GR00T N1: An Open Foundation Model for Generalist Humanoid Robots

NVIDIA[1]

Abstract

General-purpose robots need a versatile body and an intelligent mind. Recent advancements in humanoid robots have shown great promise as a hardware platform for building generalist autonomy in the human world. A robot foundation model, trained on massive and diverse data sources, is essential for enabling the robots to reason about novel situations, robustly handle real-world variability, and rapidly learn new tasks. To this end, we introduce GR00T N1, an open foundation model for humanoid robots. GR00T N1 is a Vision-Language-Action (VLA) model with a dual-system architecture. The vision-language module (System 2) interprets the environment through vision and language instructions. The subsequent diffusion transformer module (System 1) generates fluid motor actions in real time. Both modules are tightly coupled and jointly trained end-to-end. We train GR00T N1 with a heterogeneous mixture of real-robot trajectories, human videos, and synthetically generated datasets. We show that our generalist robot model GR00T N1 outperforms the state-of-the-art imitation learning baselines on standard simulation benchmarks across multiple robot embodiments. Furthermore, we deploy our model on the Fourier GR-1 humanoid robot for language-conditioned bimanual manipulation tasks, achieving strong performance with high data efficiency.

1. Introduction

Creating autonomous robots to perform everyday tasks in the human world has long been a fascinating goal and, at the same time, a significant technical undertaking. Recent progress in robotic hardware, artificial intelligence, and accelerated computing has collectively paved the ground for developing general-purpose robot autonomy. To march toward human-level physical intelligence, we advocate for a full-stack solution that integrates the three key ingredients: hardware, models, and data. First and foremost, robots are embodied physical agents, and their hardware determines their capability envelope. It makes **humanoid robots** a compelling form factor to build robot intelligence due to their human-like physique and versatility. Second, the diversity and variability of the real world demands that the robots operate on open-ended objectives and perform a wide range of tasks. Achieving this requires a **generalist robot model** sufficiently expressive and capable of handling various tasks. Third, real-world humanoid data are costly and time-consuming to acquire at scale. We need an effective data strategy to train large-scale robotic models.

In recent years, foundation models have brought forth dramatic breakthroughs in understanding and generating visual and text data. They demonstrate the effectiveness of training generalist models on web-scale data to enable strong generalization and fast adaptation to downstream tasks. The successes of foundation models in neighboring fields of AI have depicted a promising roadmap for building the "backbone" of intelligence for generalist robots, endowing them with a set of core competencies and enabling them to rapidly learn and adapt in the real world. However, unlike the digital realms of words and pixels, no Internet of humanoid robot datasets exist for large-scale pre-training. The data available for any single humanoid hardware would be orders of magnitude too small. Recent efforts in the robot learning community (Open X-Embodiment Collaboration et al., 2024) have explored cross-embodied learning to enlarge the dataset by pooling training data from many different robots. However, the great variability in robot embodiments, sensors, actuator degrees of freedom,

[1]A detailed list of contributors and acknowledgments can be found in App. A of this paper.

control modes, and other factors result in an archipelago of "data islands" rather than a coherent, Internet-scale dataset needed for training a true generalist model.

We introduce GR00T N1, an open foundation model for generalist humanoid robots. The GR00T N1 model is a Vision-Language-Action (VLA) model, which generates actions from image and language instruction input. It has cross-embodiment support from tabletop robot arms to dexterous humanoid robots. It adopts a **dual-system compositional architecture**, inspired by human cognitive processing (Kahneman, 2011). The System 2 reasoning module is a pre-trained Vision-Language Model (VLM) that runs at 10Hz on an NVIDIA L40 GPU. It processes the robot's visual perception and language instruction to interpret the environment and understand the task goal. Subsequently, a Diffusion Transformer, trained with action flow-matching, serves as the System 1 action module. It cross-attends to the VLM output tokens and employs embodiment-specific encoders and decoders to handle variable state and action dimensions for motion generation. It generates closed-loop motor actions at a higher frequency (120Hz). Both the System 1 and System 2 modules are implemented as Transformer-based neural networks, tightly coupled and jointly optimized during training to facilitate coordination between reasoning and actuation.

Figure 1: **Data Pyramid for Robot Foundation Model Training.** GR00T N1's heterogeneous training corpora can be represented as a pyramid: data quantity decreases, and embodiment-specificity increases, moving from the bottom to the top.

To mitigate the "data island" problem mentioned earlier, we structure the VLA training corpora as a **data pyramid**, illustrated in Fig. 1. Rather than treating the training datasets as a homogeneous pool, we organize heterogeneous sources by scale: large quantities of web data and human videos lay the base of the pyramid; synthetic data generated with physics simulations and/or augmented by off-the-shelf neural models form the middle layer, and real-world data collected on the physical robot hardware complete the top. The lower layers of the pyramid provide broad visual and behavioral priors, while the upper layers ensure grounding in embodied, real-robot execution.

We develop an effective **co-training** strategy to learn across the entire data pyramid in both pre- and post-training phases. To train our model with action-less data sources, such as human videos and neural-generated videos, we learn a latent-action codebook (Ye et al., 2025) and also use a trained inverse dynamics model (IDM) to infer pseudo-actions. These techniques enable us to annotate actions on action-less videos so we can effectively treat them as additional robot embodiments for model training. By unifying all data sources across the data pyramid, we construct a consistent dataset where the input consists of the robot state, visual observations, and language instruction, and the output is the corresponding motor action. We pre-train our model end-to-end across the three data layers, spanning (annotated) video datasets, synthetically generated datasets, and real-robot trajectories — by sampling training batches across this heterogeneous data mixture.

With a unified model and single set of weights, GR00T N1 can generate diverse manipulation behaviors using single-arm, bimanual, and humanoid embodiments. Evaluated on standard simulation benchmark environments, GR00T N1 achieves superior results compared to state-of-the-art imitation learning baselines. We also demonstrate GR00T N1's strong performance in real-world experiments with GR-1 humanoid robots. Our GR00T-N1-2B model checkpoint, training data, and simulation benchmarks are publicly available here: GitHub and HuggingFace Datasets.

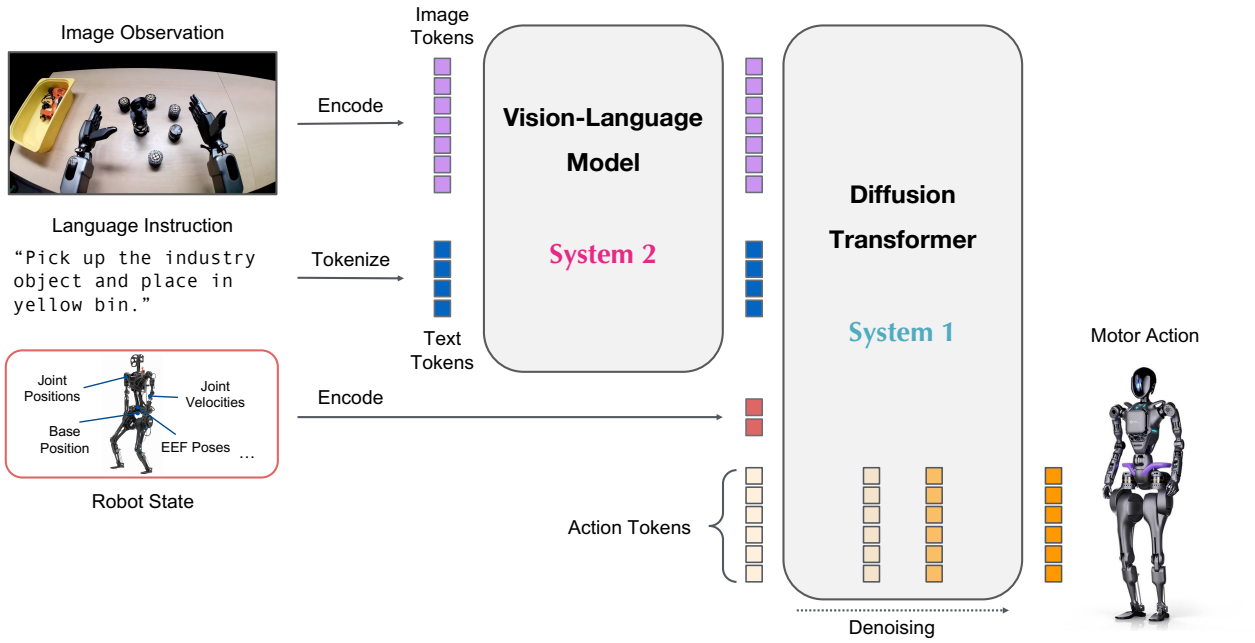


Figure 2: **GR00T N1 Model Overview.** Our model is a Vision-Language-Action (VLA) model that adopts a dual-system design. We convert the image observation and language instruction into a sequence of tokens to be processed by the Vision-Language Model (VLM) backbone. The VLM outputs, together with robot state and action encodings, are passed to the Diffusion Transformer module to generate motor actions.

2. GR00T N1 Foundation Model

GR00T N1 is a Vision-Language-Action (VLA) model for humanoid robots trained on diverse data sources. The model contains a vision-language backbone that encodes language and image input and a DiT-based flow-matching policy that outputs high-frequency actions. We use the NVIDIA Eagle-2 VLM (Li et al., 2025) as the vision-language backbone. Specifically, our publicly released GR00T-N1-2B model has 2.2B parameters in total, with 1.34B in the VLM. The inference time for sampling a chunk of 16 actions is 63.9ms on an L40 GPU using bf16. Fig. 2 provides a high-level overview of our model design. We highlight three key features of GR00T N1:

- We design a compositional model that integrates Vision-Language Model (VLM)-based reasoning module (System 2) and Diffusion Transformer (DiT)-based action module (System 1) in a unified learning framework;
- We develop an effective pre-training strategy using a mixture of human videos, simulation and neural-generated data, and real robot demonstrations (see Fig. 1) for generalization and robustness;
- We train a massively multi-task, language-conditioned policy that supports a wide range of robot embodiments and enables rapid adaptation to new tasks through data-efficient post-training.

2.1. Model Architecture

In this section, we describe the GR00T N1 model architecture, illustrated in Fig. 3. GR00T N1 uses flow-matching (Lipman et al.) to learn action generation. A diffusion transformer (DiT) processes the robot's proprioceptive state and action, which are then cross-attended with image and text tokens from the Eagle-2 VLM backbone to output the denoised motor actions. Below, we elaborate on each module in detail.

State and Action Encoders

To process states and actions of varying dimensions across different robot embodiments, we use an MLP per embodiment to project them to a shared embedding dimension as input to the DiT. As in Black et al. (2024),

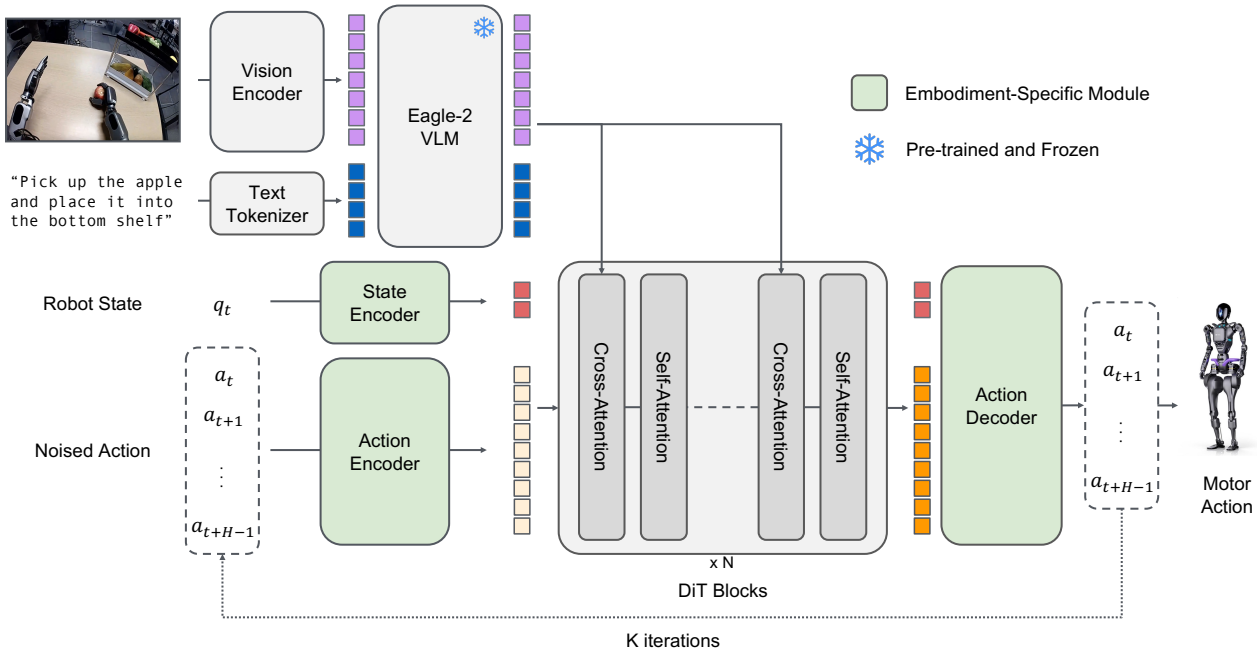


Figure 3: **GR00T N1 Model Architecture.** GR00T N1 is trained on a diverse set of embodiments ranging from single-arm robot arms to bimanual humanoid dexterous hands. To deal with different robot embodiment's state observation and action, we use DiT blocks with an embodiment-aware state and action encoder to embed the robot's state and action inputs. GR00T N1 model leverages latent embeddings of the Eagle-2 model to incorporate the robot's visual observation and language instructions. The vision language tokens will then be fed into the DiT blocks through cross-attention layers.

the Action Encoder MLP also encodes the diffusion timestep together with the noised action vector.

We use action flow matching, which samples actions through iterative denoising. The model takes as input noised actions in addition to encodings of the robot's proprioceptive state, image tokens, and text tokens. The actions are processed in chunks as in Zhao et al. (2023), meaning that at any given time t the model uses $A_t = [a_t, a_{t+1}, \ldots, a_{t+H-1}]$ which contains the action vectors of timesteps t through $t + H - 1$. We set $H = 16$ in our implementation.

Vision-Language Module (System 2)

For encoding vision and language inputs, GR00T N1 uses the Eagle-2 (Li et al., 2025) vision-language model (VLM) pretrained on Internet-scale data. Eagle-2 is finetuned from a SmolLM2 (Allal et al., 2025) LLM and a SigLIP-2 (Tschannen et al., 2025) image encoder. Images are encoded at resolution 224×224 followed by pixel shuffle (Shi et al., 2016), resulting in 64 image token embeddings per frame. These embeddings are then further encoded together with text by the LLM component of the Eagle-2 VLM. The LLM and image encoder are aligned over a broad set of vision-language tasks following the general recipe of Li et al. (2025).

