Lifting Embodied World Models for Planning and Control

Alex N. Wang[1], Trevor Darrell[2], Pavel Izmailov[1], Yutong Bai[2‡], and Amir Bar[3‡]

[1] Computer Science, New York University
[2] BAIR, UC Berkeley
[3] Imperial College London

Abstract. World models of embodied agents predict future observations conditioned on an action taken by the agent. For complex embodiments, action spaces are high-dimensional and difficult to specify: for example, precisely controlling a human agent requires specifying the motion of each joint. This makes the world model hard to control and expensive to plan with as search-based methods like CEM scale poorly with action dimensionality. To address this issue, we train a lightweight policy that maps high-level actions to sequences of low-level joint actions. Composing this policy with the frozen world model produces a *lifted world model* that predicts a sequence of future observations from a single high-level action. We instantiate this framework for a human-like embodiment, defining the high-level action space as a small set of 2D waypoints annotated on the current observation frame, each specifying a near-term goal position for a leaf joint (pelvis, head, hands). Waypoints are low-dimensional, visually interpretable, and easy to specify manually or to search over. We show that the lifted world model substantially outperforms searching directly in low-level joint space (3.8× lower mean joint error to the goal pose), while remaining more compute-efficient and generalizing to environments unseen by the policy.

1 Introduction

World models are a promising approach to understanding the physical world from data [13,22,27,45]. They learn how actions affect the environment by predicting a future observation conditioned on a current observation and an action. World models have progressed from narrow settings like simple games [13, 17, 34] to diverse environments, with actions that span text [1], navigation [4, 19], and whole-body joint control [3].

As world models target richer embodiments [3,11,26,29], their action spaces have grown high-dimensional, which presents a challenge in planning and control. For a human-like embodiment [35], actions are specified by high-dimensional, per-joint parameters [6,10,29,41,42]. For example, reaching out with one hand requires coordinated extension of the shoulder, elbow, and wrist. This high dimensionality makes world models difficult to control and especially expensive to use

‡ Equally advised.
Project page: `https://alexn.wang/lwm`

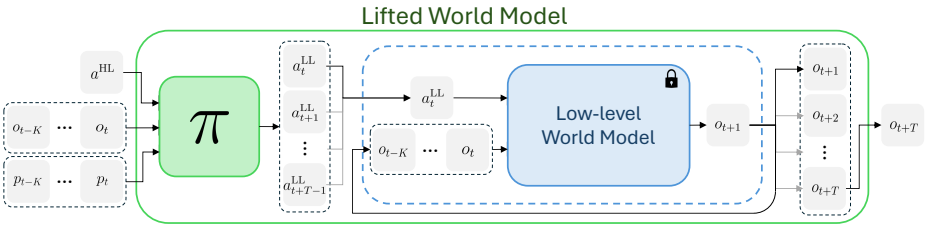

Fig. 1: Lifted World Model. The lifted world model outputs a new observation o_{t+T} given a high-level action a^{HL}. First, the policy π predicts a sequence of T low-level actions $a^{\text{LL}}_{t:t+T-1}$. Then the low-level world model autoregressively samples a sequence of new observations, one for each low-level action, appending each to the observation context for the next step.

for planning: search-based methods like the Cross-Entropy Method (CEM) [36] scale poorly in action dimensionality [5, 33].

To address this, we train a lightweight policy that maps high-level actions to sequences of low-level actions (see Figure 1). Composing this policy with the frozen world model yields a *Lifted World Model* (LWM) that predicts a sequence of observations from a single high-level action. Lifting reduces the effective dimensionality and length of the action sequence the planner must search over, without requiring any changes to the world model itself.

In our experiments, we lift PEVA [3], a world model that predicts egocentric observations conditioned on human joint-angle displacements. Inspired by human visuomotor control, where visuospatial goals in the posterior parietal cortex are translated into joint-level commands by the motor cortex [25], we design our high-level action space around visual goals expressed in the agent's current view. A waypoint is the 2D projection of a 3D goal position for a leaf joint of the embodiment (pelvis, head, or hand) that can be reached in a short sequence of joint-space actions (Figure 2). At training time, waypoints are obtained from ground-truth future poses via 3D→2D camera projection (Figure 3); at inference time, they can be specified manually or searched over with CEM. Figure 4 shows example LWM rollouts: given waypoints overlaid on the current frame, the policy generates a sequence of joint actions and the world model rolls them forward into future observations.

