$\Psi_{0.1}$: A STEERABLE HUMANOID VLA MODEL FOR LONG-HORIZON LOCO-MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language-action models predict action chunks efficiently but tend to blindly follow training trajectories when facing out-of-domain tasks. While world models demonstrate more robust generalization, they require longer inference times, which critically hinder humanoid loco-manipulation, where real-time execution is demanded. Combining the best of both, we present $\Psi_{0.1}$, a hybrid humanoid foundation model with unprecedented steerability that enables up to five minutes of autonomous execution on long-horizon, whole-body loco-manipulation tasks. Specifically, we first train a *predictor*: a time-conditioned image world model that predicts a future goal image to steer task execution, conditioned on a time horizon and a language instruction. We then re-purpose an existing VLA model, which predicts actions from solely the current visual observation, to additionally condition on a goal image. It acts as a *planner* in action space, outputting actions that steer the humanoid from the current observation toward the goal image. Finally, we implement a real-time execution pipeline that streams actions to a whole-body *controller*, achieving whole-body control capable of more diverse motions, such as kicking and bending, that are challenging for a decoupled controller. To obtain steerability, we draw on priors from the existing image generation model and further condition it on the time horizon and language instruction. In addition, we pretrain the VLA model with variable-length action chunk prediction, allowing it to develop a better understanding the dynamics between the current observation and goal image. Compared with baseline methods, $\Psi_{0.1}$, achieves better generalization, more reliable language instruction following, and higher success rates on long-horizon tasks. Through exhaustive experiments across seven real-world tasks, we demonstrate that our model achieves the highest overall performance, outperforming the strongest baseline by 25% in success rate in novel scenes with unseen objects and layouts, and by 43% in task progress on cross-task transitions, supporting whole-body loco-manipulation tasks lasting up to five minutes.

1 INTRODUCTION

Foundation models have emerged as a promising approach to general robotics (Physical Intelligence, 2025; Bjorck et al., 2025; Gemini Robotics et al., 2025), with vision language action models and world action models drawing on large and diverse datasets to connect multimodal understanding with embodied control (Ye et al., 2026; Zhang et al., 2025; Deng et al., 2025b). Central to this paradigm is generalizability, the ability to transfer learned capabilities to unfamiliar objects, environments, and tasks beyond the training distribution, which is essential for humanoid robots operating in diverse settings. One manifestation of this generalizability under unfamiliar conditions is steerability, whereby a humanoid follows language instructions or future goal images and adapts its whole body actions to achieve the intended outcome in situations beyond its training experience (Physical Intelligence, 2026). Despite substantial progress, generalizability remains a major limitation of vision language action (VLA) models. Compared with large language models, VLA models have access to far less training data (Wei et al., 2026a), making it difficult to achieve similarly broad capabilities in robotics through data scaling alone. Existing VLA models often exhibit a tendency to reproduce familiar behaviors without adequately responding to task guidance, particularly when confronted with novel tasks. In such situations, they may overlook the target object or language

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

instruction and instead execute an action pattern observed during training, revealing limited steerability beyond the training distribution.

World action models (WAMs) offer another emerging training paradigm that can improve generalizability through future video prediction, although the mechanisms underlying these gains remain unclear (Ye et al., 2026; Li et al., 2026; NVIDIA et al., 2025). One possible explanation is that predicting future observations provides dense supervision beyond action targets, supplying richer learning signals for optimization. Another is that video prediction encourages the model to learn visual representations that better support action generation (Luo et al., 2026a; Ma et al., 2026). We characterize this paradigm, which extends the model's outputs, i.e., video, while leaving its inputs unchanged, as a form of *implicit steerability*, in which future prediction shapes action generation without introducing explicit goal guidance at the input.

We define *explicit steerability* as the ability to guide a model's behavior through an additional goal image provided alongside the current observation (Nguyen et al., 2026). Building on this formulation, we introduce $\Psi_{0.1}$, a humanoid foundation model that exhibits unprecedented steerability, demonstrates generalization across multiple tasks on a physical robot, and sustains autonomous execution of whole body locomotion and manipulation up to five minutes. These capabilities are enabled by a hybrid architecture that decomposes steerability into two stages (Cai et al., 2026; GigaBrain Team et al., 2026). First, the model predicts a future goal image conditioned on a language instruction and a specified time interval, drawing on the prior knowledge of a pretrained image generation foundation model. Second, a VLA model takes both the current observation and the predicted goal image as inputs, using the latter to guide action generation. We find that the language following capabilities inherited by the image world model substantially improve generalization in task execution.

Our system comprises three components: a *predictor*, a *planner*, and a *controller*. The predictor guides execution by generating visual goals, while the planner translates this guidance from image space into action space by predicting an action chunk that moves the robot from its current state toward the state depicted in the goal image. Providing an explicit visual target helps the VLA model ground its actions in the intended outcome and reduces its tendency to reproduce familiar grasping behaviors. To develop this capability, we pretrain the planner on 800 hours of egocentric human video from EgoDex (Hoque et al., 2026), using an objective that requires the model to predict action chunks of varying lengths from pairs of current and future observations. This objective encourages the model to infer the actions connecting two states from their visual representations. To support whole body control, we build a real time execution system upon a state of the art whole body controller SONIC (Luo et al., 2026b). We apply real-time action chunking during inference to enable smoother coordination across successive action predictions (Black et al., 2025b), supporting behaviors such as bending down and pressing a trash bin pedal with the foot. However, the action token space of SONIC differs substantially from the human hand end effector space used during pretraining. To bridge this gap, we curate and retarget 100 hours of real-world robot data into the SONIC token space and validate the resulting trajectories through simulation replay.

Our contributions are threefold:

1. We introduce $\Psi_{0.1}$, a hybrid humanoid foundation model that combines a predictor, a planner, and a controller to achieve explicit steerability through language instructions and visual goals, demonstrating generalization across multiple real robot tasks and sustained autonomous execution up to five minutes.

2. We present a time-conditioned image world model that is able to robustly predict future world state that can generalize to novel language instruction and novel tasks. The conditioned time horizon is a key enabler to achieve stable steerability in the image space.

3. We propose a novel pretraining paradigm in which the model predicts variable-length action chunks from two temporal randomly sampled frames in egocentric human videos, enabling it to learn planning directly in the visual space.

2 Related Work

Vision-language-action models. Vision-language-action (VLA) models usually map the current observation and a language instruction to actions (Zitkovich et al., 2023; Black et al., 2025a; Physical Intelligence, 2025; Gemini Robotics et al., 2025), and GR00T N1 (Bjorck et al., 2025) and Ψ_0 (Wei et al., 2026a) further extend them to humanoid robots. To compensate for scarce robot data, recent work collects egocentric human video at scale (Hoque et al., 2026; Punamiya et al., 2026), which Ψ_0 uses to pre-train its VLM on human hand actions. Despite this progress, VLAs generalize poorly beyond their training distribution and tend to replay memorized trajectories while ignoring the instruction (Zhou et al., 2025; Fei et al., 2025). Our planner builds on Ψ_0 but additionally takes a future image as input, and it learns to use this image by predicting variable-length action sequences between current and future frames, mostly from human video.

World models for robot control. World action models (WAMs) add future prediction to the policy, either by generating future video together with the actions (Wu et al., 2024; Ye et al., 2026; Li et al., 2026; NVIDIA et al., 2026) or by conditioning the actions on video-model features or future latents (Ma et al., 2026; Luo et al., 2026a). The predicted future remains an output or internal feature of the policy while its inputs stay unchanged, so it shapes the actions only implicitly. Another line of work generates a goal and passes it to the policy as an input, such as subgoal images from an image-editing model in SuSIE (Black et al., 2024) or a video plan followed by inverse dynamics in UniPi (Du et al., 2023). $\pi_{0.7}$ (Physical Intelligence, 2026) introduces this idea into a generalist VLA, adding subgoal images of the current subtask to its prompt to guide long-horizon tasks. GigaBrain-0.7 (GigaBrain Team et al., 2026) further conditions its action model on the last frame of a predicted subtask video, and the concurrent τ_0-VLA (Cai et al., 2026) uses predicted outcome images to choose which subtask to execute next. Unlike these images of subtask completion, our future image lies a chosen time Δt ahead and is generated as a minimal clip, cheap enough to refresh while the planner keeps acting.