During policy training, a text description of the task, as well as (possibly multiple) images, are passed to the VLM in the chat format used during vision-language training. We then extract vision-language features of shape (batch size $\times$ sequence length $\times$ hidden dimension) from the LLM. We found that using middle-layer instead of final-layer LLM embeddings resulted in both faster inference speed and higher downstream policy success rate. For GR00T-N1-2B, we use the representations from the 12th layer.

Diffusion Transformer Module (System 1)

For modeling actions, GR00T N1 uses a variant of DiT (Peebles and Xie, 2023), which is a transformer with denoising step conditioning via adaptive layer normalization, denoted as V_θ. As shown in Fig. 3, V_θ consists of

alternating cross-attention and self-attention blocks, similar to Flamingo (Alayrac et al., 2022) and VIMA (Jiang et al., 2023). The self-attention blocks operate on noised action token embeddings A_t^τ together with state embeddings q_t, while cross-attention blocks allow conditioning on the vision-language token embeddings ϕ_t output by VLM. After the final DiT block, we apply an embodiment-specific Action Decoder, another MLP, to the final H tokens to predict the actions.

Given a ground-truth action chunk A_t, a flow-matching timestep $\tau \in [0, 1]$ and sampled noise $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the noised action chunk A_t^τ is computed as $A_t^\tau = \tau A_t + (1 - \tau)\epsilon$. The model prediction $V_\theta(\phi_t, A_t^\tau, q_t)$ aims to approximate the denoising vector field $\epsilon - A_t$ by minimizing the following loss:

$$\mathcal{L}_{fm}(\theta) = \mathbb{E}_\tau \left[\| V_\theta(\phi_t, A_t^\tau, q_t) - (\epsilon - A_t) \|^2 \right]. \tag{1}$$

As in Black et al. (2024), we use $p(\tau) = \text{Beta}(\frac{s-\tau}{s}; 1.5, 1)$, $s = 0.999$. During inference, we generate action chunks with K-step denoising. First, randomly sample $A_t^0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and then use forward Euler integration to iteratively generate the action chunk, updating as follows:

$$A_t^{\tau + 1/K} = A_t^\tau + \frac{1}{K} V_\theta(\phi_t, A_t^\tau, q_t).$$

In practice, we found $K = 4$ inference steps to work well across all embodiments.

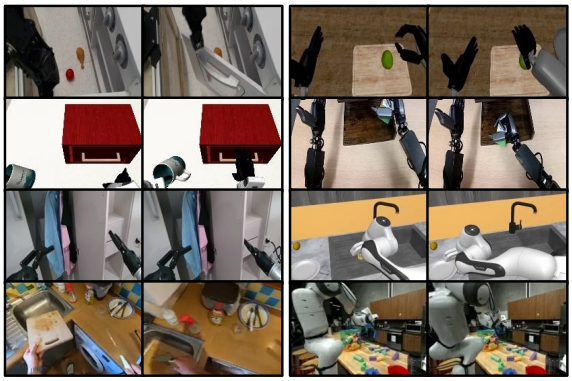
2.2. Training Data Generation

To train GR00T N1, we use a diverse set of data sources and objectives to construct the data pyramid (Fig. 1). We first source diverse human egocentric video data from open datasets, which forms the base, together with the web data used in VLM pretraining. Next, we generate synthetic *neural* trajectories using pre-trained video generation models. In this way, we $\sim$10$\times$ our in-house collected teleoperation trajectories — the "peak" of the data pyramid — from 88 hours to 827 hours, using diverse counterfactual robot trajectories with novel language instructions (see Fig. 5 for examples). We additionally generate diverse simulation trajectories, which also expand the middle part of the data pyramid.

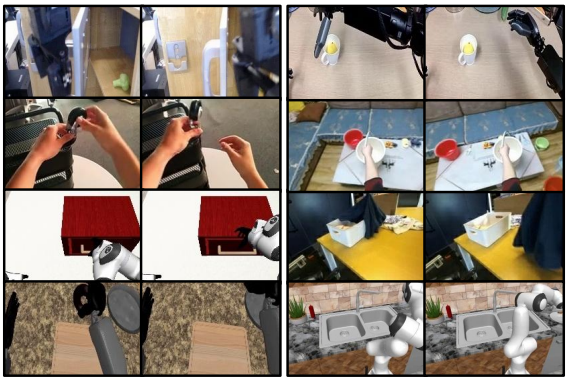
In the next paragraph, we first describe how we extract *latent* actions from videos, which we use to extract labels for web-scaled human egocentric datasets. Next, we describe how we generate *neural* and *simulated* robot trajectories, and how we obtain actions for each of these data sources.

Latent Actions

For human egocentric videos and neural trajectories, we do not have any actions that we can directly use to train GR00T N1. For these data, we instead generate latent actions by training a VQ-VAE model to extract features from consecutive image frames from videos (Ye et al., 2025). The encoder takes the current frame x_t and the future frame x_{t+H} of a video with a fixed window size H and outputs the latent action z_t. The decoder is trained to take the latent action z_t and x_t and reconstruct x_{t+H}. This model is trained with a VQ-VAE objective, where the continuous embedding from the encoder is mapped to the nearest embedding from the codebook. After training, we take the encoder and use it as an inverse dynamics model; given an x_t and x_{t+H} pair, we extract the continuous pre-quantized embedding and use this as the latent action label during pre-training, with the same flow-matching loss, but treat it as a distinct "LAPA" embodiment. Training the VQ-VAE model on all heterogeneous data together allows us to unify all of the data to share the same learned latent action space, potentially improving cross-embodiment generalization. Fig. 4 shows x_t and x_{t+H} pairs from 8 distinct embodiments including both robot and human embodiment, all retrieved from similar latent actions; the first latent action shows all embodiments *moving right arm to the left* and the second latent action shows *moving right arm to the right*.



Latent action 1: Move the right arm to the left



Latent action 2: Move the right arm to the right

Figure 4: **Latent Actions.** We retrieve similar latent embeddings across various embodiments. The left images illustrate the latent action that corresponds to moving the right arm (or hand) to the left, while the right images illustrate the latent action that corresponds to moving the right arm (or hand) to the right. Note that this general latent action is not only consistent in different robot embodiments, but also in human embodiment.

Neural Trajectories

Robot data scales linearly with human labor, since it typically requires a human operator to teleoperate the robot to produce each trajectory. Recently, **video generation models** have demonstrated significant potential for high-quality controllable video generation (Brooks et al., 2024; Lin et al., 2024; Ren et al., 2025; Wan Team, 2025; Xiang et al., 2024; Yang et al., 2024), which paves the way for building world models in the robotic domain. To harness these models, we fine-tune image-to-video generation models (Agarwal et al., 2025; Wan Team, 2025; Yang et al., 2024) on all of our 88 hours of in-house collected teleoperation data and generate 827 hours of video data given the existing initial frames with novel language prompts, augmenting it by around 10×. This enables generating training data that captures many more counterfactual scenarios in the real world without actually collecting teleoperation data for each of these cases (examples shown in Fig. 5; more examples of dream generations in Fig. 14).

To increase the diversity of our neural trajectories, we first use a commercial-grade multimodal LLM to detect the objects given initial frames and generate many more possible combinations of "*pick up* {*object*} *from* {*location A*} *to* {*location B*}", while instructing the model to only consider the physically feasible combinations. We also apply post-processing mechanisms, including filtering and re-captioning, to the generated videos. For this, we also use a commercial-grade multimodal LLM as a judge and feed the downsampled 8 frames to filter out neural trajectories that do not follow the language instruction precisely. We then caption the filtered-out videos. (More details can be found in Appendix E).

Simulation Trajectories

Scaling up real-world data collection for humanoid robots is highly expensive due to the challenge of simultaneously controlling both arms and dexterous hands. Recent research (Jiang et al., 2024; Mandlekar et al., 2023; Wang et al., 2024) has demonstrated that generating training data in simulation is a practical alternative. We use DexMimicGen (Jiang et al., 2024) to synthesize large-scale robot manipulation trajectories.

Starting with a small set of human demonstrations, DexMimicGen applies demonstration transformation and replay in simulation to expand the dataset automatically. Each task is decomposed into a sequence of object-centric subtasks. The initial human demonstrations are segmented into smaller manipulation sequences, each corresponding to a subtask involving a single object. These segments are then adapted to new environments by aligning them with the object's position, preserving the relative poses between the robot's end effector and the object. To ensure smooth execution, the system interpolates movements between the robot's current state and the transformed segment. The robot then follows the full sequence step by step, verifying task success at the

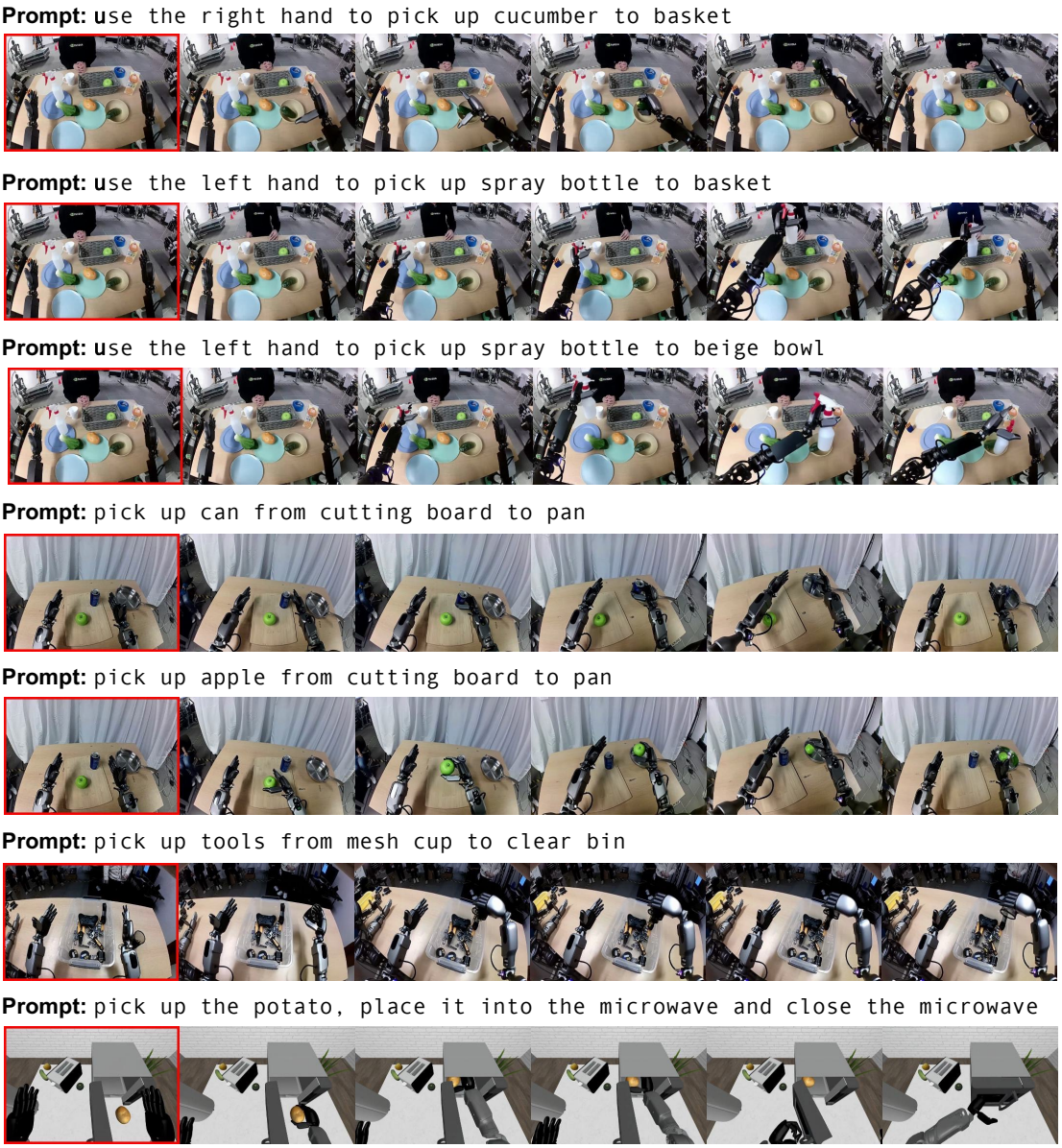


Figure 5: **Synthetically Generated Videos.** We leverage off-the-shelf video generation models to create neural trajectories to increase the quantity and diversity of our training datasets. These generated data can be used for both pre- and post-training of our GR00T N1. (1) The first three rows are generated from the same initial frames but with different prompts (change left or right, the location to place the object), (2) the following two are from the same initial frames but replace the object to pick up, (3) the next row showcases the video model generating a robot trajectory which is very challenging to generate in simulation (spilling contents inside a mesh cup into a bin), and (4) the last row is generated from an initial frame from simulation data. We use the red rectangles to indicate the initial frames.

end. Only successful demonstrations are retained, ensuring high-quality data. Using DexMimicGen, we scale a limited set of human demonstrations into a large-scale humanoid manipulation dataset. Considering the pre- and post-training datasets, we have generated 780,000 simulation trajectories — equivalent to 6,500 hours, or nine continuous months, of human demonstration data — in just 11 hours. These simulation data significantly supplement the real-robot data with minimal human costs.

2.3. Training Details

Pre-training

During the pre-training phase, GR00T N1 is trained via flow-matching loss (Equation 1) on a diverse collection of embodiments and data sources, encompassing various real and synthetic robot datasets as well as human motion data. We refer readers to Sec. 3 for a detailed description of the datasets.

For human videos, in the absence of ground-truth actions, we extract learned latent actions and use them as flow-matching targets (see Sec. 2.2). For robot datasets such as our GR-1 humanoid data or Open X-Embodiment data, we use both ground-truth robot actions as well as learned latent actions as flow-matching targets. In the case of neural trajectories (Sec. 2.2) used to augment our robot datasets, we use both latent actions as well as predicted actions from an inverse-dynamics model trained on the real robot data. Pre-training hyper-parameters are listed in Table 6 in the Appendix.

Post-training

In the post-training phase, we fine-tune our pre-trained model on datasets corresponding to each single embodiment. As in pretraining, we keep the language component of the VL backbone frozen and fine-tune the rest of the model. Post-training hyperparameters are given in Table 6 in the Appendix.