In Section 4.1 we show that waypoints are an effective goal-conditioning signal for generating joint actions, while goal observations are not: in egocentric video the agent's own body is mostly out of frame, so a goal image carries little information about the target pose, and conditioning on it barely improves over an unconditioned policy. Waypoints, by contrast, express goals directly in the input frame and meaningfully steer the generated actions toward the target. We further find that the policy responds sensibly to sparse and manually specified waypoints (Figure 6) and uses image context to interpret the same waypoints differently depending on the scene (Figure 7).

In Section 4.2 we use the LWM for search-based planning with CEM on hybrid navigation and interaction tasks from the Nymeria dataset [23]. We find that searching in waypoint space yields solutions 3.8× closer to the goal in mean

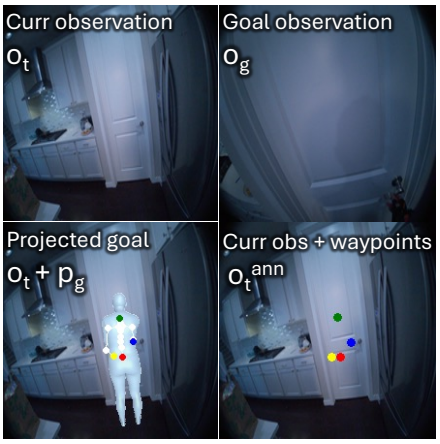

Fig. 2: Specifying goals using waypoints. A subset of joints from goal pose p_g is drawn as waypoints on o_t to create o_t^{ann}.

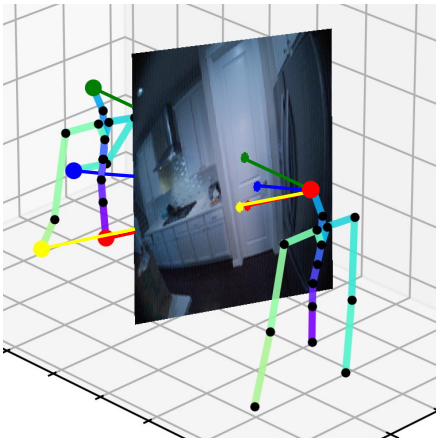

Fig. 3: To obtain waypoints, the goal pose p_g is projected onto current observation o_t seen by pose p_t.

joint error than searching directly in low-level joint space. Planning with high level actions is more compute-efficient, performs better on long horizon tasks, and generalizes to environments not seen by the policy during training.

In summary, our contributions are:

1. We propose a method for lifting a low-level world model to a higher abstraction level by training a lightweight policy that maps high-level actions to sequences of low-level actions, without modifying the base world model.
2. We instantiate this framework for a human-like embodiment with a new high-level action space of waypoints: 2D image-space goal positions for the agent's leaf joints, which are low-dimensional, visually interpretable, and well-suited to search.
3. We show that waypoints are a more effective goal-conditioning signal for egocentric policies on a human-like embodiment than goal observations.
4. Using the resulting Lifted World Model, we improve CEM planning by 3.8× in mean joint error to the goal compared to CEM in joint space, at lower compute cost, and with generalization to environments unseen by the policy.

2 Preliminaries

Here we define world models and the representations used for poses and actions.

World Models. A world model predicts the future observation o_{t+1} given an action a_t and a current observation o_t. A model may condition on K previous observations, $\boldsymbol{o}_t = \{o_{t-K}, \ldots, o_t\}$; we refer to K as the *world model context length*. The world model prediction is given by:

$$o_{t+1} = f_\phi(\boldsymbol{o}_t, a_t). \tag{1}$$

[4] We render the mesh for illustration purposes only. The overlaid skeleton shows the exact pose of the embodied agent.