Humanoid whole-body control. A humanoid VLA needs an interface through which its actions reach the whole body. However, previous VLA models (Wei et al., 2026a; Bjorck et al., 2025) predict upper-body joint targets while delegating locomotion and posture control to a separate lower-body controller, AMO (Li et al., 2025). This decoupled control architecture limits the range of whole-body motions they can perform Wei et al. (2026b). In contrast, whole-body teleoperation systems such as TWIST2 (Ze et al., 2025) retarget full-body human motion to humanoid robots, producing datasets such as UnifoLM-WBT (Unitree Robotics, 2026) that contain demonstrations of coordinated whole-body behaviors. SONIC (Luo et al., 2026b) further scales whole-body motion tracking and provides a token-based interface through which either VR teleoperation or a VLA can control the entire body. This interface enables coordinated behaviors such as bending down to carry a large box or lifting a foot to press a pedal that opens a trash bin. We therefore adopt the SONIC interface: our planner predicts SONIC tokens alongside hand and neck targets and executes them using real-time chunking (Black et al., 2025b). This design enables coordinated motions that are difficult to achieve with decoupled whole-body control architectures.

3 Method

3.1 Problem Formulation

We consider a humanoid robot performing a long-horizon task specified by a language instruction L. At time t, the robot receives an observation $o_t = (I_t, q_t)$, consisting of an egocentric image I_t and its proprioceptive state q_t, and predicts an action chunk $a_{t:t+H} = (a_t, \dots, a_{t+H-1})$ of whole-body commands, which a low-level controller executes on the robot.

A standard VLA models $\pi(a_{t:t+H} \mid o_t, L)$ and has to infer from the instruction alone which object to handle, where to go, and when a stage is finished. World action models build on pretrained video generation models and generate future frames together with the actions, so the co-generated video guides the actions only implicitly. $\Psi_{0.1}$ instead gives the future to the policy explicitly as an input, an image $I_{t+\Delta t}$ of what the robot should see Δt seconds later, and models $\pi(a_{t:t+H} \mid o_t, L, I_{t+\Delta t})$.

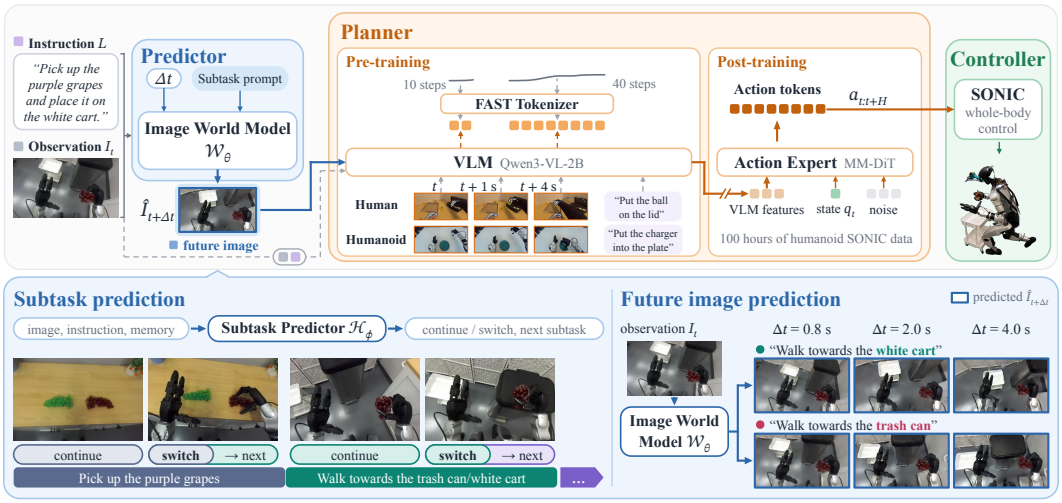


Figure 1: Overview of $\Psi_{0.1}$. Top: the predictor turns the instruction and the current image I_t into a future image $\hat{I}_{t+\Delta t}$, which steers the planner. The planner's VLM is pre-trained to predict variable-length actions between current and future frames and then frozen; The action expert is post-trained to generate SONIC motion tokens $a_{t:t+H}$ that the whole-body controller executes. Bottom left: the subtask predictor $\mathcal{H}_\phi$ decides whether to continue or switch to the next subtask prompt. Bottom right: future images predicted from one observation for two prompts at $\Delta t = 0.8$, 2.0, and 4.0 s.

The future image grounds the instruction in the current scene: it shows which object to handle and where to bring it, and together with Δt it indicates how far the task should progress by then.

Recorded future frames are available in demonstrations but not when the robot executes a new task. $\Psi_{0.1}$ therefore splits the model into a predictor P that generates the future image and a planner π that turns it into actions (Fig. 1),

$$\hat{I}_{t+\Delta t} \sim P(\cdot \mid I_t,\, L,\, \Delta t)\,, \qquad a_{t:t+H} \sim \pi\Big(\cdot \mid o_t,\, L,\, \hat{I}_{t+\Delta t}\Big)\,. \tag{1}$$

These two models run asynchronously, with the planner acting on the latest future image while the predictor refreshes it. Sections 3.2–3.4 describe the predictor, the planner, and the whole-body controller that executes the planner's actions.

3.2 Predictor: From Instructions to Future Images

The instruction steers the humanoid robot through a future image, and the predictor produces it in two steps (Fig. 1, bottom). A subtask predictor $\mathcal{H}_\phi$ turns the instruction into the prompt ℓ_k of the step the robot is on. An image world model $\mathcal{W}_\theta$ then renders how the scene should look Δt seconds later while this step is carried out, $\hat{I}_{t+\Delta t} \sim \mathcal{W}_\theta(\cdot \mid I_t,\, \ell_k,\, \Delta t)$. Different prompts thus produce different future images and eventually steer the humanoid action (Fig. 1, bottom right).

Predicting the next subtask. A long instruction consists of many steps, but the world model needs the prompt of the current one. $\mathcal{H}_\phi$ is a vision-language model (Qwen3-VL-4B (Bai et al., 2025)) that predicts the future subtask: from the current image, the instruction, and the subtasks started so far, it decides whether the current subtask is still going on or is done, and gives the prompt of the next one, $(d_t,\ \ell_{k+1}) = \mathcal{H}_\phi(I_t,\, L,\, \ell_{1:k})\,, d_t \in \{\texttt{continue}, \texttt{switch}\}$. When it answers `switch`, ℓ_{k+1} becomes the active prompt (Fig. 1, bottom left). The model also outputs a rough estimate of the time left in the current subtask. The input also includes the embodiment, since different embodiments move at different speeds. We train $\mathcal{H}_\phi$ with the next-token loss on subtask annotations of about 70 hours of humanoid demonstrations with about 37k subtasks. About 35% of the data is manually annotated and the rest is annotated by an automatic annotator that uses a vision-language model coarsely locates each subtask boundary, and the joint actions of the humanoid pin it to the frame where the subtask ends (Appendix B).

Predicting the future image. $\mathcal{W}_\theta$ is built on the unified model Cosmos3-Nano (NVIDIA et al., 2026). Because of the image-to-video pretraining of Cosmos3-Nano, $\mathcal{W}_\theta$ inherits the ability to render a plausible future of the current scene for a given instruction. Unlike the video branch of a WAM, which generates a dense clip over a short time window, $\mathcal{W}_\theta$ needs only one image further ahead, as a result, it generates one single latent frame. The time delta enters in two ways: the caption states it in seconds, and the frame rate of the clip, set by one of three horizon bins (0.8, 1.6 and 3.2 s), places the frames along the time axis of the rotary position embedding of Cosmos3. Trained over a continuous range of Δt, the model acquires a sense of time: from the same observation, a larger Δt gives more progress (Fig. 1, bottom right), so one model serves tasks of different pace, with Δt chosen per task (Appendix E). The caption also names the embodiment, since different embodiments move at different speeds. We train only the generation tower, mostly on about 1.75k hours of human egocentric video (EgoDex (Hoque et al., 2026) and EgoVerse (Punamiya et al., 2026)) together with humanoid data (Humanoid Everyday (Zhao et al., 2025) and UnifoLM (Unitree Robotics, 2026)), and then fine-tune briefly on our in-house humanoid data (Appendix C).

