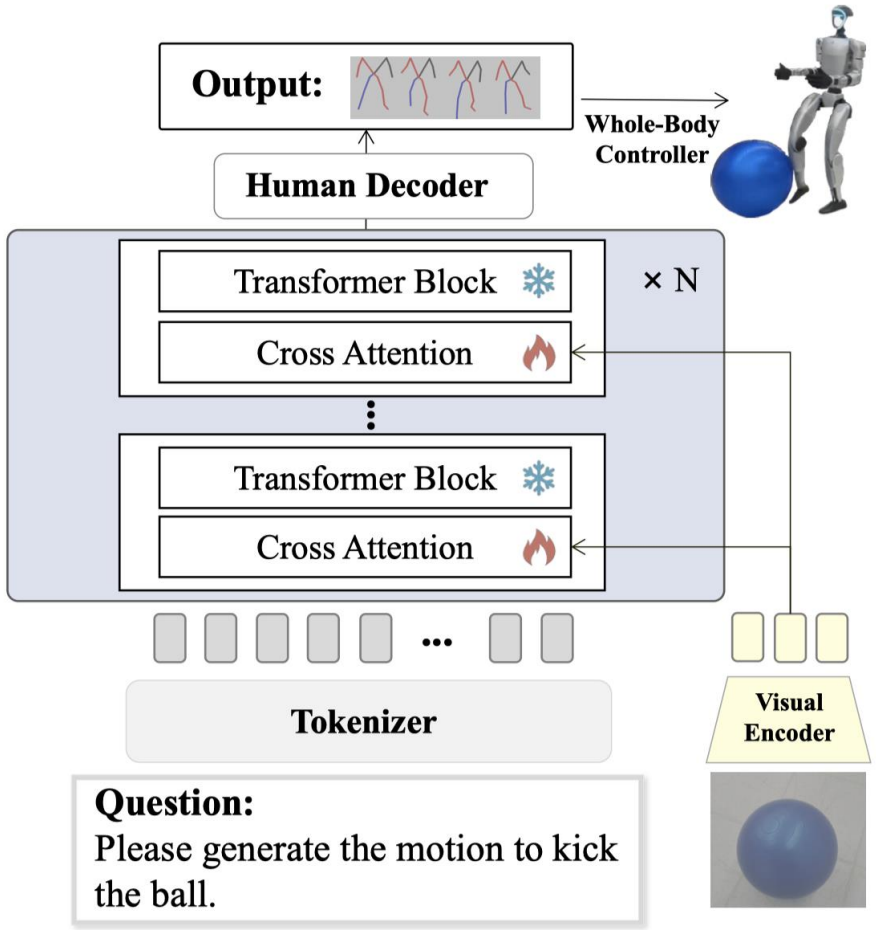


Vision-Conditioned Fine-Tuning

Goal and Purpose:

- ✓ Adds egocentric visual information for object-aware behavior
- ✓ Collect real-world Mocap + visual data
- ✓ Transfer language-motion alignment to vision-grounded humanoid tasks

Methodology: Vision-Conditioned Fine-Tuning



Vision-Conditioned Fine-Tuning

Cross-Attention Fusion for Vision-Language

1. Freeze transformer layers from pre-alignment phase
2. Add vision encoder + cross-attention layers
3. For each layer l :

$$Q_l = X_d^l W_Q^l, \quad K_l = X_v^l W_K^l, \quad V_l = X_v^l W_V^l$$

$$X_u^l = \text{Softmax}\left(\frac{Q_l K_l^T}{\sqrt{D}}\right) V_l$$

4. Combines visual features X_v + language features X_d into unified embedding X_u

Fine-Tuning Objective

- Optimize loss function same as pre-alignment:
  - Maximize consistency between predicted & ground-truth motion
  - Maintain temporal & semantic alignment
- Output: Vision-conditioned motion-language transformer

Experiments

Quantitative Evaluation — Kinematic Fidelity

Datasets:
- HumanML3D: locomotion tasks (run, swim, dance)
- Humanoid-S: complex, manually annotated actions (4646 clips)