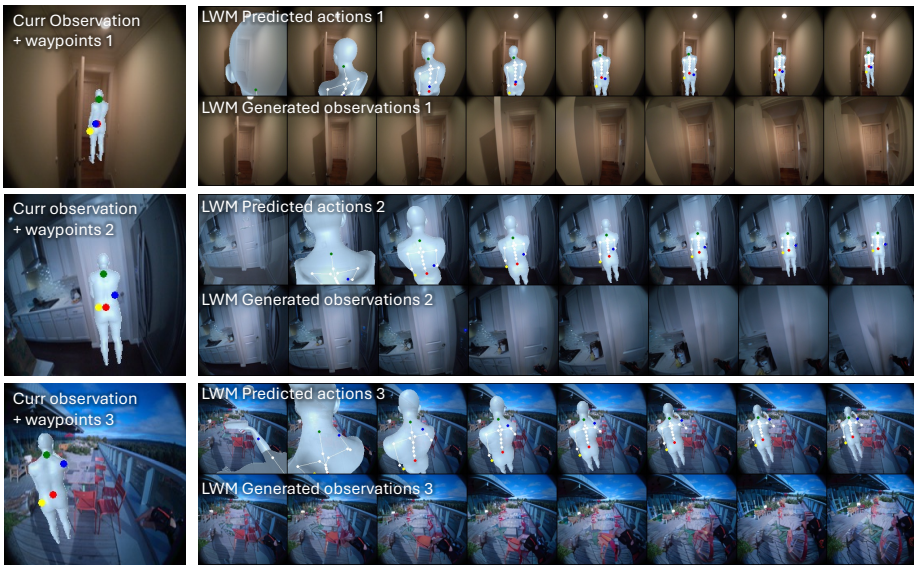

Fig. 4: Visualization of LWM rollouts and paired actions[4]. Left→right on each row: the input goal, the sequence of low-level actions predicted by the policy (upper row), and the observations generated by the world model (lower row). The goal is shown by waypoints (pelvis ●, head ●, left-hand ●, right-hand ●) plotted on current observation o_t, where each dot represents the goal-position for that joint; the mesh is shown for visualization only. **Ex 1:** advancing through the doorway. **Ex 2:** agent walking up and grasping the doorknob. **Ex 3:** avoiding obstacles and moving forward holding a camera.

Notation. We use bold symbols (e.g., $\boldsymbol{o}_t, \boldsymbol{p}_t$) to denote sequences ending at time t, and non-bold symbols (e.g., o_t, p_t) to denote individual elements at time t. We also use non-bold symbols for vector-valued quantities at a single timestep (e.g., a 3D position x or a vector of Euler angles ϕ).

Pose representation. The human-like embodiment is represented using the XSens [35] model following PEVA [3]. A pose p consists of the 3D position x_{pelvis} of the pelvis and Euler angles $\phi_{(\cdot)}$ for each joint. In our experiments we use the upper-body PEVA checkpoint with 15 joints and the pose representation:

$$p = [x_{\text{pelvis}}^\top | \phi_{\text{pelvis}}^\top | \phi_{\text{lumbar5}}^\top | \dots | \phi_{\text{left\_hand}}^\top] \in \mathbb{R}^{48}. \tag{2}$$

The pelvis position x_{pelvis} localizes the pose in a reference frame while the Euler angles $\phi_{(\cdot)}$ describe the orientation of each joint. The model assumes rigid bones of constant length.

A single pose p_t is defined with respect to its own pelvis frame; therefore at start time t the position and Euler angles of the pelvis are both zero: $x_{\text{pelvis}}, \phi_{\text{pelvis}} = \mathbf{0}$. In contrast, a sequence of poses $p_t, p_{t+1} \dots$ are defined with respect to the pelvis frame of the first pose of the sequence.

Actions. An action a_t is the change between poses p_t, p_{t+1}:

$$a_t = [\delta x_{\text{pelvis}}^\top | \delta \phi_{\text{pelvis}}^\top | \delta \phi_{\text{lumbar5}}^\top | \dots | \delta \phi_{\text{left\_hand}}^\top] \in \mathbb{R}^{48}. \tag{3}$$

Non-pelvis joints are defined in the frame of their parent joint, so only angular displacements are needed; their 3D positions are recovered via forward kinematics (Appendix J).

$$\delta x_{\text{pelvis}} = x_{\text{pelvis},t+1} - x_{\text{pelvis},t}, \tag{4}$$

$$\delta \phi_{(\cdot)} = \mathrm{R}^{-1}(\mathrm{R}(\phi_{t,(\cdot)})^{\top}\mathrm{R}(\phi_{t+1,(\cdot)})). \tag{5}$$

R and R^{-1} transform Euler angles to and from rotation matrices in SO(3).

3 Lifting an Ego World Model

Here we introduce the three components of our method. In Section 3.1, we describe the high-level waypoint action space for controlling egocentric, embodied agents. In Section 3.2, we present the waypoint-conditioned policy that predicts sequences of low-level joint actions. Finally, in Section 3.3, we combine the policy with a low-level world model to form a *Lifted World Model* (LWM) and describe its use in planning.

3.1 High-level Action Space

The high-level action space should be effective, low-dimensional and intuitive. We express high-level actions a^{HL} in the current observation o_t as a set of *waypoints*. A waypoint is the 2D projection into the current observation o_t of a near-term 3D goal position for one leaf joint of the embodied agent. We incorporate them by annotating o_t with each waypoint to obtain o_t^{ann} (see Figure 2).