3.3 Planner: From Future Images to Whole-Body Actions

The planner follows the architecture of Ψ_0 (Wei et al., 2026a). A VLM backbone (Qwen3-VL-2B (Bai et al., 2025)) extracts visual-language features, and a flow-based MM-DiT action expert generates action chunks from these features and the proprioceptive state. We extend the VLM input to the current image, the instruction, and the future image, in this order, so that under causal attention the future-image tokens attend to both the current observation and the instruction. The planner is trained in three stages: we pre-train the VLM to predict the actions between current and future images, post-train the action expert on whole-body humanoid data, and fine-tune it on the target tasks.

Pre-training with variable-length action chunks. A future image shows the state the robot should reach, and the actions needed to reach it depend on how far ahead that state lies. We therefore train the VLM to predict exactly the actions that connect the current observation to the future image. For a current frame I_t, we sample a future frame $I_{t+\Delta t}$ from the same trajectory with $\Delta t \in [1, 4]$ s, and the target is the action sequence $a_{t:t+\Delta t}$ in between. Paired with nearer or farther futures, the same observation leads to shorter or longer action sequences with different content, so the VLM has to read the future image to decide which motion to perform and how far to proceed.

We start from 829 hours of human egocentric video in EgoDex (Hoque et al., 2026). Following Ψ_0, we represent actions in a 48-dimensional end-effector space of wrist poses and fingertip positions of both hands, expressed relative to the current frame. The actions are downsampled from 30 Hz to 10 Hz, giving sequences of 10–40 steps, and tokenized with a FAST tokenizer (Pertsch et al., 2025) fitted on sequences of up to 40 steps, which yields about 300 tokens per sequence on average. The VLM is trained with the next-token prediction loss on these action tokens y,

$$\mathcal{L}_{\text{pre}} = -\sum_j \log p(y_j \mid y_{<j},\, I_t,\, L,\, I_{t+\Delta t})\,. \tag{2}$$

We then continue pre-training on real-world humanoid demonstrations from Humanoid Everyday (Zhao et al., 2025) and UnifoLM-WBT (Unitree Robotics, 2026) with the same action representation and objective. At deployment, however, the planner is guided by predicted rather than real future frames. We therefore generate future images for these demonstrations with our image world model and use them in place of the ground-truth frames with probability 0.5, which exposes the VLM to the imperfections of generated images. In addition, for 25% of the samples, we use the end frame of the current subtask (Sec. 3.2) as the future image, clipped to 1–4 s ahead, which gives the VLM a prior on what the completion of a step looks like.

Post-training on whole-body humanoid data. We then freeze the VLM and train the action expert to turn its features into whole-body actions. The training data consists of about 100 hours of humanoid demonstrations from Humanoid Everyday, UnifoLM-WBT, and Ψ_0, together with a small portion of real-world datasets teleoperated directly through SONIC. Since most of these demonstrations are not recorded in the action space of SONIC, we first convert their whole-body motions into SONIC motion tokens with the SONIC v1.1 encoder (Luo et al., 2026b) (Appendix D). The action expert then learns to generate these tokens, which the controller executes directly (Sec. 3.4).

Following Ψ_0, the action expert is trained with flow matching (Lipman et al., 2023) to generate action chunks of length $H = 30$ from the VLM features and the proprioceptive state. We double the depth of the action header to 12 blocks to increase its capacity, and let each block attend to features from a different VLM layer from shallow to deep, so that action generation can draw on both fine-grained visual detail and high-level semantics. We also project the proprioceptive state into a token prepended to the action tokens, so that state and actions interact in every block. During post-training, generated future images again replace ground-truth ones with probability 0.5, and the state is dropped with probability 0.5 to keep the action expert from relying on proprioception alone.

Fine-tuning on multiple tasks. Finally, we fine-tune the VLA model on demonstrations of various tasks, including long-horizon whole-body tasks as well as short-horizon tasks that emphasize steerability. During fine-tuning, the VLM remains frozen, and only the action expert is optimized. Unlike the earlier stages, fine-tuning uses only ground-truth future frames sampled 1–4 s ahead. We empirically found that this avoids the computational cost of running the predictor without a performance drop.

3.4 Controller: Real-Time Whole-Body Execution

Whole-body control through SONIC. Humanoid VLAs often rely on a decoupled controller that tracks upper-body targets and locomotion commands separately, which cannot produce whole-body motions such as kneeling or stepping on a pedal with the foot to pop open a trashcan. We instead control the whole body through SONIC (Luo et al., 2026b), whose encoder maps human motion into a 64-dimensional motion token and whose control policy decodes the token into joint-position targets for the 29 body joints. The planner therefore outputs actions $a_t = [\, q_t^{\text{hand}};\, z_t;\, q_t^{\text{neck}} \,] \in \mathbb{R}^{80}$, consisting of 14-dimensional hand joint targets, a SONIC motion token z_t, and 2-dimensional neck joint targets, and its proprioceptive state q_t contains the positions of all 45 body, hand, and neck joints.

Real-time action chunking. To execute action chunks asynchronously without latency-induced discontinuities at chunk boundaries, we adopt inference-time Real-Time Chunking (RTC) (Black et al., 2025b), which we empirically found to overfit less than training-time RTC (Black et al., 2025c). RTC generates each new chunk as an inpainting problem conditioned on the actions still being executed, and we normalize its guidance term by its own magnitude to keep the guidance stable across noise levels (Appendix F).

4 Experiments

We evaluate $\Psi_{0.1}$ on a real humanoid from three aspects: in Sec. 4.2, we compare how reliably $\Psi_{0.1}$, VLAs, and WAMs follow instructions, particularly with novel objects, containers, and layouts; in Sec. 4.3, we use the same steering to compose separate tasks into a routine of several minutes and to steer it into an early transition unseen in training; and in Sec. 4.4, we measure how much the future-image input, variable-length pre-training, and humanoid post-training each contribute.

4.1 Implementation

Hardware platform. We conduct all real-world experiments on a Unitree G1 humanoid with 29 body degrees of freedom (DoFs) and two Dex3-1 dexterous hands, each with 7 DoFs. We equip the robot with the open-source 2-DoF neck from TWIST2 (Ze et al., 2025) and mount a ZED Mini camera on it to provide egocentric visual observations. The actuated neck allows the robot to reorient its camera, extending visual coverage during locomotion and manipulation.

Training details. The VLM is pre-trained on human and real robot data for 20 days on 32 NVIDIA H100 GPUs, with a constant learning rate of 10^{-4} and a global batch size of 1,024, reaching an action prediction L1 error of about 0.025. Given a ground-truth future image 2.4 s ahead, the duration of its predicted action sequence deviates from 2.4 s by only about 0.3 s, indicating that the VLM infers from the future image how long the robot should act. The action expert, with about 670M parameters, is post-trained for three days on 16 H100 GPUs with a global batch size of 512 and

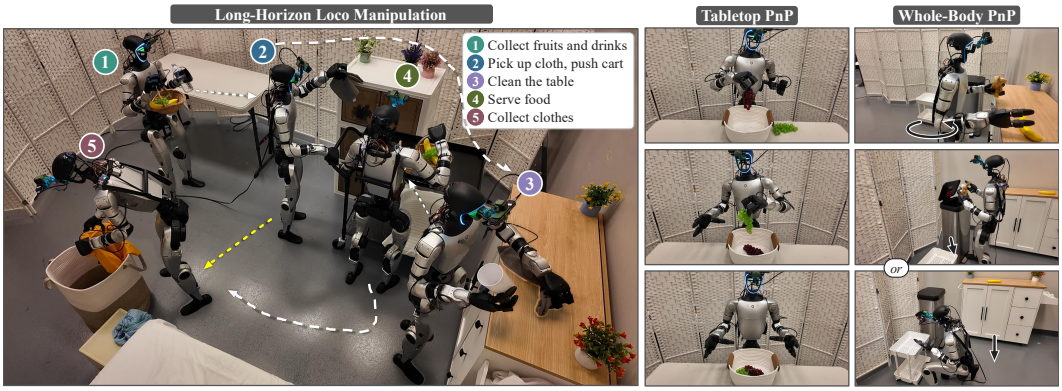


Figure 2: Real-world tasks performed by $\Psi_{0.1}$. Left: the long-horizon routine, with the robot composited at the key moment of each task; white dashed arrows give the task order, and the yellow arrow marks the unseen early transition from Task 2 directly to Task 5. Right: tabletop PnP and whole-body PnP, where the robot carries an object from the island to the trash bin or the cart. Arrows mark motions that a single frame does not show.

a cosine learning rate schedule that peaks at 10^{-4} after 1,000 warmup steps. Fine-tuning runs for 40,000 steps with the same schedule and adds Gaussian noise to the proprioceptive input for robustness to state variations at deployment. Training details of the predictor are given in Appendices B and C.