Post-training with Neural Trajectories

To overcome the challenge of data scarcity during post-training, we explore augmenting the data for each downstream task by generating neural trajectories, similar to the procedure described in Sec. 2.2. For downstream tasks that are conditioned on multiple views, we finetune the video model to generate multiple subimages in a grid (Fig. 14). For simulation tasks, we collect diverse initial frames from the randomly initialized environment. For real robot tasks, we randomly initialize object poses manually and record the robot's initial observation. Novel initial frames could also be created automatically using img2img diffusion (example shown in Fig. 14), but we leave further exploration for future work. We also demonstrate examples of (1) multi-round video generation for generating long-horizon trajectories composed of atomic tasks and (2) neural trajectories of liquids and articulated objects, known to be extremely challenging to simulate, though we leave quantitative evaluation of downstream tasks for future work.

For our post-training pipeline with neural trajectories, we restrict ourselves to fine-tuning the video generation model *only* on the human-collected trajectories for simulation tasks and only 10% of the data from the real-world benchmark collected for post-training, to match the realistic scenario that we only have access to limited number of teleoperation data. Since the generated videos do not have action labels, we use either latent or inverse dynamics models (IDM) labeled actions (Baker et al., 2022) and train the policy model to treat these pseudo-actions as action labels for a different embodiment. In low-data regime scenarios, we also restrict ourselves on training the IDM models only on the low-data, to facilitate realistic scenarios. Details of how we train the IDM models are provided in Appendix E. Some empirical comparisons between latent and IDM-labeled actions are made in Sec. 4.4. During post-training, we co-train the policy with real-world trajectories with neural trajectories with a 1:1 sampling ratio.

Training Infrastructure

We train GR00T N1 on a cluster managed via NVIDIA OSMO (NVIDIA, 2025), an orchestration platform for scaling complex robotics workloads. The training cluster is equipped with H100 NVIDIA GPUs connected via NVIDIA Quantum-2 InfiniBand in a fat-tree topology. We facilitate fault-tolerant multi-node training and data ingestion via a custom library built on top of the Ray distributed computing library (Moritz et al., 2018). We use up to 1024 GPUs for a single model. GR00T-N1-2B used roughly 50,000 H100 GPU hours for pretraining.

Compute-constrained finetuning was tested in the context of a single A6000 GPU. If only tuning the adapter layers (action and state encoders + action decoder) and DiT, a batch size up to 200 can be used. When tuning the vision encoder, a batch size of up to 16 can be used.

3. Pre-Training Datasets

We structure our pre-training corpus into three main categories: real-robot datasets (Sec. 3.1), synthetic datasets (Sec. 3.2), and human video datasets (Sec. 3.3). These roughly correspond to the peak, middle, and base of the data pyramid (Fig. 1), respectively. The synthetic datasets consist of both simulation trajectories and neural trajectories. Table 1 summarizes our training data generation strategies in Sec. 2.2 and their applicable data sources correspondingly. We provide the full statistics (# of frames, hours, and camera views) of our pretraining datasets in Table 7.

Table 1: **Training Data Generation.** Our data generation strategies leverage different data sources. The latent-action learning technique is broadly applied to diverse video datasets. Neural trajectories can be generated from datasets containing robot actions, while simulation trajectories rely on a physics simulator and utilize our DexMimicGen-based automated data generation system.

	Latent Actions	Neural Trajectories	Simulation Trajectories
Real-Robot Datasets	✓	✓	✓
Simulated Robot Datasets	✓	✓	
Human Video Datasets	✓		

3.1. Real-World Datasets

We use the following real-world robot datasets:

1. **GR00T N1 Humanoid Pre-Training Dataset.** Our internally collected dataset covers a broad range of general manipulation tasks, focused on Fourier GR1 through teleoperation. We leverage the VIVE Ultimate Tracker to capture the teleoperator's wrist poses while Xsens Metagloves track finger movements. We also explored other teleoperation hardware options, including Apple Vision Pro and Leap Motion (see Fig. 6). The recorded human movements are then retargeted to humanoid actions via inverse kinematics. The real-time teleoperation operates at a control frequency of 20Hz. Alongside the robot's actions, we capture images from a head-mounted camera at each step, as well as the human's low-dimensional proprioception and actions. The dataset includes fine-grained annotations, which detail atomic actions such as grasping, moving, and placing, and coarse-grained annotations, which aggregate sequences of fine-grained actions into higher-level task representations. This hierarchical structure supports learning both precise motion control and high-level task reasoning.
2. **Open X-Embodiment.** Open X-Embodiment Collaboration et al. (2024) is a widely used cross-embodiment dataset for robot manipulation. We include the RT-1 (Brohan et al., 2022), Bridge-v2 (Walke et al., 2023), Language Table (Lynch et al., 2022), DROID (Khazatsky et al., 2024), MUTEX (Shah et al., 2023), RoboSet (Bharadhwaj et al., 2024) and Plex (Thomas et al., 2023), providing diverse datasets covering various manipulation tasks, language-conditioned control, and robot-environment interactions.
3. **AgiBot-Alpha.** AgiBot-World-Contributors et al. (2025) is a large-scale dataset of trajectories from 100 robots. We used the 140,000 trajectories available at the time of launching our training run. The dataset covers fine-grained manipulation, tool usage, and multi-robot collaboration.

3.2. Synthetic Datasets

Our synthetic datasets include 1) simulation trajectories automatically multiplied from a small number of human demonstrations within physics simulators and 2) neural trajectories derived from videos produced by off-the-shelf neural generation models.

Simulation Trajectories

In addition to real-world datasets, we feature large-scale synthetic datasets generated in simulation as described in Sec. 2.2. Our simulation tasks comprise humanoid robots performing a broad range of tabletop rearrangement

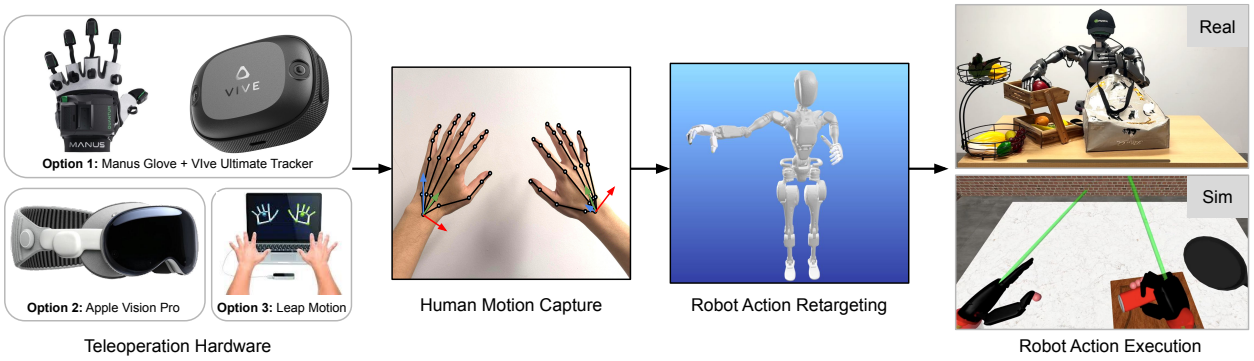


Figure 6: **Data Collection via Teleoperation.** Our teleoperation infrastructure supports multiple devices to capture human hand motion, including 6-DoF wrist poses and hand skeletons. Robot actions are produced through retargeting and executed on robots in real and simulation environments.

tasks and feature a large array of realistic 3D assets. We build these tasks under the RoboCasa simulation framework (Nasiriany et al., 2024). Broadly, our tasks follow the behavior "rearrange A from B to C", where A corresponds to an object, and B and C represent the source and target locations in the environment. The source and target locations are receptacles such as plates, baskets, placemats, and shelves, and the robot must rearrange objects across different combinations of source and target receptacles. Overall, our pre-training simulation datasets feature 54 unique combinations of source and target receptacle categories. We place the objects and receptacles in randomized locations throughout the table and additionally incorporate distractor objects and receptacles in the scene. The distractors require the model to pay attention to the task language to perform the desired behavior.

We generate diverse, high-quality training datasets at a massive scale using DexMimicGen. Our datasets feature the GR-1 humanoid robot, but we can adopt the system for a wide range of robots. We begin by collecting a few dozen source demonstrations via teleoperation using the Leap Motion device. The Leap Motion device tracks the 6-DoF wrist poses and finger poses, and we retarget these values and send them to the whole-body IK controller based on mink (Zakka, 2024). Given human demonstrations, DexMimicGen processes the demonstrations into object-centric segments and then transforms and combines these segments to generate new demonstrations. Using this system, we generate 10,000 new demonstrations for each (source, target) receptacle pair in our pre-training task regime, resulting in 540k total demonstrations.

Neural Trajectories

To generate neural trajectories, we fine-tune open-source image-to-video models on our real-world GR00T N1 Humanoid Pre-Training dataset, as described in Sec. 2.2. We trained the models for 100 epochs on a dataset comprising 3,000 real-world robot data samples with language annotations, each recorded at 480P resolution and consisting of 81 frames. As illustrated in Fig. 5, our model can generate high-quality counterfactual trajectories given novel language prompts. Moreover, the model, trained on Internet-scale video data, demonstrates strong generalization capabilities in handling unseen initial frames, novel objects, and new motion patterns. These videos are further labeled with latent actions and IDM-based pseudo-actions for model training. We generate a total of around 827 hours of videos; it takes 2 minutes to generate a one-second video on an L40 GPU, and required approximately 105k L40 GPU hours (∼1.5 days) on 3,600 L40 GPUs.

3.3. Human Video Datasets

We include a diverse set of human video datasets. These do not include explicit action labels but contain extensive sequences of human-object interactions, capturing affordances, task semantics, and natural motion patterns. These datasets cover a wide range of real-world human behaviors, including grasping, tool use, cooking, assembly, and other task-oriented activities performed in natural environments, and provide detailed

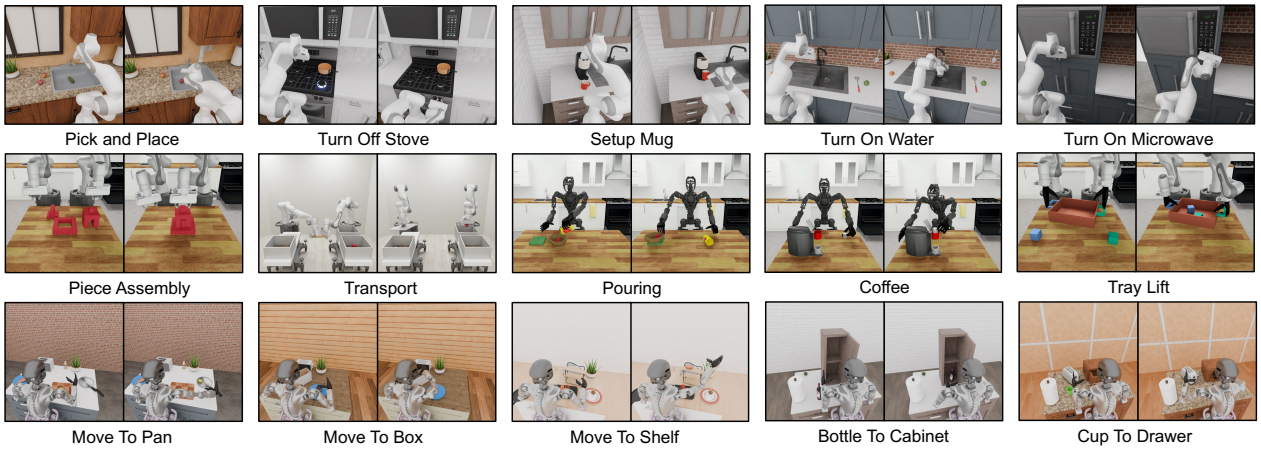


Figure 7: **Simulation Tasks.** Our simulation experiments use tasks from two open-source benchmarks (RoboCasa (Nasiriany et al., 2024) in the top row and DexMimicGen (Jiang et al., 2024) in the middle row) and a newly developed suite of tabletop manipulation tasks that closely resemble our real-world tasks (bottom row). We provide Omniverse renderings of the tasks above.

first-person perspectives of hand-object interactions (examples shown in Figure 11). Our video datasets include the following:

- **Ego4D** is a large-scale egocentric video dataset that includes diverse recordings of everyday activities (Grauman et al., 2022);
- **Ego-Exo4D** adds complementary exocentric (third-person) views alongside first-person recordings (Grauman et al., 2024);
- **Assembly-101** focuses on complex assembly tasks by providing detailed videos of step-by-step object assembly (Sener et al., 2022);
- **EPIC-KITCHENS** includes first-person footage of culinary activities (Damen et al., 2018);
- **HOI4D** captures human-object interactions with frame-wise annotations for segmentation, hand and object poses, and actions (Liu et al., 2022);
- **HoloAssist** captures collaborative and assistive tasks within augmented reality environments (Wang et al., 2023);
- **RH20T-Human** includes recordings of fine-grained manipulation tasks with an emphasis on natural hand-object interactions across diverse real-world scenarios (Fang et al., 2023).

4. Evaluation

We evaluate our GR00T N1 models in a diverse set of simulated and real-world benchmarks. Our simulation experiments are conducted on three distinct benchmarks designed to systematically assess the effectiveness of our model across various robot embodiments and manipulation tasks. In our real-world experiments, we investigate the model's capability on a suite of tabletop manipulation tasks with the GR-1 humanoid robot. These experiments aim to demonstrate GR00T N1's ability to acquire new skills from a limited number of human demonstrations.

4.1. Simulation Benchmarks

Our simulation experiments comprise two open-source benchmarks from prior work (Jiang et al., 2024; Nasiriany et al., 2024), as well as a newly developed suite of tabletop manipulation tasks designed to closely mirror our real-world task settings. We meticulously choose these benchmarks for evaluating our models across different robot embodiments and diverse manipulation tasks. Our model checkpoints, together with the publicly available simulation environments and datasets, ensure the reproducibility of our key results. Fig. 7

illustrates some example tasks from these three benchmarks.