Metrics:
- FID↓ : distribution similarity (lower -> better realism)
- DIV↑ : motion diversity (higher -> richer motion)

Baselines:
- MDM (diffusion), T2M-GPT (transformer + VQ-VAE)

Results:
- Humanoid-VLA achieves lowest FID (0.467), highest DIV (4.585), +47.5% improvement over MDM, +12% over T2M-GPT

Method	HumanML3D		Humanoid-S	
	FID↓	DIV↑	FID↓	DIV↑
MDM	$0.889^{\pm.026}$	$3.855^{\pm.053}$	$2.351^{\pm.590}$	$4.111^{\pm.261}$
T2M-GPT	$0.531^{\pm.020}$	$4.555^{\pm.058}$	$1.101^{\pm.189}$	$4.199^{\pm.218}$
Humanoid-VLA	$\mathbf{0.467}^{\pm.018}$	$\mathbf{4.585}^{\pm.086}$	$\mathbf{1.037}^{\pm.147}$	$\mathbf{4.466}^{\pm.213}$

Table 3: **Kinematic fidelity of generated motion in HumanML3D and Humanoid-S.** We use FID score and Diversity to evaluate the quality of the motion generated by the model, where bold values indicate the best results.

Experiments

Quantitative Evaluation — Kinematic Fidelity

- **Setup:**
  Evaluated in IsaacGym simulator
  Measures how well humanoid executes generated trajectories

Results:

- Joint errors < 40mm, best 31.07mm under medium difficulty
- Empjpe=1.18, Eaccel=27.84, Evel=14.76
- Demonstrates smooth & physically consistent motion

Ablation:

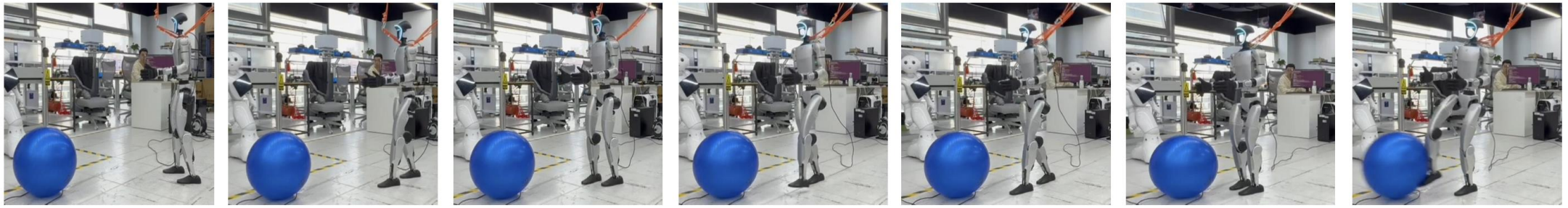
- Adding large-scale video data improves FID from 0.557 to 0.467 (+16%)
- Confirms effectiveness of self-supervised data augmentation

Types	Input	Accuracy			
		$E^{g}_{\text{mpjpe}} \downarrow$	$E^{\text{pa}}_{\text{mpjpe}} \downarrow$	$E_{\text{accel}} \downarrow$	$E_{\text{vel}} \downarrow$
Easy	D	36.13	1.53	34.42	18.73
	T	36.57	1.48	35.10	18.53
	A	39.02	1.32	34.32	17.91
	S_n	36.29	1.55	34.93	18.88
Medium	D + T	31.07	1.18	27.84	14.76
	D + A	36.98	1.30	34.87	18.16
	D + S_n	35.75	1.18	33.41	17.18
Hard	D + S_1 + S_N	37.14	1.34	34.69	18.08

Low-quality data w aug	High-quality Data w aug	FID↓	DIV↑
✓		$0.698^{\pm.037}$	$4.576^{\pm.098}$
	✓	$0.557^{\pm.016}$	$3.867^{\pm.062}$
✓	✓	$\mathbf{0.467}^{\pm.018}$	$\mathbf{4.585}^{\pm.086}$