Deployment. $\Psi_{0.1}$ runs locally on two NVIDIA GeForce RTX 5090 GPUs, one for the planner and one for the predictor. The subtask predictor is queried once per second and the world model every 0.8 s, generating future images $\Delta t = 2.4$ s ahead. The planner always conditions on the most recent future image, and its action chunks are executed by SONIC with real-time chunking (Sec. 3.4). Further details are given in Appendix E.

Baselines. We compare against two WAMs, Cosmos3 (NVIDIA et al., 2026) and DiT4DiT (Ma et al., 2026), and two humanoid VLAs, Ψ_0 (Wei et al., 2026a) and GR00T N1.7 (Bjorck et al., 2025). Cosmos3 generates future video and actions jointly within one multimodal model, while DiT4DiT conditions an action diffusion transformer on intermediate features of a video diffusion transformer. Ψ_0 pre-trains its VLM on egocentric human video and trains a flow-based action expert on humanoid data, and GR00T N1.7 combines a reasoning VLM with a diffusion action head and learns manipulation priors from large-scale egocentric human data. All baselines start from their released pre-trained weights and are fine-tuned on the same multi-task demonstrations for the same 40,000 steps as $\Psi_{0.1}$.

4.2 Language Steerability and Generalization

We evaluate language steering on two pick-and-place (PnP) tasks (Fig. 2, right). In tabletop PnP, the robot stands at a table with two objects and a container, and the instruction specifies which object to place and in which order. In whole-body PnP, the robot picks up the instructed object and carries it to the instructed destination. It either walks to a trash bin, opens the lid by stepping on its pedal, and discards the object, or walks to a low cart, kneels on one knee to place the object, and stands back up. Beyond the in-domain setting, we test novel objects and containers in tabletop PnP, and novel objects and layouts of the bin and cart in whole-body PnP (Appendix A). We run 10 trials per in-domain setting and 6 per novel setting and report success rate (SR), which requires completing all stages without assistance, and Task Progress, the fraction of completed milestones. When a stage fails, an evaluator intervenes so that later stages can still be evaluated, without crediting the assisted stage.

As shown in Fig. 3(a), $\Psi_{0.1}$ achieves the highest or tied-highest success rate in every setting. In the in-domain settings, it performs slightly better than the strongest baseline, whereas in the novel settings it outperforms all baselines by a large margin and performs nearly as well as in-domain. Ψ_0

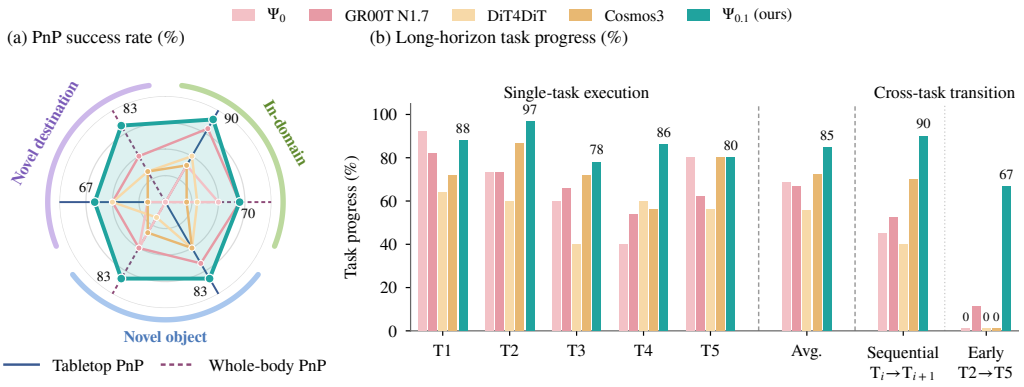


Figure 3: Real-world results. (a) Success rate on tabletop and whole-body PnP. The outer arcs group the settings into in-domain, novel object, and novel destination (a novel container for tabletop PnP and a novel layout for whole-body PnP), with 10 trials per in-domain setting and 6 per novel setting. (b) Task progress on the long-horizon routine, evaluated on single-task execution (T1–T5), on average over the full routine (Avg.), and on cross-task transitions, either in the original order (Sequential) or from Task 2 directly to Task 5 (Early).

fails completely with novel tabletop objects and novel layouts, mechanically repeating motions from training instead of following the instruction, and the two WAMs do not consistently outperform the VLAs, suggesting that co-generated video alone does not make a policy follow instructions well in unfamiliar scenes. Guided by a future image of the current scene, $\Psi_{0.1}$ remains steerable when the objects and destinations change.

4.3 Long-Horizon Planning and Task Composition

We next test whether steering can chain tasks that are demonstrated separately into a long-horizon routine. In one household environment, we collect 40 demonstrations for each of five tasks (Fig. 2, left; see Fig. 6 in the appendix for the full procedure): collecting fruits and drinks (Task 1), picking up a cloth and pushing a cart (Task 2), cleaning the table (Task 3), pouring water and serving food (Task 4), and collecting clothes into a laundry basket (Task 5). Each demonstration covers a single task, so no demonstration shows the full routine that the five tasks form when performed in order. $\Psi_{0.1}$ completes this routine in about five minutes on average.

Beyond the sequential transitions between consecutive tasks, we test an early transition, in which the robot is instructed to go from Task 2 directly to Task 5, skipping Tasks 3 and 4 (yellow arrow in Fig. 2). In the training data, the end of Task 2 is always followed by the start of Task 3, so the scene from which the robot must begin Task 5 is unseen. We therefore provide a single demonstration of the early transition as one-shot guidance, add it to the multi-task data, and continue fine-tuning each method for 2,000 steps. For $\Psi_{0.1}$, only the planner is fine-tuned, and the predictor is used as is. This setting tests whether a policy can be steered into a new task order from minimal data.

We report Task Progress for each task and its average over the full routine, where the average weights the milestones of all stages. For a cross-task transition, we check whether the robot completes the first milestone of the next task after finishing the current one, and report this for the sequential transitions, averaged over the four transitions of the original order (Task 1→2 through Task 4→5), and for the early transition from Task 2 to Task 5.

As shown in Fig. 3(b), $\Psi_{0.1}$ outperforms all baselines on average in single-task execution, reflecting precise execution of the in-domain tasks. On cross-task transitions, it surpasses all baselines by a wide margin, both on the sequential transitions and on the early transition, where a single demonstration suffices to steer it into the new task order. These results highlight how much explicit visual guidance contributes to steering the policy across long-horizon tasks.

Table 1: Ablation of the planner. Each row adds or removes one component; row (f) is the full $\Psi_{0.1}$. Success is reported as successful trials out of 10.

		VLM pre-training		Planner		Success	
	VLM	Images	Action target	Future-image input	Humanoid post-training	Tabletop PnP	Whole-body PnP
(a)	Ψ_0	Current only	Next action	✗	✗	1/10	2/10
(b)	Qwen3-VL	–	–	✓	✗	5/10	3/10
(c)	Ψ_0	Current only	Next action	✓	✗	5/10	4/10
(d)	Ours	Current, future	Fixed-length	✓	✗	4/10	4/10
(e)	Ours	Current, future	Variable-length	✓	✗	7/10	7/10
(f)	Ours	Current, future	Variable-length	✓	✓	**8/10**	**8/10**

4.4 Ablation Study

We ablate the planner on tabletop and whole-body PnP with 10 trials per task, 5 in the in-domain setting and 5 in the novel settings of Sec. 4.2, using the same success criterion. Table 1 lists the variants by how the VLM is pre-trained (on the current image only or on current and future images, and with which action target), whether the planner receives a future image as input, and whether the action expert is post-trained on humanoid data. In all variants, the VLM is frozen and only the action expert is fine-tuned on the same task data.