- **RoboCasa Kitchen (24 tasks, RoboCasa)**
  RoboCasa (Nasiriany et al., 2024) features a collection of tasks in simulated kitchen environments. We focus on 24 "atomic" tasks that involve foundational sensorimotor skills such as pick-and-place, door opening and closing, pressing buttons, turning faucets, and more. For each task, we use the publicly available dataset of 3000 demonstrations featuring the Franka Emika Panda arm, all generated with MimicGen (Mandlekar et al., 2023). The observation space includes three RGB images captured from cameras positioned on the left, right, and at the wrist. The state representation comprises the position and rotation of both the end-effector and the robot base, as well as the gripper's state. The action space is defined by the relative position and rotation of the end-effector along with the gripper state. We follow the same training and evaluation protocol outlined by Nasiriany et al. (2024).
- **DexMimicGen Cross-Embodiment Suite (9 tasks, DexMG)**
  DexMimicGen (Jiang et al., 2024) includes an array of nine bimanual dexterous manipulation tasks requiring precise two-arm coordination. Together, these tasks cover three bi-manual robot embodiments: (1) *Bimanual Panda Arms with Parallel-Jaw Grippers*: tasks include threading, piece assembly, and transport. The state/action space consists of the end-effector position and rotation of both arms, as well as the gripper states; (2) *Bimanual Panda Arms with Dexterous Hands*: tasks include box cleanup, drawer cleanup, and tray lifting. The state/action space consists of the end-effector position and rotation of both arms and hands; (3) *GR-1 Humanoid with Dexterous Hands*: tasks include pouring, coffee preparation, and can sorting. The state/action space consists of the joint position and rotation of both arms and hands, along with the waist and neck. We generate 1000 demonstrations for each task using the DexMimicGen data generation system and evaluate the model's ability to generalize to novel object configurations.
- **GR-1 Tabletop Tasks (24 tasks, GR-1)**
  This dataset serves as a digital counterpart to real-world humanoid datasets, enabling systematic evaluations that inform the performance of real-robot deployment. This benchmark focuses on dexterous hand control using the GR-1 humanoid robot equipped with Fourier dexterous hands. Compared to DexMG, this benchmark features a significantly larger variety of objects with diverse placements. We model a total of 18 rearrangement tasks, which have a similar structure to the pre-training tasks outlined in Sec. 3.2, *i.e.*, rearranging objects from a source to a target receptacle. Each task involves a unique combination of receptacles, and these combinations are unseen in our pre-training data. Like the pre-training tasks, most tasks involve distractor objects and receptacles that require the model to pay attention to the task language. We additionally feature six tasks that involve placing objects into articulated objects (*i.e.*, cabinets, drawers, and microwaves) and closing them. The observation space includes one RGB image captured from an egocentric camera positioned on the robot's head. The state/action space consists of the joint position and rotation of both arms and hands, along with the waist and neck. We optionally include in our datasets the end effector-based actions for controlling the arms, as the native action space for controlling the whole-body IK controller is end effector-based. We generate 1000 demonstrations for each task using the DexMimicGen system.

4.2. Real-World Benchmarks

We introduce a diverse and meticulously designed set of tabletop manipulation tasks, aimed at evaluating and post-training our models on human demonstrations. These tasks emphasize critical aspects of real-world dexterity, including precise object manipulation, spatial reasoning, bimanual coordination, and multi-agent collaboration. We carefully categorize our benchmarks into four distinct types, ensuring a rigorous evaluation of model performance. We show some example tasks from our real-world benchmarks in Fig. 8.

- **Object-to-Container Pick-and-Place (5 tasks, Pick-and-Place)**
  This category evaluates the model's ability to grasp objects and place them into designated containers, a fundamental capability for robotic manipulation. Tasks include transferring objects between common

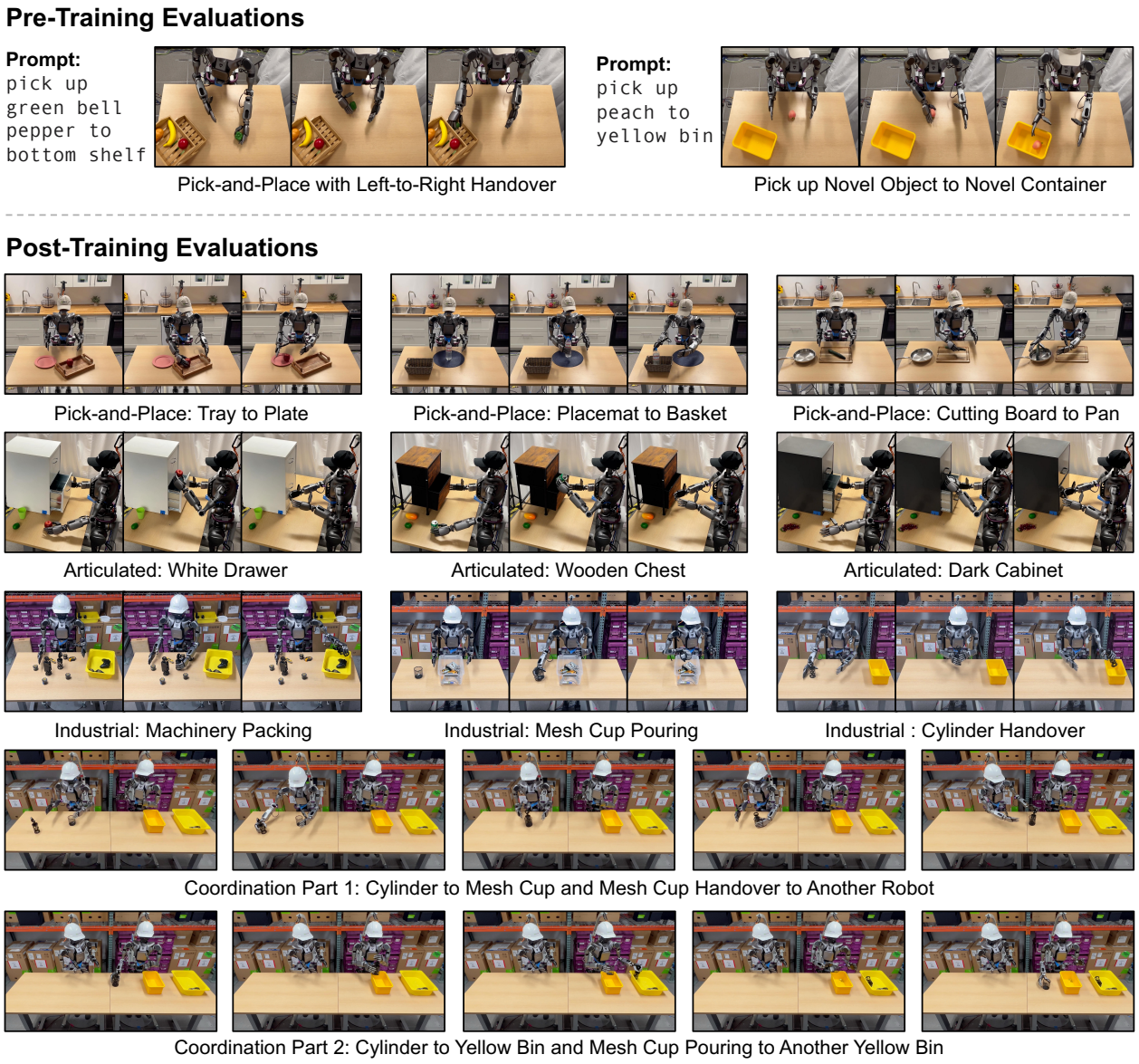


Figure 8: **Real-World Tasks.** All images are captured from policy rollouts of GR00T-N1-2B and models post-trained from GR00T-N1-2B. **(Top) Pre-training evaluations.** We design two manipulation tasks to assess our pretrained models. The left image shows a left-to-right handover, while the right image illustrates the placement of novel objects into an unseen target container. **(Bottom) Post-training evaluations.** We introduce four distinct task categories. From top to bottom, we present examples of object-to-container pick-and-place, articulated object manipulation, industrial object manipulation, and multi-agent coordination.

household containers such as trays, plates, cutting boards, baskets, placemats, bowls, and pans. These scenarios test fine motor skills, spatial alignment, and adaptability to different object geometries. To rigorously assess generalization, we evaluate models on both seen and unseen objects.

- **Articulated Object Manipulation (3 tasks, Articulated)**
  These tasks assess the model's ability to manipulate articulated storage compartments. The model must grasp an object, place it into a storage unit such as a wooden chest, dark cabinet, or white drawer, and then close the compartment. These tasks introduce challenges in constrained motion control and precise placement within limited spaces. Generalization is tested with both seen and unseen objects.
- **Industrial Object Manipulation (3 tasks, Industrial)**

We design this category for industrial scenarios, which involve three structured workflows and tool-based interactions: 1) *Machinery Packing*: Pick up various machinery parts and tools and place them into a designated yellow bin; 2) *Mesh Cup Pouring:* Grasp a mesh cup containing small industrial components (*e.g.*, screws and bolts) and pour its contents into a plastic bin; and 3) *Cylinder Handover*: Pick up a cylindrical object, transfer it from one hand to the other, and place it into a yellow bin. These tasks closely mirror real-world industrial applications, making them highly relevant benchmarks for assessing dexterity in structured environments.

- **Multi-Agent Coordination (2 tasks, Coordination)**
  Collaborative tasks require synchronization between multiple agents, emphasizing role coordination and adaptive decision-making: 1) *Coordination Part 1*: Pick up a cylinder, place it into a mesh cup, and hand it over to another robot; and 2) *Coordination Part 2*: The receiving robot places the cylinder into one yellow bin, then pours the remaining contents of the mesh cup into another yellow bin.

These carefully designed benchmarks introduce structured, goal-driven interactions to test whether a model can seamlessly adapt to real-world applications. To build a high-quality post-training dataset, we let human operators collect task-specific data for durations ranging from 15 minutes to 3 hours, depending on task complexity. We then filter out low-quality trajectories to maintain data integrity. By incorporating a diverse set of task requirements — spanning precise single-agent manipulation to complex multi-agent coordination—our benchmark provides a rigorous testbed for evaluating generalization, adaptability, and fine-tuned control in human-like manipulation tasks.

4.3. Experiment Setup

Our evaluation experiment consists of post-training GR00T N1 and baseline models as described in Sec. 2.3 in a data-limited setting and evaluating the policy success rate in our simulated and real benchmarks described in Sections 4.1 and 4.2, respectively. By default we use a global batch size of 1024 and train for 60k steps. For the DexMimicGen Cross-Embodiment Suite, where each embodiment contains relatively few tasks and the overall training data is limited, we used a smaller batch size of 128 for GR00T-N1-2B.

Baselines

To demonstrate the effectiveness of diverse pretraining of GR00T N1, we compare with two established baselines, BC-Transformer (Mandlekar et al., 2021) and Diffusion Policy (Chi et al., 2024). We describe the details of these two methods below:

- **BC-Transformer** is a Transformer-based behavior cloning policy in RoboMimic (Mandlekar et al., 2021). It consists of a Transformer architecture for processing observation sequences and a Gaussian Mixture Model (GMM) module for modeling action distributions. The policy takes 10 observation frames as input and predicts the next 10 actions.
- **Diffusion Policy** (Chi et al., 2024) models action distributions through a diffusion-based generative process. It employs a U-Net architecture that progressively removes noise from random samples to generate precise robot actions conditioned on observation sequences. It takes a single frame of observations as input and produces 16 action steps in one inference pass.

Evaluation Protocol

For simulated benchmark evaluation, we report the average success rate over 100 trials, taking the maximum score of the last 5 checkpoints, where checkpoints are written every 500 training steps, following the protocol from RoboCasa (Nasiriany et al., 2024).

For real robot evaluation, we employ a partial scoring system to capture model behavior across different execution phases, ensuring a fine-grained assessment of performance. We report the average success rate over 10 trials for each task, except for the task of *Pack Machinery*; for this task, we report the success rate of how many objects out of the 5 machinery parts and tools are placed into the bin, given a time-limit of 30 seconds.

We conduct only 5 trials due to the time constraint. Additionally, to assess the model's efficiency in a low-data regime, we subsample 10% of the full dataset for each task and evaluate whether the model can still learn effective behaviors.

4.4. Quantitative Results

Pre-training Evaluations

To evaluate the generalization capabilities of our pretrained checkpoint, we design two tasks on the real GR-1 humanoid robot (Fig. 8). In the first task, the robot is instructed to place an object on the bottom shelf. However, the object is intentionally positioned to the left of its left hand, requiring a coordinated bimanual strategy. The robot must first grasp the object with its left hand, transfer it within reach of the right hand, and then complete the placement onto the shelf. In the second task, the robot is instructed to place a novel object into an unseen target container. For each task, we evaluate the pretrained GR00T-N1-2B model using five different objects, with three trials per object. GR00T-N1-2B achieves a success rate of 76.6% (11.5/15) in the first coordinated setting and 73.3% (11/15) in the second setting involving novel object manipulation. 0.5 stands for grasping the object correctly but failing to place the object into the container. The high performance under these two evaluation settings illustrates the effectiveness of large-scale pre-training.

Post-training Evaluations

In simulation, we compare the quantitative results for our post-trained GR00T N1 models against from-scratch baselines in the three simulation benchmarks (Table 2). For each benchmark, we post-train using 30, 100, and 300 demonstrations per task (24 tasks for RoboCasa, 9 tasks for DexMG, and 24 tasks for GR-1). We observe that GR00T N1 consistently outperforms the baseline models across benchmark tasks and dataset sizes. In Appendix B, we include the full results and a bar plot (Fig. 10) for comparison.

Table 2: **Simulation Results.** Average success rate across three simulation benchmarks, using 100 demonstrations per task. GR00T N1 outperforms both baselines, especially on the GR-1 task where it outperforms by more than 17 %.

	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%**

On the real robot, we compare GR00T-N1-2B against Diffusion Policy, training on 10% of the human teleoperation dataset and the full dataset (Table 3 and Fig. 9). GR00T-N1-2B, achieves a significantly higher success rate across all tasks, outperforming Diffusion Policy by 32.4% in the 10% Data setting and by 30.4% in the Full Data setting. Notably, GR00T-N1-2B trained on just 10% of the data performs only 3.8% lower than Diffusion Policy trained on the full dataset, highlighting its data efficiency.

Table 3: **Real-World Results.** Average policy success rate on real-world tasks with the GR-1 humanoid robots. GR00T N1 beats the diffusion policy baseline and shows strong results even with very little data.