Experiments

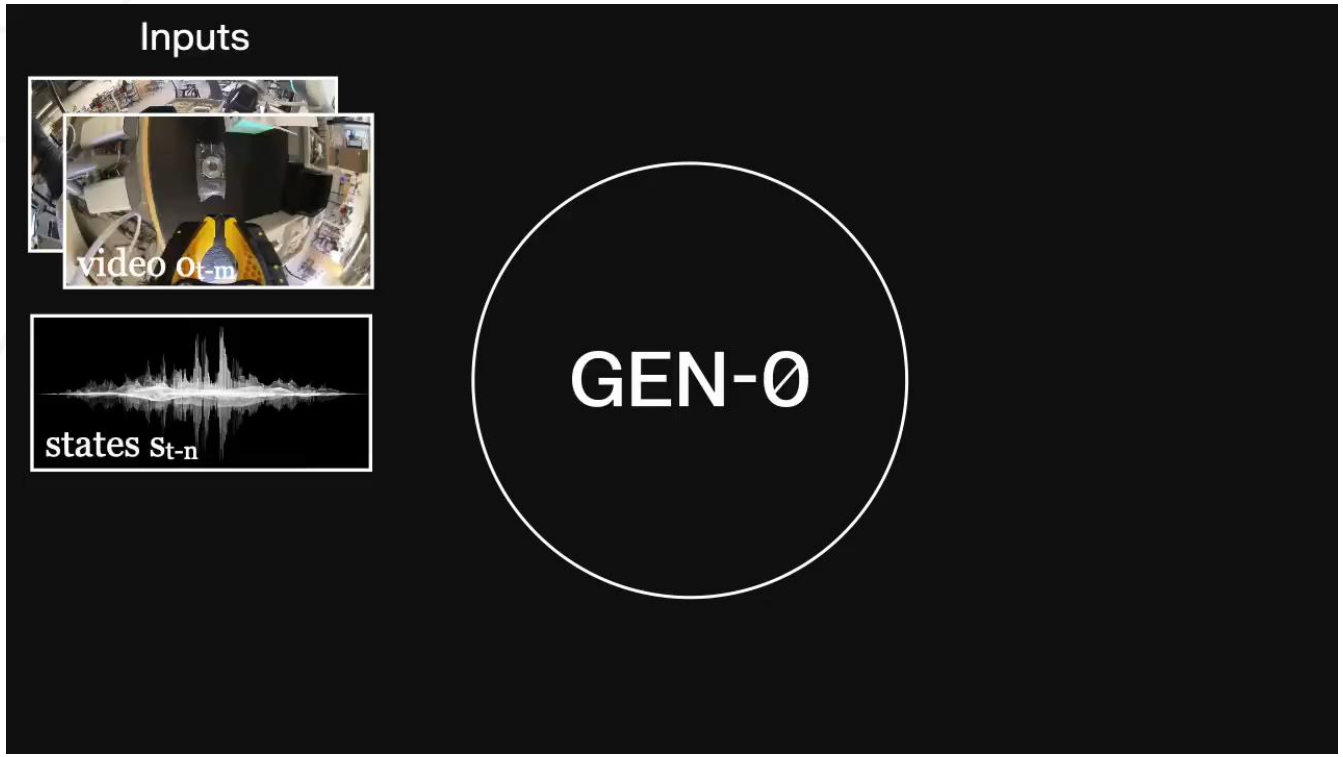
Kick Ball



Avoid Obstable

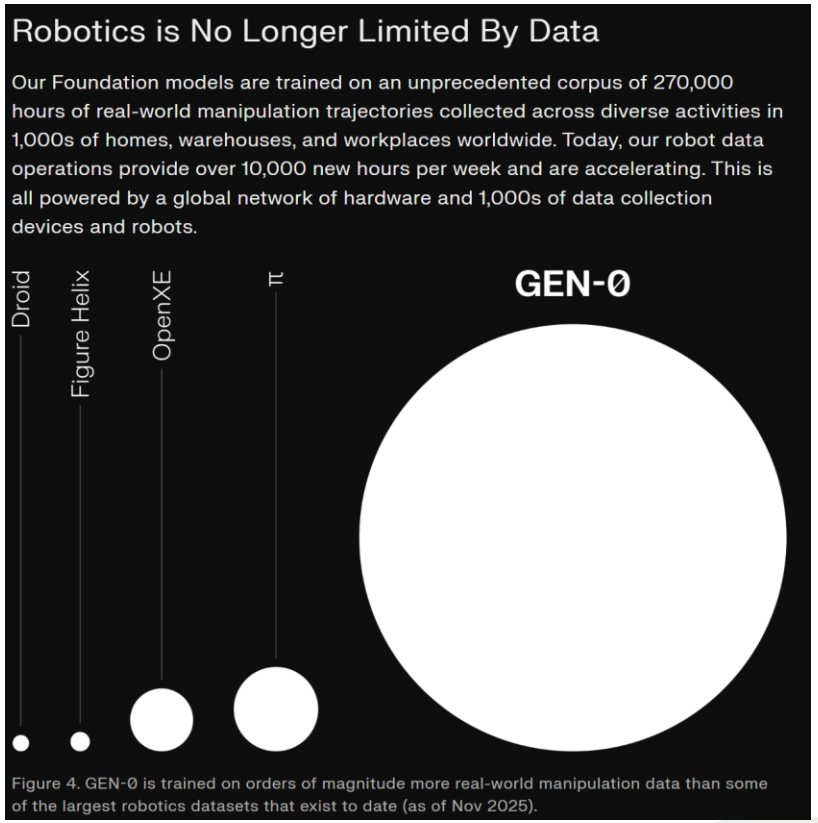
Last week …

GEN-0 / Embodied Foundation Models That Scale with Physical Interaction