Future-image input. With the same Ψ_0 VLM, adding the future image (a→c) raises success from 1/10 to 5/10 on tabletop PnP and from 2/10 to 4/10 on whole-body PnP. The future image already steers the planner even though this VLM was never pre-trained on current and future image pairs.

VLM pre-training. Starting from Qwen3-VL without action pre-training, the single-image pre-training of Ψ_0 (b→c) improves whole-body success from 3/10 to 4/10. Surprisingly, pre-training on current and future image pairs with a fixed-length action target (d) brings no further gain and even lowers tabletop success to 4/10. One explanation is that a fixed-length target stays the same whether the future image lies 1 or 4 s ahead, so the image adds little to the prediction. The VLM can then learn to ignore it as noise and fall back on the current image, losing the temporal cues that the future image is meant to provide. Tying the length of the action sequence to the future image (e) instead raises success to 7/10 on both tasks, suggesting that variable-length pre-training teaches the VLM to read the future image and gives the planner stronger steerability.

Humanoid post-training. With our VLM and the future image, post-training the action expert on humanoid data (e→f) further raises success from 7/10 to 8/10 on both tasks, as the action expert learns to turn the visual guidance into whole-body motion before task-specific fine-tuning. Ablations of the predictor, including its inference speed, future-image quality, and subtask prediction, are reported in Appendix G.

5 Conclusion

We presented $\Psi_{0.1}$, a humanoid foundation model that makes whole-body behavior explicitly steerable through future images. Instead of letting co-generated video guide the actions implicitly, $\Psi_{0.1}$ feeds a predicted future image to the policy as an input: a predictor renders how the scene should look after a time delta, a planner pre-trained with variable-length action chunks turns this image into actions, and a real-time controller executes them as whole-body motion. In real-world experiments, this explicit visual guidance keeps the humanoid steerable with novel objects and destinations, chains separately demonstrated tasks into routines of up to five minutes, and steers the robot into a new task order from a single demonstration. The performance of $\Psi_{0.1}$ is currently bounded by the quality of the predicted future images, and the predictor needs to scale to more data to generate reliable future images in more diverse novel scenes. In future work, we plan to integrate an agent for high-level reasoning into the predictor, moving toward a general-purpose predictor that reasons about tasks and generates the corresponding future images.

AI USE STATEMENT

We did not use generative AI tools for any of the tasks that require disclosure under the ICLR 2027 AI Policy for Authors. We used generative AI tools to edit the paper for readability and to help create and refine scientific figures. We have reviewed all AI-assisted work, and we take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

Kevin Black, Mitsuhiko Nakamoto, Pranav Atreya, Homer Walke, Chelsea Finn, Aviral Kumar, and Sergey Levine. Zero-shot robotic manipulation with pretrained image-editing diffusion models. In *International Conference on Learning Representations*, 2024.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, et al. π_0: A vision-language-action flow model for general robot control. In *Robotics: Science and Systems*, 2025a. doi: 10.15607/RSS.2025.XXI.010.

Kevin Black, Manuel Galliker, and Sergey Levine. Real-time execution of action chunking flow policies. *Advances in Neural Information Processing Systems*, 38:37596–37620, 2025b.

Kevin Black, Allen Z Ren, Michael Equi, and Sergey Levine. Training-time action conditioning for efficient real-time chunking. *arXiv preprint arXiv:2512.05964*, 2025c.

Xiaowei Cai, Yunuo Cai, Bingao Chen, Jingxiao Chen, Zhi Chen, Siyuan Feng, Tengyu Hou, Jingshun Huang, Han Jiang, Runkun Ju, et al. τ_0-VLA: a hierarchical robot foundation model with world-model-guided test-time computation. *arXiv preprint arXiv:2608.16885*, 2026.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, Guang Shi, and Haoqi Fan. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025a.

Shengliang Deng, Mi Yan, Songlin Wei, Haixin Ma, Yuxin Yang, Jiayi Chen, Zhiqi Zhang, Taoyu Yang, Xuheng Zhang, Heming Cui, Zhizheng Zhang, and He Wang. GraspVLA: a grasping foundation model pre-trained on billion-scale synthetic action data. In *9th Annual Conference on Robot Learning*, 2025b. URL https://openreview.net/forum?id=zEC8TOXDkH.

Yilun Du, Sherry Yang, Bo Dai, Hanjun Dai, Ofir Nachum, Josh Tenenbaum, Dale Schuurmans, and Pieter Abbeel. Learning universal policies via text-guided video generation. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1d5b9233ad716a43be5c0d3023cb82d0-Abstract-Conference.html.

Senyu Fei, Siyin Wang, Junhao Shi, Zihao Dai, Jikun Cai, Pengfang Qian, Li Ji, Xinzhe He, et al. LIBERO-Plus: In-depth robustness analysis of vision-language-action models. *arXiv preprint arXiv:2510.13626*, 2025.

Gemini Robotics, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Travis Armstrong, Ashwin Balakrishna, Robert Baruch, Maria Bauza, Michiel Blokzijl, et al. Gemini Robotics: Bringing AI into the physical world. *arXiv preprint arXiv:2503.20020*, 2025.

GigaBrain Team, Angen Ye, Axiang Sun, Can Jin, Chenxi Cheng, Chong Shi, Dengke Shang, Dingqian Zhang, Guan Huang, Guangqiang Wang, et al. GigaBrain-0.7: Scaling embodied foundation models to emergent capabilities with a three-system architecture. *arXiv preprint arXiv:2608.15875*, 2026.

Ryan Hoque, Peide Huang, David J. Yoon, Mouli Sivapurapu, and Jian Zhang. EgoDex: Learning dexterous manipulation from large-scale egocentric video. In *International Conference on Learning Representations*, 2026.

Jialong Li, Xuxin Cheng, Tianshu Huang, Shiqi Yang, Ri-Zhao Qiu, and Xiaolong Wang. AMO: Adaptive motion optimization for hyper-dexterous humanoid whole-body control. In *Robotics: Science and Systems*, 2025. doi: 10.15607/RSS.2025.XXI.061.

Lin Li, Qihang Zhang, Yiming Luo, Shuai Yang, Ruilin Wang, Fei Han, Mingrui Yu, Zelin Gao, Nan Xue, Xing Zhu, Yujun Shen, and Yinghao Xu. Causal world modeling for robot control. *arXiv preprint arXiv:2601.21998*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Hao Luo, Wanpeng Zhang, Yicheng Feng, Sipeng Zheng, Haiweng Xu, Chaoyi Xu, Ziheng Xi, Yuhui Fu, and Zongqing Lu. Being-H0.7: A latent world-action model from egocentric videos. *arXiv preprint arXiv:2605.00078*, 2026a.

Zhengyi Luo, Ye Yuan, Tingwu Wang, Chenran Li, Fernando Castañeda, Sirui Chen, Zi-Ang Cao, Jiefeng Li, David Minor, Qingwei Ben, Jinhyung Park, David Sami, Zi Wang, Xingye Da, Runyu Ding, Cyrus Hogg, Lina Song, Edy Lim, Eugene Jeong, Tairan He, Haoru Xue, Wenli Xiao, Simon Yuen, Jan Kautz, Yan Chang, Umar Iqbal, Linxi "Jim" Fan, and Yuke Zhu. SONIC: Supersizing motion tracking for natural humanoid whole-body control. *Science Robotics*, 11 (117), August 2026b. ISSN 2470-9476. doi: 10.1126/scirobotics.aed4592. URL http://dx.doi.org/10.1126/scirobotics.aed4592.