	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%**

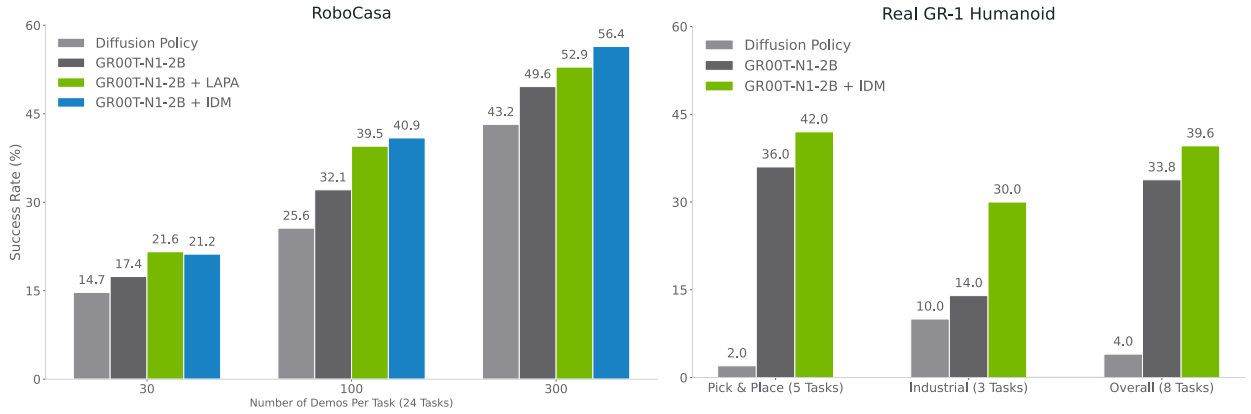


Figure 9: **Average Success Rate (%) across 24 Tasks in simulation and 8 tasks in the real world.** In the RoboCasa simulation, we show all post-training results using 30, 100, and 300 demonstrations per task. In the real world, we show results only on the low-data regime (10% of the demonstrations). We co-train with 3k neural trajectories per task for RoboCasa and 100 neural trajectories per task for real-world tasks. We explore using both latent and IDM-labeled actions in simulation and only IDM-labeled actions for the real robot.

Post-training w/ Neural Trajectories Evaluations

We show some preliminary results of using neural trajectories during post-training for the RoboCasa benchmark for simulation evaluation and Pick-and-Place (seen) and Industrial for the real-world evaluation in Figure 9. We observe that GR00T N1 co-trained with neural trajectories consistently results in substantial gains compared to GR00T N1 only trained on real-world trajectories: +4.2%, +8.8%, +6.8% on average for 30, 100, and 300 data-regimes, respectively, for RoboCasa and +5.8% on average across the 8 tasks with the GR-1 Humanoid.

When comparing LAPA and IDM labels in RoboCasa, an interesting pattern emerges: LAPA slightly outperforms IDM in the relatively low-data regime (30), but as more data becomes available (100 and 300), the performance gap between LAPA and IDM widens. This trend is intuitive— with more data for IDM training, the pseudo-action labels become increasingly aligned with real-world actions, leading to stronger positive transfer. Since GR-1 Humanoid is a relatively "high-data" regime for us, we only utilize IDM actions for neural trajectory co-training in the real world.

4.5. Qualitative Results

How does this behavior look qualitatively? To answer this, we consider the task "Turn Sink Spout" in RoboCasa — in the 100 sample regime, the DP baseline gets 11.8% success rate whereas GR00T N1 gets 42.2%. The DP baseline often gets confused about the semantics of the tasks. From Table 2, we see that GR00T N1 has strong results in the low-data regime. It is natural, in the limit of large fine-tuning datasets, that the effect of pre-training dwindles.

When prompting the pre-trained GR00T N1 model with the task instruction "Pick up the red apple and place it in the basket," one of the tasks in our post-training benchmark, we observe interesting behavioral patterns. In this scenario, we intentionally position the apple to the left of the humanoid hand. Despite seeing few similar tasks during pretraining and exhibiting jerkier motions, the pretrained checkpoint uses its left hand to grasp the apple, hands it over to the right hand, and then places it into the basket. We provide the visualization of this behavior in Fig. 12. In contrast, the post-trained checkpoint fails in this scenario. Since all post-training data exclusively involve the right hand without any inter-hand transfer, the post-trained policy loses the capability to perform this behavior.

For post-trained GR00T N1, we observed that, compared to the baseline Diffusion Policy, its motion is generally much smoother, and its grasping accuracy is significantly higher. In contrast, the Diffusion Policy baseline

suffers from immobility during the initial frames and frequently exhibits inaccurate grasping, resulting in a low success rate in our real-world benchmarks. We provide visualizations of two policy rollout examples in Fig. 13.

4.6. Limitations

Currently, our GR00T N1 model focuses primarily on short-horizon tabletop manipulation tasks. In future work, we aim to extend its capabilities to tackle long-horizon loco-manipulation, which will require advancements in humanoid hardware, model architecture, and training corpora. We anticipate a stronger vision-language backbone will enhance the model's spatial reasoning, language understanding, and adaptability. Our synthetic data generation techniques — leveraging video generation models and automated trajectory synthesis systems — have shown great promise. However, existing methods still face challenges in generating diverse and counterfactual data, while adhering to the laws of physics, limiting the quality and variability of synthetic datasets. We aim to enhance our synthetic data generation techniques to further enrich our data pyramid for model training. Furthermore, we plan to explore novel model architectures and pre-training strategies to improve the robustness and generalization capabilities of our generalist robot models.

5. Related Work

Foundation Models in Robotics. Developing and using foundation models (Bommasani et al., 2021) for robotics has been of great interest recently. One common approach is to leverage existing pre-trained foundation models as high-level black-box reasoning modules in conjunction with low-level robot-specific policies (Brohan et al., 2023; Driess et al., 2023; Huang et al., 2023; Liang et al., 2023; Lin et al., 2023; Singh et al., 2023). This approach allows the robot to plan sequences of low-level skills or motions using the pre-trained foundation model. However, it assumes the availability of these low-level policies and a sufficient interface to connect them to the black-box foundation models. An alternative approach is to finetune pre-trained foundation models on robotics data to build Vision-Language-Action (VLA) models (Black et al., 2024; Brohan et al., 2022, 2023; Cheang et al., 2024; Huang et al., 2024; Kim et al., 2024; Li et al., 2023; Wen et al., 2024; Yang et al., 2025; Ye et al., 2025; Zhen et al., 2024; Zheng et al., 2025). Instead of enforcing a rigid hierarchy between high-level VLM planning and low-level control, these VLA models allow for end-to-end optimization toward the downstream deployment tasks. We take a similar approach to train GR00T N1 and use the Eagle-2 model (Li et al., 2025) as our base Vision Language Model (VLM). We fine-tune our VLM together with a flow-matching (Hu et al., 2024; Lipman et al.; Liu et al., 2022) action generation model with action chunking (Zhao et al., 2023). In contrast to prior VLA models (Black et al., 2024) that use a mixture-of-experts architecture to bridge the base VLM model with the action generation model, we use a simple cross-attention mechanism. This approach provides flexibility regarding the exact architecture of the VLM model and the action generation model we can use. Furthermore, we use embodiment-specific state and action projector modules, which support different robot embodiments, including latent (Ye et al., 2025) and IDM-based (Baker et al., 2022) actions. The use of these projectors is similar to those in Octo Model Team et al. (2024), though that work did not fine-tune the VLM models.

Datasets for Robot Learning. A core challenge in robot learning is the scarcity of large-scale, diverse, and embodied datasets necessary to train generalist robots. One common approach is to use robot teleoperation (Aldaco et al., 2024; Dass et al., 2024; Fu et al., 2024; Iyer et al.; Mandlekar et al., 2018, 2019, 2020; Wu et al., 2023; Zhang et al., 2018; Zhao et al., 2023), where a human uses a device such as a smartphone or Virtual Reality (VR) controller, to control a robot to perform tasks of interest. The robot sensor streams and robot controls during operation are logged to a dataset, allowing for high-quality task demonstrations to be collected. Recently, this approach has been scaled by utilizing large teams of human operators and robot fleets over extended periods of time (*e.g.*, months), resulting in large-scale robot manipulation datasets with thousands of hours of demonstrations (AgiBot-World-Contributors et al., 2025; Black et al., 2024; Brohan et al., 2022, 2023; Ebert et al., 2022; Lynch et al., 2023; O'Neill et al., 2024). However, collecting data this way requires extensive cost and human effort. Another line of work, instrumented human demonstrations, uses special

hardware to capture robot-relevant observation and action data without explicitly teleoperating the target robot. For example, Chi et al. (2024); Seo et al. (2025) use hand-held robot grippers, Fang et al. (2024) uses a robot-like exoskeleton, and Kareer et al. (2024) uses special glasses to capture human hand motions, which are retargeted to robot action data. These approaches tend to result in faster data collection, though they have a mismatch with the downstream robot compared to direct robot teleoperation. A separate line of work makes use of human video datasets (Damen et al., 2018; Goyal et al., 2017; Grauman et al., 2022, 2024; Miech et al., 2019), which are plentiful and substantially easier to collect than on-robot data, as a source of training data for robots. Some works (Karamcheti et al., 2023; Nair et al., 2022; Wu et al., 2023) use human video datasets to pre-train representations that are then used as a feature space for training policies on downstream robot datasets. Other works Bharadhwaj et al. (2024,); Ren et al. (2025) try to jointly use human video data and robot data through intermediate representations for the motions in the video. Ye et al. (2025) shows that pretraining VLAs with *latent* actions only on human videos yields positive transfer to downstream robotic tasks. Rather than relying on a single type of training data, we developed techniques to effectively learn from a diverse assortment of real-world robot data, human video data, and synthetic data.

Synthetic Data Generation in Robotics. Real-world robot data collection requires large amounts of time and considerable human cost. By contrast, data collection in simulation can be substantially more efficient and less painful, making it a compelling alternative. Recently, several works (Dalal et al., 2023; Garrett et al., 2024; Gu et al., 2023; Ha et al., 2023; James et al., 2020; Jiang et al., 2024; Mandlekar et al., 2023; Nasiriany et al., 2024; Wang et al., 2024; Yang et al., 2025) have proposed automated data generation pipelines that can leverage simulation to produce thousands of task demonstrations with minimal human effort. This makes it easy to generate large-scale datasets; however, utilizing these datasets can be challenging due to the simulation-to-reality gap.

Another promising avenue has been using neural generative models to augment existing sets of robot demonstrations (Chen et al., 2023; Mandi et al., 2022; Yu et al., 2023). However, previous work have been limited to utilizing in-painting or text-to-image diffusion models to augment the training data. In our work, we leverage the recent advancements in video generative models (Agarwal et al., 2025; Wan Team, 2025) to create entire neural trajectories, at a scale that has never been explored before: ∼300k neural trajectories which amounts to 827 hours of robot trajectories.

In our model, we make use of large synthetic simulation datasets generated by MimicGen (Mandlekar et al., 2023) and DexMimicGen (Jiang et al., 2024), as well as neural-generated video datasets with state-of-the-art video generation models. Our way of co-training with synthetically generated and real-world data sets us from other large-scale VLA efforts.

6. Conclusions

We have presented GR00T N1, an open foundation model for generalist humanoid robots. GR00T N1 features a dual-system model design, leverages heterogeneous training data, and supports multiple robot embodiments. We systematically evaluate it as a generalist policy across simulation benchmarks and on the real GR-1 humanoid robot. Our experiments demonstrate its strong generalization capabilities, enabling robots to learn diverse manipulation skills with high data efficiency. We hope that our open GR00T-N1-2B model, alongside its training datasets and simulation environments, will accelerate the community's progress toward building and deploying generally capable humanoid robots in the wild.

A. Contributors and Acknowledgments

A.1. Core Contributors

Model Training
Scott Reed, Ruijie Zheng, Guanzhi Wang, Johan Bjorck, Joel Jang, Ao Zhang, Jing Wang, Yinzhen Xu, Fengyuan Hu, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang, Loic Magne, Zhiding Yu, Zhiqi Li

Real-Robot and Teleoperation Infrastructure
Zhenjia Xu, Zu Wang, Xinye (Dennis) Da, Fernando Castañeda

Real-Robot Experiments
Guanzhi Wang, Yinzhen Xu, Joel Jang, You Liang Tan, Ruijie Zheng

Simulation Infrastructure
Yu Fang, Nikita Cherniadev, Runyu Ding, Soroush Nasiriany, Zhenyu Jiang, Kevin Lin, Yuqi Xie

Simulation Experiments
Soroush Nasiriany, Zhenyu Jiang, Yuqi Xie, Kevin Lin, Yu Fang, Runyu Ding, Nikita Cherniadev, Johan Bjorck, Jing Wang

Video Generation Models and Latent Actions
Joel Jang, Seonghyeon Ye, Zongyu Lin, Jiannan Xiang

Data Infrastructure and Curation
Fengyuan Hu, Yinzhen Xu, Avnish Narayan, Loic Magne

Compute Infrastructure and Open-Sourcing
Avnish Narayan, You Liang Tan, Kaushil Kundalia, Fengyuan Hu

Program Management and Operations
Qi Wang, Lawrence Lao

Product Lead
Spencer Huang

Research Leads
Linxi “Jim” Fan, Yuke Zhu

A.2. Contributors

Ajay Mandlekar, Jan Kautz, Dieter Fox, Edith Llontop, Yizhou Zhao, Hao Zhang, Guilin Liu

A.3. Acknowledgments

We thank the 1X team, including Bernt Børnich, Eric Jang, Jorge Milburn, Darien Sleeper, Ralf Mayet, Mohi Khansari, Austin Wang, Vlad Lialin, George Joseph, and Turing Zelsnack for providing support with their humanoid robot hardware and technical support. We thank the Fourier team, including Jie Gu, Roger Cai, Yuxiang Gao, Victor Suen, Hengxin Chen, and Fangzhou Shi, for the hardware support and maintenance of the Fourier GR-1 robots.

We thank Max Fu, Zhengyi Luo, Annika Brundyn, Aastha Jhunjhunwala, Jeff Smith, Yunze Man, and Guo Chen for their technical discussion and assistance, and Marco Pavone, Soha Pouya, Shiwei Sheng, Di Zeng, Yan Chang, Chirag Majithia, John Welsh, Stephan Pleines, Joydeep Biswas for their feedback on our paper draft.

We thank Erwin Coumans, Billy Okal, John Welsh, Pulkit Goyal, Stephan Pleines, Vishal Kulkarni, Chirag Majithia, Di Zeng, Yan Chang, Soha Pouya, Wei Liu, Rushane Hua, Benjamin Butin, Lionel Gulich, Lakshmi Ramesh, Peter Mcaughan, Piyush Medikeri for internal codebase review and testing.

We thank Kyle Yumen, Jeremy Chimienti, Gianna Calderon, Isabel Zuluaga, Juan Zuluaga, Ivy Tam, Jazmin Sanchez, Jesse Yang, Leilee Naderi, Patrick Lee, Tri Cao, Jenna Diamond, Andrew Mathau, Marina Davila, and Sarah Stoddard for working with us on robot teleoperation data collection and annotation.