Robotics is No Longer Limited By Data

—— really?

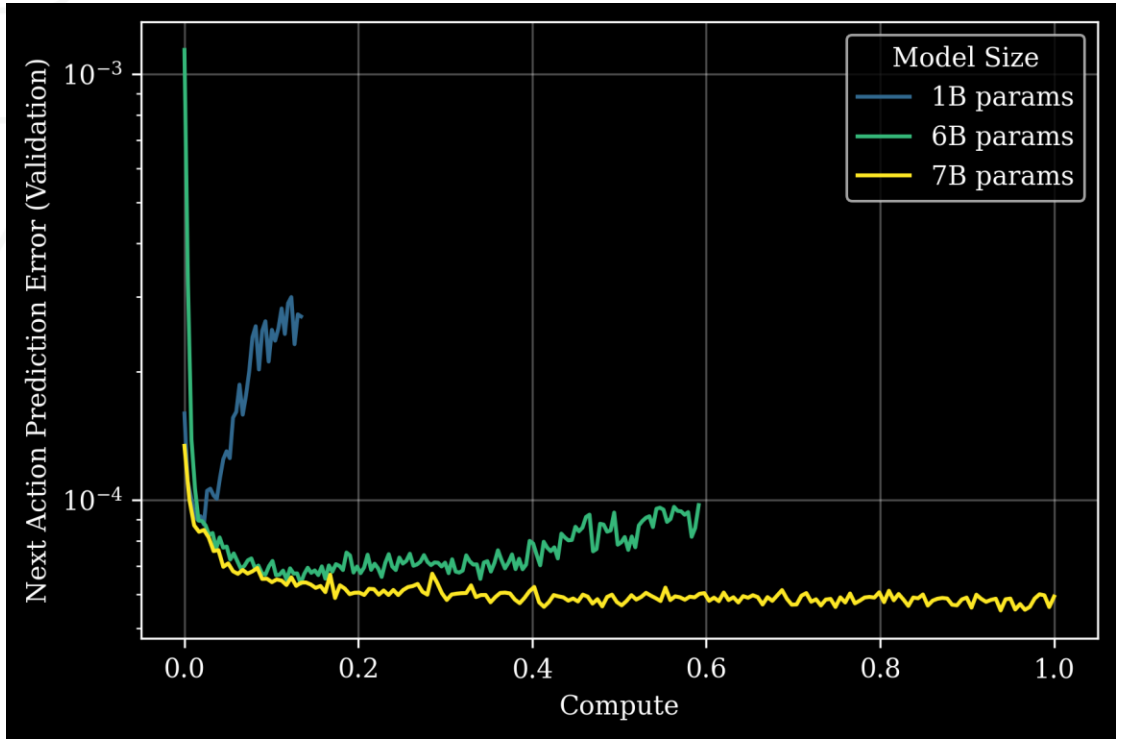


Generalist AI Team, "GEN-0: Embodied Foundation Models That Scale with Physical Interaction", Generalist AI Blog, Nov 2025.