Teli Ma, Jia Zheng, Zifan Wang, Chunli Jiang, Andy Cui, Junwei Liang, and Shuo Yang. DiT4DiT: Jointly modeling video dynamics and actions for generalizable robot control. *arXiv preprint arXiv:2603.10448*, 2026.

Toan Nguyen, Weiduo Yuan, Songlin Wei, Hui Li, Daniel Seita, and Yue Wang. ICLR: In-context imitation learning with visual reasoning. *arXiv preprint arXiv:2603.07530*, 2026.

NVIDIA, 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.

NVIDIA, Aditi, Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026.

Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. FAST: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

Physical Intelligence. $\pi_{0.5}$: a vision-language-action model with open-world generalization. *arXiv preprint arXiv:2504.16054*, 2025.

Physical Intelligence. $\pi_{0.7}$: A steerable generalist robotic foundation model with emergent capabilities. *arXiv preprint arXiv:2604.15483*, 2026.

Ryan Punamiya, Simar Kareer, Zeyi Liu, Josh Citron, Ri-Zhao Qiu, Xiongyi Cai, Alexey Gavryushin, Jiaqi Chen, Davide Liconti, Lawrence Y Zhu, et al. EgoVerse: An egocentric human dataset for robot learning from around the world. *arXiv preprint arXiv:2604.07607*, 2026.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International conference on learning representations*, 2023.

Unitree Robotics. UnifoLM-WBT-Dataset. https://huggingface.co/collections/unitreerobotics/unifolm-wbt-dataset, 2026. Humanoid whole-body teleoperation dataset.

Songlin Wei, Hongyi Jing, Boqian Li, Zhenyu Zhao, Jiageng Mao, Zhenhao Ni, Sicheng He, Sheng Zang, Xiawei Liu, Kaidi Kang, Jie Liu, Weiduo Yuan, Marco Pavone, Di Huang, and Yue Wang. Ψ_0: An open foundation model towards universal humanoid loco-manipulation. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2026a. doi: 10.15607/RSS.2026.XXII.021.

Songlin Wei, Zhenhao Ni, Jie Liu, Zhenyu Zhao, Junjie Ye, Hongyi Jing, Junkai Xia, Xiawei Liu, Michael Leong, Liang Heng, et al. Simple: Simulation-based policy learning and evaluation for humanoid loco-manipulation. *arXiv preprint arXiv:2606.08278*, 2026b.

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. In *International Conference on Learning Representations*, 2024.

Seonghyeon Ye, Yunhao Ge, Kaiyuan Zheng, Shenyuan Gao, Sihyun Yu, George Kurian, Suneel Indupuru, You Liang Tan, Chuning Zhu, Jiannan Xiang, et al. World action models are zero-shot policies. *arXiv preprint arXiv:2602.15922*, 2026.

Yanjie Ze, Siheng Zhao, Weizhuo Wang, Angjoo Kanazawa, Rocky Duan, Pieter Abbeel, Guanya Shi, Jiajun Wu, and C. Karen Liu. TWIST2: Scalable, portable, and holistic humanoid data collection system, 2025. URL https://arxiv.org/abs/2511.02832.

Jiazhao Zhang, Kunyu Wang, Shaoan Wang, Minghan Li, Haoran Liu, Songlin Wei, Zhongyuan Wang, Zhizheng Zhang, and He Wang. Uni-NaVid: A Video-based Vision-Language-Action Model for Unifying Embodied Navigation Tasks. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2025. doi: 10.15607/RSS.2025.XXI.013.

Zhenyu Zhao, Hongyi Jing, Xiawei Liu, Jiageng Mao, Abha Jha, Hanwen Yang, Rong Xue, Sergey Zakharov, Vitor Guizilini, and Yue Wang. Humanoid Everyday: A comprehensive robotic dataset for open-world humanoid manipulation. *arXiv preprint arXiv:2510.08807*, 2025.

Xueyang Zhou, Yangming Xu, Guiyao Tie, Yongchao Chen, Guowen Zhang, Duanfeng Chu, Pan Zhou, and Lichao Sun. LIBERO-PRO: Towards robust and fair evaluation of vision-language-action models beyond memorization. *arXiv preprint arXiv:2510.03827*, 2025.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183, 2023. URL https://proceedings.mlr.press/v229/zitkovich23a.html.

A Environment Setups

All experiments run on the hardware described in Sec. 4.1. This section describes the scene of each task and how the generalization settings change it.

A.1 Tabletop pick-and-place.

The robot stands at a table with one object on its left, one on its right and a container between them (Fig. 4). The instruction names the object to put into the container, or the order in which to put both. The in-domain objects are a banana, a coffee can, a green apple, green grapes, a plum, a water bottle and a yellow toy lion, and the in-domain container is a white basket. The novel-object setting uses purple grapes, a red pepper, yellow corn and a bottle of iced tea; the novel-container setting uses a pink box, an orange plate and a white plate. Neither appears in the demonstrations of this task.

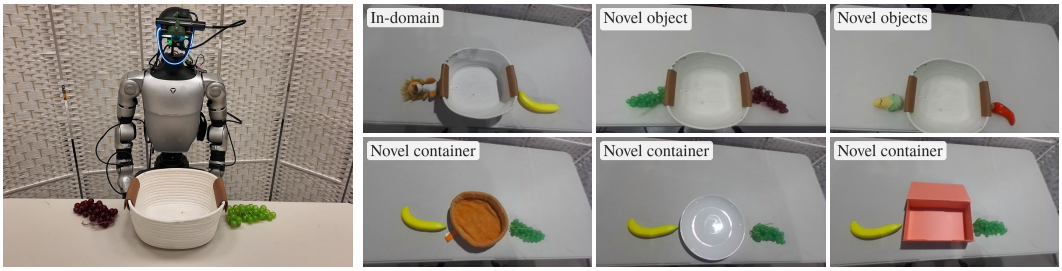


Figure 4: Tabletop pick-and-place. Left: third-person view. Right: the robot's head camera at the start of an episode, in the in-domain setting and with novel objects or containers.

A.2 Whole-body pick-and-place.

The robot stands at one end of a kitchen island that holds the objects. Behind it stand a trash bin with a foot pedal and a low two-tier cart. The instruction sets two choices: which object to take from the island, and where to bring it (Fig. 5). Throwing the object means turning, walking to the bin and stepping on the pedal to open the lid; placing it on the cart means walking to the cart and kneeling on one knee. The in-domain objects are those of the tabletop task. The novel-object setting uses purple grapes, yellow corn, a red pepper and three toys (a grey hippo, a zebra and a brown monkey). In training, the bin and the cart stand so that the robot turns left by about 90° to face them. The novel-layout setting moves them so that the robot has to turn by about 130° or 180° (Fig. 5, left), or swaps their places.

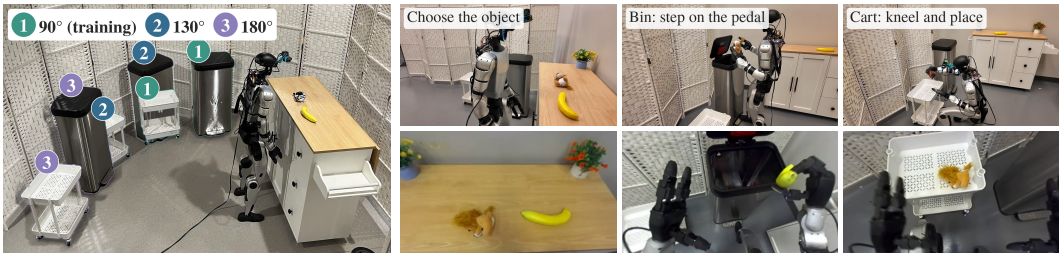


Figure 5: Whole-body pick-and-place. Left: the scene, with the kitchen island in front of the robot and the trash bin and the cart behind it; the three placements of the bin and the cart were photographed from the same viewpoint and composited into one image, and the legend gives how far the robot turns to face them (90° is the training layout). Right: the two choices of the task from a third-person view (top) and from the robot's head camera in a demonstration (bottom).

A.3 LONG-HORIZON HOUSEHOLD ROUTINE.