We thank Arun Shamanna Lakshmi, Eric Colter, Ryan Li, Trasha Dewan, Ethan Yu, Xutong Ren, Fernando Luo, for the OSMO compute infrastructure support, Zhe Zhang for training infrastructure support, Alexis Bjorlin for compute resource support, Amit Goel, Amit, Sandra, Leela Karumbunathan for humanoid robot ecosystem support. We thank Ming-Yu Liu, Yen-Chen Lin, Jinwei Gu, Lyne Tchapmi, Qinsheng Zhang for Cosmos technical support, Madison Huang, Douglas Chang, Kalyan Vadrevu, Oyindamola Omotuyi for marketing support, Sangeeta Subramanian, Shri Sundaram, Vishal Kulkarni for Isaac product support, and Bill Dally and Jensen Huang for their leadership, vision, and guidance.

B. Detailed Experiment Results

Table 4 and Table 5 present a detailed per-task comparison of our GR00T-N1-2B and the Diffusion Policy baseline across our simulation benchmarks and real-world benchmarks, respectively. We train both models on datasets of varying sizes — 30, 100, and 300 demonstrations for simulation benchmarks, and 10% and full data for real-world benchmarks. As expected, performance improves steadily for both models with increasing dataset sizes. Meanwhile, our model consistently outperforms the baseline across all benchmarks and dataset sizes, indicating better generalization and sample efficiency.

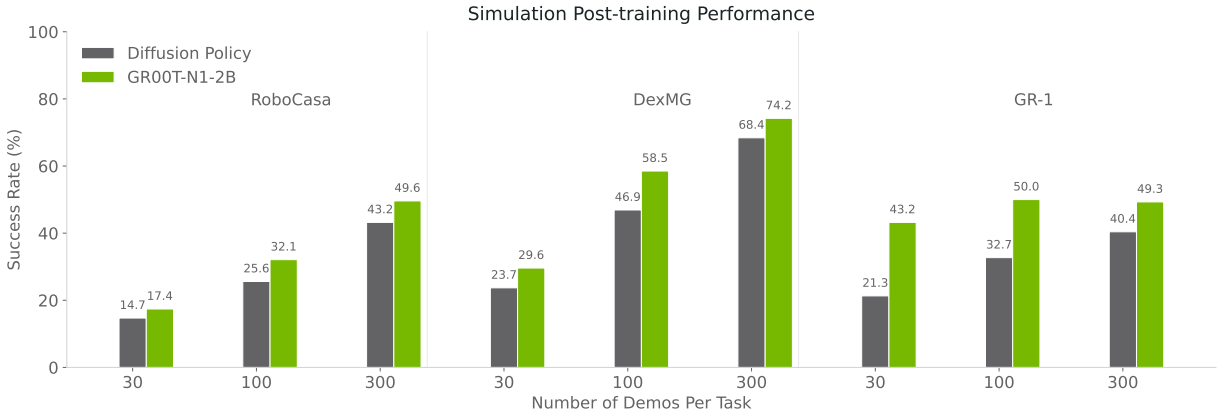


Figure 10: Average policy success rate on simulated manipulation tasks with varying numbers of demonstrations.

C. Hyperparameters

We report important hyperparameters used for the pre- and post-training phases in Table 6. Overall, these two phases share the same values for most of the hyperparameters. For post-training, we use smaller batch sizes to avoid overfitting when fine-tuning in data-limited settings.

D. System Design

D.1. Dataset Formats

Our training corpora build upon the LeRobot dataset format (Cadene et al., 2024), a widely adopted standard in the open-source robotics community. Developed by Hugging Face, LeRobot aims to lower the barrier to entry for robotics research by providing a standardized format for storing, sharing, and utilizing robot demonstration

data. The format has gained significant traction due to its flexibility and the extensive collection of pretrained models and datasets available through the Hugging Face hub.

At its core, the LeRobot dataset format employs a combination of established file formats for efficient storage and access:

1. **Tabular Data:** Robot states, actions, and metadata are stored in parquet files, which offer efficient columnar storage and fast data retrieval. This format enables quick filtering and slicing operations essential for training deep learning models.
2. **Image and Video Data:** Visual observations are encoded as MP4 video files (or alternatively as PNG image sequences), with references stored in the parquet files. This approach significantly reduces storage requirements while maintaining data accessibility.
3. **Metadata:** Dataset statistics, episode indices, and other metadata are stored in structured JSON files, providing machine-readable information about the dataset's characteristics.

The format organizes demonstration data into episodes, with each frame containing synchronized observation and action pairs. Each observation typically includes camera imagery (`observation.images.*`) and robot state information (`observation.state`), while actions represent the control commands sent to the robot. This organization facilitates both imitation learning, where models learn to predict actions from observations, and reinforcement learning, where models learn to optimize for specific outcomes.

While the LeRobot format provides a solid foundation, our work with cross-embodiment data necessitated additional structure to support richer modality information and more sophisticated training regimes. We have extended the LeRobot format with the following constraints:

1. **Modality configuration file:** We require a `modality.json` configuration file in the `meta` directory that explicitly defines the structure of state and action vectors, mapping each dimension to a semantic meaning and provides additional modality-specific information.
2. **Fine-grained modality specification:** Unlike the standard LeRobot format, which treats state and action as monolithic vectors, our extension splits these vectors into semantically meaningful fields (*e.g.*, end-effector position, orientation, gripper state), each with their own metadata including data types, ranges, and transformation specifications.
3. **Multiple annotation support:** We extend the format to support multiple annotation types (*e.g.*, task descriptions, validity flags, success indicators) within a single dataset, following the LeRobot convention of storing indices in the parquet file with the actual content in separate JSON files.
4. **Rotation type specification:** Our format explicitly specifies the representation used for rotational data (*e.g.*, quaternions, Euler angles, axis-angle), enabling proper handling of rotational transformations during training.

Our extended format offers several key benefits for training VLA models:

1. **Semantic clarity:** By explicitly defining the structure and meaning of each dimension in state and action vectors, our format enhances interpretability and reduces errors during data preprocessing and model training.
2. **Flexible transformations:** The fine-grained modality specification enables sophisticated, field-specific normalization and transformation during training. For example, rotational data can be properly normalized and augmented according to its specific representation.
3. **Multi-modal learning support:** The extended format naturally accommodates the diverse data types required for VLA models, including visual observations, state information, action commands, and language annotations, while maintaining clear relationships between these modalities.
4. **Improved data validation:** The explicit structure enables more thorough validation of datasets, reducing the risk of training with malformed or inconsistent data.

5. **Enhanced interoperability:** While adding constraints, our format maintains backward compatibility with the LeRobot ecosystem, allowing us to leverage existing tools and datasets while enabling more sophisticated modeling approaches.

The extended format strikes a balance between standardization and flexibility, providing a clear structure for common robotics data while accommodating the specific needs of VLA models. This approach has proven valuable in our work, enabling more efficient training and improved model performance while maintaining compatibility with the broader robotics research community.

D.2. Standardized Action Spaces

For the above datasets, we make a **best-effort unification** of action and state spaces to ensure consistency across different embodiments and control modalities. Several key practices are applied to achieve this standardization:

1. **End-effector rotation state normalization:** State end-effector rotations are converted to a *6D rotation representation* to avoid singularities and discontinuities in traditional Euler angles.
2. **End-effector rotation action standardization:** End-effector rotation actions are expressed in *axis-angle representation*, providing a compact and smooth parameterization for rotation control.
3. **State and action scaling:** *Min-max normalization* is applied to joint states, joint actions, end-effector state positions, and end-effector action positions and rotations, ensuring uniform value ranges across different robots.
4. **Consistent ordering:** The arrangement of state and action vectors follows a standardized sequence: *end-effector rotation, end-effector position, and gripper closeness*, ordered from the *left arm to the right arm* (if applicable).

E. Additional Training Details

Auxiliary Object Detection Loss

To enhance the model's spatial understanding, we introduce an auxiliary object detection loss during training. In addition to predicting actions, the model must also localize the object of interest based on the given language instruction. Specifically, for each frame in a trajectory segment, we annotate the bounding box of the target object using the OWL-v2 object detector (Minderer et al., 2023). We then compute the normalized center coordinates of the bounding box, x_{gt}, by dividing its x and y coordinates by the image width and height, respectively. To predict the 2D coordinates, we append a linear layer atop the final vision-language embedding tokens and optimize using a squared loss: $L_{det} = \|\mathbf{x}_{pred} - \mathbf{x}_{gt}\|^2$. Thus, the final loss is given by: $L = L_{fm} + L_{det}$.

Neural Trajectory Generation

We finetune WAN2.1-I2V-14B (Wan Team, 2025) using LoRA (Hu et al., 2022) on collected teleoperation trajectories. The trajectories are uniformly downsampled to 81 frames at 480P resolution for finetuning. The resulting image-to-video model generates neural trajectories that capture all possible "counterfactual scenarios" in the real world. To ensure quality, we filter out generated videos that do not accurately follow the given language instructions. Specifically, we sample 8 frames from each video and prompt a commercial-grade multimodal LLM to assess whether it adheres to the instructions. Videos that fail this criterion undergo re-captioning, with the videos downsampled to 16 frames at 256P resolution for this process.

IDM Model Training

We train an inverse dynamics model (IDM) by conditioning on two images (current and the future frame) within a trajectory and train to generate action chunks between the two image frames. From preliminary experiments, we observed that adding state information or more image frames did not significantly improve the action prediction performance on the validation set. For the IDM model architecture, we use the Diffusion Transformer module (System 1) with SigLIP-2 vision embeddings and train with a flow-matching objective. We train the IDM model for each embodiment for 30K or 60K, depending on the size of the training set. After

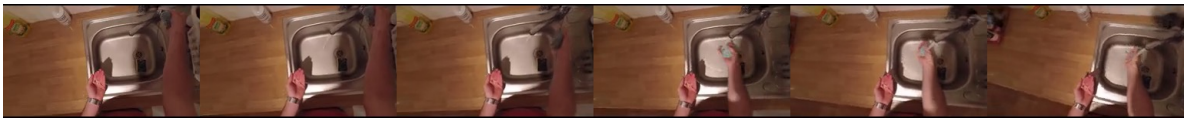
training, we pseudo-label the actions given the two images (with the same action horizon as training) for each step of the neural trajectories.

(Ego4D) Language Annotation: `drops the hand dryer in the cabinet with her right hand.`

(EgoeXO-4D) Language Annotation: `pours the garlic into the bowl with her right hand.`

(HOI4D) Language Annotation: `pick and place stapler.`

(EPIC-KITCHENS) Language Annotation: `turn on tap`



(Assembly-101) Language Annotation: `attach wheel`

(HoloAssist) Language Annotation: `The student inspects the GoPro.`

(RH20T-Human) Language Annotation: `Turn the knob to increase the volume of the speaker`

Figure 11: **Human Egocentric Video Dataset Samples.** We use seven human video datasets for pre-training. The images above show examples from each of the seven datasets with their corresponding language annotations.

F. Additional Qualitative Results

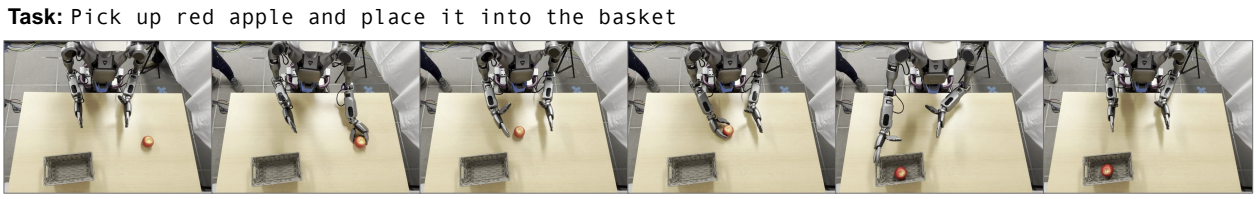


Figure 12: **Pre-training Qualitative Example.** While prompting the pretrained GR00T-N1-2B model with a post-training task instruction, we even increase the difficulty by placing the apple to the left of both hands. Despite not having encountered this setup during training, the model successfully places the red apple into the basket via a two-handed handover, albeit with jerkier motion.

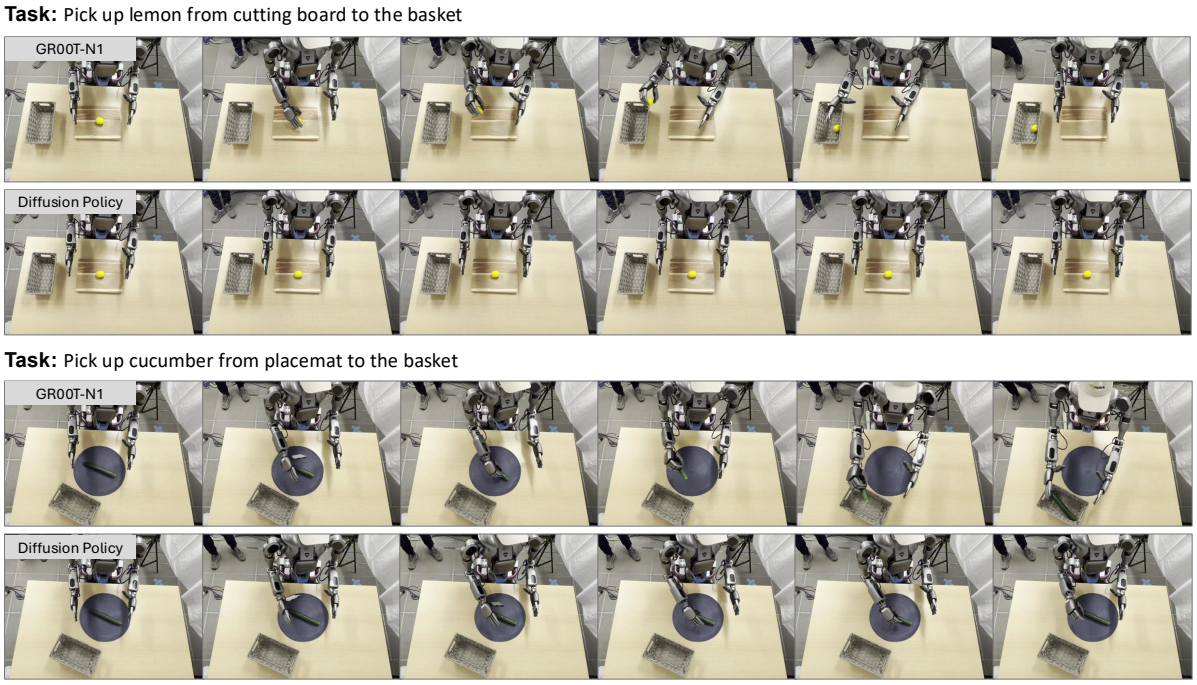


Figure 13: **Post-training Qualitative Example.** (Top) Post-trained GR00T-N1-2B successfully places the cucumber into the basket, whereas the Diffusion Policy fails due to an inaccurate grasp. (Bottom) The post-trained model successfully picks the lemon from the cutting board and puts it into the pan while the Diffusion Policy remains stuck.