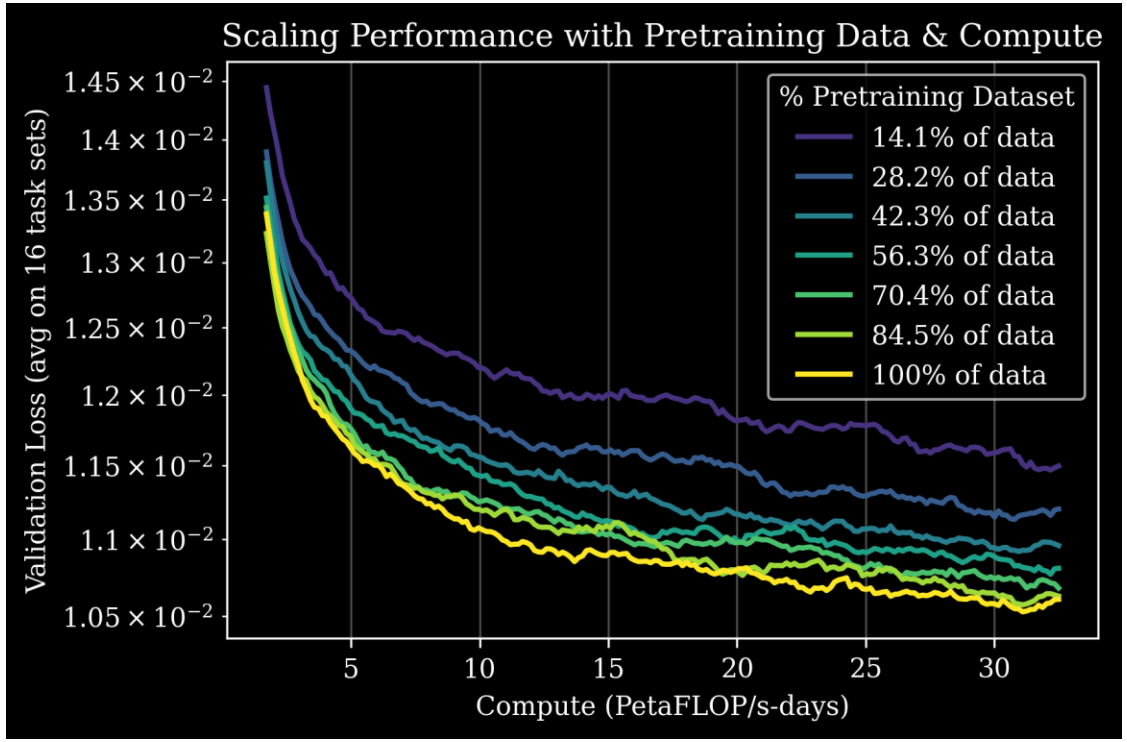
Last week …

GEN-0 / Embodied Foundation Models That Scale with Physical Interaction

Surpassing the Intelligence Threshold



Scaling Laws for Robotics



Generalist AI Team, "GEN-0: Embodied Foundation Models That Scale with Physical Interaction", Generalist AI Blog, Nov 2025.

Discussions

- *Will humanoid remain research platforms, or can they evolve into truly useful co-worker in read world (if so, when)? Is the bottleneck mainly in hardware or in the intelligence layer?*
- *Do we really need a human-like body to achieve general-purpose intelligence? Could other embodiments achieve better efficiency and scalability?*

- *Synthetic Data Dependence —*

How reliable are synthetic data sources?

Can they ever replace real robot demonstrations?

- *Is Data Alone the Ultimate Solution? —*

Will scaling data and compute alone eventually solve general-purpose robotics?

Where should model-based methods sit in the era of data-driven robotics?