Five tasks take place in one room (Fig. 6): a white folding table with a fruit basket and a bottle, a black two-tier cart, a storage cabinet, a kitchen island, and a bed next to a laundry basket. The robot brings the fruit basket and the bottle from the folding table to the cart (Task 1), takes a cloth from the cabinet and pushes the cart to the island (Task 2), cleans the island (Task 3), serves the basket and the drink beside the cup on the island (Task 4), and puts the clothes from the bed and a stool into the laundry basket (Task 5). Fig. 2 (left) shows the room from above, and Fig. 6 (top) shows each task in three frames. In the one-shot setting, the robot goes from Task 2 directly to Task 5, skipping the cleaning and the serving (Fig. 6, bottom).

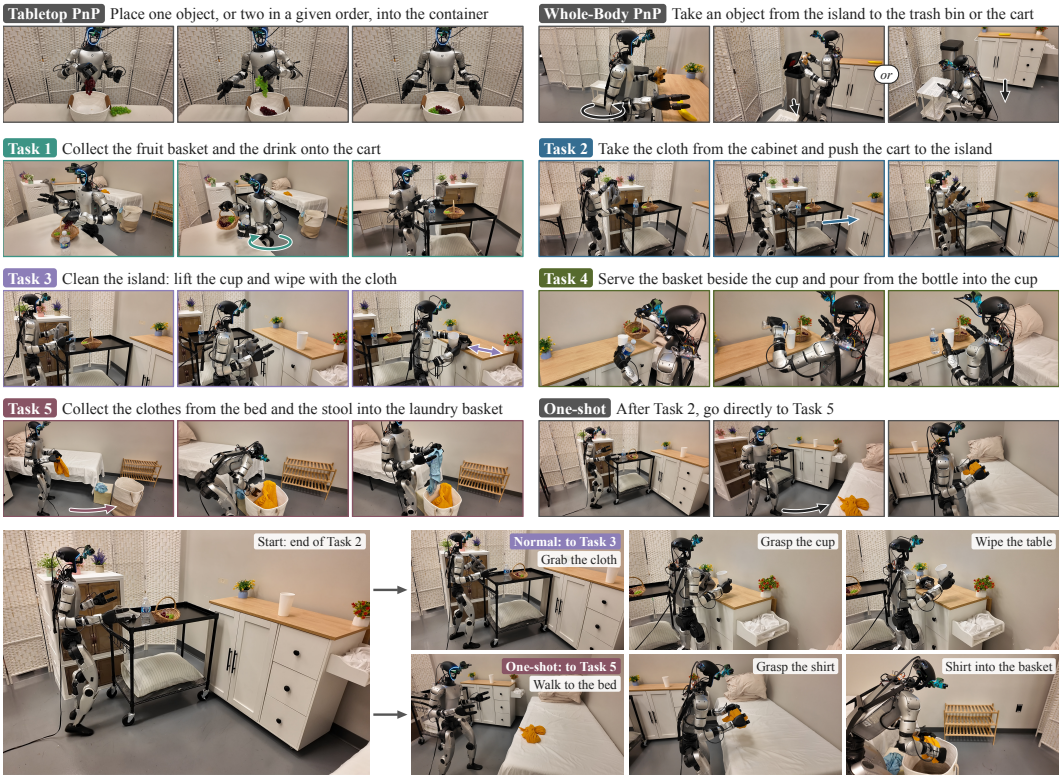


Figure 6: The tasks in more detail. Top: each task in three third-person frames: tabletop and whole-body pick-and-place (in the whole-body row, the second and third frames show the two destinations, from two episodes), the five tasks of the long-horizon routine, and the one-shot transition; arrows mark motions that a single frame does not show. Bottom: the one-shot setting. Both runs start at the end of Task 2 (left); the normal routine continues with Task 3 (top row), and the one-shot run goes directly to Task 5 (bottom row).

B SUBTASK PREDICTOR

Data and annotation. The subtask predictor is trained on 72 hours of humanoid demonstrations (11,810 episodes) with 36,570 annotated subtasks, about 7 s each on average. About 25 hours are annotated by hand: we mark the subtask boundaries of Humanoid Everyday (Zhao et al., 2025) and of our own teleoperation data. The remaining 47 hours are annotated automatically, based on robot joint trajectories matching and a vision-language model.

Training. A frame within 0.7 s before the annotated end of a subtask is labeled `switch`, with the prompt of the next subtask; every other frame is labeled `continue`. We fine-tune Qwen3-VL-4B-Instruct (Bai et al., 2025) with the vision encoder frozen, using the next-token loss, AdamW, bf16

and a global batch of 256: first for 8,000 steps (learning rate 2×10^{-5}) on the full mixture. During training, instructions are paraphrased and the list of earlier subtasks is dropped at random.

C World Model Training

The world model starts from Cosmos3-Nano (NVIDIA et al., 2026). It is first trained for 130k steps (global batch 1,024) on about 1.75k hours of human egocentric video, from EgoDex (Hoque et al., 2026) (about 700 h) and EgoVerse (Punamiya et al., 2026) (about 1,050 h). It is then fine-tuned for 5,000 steps (global batch 512, learning rate 1.4×10^{-5}) on about 60 hours of humanoid data: UnifoLM (Unitree Robotics, 2026) (about 44 h), the Ψ_0 data and our own teleoperation. The canvas is 448×256. The time delta is drawn from three bins (0.8, 1.6 and 3.2 s, sampled with weights 0.5, 0.3 and 0.2) and then redrawn uniformly within its bin, so that the model sees every Δt between 0.8 and 4.0 s. A future target must lie within 90% of the time left in the current subtask. The caption is structured and states the embodiment, the camera, the task or subtask text, and Δt in seconds.

D SONIC Token Conversion

SONIC (Luo et al., 2026b) encodes whole-body motion into a shared token space. Its encoder map either a human SMPL motion sequence or a sequence of G1 joint states to the same motion tokens, which its control policy decodes into joint-position targets. We use the G1 v1.1 encoder to convert the whole-body motions of existing humanoid datasets into SONIC tokens offline. How we apply it depends on the controller with which each dataset was recorded.

Datasets recorded with decoupled controllers. In Humanoid Everyday (Zhao et al., 2025) and the real-robot dataset of Ψ_0 (Wei et al., 2026a), the lower body is driven by a decoupled locomotion controller, so the robot walks with a stepping gait rather than a natural, human-like gait. SONIC has not seen this gait during training: when we encode the recorded states directly and replay the resulting tokens in simulation, the robot shuffles in place instead of walking. We therefore resynthesize the lower-body motion. SONIC also provides a motion generator that produces natural walking and turning from velocity and heading commands. We feed it the velocity and turning commands recorded in the original data to synthesize the lower-body motion, overwrite the upper body with the recorded joint positions, and encode the combined sequence into SONIC tokens.

Datasets recorded with whole-body controllers. UnifoLM-WBT (Unitree Robotics, 2026) is recorded with a whole-body controller whose walking gait is already natural. We therefore encode its state sequences into SONIC tokens directly, and the resulting tokens walk correctly when replayed in simulation.

Hand retargeting and validation. The datasets use different dexterous hands, which we retarget to the three-finger Dex3-1 hands of our G1. After conversion, we sample episodes from every task and replay their tokens in simulation to verify that the converted motions are executed correctly. This yields close to 100 hours of real-robot data in the SONIC token space.

E Deployment

$\Psi_{0.1}$ runs on two RTX 5090 GPUs, one for the planner and one for the predictor. The subtask predictor is queried once per second and the world model every 0.8 s, with $\Delta t = 2.4$ s. The world model samples with 8 steps, classifier-free guidance 3 and flow shift 3, with FP8 weights, `torch.compile` and CUDA graphs, and takes about 0.3 s per image (Table 2). Compared with 20-step sampling in bf16 at the same guidance and flow shift, this changes PSNR and LPIPS on the validation queries of Table 3 by less than 0.1 dB and 0.002. The planner always uses the most recent future image, and its action chunks are executed with real-time chunking (Black et al., 2025b).