Table 4: Simulation Evaluation Results with Models Trained with Different Dataset Sizes.

Task	Diffusion Policy			GR00T N1		
	30 demos	100 demos	300 demos	30 demos	100 demos	300 demos
RoboCasa Kitchen (24 tasks, PnP = Pick-and-Place)						
Close Double Door	1.7	26.5	60.8	0.0	43.1	74.5
Close Drawer	57.5	88.2	94.1	76.9	96.1	99.0
Close Single Door	21.7	46.1	72.6	49.1	67.7	83.3
Coffee Press Button	32.5	46.1	91.2	27.8	56.9	85.3
Coffee Serve Mug	6.7	28.4	66.7	3.7	34.3	72.6
Coffee Setup Mug	0.0	19.6	32.4	0.0	2.0	22.6
Open Double Door	0.0	9.8	18.6	0.0	12.8	14.7
Open Drawer	15.8	42.2	61.8	9.3	42.2	79.4
Open Single Door	36.7	42.2	57.8	20.4	54.9	58.8
PnP from Cab to Counter	2.5	4.9	9.8	0.9	3.9	19.6
PnP from Counter to Cab	0.0	2.9	10.8	1.9	6.9	36.3
PnP from Counter to Microwave	0.0	2.0	8.8	0.0	0.0	12.8
PnP from Counter to Sink	0.0	0.0	13.7	0.0	1.0	9.8
PnP from Counter to Stove	0.0	1.0	17.7	0.0	0.0	23.5
PnP from Microwave to Counter	0.0	2.0	11.8	0.0	0.0	15.7
PnP from Sink to Counter	4.2	8.8	42.2	0.0	5.9	33.3
PnP from Stove to Counter	1.7	2.9	23.5	0.0	0.0	29.4
Turn Off Microwave	63.3	53.9	52.0	47.2	57.8	70.6
Turn Off Sink Faucet	21.7	63.7	72.6	49.1	67.7	72.6
Turn Off Stove	5.0	10.8	19.6	4.6	15.7	26.5
Turn On Microwave	30.0	51.0	75.5	55.6	73.5	78.4
Turn On Sink Faucet	31.7	27.5	63.7	33.3	59.8	62.8
Turn On Stove	12.5	22.6	36.3	14.8	25.5	55.9
Turn Sink Spout	8.3	11.8	23.5	24.1	42.2	52.9
RoboCasa Average	**14.7**	**25.6**	**43.2**	**17.4**	**32.1**	**49.6**
DexMimicGen Cross-Embodiment Suite (9 tasks)						
Can Sort	82.8	93.1	99.4	94.8	98.0	98.0
Coffee	35.5	68.1	79.7	44.9	79.4	73.5
Pouring	37.0	62.3	68.8	54.4	71.6	87.3
Threading	4.2	18.3	27.5	3.9	37.3	60.8
Three Piece Assembly	10.0	32.5	63.3	10.8	43.1	69.6
Transport	7.5	25.0	53.3	7.8	48.0	61.8
Box Cleanup	30.0	80.8	97.5	33.3	29.4	95.1
Drawer Cleanup	1.7	16.7	52.5	10.8	42.2	55.9
Lift Tray	5.0	25.0	73.3	5.8	77.5	65.7
DexMG Average	**23.7**	**46.9**	**68.4**	**29.6**	**58.5**	**74.2**
GR-1 Tabletop (24 Tasks)						
Cutting Board to Pot	22.6	37.3	48.0	58.8	57.8	57.8
Cutting Board to Basket	19.6	42.2	29.4	43.1	61.8	56.9
Cutting Board to Tiered Basket	13.7	13.7	18.6	13.7	23.5	34.3
Cutting Board to Pan	28.4	48.0	57.8	67.7	65.7	68.6
Cutting Board to Cardboard Box	11.8	15.7	22.6	31.4	30.4	33.3
Placemat to Bowl	14.7	18.6	23.5	31.4	39.2	39.2
Placemat to Plate	15.7	23.5	37.3	33.3	37.3	49.0
Placemat to Basket	15.7	25.5	41.2	50.0	46.1	55.9
Placemat to Tiered Shelf	6.9	5.9	11.8	11.8	21.6	19.6
Plate to Pan	13.7	17.7	35.3	35.3	48.0	52.9
Plate to Cardboard Box	12.8	13.7	27.5	34.3	38.2	32.4
Plate to Bowl	15.7	18.6	31.4	41.2	42.2	34.3
Plate to Plate	25.5	39.2	61.8	72.6	85.3	68.6
Tray to Tiered Shelf	2.0	6.9	15.7	17.7	27.5	14.7
Tray to Tiered Basket	12.8	34.3	39.2	33.3	49.0	45.1
Tray to Plate	26.5	41.2	49.0	53.9	68.6	62.8
Tray to Cardboard Box	21.6	37.3	40.2	51.0	55.9	54.9
Tray to Pot	21.6	48.0	52.9	52.0	59.8	65.7
Wine to Cabinet	43.1	55.9	60.8	57.8	53.9	62.8
Place Bottle to Cabinet	40.2	62.8	60.8	60.8	81.4	74.5
Place Milk to Microwave	37.3	41.2	51.0	42.2	58.8	49.0
Potato to Microwave	17.7	30.4	41.2	30.4	26.5	34.3
Cup to Drawer	24.5	32.4	36.3	36.3	44.1	40.2
Can to Drawer	48.0	74.5	75.5	77.5	76.5	75.5
GR-1 Average	**21.3**	**32.7**	**40.4**	**43.2**	**50.0**	**49.3**

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%**

Table 6: Training hyperparameters. Pre- and post-training use the same hyperparameters unless specified.

Hyperparameter	**Pre-training Value**	**Post-training Value**
Learning rate	1e-4	
Optimizer	AdamW	
Adam beta1	0.95	
Adam beta2	0.999	
Adam epsilon	1e-8	
Weight decay	1e-5	
LR scheduler	cosine	
Warmup ratio	0.05	
Batch size	16,384	128 or 1024
Gradient steps	200,000	20,000 – 60,000
Backbone's vision encoder	unfrozen	
Backbone's text tokenizer	frozen	
DiT	unfrozen	

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	–	–	–

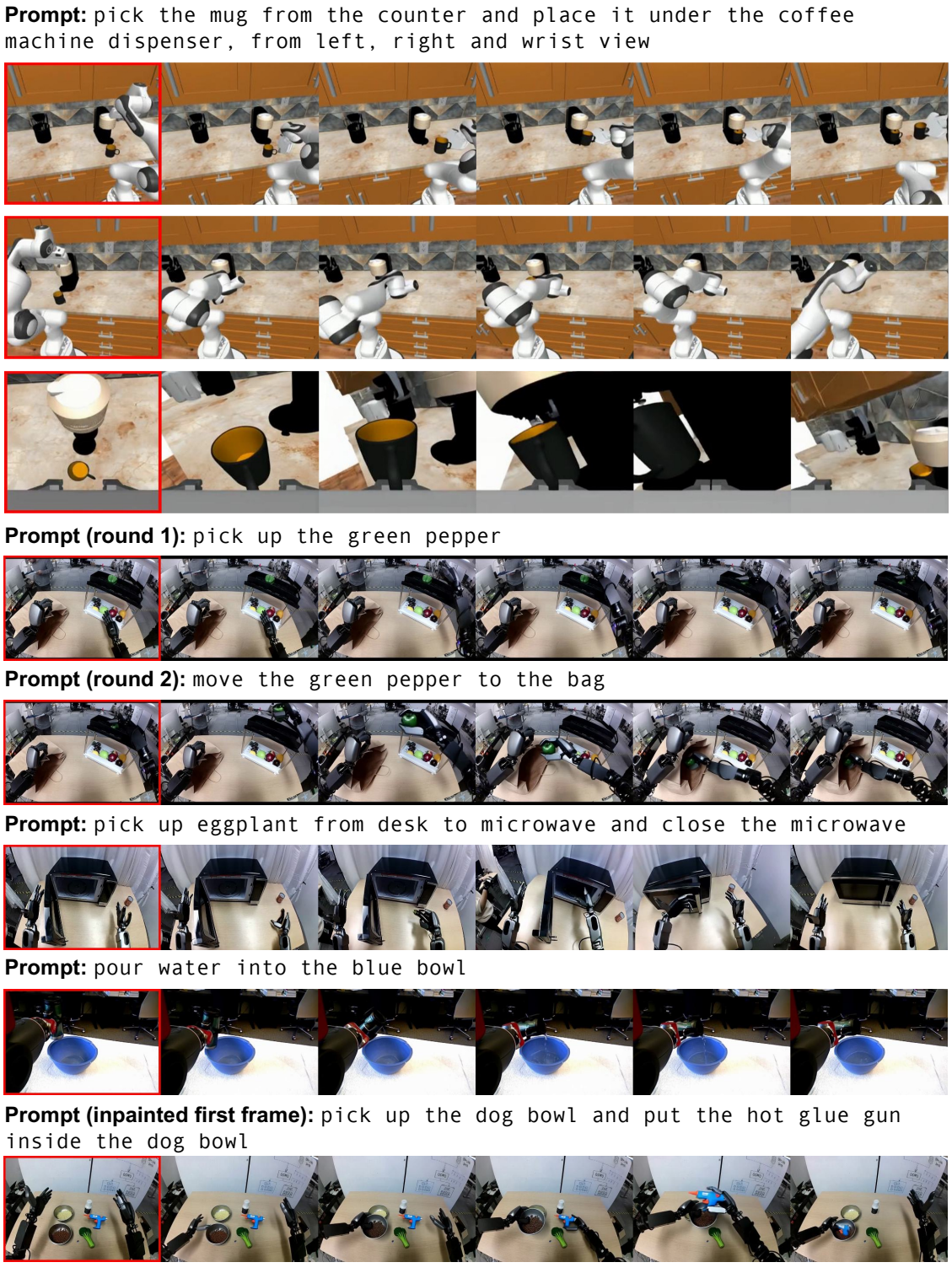


Figure 14: **More Examples of Neural Generated Trajectories.** We highlight 4 key capabilities of neural trajectories: (1) The first three rows shows an example of a multi-view trajectory generated for the post-training in RoboCasa; we achieve this by concatenating the views as a 4-grid video during training. (2) The following two rows show two consecutive sequences, where the initial frame of the latter is from the end of the former, highlighting the possibility of generating trajectories of tasks requiring composition of atomic tasks. (3) The following two rows illustrate the capability of our models to generate trajectories with articulated objects and liquids, which are known to be very challenging in simulation. (4) The last row is generated from an in-painted initial frame, showcasing that we can generate more diverse videos without having to collect novel initial frames in the real-world and be bound by human labor. We use the red rectangles to indicate the initial frames.

References

[1] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025. 6, 18

[2] AgiBot-World-Contributors et al. AgiBot World Colosseo: A Large-scale Manipulation Platform for Scalable and Intelligent Embodied Systems. *arXiv preprint arXiv:2503.06669*, 2025. 9, 17

[3] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, 2022. 5

[4] Jorge Aldaco, Travis Armstrong, Robert Baruch, Jeff Bingham, Sanky Chan, Kenneth Draper, Debidatta Dwibedi, Chelsea Finn, Pete Florence, Spencer Goodrich, et al. Aloha 2: An enhanced low-cost hardware for bimanual teleoperation. *arXiv preprint arXiv:2405.02292*, 2024. 17

[5] Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025. 4

[6] Bowen Baker, Ilge Akkaya, Peter Zhokov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos. *Advances in Neural Information Processing Systems*, 35:24639–24654, 2022. 8, 17

[7] Homanga Bharadhwaj, Debidatta Dwibedi, Abhinav Gupta, Shubham Tulsiani, Carl Doersch, Ted Xiao, Dhruv Shah, Fei Xia, Dorsa Sadigh, and Sean Kirmani. Gen2act: Human video generation in novel scenarios enables generalizable robot manipulation. *arXiv preprint arXiv:2409.16283*, 2024. 18

[8] Homanga Bharadhwaj, Roozbeh Mottaghi, Abhinav Gupta, and Shubham Tulsiani. Track2act: Predicting point tracks from internet videos enables diverse zero-shot robot manipulation. *arXiv e-prints*, pages arXiv–2405, 2024. 18

[9] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 9

[10] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 3, 5, 17

[11] Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021. 17

[12] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir

Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022. 9, 17

[13] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 17

[14] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023. 17

[15] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning*, pages 287–318. PMLR, 2023. 17

[16] Anthony Brohan, Yevgen Chebotar, Chelsea Finn, Karol Hausman, Alexander Herzog, Daniel Ho, Julian Ibarz, Alex Irpan, Eric Jang, Ryan Julian, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on robot learning*, pages 287–318. PMLR, 2023. 17

[17] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, et al. Video generation models as world simulators. 2024. *URL https://openai.com/research/video-generation-models-as-world-simulators*, 3:1, 2024. 6

[18] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024. 20

[19] Chi-Lam Cheang, Guangzeng Chen, Ya Jing, Tao Kong, Hang Li, Yifeng Li, Yuxiao Liu, Hongtao Wu, Jiafeng Xu, Yichu Yang, Hanbo Zhang, and Minzhao Zhu. Gr-2: A generative video-language-action model with web-scale knowledge for robot manipulation. *arXiv preprint arXiv:2410.06158*, 2024. 17

[20] Zoey Chen, Sho Kiami, Abhishek Gupta, and Vikash Kumar. Genaug: Retargeting behaviors to unseen situations via generative augmentation. *arXiv preprint arXiv:2302.06671*, 2023. 18

[21] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024. 14

[22] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. In *Proceedings of Robotics: Science and Systems (RSS)*, 2024. 18

[23] Murtaza Dalal, Ajay Mandlekar, Caelan Reed Garrett, Ankur Handa, Ruslan Salakhutdinov, and Dieter Fox. Imitating task and motion planning with visuomotor transformers. In *Conference on Robot Learning*, pages 2565–2593. PMLR, 2023. 18

[24] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European conference on computer vision (ECCV)*, pages 720–736, 2018. 11, 18

[25] Shivin Dass, Wensi Ai, Yuqian Jiang, Samik Singh, Jiaheng Hu, Ruohan Zhang, Peter Stone, Ben Abbatematteo, and Roberto Martín-Martín. Telemoma: A modular and versatile teleoperation system for mobile manipulation. In *2nd Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024*, 2024. 17

[26] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 17

[27] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge Data: Boosting Generalization of Robotic Skills with Cross-Domain Datasets. In *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022. doi: 10.15607/RSS.2022.XVIII.063. 17