Defining high-level actions as 2D points in o_t makes them visually interpretable and easy to specify manually at test time. Even more importantly, because waypoints are finite and low dimensional, they avoid the exponential scaling with action dimensionality in search-based planning such as CEM. Our method is in contrast to specifying goals using a goal image o_g used in NoMaD [38] and Diffusion Policy [7]. Their approach requires the availability of goal images while also being high dimensional and ill-suited to search.

Waypoints for a Human Embodiment. For the XSens embodiment used by PEVA, we define a^{HL} using four waypoints — for the pelvis, head, left and right hands:

$$a^{\text{HL}} = \{w_{\text{pelvis}}, w_{\text{head}}, w_{\text{left\_hand}}, w_{\text{right\_hand}}\}. \tag{6}$$

These are the leaf joints in the embodiment, affording navigation, interaction and camera control (see Figure 2). When plotting the waypoints on image o_t, we differentiate them by assigning each a unique color: pelvis ●, head ●, left hand●, right hand ●. While 2D image points do not fully specify a 3D goal, the image o_t and the spacing between waypoints provide context for inferring the goal pose.

Computing Ground-truth Waypoints. During training, high-level action labels are computed by projecting the ground-truth goal pose p_g onto the current image o_t. 3D positions are computed from the pose representation with forward

kinematics (see Appendix J), which are then projected into the image plane with the camera matrix $\mathbf{P}$ (Figure 3)

$$\{x_{\text{pelvis}}, \ldots, x_{\text{left\_hand}}\} = \text{forward\_kinematics}(p_g), \tag{7}$$

$$w_{(\cdot)} = \mathbf{P} x_{(\cdot)}.^{5} \tag{8}$$

3.2 High-level Action Conditioned Policy

We train a lightweight policy to map high-level actions into the low-level action space. At time t the policy predicts a sequence of low-level actions $a^{\text{LL}}_{t:t+T-1} \in \mathbb{R}^{T\times 48}$ conditioned on context observations $\boldsymbol{o}_t = \{o_{t-K_\pi}, \ldots, o_t\} \in [0,1]^{(K+1)\times CHW}$, poses $\boldsymbol{p}_t = \{p_{t-K_\pi}, \ldots, p_t\} \in \mathbb{R}^{(K+1)\times 48}$, and high-level action $a^{\text{HL}}_t \in [0,1]^8$:

$$a^{\text{LL}}_{t:t+T-1} \sim \pi_\theta(\boldsymbol{o}_t, \boldsymbol{p}_t, a^{\text{HL}}_t). \tag{9}$$

We allow for sparse actions where waypoints are out-of-frame *or* masked, allowing the policy to infer the final joint positions. Masking all waypoints removes goal conditioning and samples from the unconditioned distribution $p(a^{\text{LL}}_{t:t+T-1}|\boldsymbol{o}_t, \boldsymbol{p}_t)$. The policy context size K_π can be different from the world model context K. By training a policy, we capture short-term motion patterns like locomotion, reaching and grasping. See Appendix D for architecture details.

Waypoint Masking. Waypoints are masked during training so the policy can handle sparse inputs at inference. For example, only a pelvis waypoint can be used to navigate, leaving the other joint targets unspecified. Masking also teaches the policy to infer the target joint position when a waypoint is missing, such as when it is out of frame. During training, half the time no waypoints are masked, and half the time each waypoint is independently masked with probability 0.5.

Our method adds pose context, a DINOv3 encoder, later spatial pooling, 3D positional embeddings, and a larger denoising network to a base diffusion policy [7, 38]. See Appendix K for an architecture figure.

3.3 Lifted World Model

Using the policy, we can lift the low-level world model to use inputs from the high-level action space (Figure 1). The *Lifted World Model* (LWM) predicts a future observation o_{t+T} from image, pose context, and a single high-level action:

$$o_{t+T} = f^{\text{HL}}(\boldsymbol{o}_t, \boldsymbol{p_t}, a^{\text{HL}}_t). \tag{10}$$

Internally, the policy predicts a sequence of low-level actions that are used to autoregressively sample a sequence of internal observations $o_{t+1:t+T-1}$ from the low-level world model.

$$a^{\text{LL}}_{t:t+T-1} = \pi_\theta(\boldsymbol{o}_t, \boldsymbol{p}_t, a^{\text{HL}}_t) \tag{11}$$

$$o_{\tau+1} = f_\phi(\boldsymbol{o}_\tau, a^{\text{LL}}_\tau) \tag{12}$$

for $\tau \in \{t, \ldots, t+T-1\}$. Generated observations are appended to the observation context $\boldsymbol{o}_\tau$ for subsequent steps $\tau > t$.