F Real-Time Chunking

We execute action chunks with the inference-time RTC scheme of Black et al. (2025b), which casts the generation of a new chunk as an inpainting problem conditioned on the chunk currently being executed. Let H denote the prediction horizon, d the inference delay (the number of actions from the previous chunk $\mathbf{Y}$ that will be executed before the new chunk becomes available), and s the execution horizon i.e. the number of steps between consecutive inference starts. Following Eq. (5) of Black et al. (2025b), a per-timestep guidance weight $\mathbf{W}$ is constructed that is 1 on the frozen prefix $i < d$, decays over the overlap region $d \le i < H - s$ according to an exponentially decaying schedule, and is 0 on the free region $i \ge H - s$. At each of the N Euler steps of the flow-matching sampler, the action head predicts the velocity $\mathbf{v}_\theta(\mathbf{A}^\tau, \tau)$ from the noisy chunk $\mathbf{A}^\tau$, and the corresponding clean-sample estimate is obtained by $\hat{\mathbf{A}}^0 = \mathbf{A}^\tau - \tau \mathbf{v}_\theta$. The inpainting constraint is enforced through a pseudo-inverse guidance (Song et al., 2023). The masked residual $\mathbf{e} = \mathbf{W} \odot (\mathbf{Y} - \hat{\mathbf{A}}^0)$ is back-propagated through $\hat{\mathbf{A}}^0$ with respect to $\mathbf{A}^\tau$ to yield the guidance term $\mathbf{g} = (\partial \hat{\mathbf{A}}^0 / \partial \mathbf{A}^\tau)^\top \mathbf{e}$. Rather than using a fixed guidance weight, we empirically normalize the correction by its own magnitude, $\mathbf{v} \leftarrow \mathbf{v}_\theta - \alpha \frac{|\mathbf{e}|}{\tau |\mathbf{g}|} \mathbf{g}$ with $\alpha = 0.9$, so that the strength of the guidance is tied to the size of the residual and is well-behaved across noise levels. To achieve stable guidance, the correction is disabled at $\tau = 1$, where the sample is still pure noise and carries no information to guide.

G Predictor Ablations

Future Image Inference speed. The single latent chunk keeps each prediction cheap. Table 2 measures the latency of the image world model under the deployed inference setting (8 sampling steps with classifier-free guidance, FP8 weights, `torch.compile` with CUDA graphs, cached text features after the first step, batch size one, 448×256 canvas) while the length of the generated clip grows from the two latent frames of our design (a five-frame clip) to 25 latent frames (a 97-frame clip, the length of a typical video clip). On the RTX 5090 used at deployment, the generator is already compute-bound at two latent frames, and its latency grows roughly linearly with the clip length: the 97-frame clip takes $14\times$ longer than our five-frame clip (4.25 s vs. 0.30 s). The latter is consistent with the median latency of 0.31 s logged during our real-robot experiments.

Table 2: Latency of the image world model on the RTX 5090 used at deployment, under the deployed inference setting, for generated clips of different lengths. Generation time covers the 8 denoising steps and VAE decoding; the per-step time excludes the first step. Medians of 10 queries after 3 warm-up queries.

Latent frames	Generation time (ms)	Time per denoising step (ms)
2 (ours)	298	21.6
4	595	40.2
8	1148	72.8
13	2012	134.6
25	4254	303.8

Future-image quality. We compare the image world model with an ablation of its time conditioning and with two general-purpose generators (Table 3). All models are queried on the same 268 frames of our validation set at nine horizons from 0.8 to 4.0 s, 2,412 queries per model, and each generated image is compared with the real frame at $t + \Delta t$. Absolute scores are low for every model: the head camera moves with the robot and the future is uncertain, so pixel metrics against a single recorded future are strict, and we use them only to compare models on the same queries. Giving the time delta through both the caption and the frame rate of the clip improves over the caption alone (PSNR $+0.45$ dB, 95% CI $[0.36, 0.54]$; LPIPS -0.018, CI $[-0.024, -0.013]$; paired bootstrap over frames). Cosmos3-Nano and BAGEL (Deng et al., 2025a), used as released without training on our data, fall behind by 1.5–2 dB PSNR, and they render about the same moment for every requested Δt.

Table 3: Quality of predicted future images on our validation set (268 frames, each at nine horizons; 2,412 queries per model), measured against the real frame at $t + \Delta t$.

Model	Trained on our data	Steps	PSNR ↑	LPIPS ↓
Ours (time in caption and frame rate)	✓	20	**13.26**	**0.488**
Ours, time in caption only	✓	20	12.81	0.506
Cosmos3-Nano, two latent frames	–	20	11.76	0.598
Cosmos3-Nano, 97-frame clip	–	20	11.47	0.605
BAGEL-7B-MoT (image editing)	–	50	11.23	0.615

Subtask Prediction with Large LLMs. We compare $\mathcal{H}_\phi$ with three state-of-the-art LLM models used without any training: Gemini Robotics-ER 2, GPT-6 Astra and Claude Fable 5.1. They were queried in September 2026 as `gemini-robotics-er-2-preview` through the Gemini API, as `gpt-6-astra` through the Codex command-line tool at medium reasoning effort, and as `claude-fable-5-1` through the Claude Code command-line tool, with tools disabled. Each model answers each query once with its default settings; $\mathcal{H}_\phi$ decodes greedily. We use 200 episodes of the predictor's validation set, drawn at random in proportion to their data source: 159 from Humanoid Everyday and 41 from the Ψ_0 task data and our own in-house data, with 488 annotated subtask boundaries in total. The recorded head-camera video of each episode is replayed at one query per second. At every query, a model sees the current frame, the task instruction and the subtasks it has started so far, and answers whether to continue the current subtask or switch, together with the prompt of the next subtask. A switch adds that subtask to the model's memory, so its own mistakes carry over, but the video does not react to its answers. The three prompted models receive the same prompt. It carries the same information as the input of $\mathcal{H}_\phi$ and adds one sentence on what a subtask is (a short step of a few seconds, written as an imperative phrase) with four example prompts. The replay waits for every answer, so latency does not affect the scores. Each annotated boundary is matched to the first switch between the middle of the subtask before it and the middle of the subtask after it. A boundary without such a switch is missed, and any other switch is counted as extra. A boundary is on time if its switch comes within 1 s of it. For matched switches, Claude Sonnet 5 judges whether the new prompt means the same as the annotated one, without knowing which model gave it. Progress is the share of an episode's annotated subtasks that a model starts in order and with a correct prompt, up to its first miss or wrong prompt, averaged over episodes.

Table 4: Subtask prediction on 200 validation episodes (488 annotated subtask boundaries), replayed at one query per second. On time: share of boundaries with a switch within 1 s. Missed and extra: numbers of switches. Next subtask: share of matched switches whose new prompt is judged correct. 95% bootstrap intervals over episodes are at most ± 0.07 for on time, next subtask and progress, and never overlap between our predictor and the prompted models. Latency is the median per query; for our predictor, it is measured on the robot during our real-robot experiments (9,537 queries).

Model	On time ↑	Missed ↓	Extra ↓	Next subtask ↑	Progress ↑	Latency (s) ↓
Ours (Qwen3-VL-4B)	**0.56**	**129**	**15**	**0.94**	**0.81**	**0.56**
Gemini Robotics-ER 2	0.22	239	118	0.63	0.34	3.17
GPT-6 Astra	0.20	229	126	0.57	0.37	4.90
Claude Fable 5.1	0.18	223	81	0.52	0.25	2.58

Trained on our annotations, $\mathcal{H}_\phi$ does better on these in-domain episodes (Table 4). It switches on time at 56% of the boundaries, compared with 18–22% for the prompted models, and reaches a progress of 0.81, compared with 0.25–0.37. The prompted models are nevertheless strong. Without any training, they break each task into sensible subtasks, and 52–63% of their next-subtask prompts are judged correct. Most of the gap lies in when to switch and how finely to split. They often split one annotated subtask into two, for example "Reach toward the kettle lid" followed by "Press the kettle lid down until shut" where the annotation has "Close the kettle lid with the right hand", and they often declare the task finished before its last subtask. They are also slower, with a median latency of 2.6–4.9 s per query, compared with 0.56 s for $\mathcal{H}_\phi$. As these models keep improving, an agentic planner built on them is a promising direction for future work.