[28] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A robotic dataset for learning diverse skills in one-shot. In *RSS 2023 Workshop on Learning for Task and Motion Planning*, 2023. 11

[29] Hongjie Fang, Hao-Shu Fang, Yiming Wang, Jieji Ren, Jingjing Chen, Ruo Zhang, Weiming Wang, and Cewu Lu. Airexo: Low-cost exoskeletons for learning whole-arm manipulation in the wild. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15031–15038. IEEE, 2024. 18

[30] Zipeng Fu, Tony Z Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 17

[31] Caelan Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillmimicgen: Automated demonstration generation for efficient skill learning and deployment. *arXiv preprint arXiv:2410.18907*, 2024. 18

[32] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 18

[33] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 18995–19012, 2022. 11, 18

[34] Kristen Grauman, Andrew Westbury, Lorenzo Torresani, Kris Kitani, Jitendra Malik, Triantafyllos Afouras, Kumar Ashutosh, Vijay Baiyya, Siddhant Bansal, Bikram Boote, et al. Ego-exo4d: Understanding skilled human activity from first-and third-person perspectives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19383–19400, 2024. 11, 18

[35] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. In *The Eleventh International Conference on Learning Representations*, 2023. 18

[36] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In *Conference on Robot Learning*, pages 3766–3777. PMLR, 2023. 18

[37] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 22

[38] Xixi Hu, Bo Liu, Xingchao Liu, and Qiang Liu. Adaflow: Imitation learning with variance-adaptive flow-based policies. *arXiv preprint arXiv:2402.04292*, 2024. 17

[39] Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2024. 17

[40] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *6th Annual Conference on Robot Learning*. 17

[41] Wenlong Huang, Fei Xia, Dhruv Shah, Danny Driess, Andy Zeng, Yao Lu, Pete Florence, Igor Mordatch, Sergey Levine, Karol Hausman, et al. Grounded decoding: Guiding text generation with grounded models for embodied agents. *Advances in Neural Information Processing Systems*, 36:59636–59661, 2023. 17

[42] Aadhithya Iyer, Zhuoran Peng, Yinlong Dai, Irmak Guzey, Siddhant Haldar, Soumith Chintala, and Lerrel Pinto. Open teach: A versatile teleoperation system for robotic manipulation. In *CoRL 2024 Workshop on Mastering Robot Manipulation in a World of Abundant Data*. 17

[43] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 18

[44] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: robot manipulation with multimodal prompts. In *Proceedings of the 40th International Conference on Machine Learning*, pages 14975–15022, 2023. 5

[45] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning. 2024. 6, 11, 12, 18

[46] Daniel Kahneman. *Thinking, fast and slow*. 2011. 2

[47] Siddharth Karamcheti, Suraj Nair, Annie S Chen, Thomas Kollar, Chelsea Finn, Dorsa Sadigh, and Percy Liang. Language-driven representation learning for robotics. *arXiv preprint arXiv:2302.12766*, 2023. 18

[48] Simar Kareer, Dhruv Patel, Ryan Punamiya, Pranay Mathur, Shuo Cheng, Chen Wang, Judy Hoffman, and Danfei Xu. EgoMimic: Scaling imitation learning via egocentric video, 2024. 18

[49] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, Peter David Fagan, Joey Hejna, Masha Itkina, Marion Lepert, Yecheng Jason Ma, Patrick Tree Miller, Jimmy Wu, Suneel Belkhale, Shivin Dass, Huy Ha, Arhan Jain, Abraham Lee, Youngwoon Lee, Marius Memmel, Sungjae Park, Ilija Radosavovic, Kaiyuan Wang, Albert Zhan, Kevin Black, Cheng Chi, Kyle Beltran Hatch, Shan Lin, Jingpei Lu, Jean Mercat, Abdul Rehman, Pannag R Sanketi, Archit Sharma, Cody Simpson, Quan Vuong, Homer Rich Walke, Blake Wulfe, Ted Xiao, Jonathan Heewon Yang, Arefeh Yavary, Tony Z. Zhao, Christopher Agia, Rohan Baijal, Mateo Guaman Castro, Daphne Chen, Qiuyu Chen, Trinity Chung, Jaimyn Drake, Ethan Paul Foster, Jensen Gao, David Antonio Herrera, Minho Heo, Kyle Hsu, Jiaheng Hu, Donovon Jackson, Charlotte Le, Yunshuang Li, Kevin Lin, Roy Lin, Zehan Ma, Abhiram Maddukuri, Suvir Mirchandani, Daniel Morton, Tony Nguyen, Abigail O'Neill, Rosario Scalise, Derick Seale, Victor

Son, Stephen Tian, Emi Tran, Andrew E. Wang, Yilin Wu, Annie Xie, Jingyun Yang, Patrick Yin, Yunchu Zhang, Osbert Bastani, Glen Berseth, Jeannette Bohg, Ken Goldberg, Abhinav Gupta, Abhishek Gupta, Dinesh Jayaraman, Joseph J Lim, Jitendra Malik, Roberto Martín-Martín, Subramanian Ramamoorthy, Dorsa Sadigh, Shuran Song, Jiajun Wu, Michael C. Yip, Yuke Zhu, Thomas Kollar, Sergey Levine, and Chelsea Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024. 9

[50] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 17

[51] Xinghang Li, Minghuan Liu, Hanbo Zhang, Cunjun Yu, Jie Xu, Hongtao Wu, Chilam Cheang, Ya Jing, Weinan Zhang, Huaping Liu, et al. Vision-language foundation models as effective robot imitators. *arXiv preprint arXiv:2311.01378*, 2023. 17

[52] Zhiqi Li, Guo Chen, Shilong Liu, Shihao Wang, Vibashan VS, Yishen Ji, Shiyi Lan, Hao Zhang, Yilin Zhao, Subhashree Radhakrishnan, et al. Eagle 2: Building post-training data strategies from scratch for frontier vision-language models. *arXiv preprint arXiv:2501.14818*, 2025. 3, 4, 17

[53] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023. 17

[54] Kevin Lin, Christopher Agia, Toki Migimatsu, Marco Pavone, and Jeannette Bohg. Text2motion: from natural language instructions to feasible plans. *Autonomous Robots*, 2023. 17

[55] Zongyu Lin, Wei Liu, Chen Chen, Jiasen Lu, Wenze Hu, Tsu-Jui Fu, Jesse Allardice, Zhengfeng Lai, Liangchen Song, Bowen Zhang, et al. Stiv: Scalable text and image conditioned video generation. *arXiv preprint arXiv:2412.07730*, 2024. 6

[56] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*. 3, 17

[57] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 17

[58] Yunze Liu, Yun Liu, Che Jiang, Kangbo Lyu, Weikang Wan, Hao Shen, Boqiang Liang, Zhoujie Fu, He Wang, and Li Yi. Hoi4d: A 4d egocentric dataset for category-level human-object interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21013–21022, June 2022. 11

[59] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time, 2022. 9

[60] Corey Lynch, Ayzaan Wahid, Jonathan Tompson, Tianli Ding, James Betker, Robert Baruch, Travis Armstrong, and Pete Florence. Interactive language: Talking to robots in real time. *IEEE Robotics and Automation Letters*, 2023. 17

[61] Zhao Mandi, Homanga Bharadhwaj, Vincent Moens, Shuran Song, Aravind Rajeswaran, and Vikash Kumar. Cacti: A framework for scalable multi-task multi-scene visual imitation learning. *arXiv preprint arXiv:2212.05711*, 2022. 18

[62] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. RoboTurk: A Crowdsourcing Platform for Robotic Skill Learning through Imitation. In *Conference on Robot Learning*, 2018. 17

[63] Ajay Mandlekar, Jonathan Booher, Max Spero, Albert Tung, Anchit Gupta, Yuke Zhu, Animesh Garg, Silvio Savarese, and Li Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1048–1055. IEEE, 2019. 17

[64] Ajay Mandlekar, Danfei Xu, Roberto Martín-Martín, Yuke Zhu, Li Fei-Fei, and Silvio Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020. 17

[65] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021. 14

[66] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, 2023. 6, 12, 18

[67] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2630–2640, 2019. 18

[68] Matthias Minderer, Alexey A. Gritsenko, and Neil Houlsby. Scaling open-vocabulary object detection. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 22

[69] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577, 2018. 8

[70] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. *arXiv preprint arXiv:2203.12601*, 2022. 18

[71] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots. In *Robotics: Science and Systems (RSS)*, 2024. 10, 11, 12, 14, 18

[72] NVIDIA. Osmo platform, 2025. URL https://developer.nvidia.com/osmo. Accessed: 2025-03-12. 8

[73] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. 17

[74] Open X-Embodiment Collaboration et al. Open X-Embodiment: Robotic learning datasets and RT-X models. International Conference on Robotics and Automation, 2024. 1, 9

[75] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. 17

[76] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4195–4205, 2023. 4

[77] Juntao Ren, Priya Sundaresan, Dorsa Sadigh, Sanjiban Choudhury, and Jeannette Bohg. Motion tracks: A unified representation for human-robot transfer in few-shot imitation learning. *arXiv preprint arXiv:2501.06994*, 2025. 18

[78] Zhongwei Ren, Yunchao Wei, Xun Guo, Yao Zhao, Bingyi Kang, Jiashi Feng, and Xiaojie Jin. Videoworld: Exploring knowledge learning from unlabeled videos. *arXiv preprint arXiv:2501.09781*, 2025. 6

[79] F. Sener, D. Chatterjee, D. Shelepov, K. He, D. Singhania, R. Wang, and A. Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. *CVPR 2022*, 2022. 11

[80] Mingyo Seo, H. Andy Park, Shenli Yuan, Yuke Zhu, , and Luis Sentis. Legato: Cross-embodiment imitation using a grasping tool. *IEEE Robotics and Automation Letters (RA-L)*, 2025. 18

[81] Rutav Shah, Roberto Martín-Martín, and Yuke Zhu. Mutex: Learning unified policies from multimodal task specifications. In *7th Annual Conference on Robot Learning*, 2023. 9

[82] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016. 4

[83] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023. 17

[84] Garrett Thomas, Ching-An Cheng, Ricky Loynd, Felipe Vieira Frujeri, Vibhav Vineet, Mihai Jalobeanu, and Andrey Kolobov. Plex: Making the most of the available data for robotic manipulation pretraining. In *CoRL*, 2023. 9

[85] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025. 4

[86] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023. 9

[87] Wan Team. Wan: Open and advanced large-scale video generative models. 2025. 6, 18, 22

[88] Xin Wang, Taein Kwon, Mahdi Rad, Bowen Pan, Ishani Chakraborty, Sean Andrist, Dan Bohus, Ashley Feniello, Bugra Tekin, Felipe Vieira Frujeri, Neel Joshi, and Marc Pollefeys. Holoassist: an egocentric human interaction dataset for interactive ai assistants in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023. 11

[89] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation. In *International Conference on Machine Learning*, 2024. 6, 18

[90] Junjie Wen, Yichen Zhu, Jinming Li, Minjie Zhu, Kun Wu, Zhiyuan Xu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, et al. Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation. *arXiv preprint arXiv:2409.12514*, 2024. 17

[91] Hongtao Wu, Ya Jing, Chilam Cheang, Guangzeng Chen, Jiafeng Xu, Xinghang Li, Minghuan Liu, Hang Li, and Tao Kong. Unleashing large-scale video generative pre-training for visual robot manipulation. *arXiv preprint arXiv:2312.13139*, 2023. 18

[92] Philipp Wu, Yide Shentu, Zhongke Yi, Xingyu Lin, and Pieter Abbeel. Gello: A general, low-cost, and intuitive teleoperation framework for robot manipulators, 2023. 17

[93] Jiannan Xiang, Guangyi Liu, Yi Gu, Qiyue Gao, Yuting Ning, Yuheng Zha, Zeyu Feng, Tianhua Tao, Shibo Hao, Yemin Shi, et al. Pandora: Towards general world model with natural language actions and video states. *arXiv preprint arXiv:2406.09455*, 2024. 6

[94] Jianwei Yang, Reuben Tan, Qianhui Wu, Ruijie Zheng, Baolin Peng, Yongyuan Liang, Yu Gu, Mu Cai, Seonghyeon Ye, Joel Jang, Yuquan Deng, Lars Liden, and Jianfeng Gao. Magma: A foundation model for multimodal AI agents, 2025. 17

[95] Lujie Yang, HJ Suh, Tong Zhao, Bernhard Paus Graesdal, Tarik Kelestemur, Jiuguang Wang, Tao Pang, and Russ Tedrake. Physics-driven data generation for contact-rich manipulation via trajectory optimization. *arXiv preprint arXiv:2502.20382*, 2025. 18

[96] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv preprint arXiv:2408.06072*, 2024. 6

[97] Seonghyeon Ye, Joel Jang, Byeongguk Jeon, Se June Joo, Jianwei Yang, Baolin Peng, Ajay Mandlekar, Reuben Tan, Yu-Wei Chao, Bill Yuchen Lin, Lars Liden, Kimin Lee, Jianfeng Gao, Luke Zettlemoyer, Dieter Fox, and Minjoon Seo. Latent action pretraining from videos. In *The Thirteenth International Conference on Learning Representations*, 2025. 2, 5, 17, 18

[98] Tianhe Yu, Ted Xiao, Austin Stone, Jonathan Tompson, Anthony Brohan, Su Wang, Jaspiar Singh, Clayton Tan, Jodilyn Peralta, Brian Ichter, et al. Scaling robot learning with semantically imagined experience. *arXiv preprint arXiv:2302.11550*, 2023. 18

[99] Kevin Zakka. Mink: Python inverse kinematics based on MuJoCo, July 2024. URL https://github.com/kevinzakka/mink. 10

[100] Tianhao Zhang, Zoe McCarthy, Owen Jow, Dennis Lee, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 17

[101] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning Fine-Grained Bimanual Manipulation with Low-Cost Hardware. In *Proceedings of Robotics: Science and Systems*, 2023. 4, 17

[102] Haoyu Zhen, Xiaowen Qiu, Peihao Chen, Jincheng Yang, Xin Yan, Yilun Du, Yining Hong, and Chuang Gan. 3d-vla: 3d vision-language-action generative world model. *arXiv preprint arXiv:2403.09631*, 2024. 17

[103] Ruijie Zheng, Yongyuan Liang, Shuaiyi Huang, Jianfeng Gao, Hal Daumé III, Andrey Kolobov, Furong Huang, and Jianwei Yang. TraceVLA: Visual trace prompting enhances spatial-temporal awareness for generalist robotic policies. In *The Thirteenth International Conference on Learning Representations*, 2025. 17