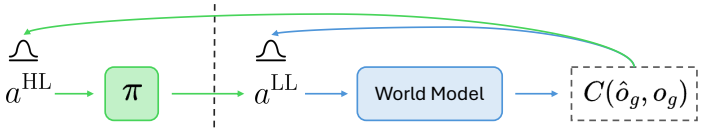

Fig. 5: Planning using the Lifted World Model. The action prior, sampling, and updates to the distribution occur in high-level action space. In contrast, planning with the low-level world model operates in the low-level action space.

Search-based Planning. Now with the LWM, we can run cross-entropy method (CEM) in the high-level action space. Given a starting state and a goal, we sample a sequence of actions, simulate future observations using the world model and select the trajectory that arrives closest to the goal. Using a low-level world model, actions are sampled in the low-level space directly. The number of samples required for effective search grows exponentially with action dimensionality times sequence length. With a lifted world model, actions can be sampled in the smaller (8 vs. 48 dimensions per step), high-level action space a^{HL}, greatly reducing both dimensionality and sequence length. The selected high-level action is then mapped to low-level joint actions via the policy for execution. See Figure 5 for a visualization.

Formally, with a goal observation o_g, perceptual similarity cost function $C(\cdot)$, and a distribution over low-level actions $p_{\text{LL}}(\cdot)$, the CEM planning objective can be formulated as:

$$\boldsymbol{a}^{\text{LL}*} = \underset{\boldsymbol{a}^{\text{LL}}=a_t,\ldots,a_{t+T-1}}{\arg\min} \mathbb{E}_{\hat{o}_g \sim f_\phi(\boldsymbol{a}^{\text{LL}})}[C(\hat{o}_g, o_g)], \tag{13}$$

while planning with a lifted world model uses the high-level prior $p_{\text{HL}}(\cdot)$ and is written as

$$\boldsymbol{a}^{\text{HL}*} = \underset{\boldsymbol{a}^{\text{HL}}=a_t^{\text{HL}},\ldots,a_{t+L*T}^{\text{HL}}}{\arg\min} \mathbb{E}_{\hat{o}_g \sim f_\phi(\boldsymbol{a}^{\text{LL}}), \boldsymbol{a}^{\text{LL}} \sim \pi_\theta(\boldsymbol{a}^{\text{HL}})}[C(\hat{o}_g, o_g)]. \tag{14}$$

Image inputs and autoregressive sampling for the world model and policy are omitted for clarity. L indexes the high-level planning horizon. Optimization proceeds iteratively: at each iteration, N action samples are drawn, simulated, and assigned a cost based on the perceptual distance to the goal at the final predicted frame. The M samples with the lowest cost are used to update the prior distribution for the next iteration of the algorithm.

4 Experiments

We present our quantitative and qualitative results for the policy conditioned on high-level actions and the Lifted World Model. Our experiments show the effectiveness of conditioning on the high-level action space compared to goal observations. We also demonstrate the improved performance and efficiency when planning in high-level action space using the LWM in comparison to PEVA.

[5] Homogeneous coordinates and the camera frame transform are omitted for clarity.

Base World Model and Nymeria Dataset. We use PEVA [3] trained on the Nymeria [23] dataset as a low-level world model. Nymeria consists of first-person videos recorded using Project Aria [9] glasses and XSens [35] motion capture suits in continuous 15-minute sessions in 50 indoor and outdoor environments. The XSens suits provide 3D body motion and joint angle data that are used as the low-level, high-dimensional action space. Compared to other datasets, Nymeria presents a combination of a flexible human-like embodiment and challenging hybrid navigation + interaction tasks.

Metrics. We evaluate the policy and planning tasks by how closely the actions bring the initial pose p_t to the goal pose p_g. The distance between the final predicted pose $\hat{p}_g$ and the ground-truth p_g is measured by the mean joint error (MJE) [20] in meters of the *leaf* joints (pelvis, head, and hands), the *intermediate* joints between them, and *all* joints together. This provides a ground-truth physical metric for both policy and world model planning performance.

To compute MJE we integrate the predicted actions starting from the initial pose p_t to obtain the predicted final pose $\hat{p}_g$:

$$x_{t+1,\text{pelvis}} = x_{t,\text{pelvis}} + \mathrm{R}(\phi_{t,\text{pelvis}})\delta x_{t,\text{pelvis}} \tag{15}$$

$$\phi_{t+1,(\cdot)} = \mathrm{R}^{-1}(\mathrm{R}(\delta\phi_{t,(\cdot)})\mathrm{R}(\phi_{t,(\cdot)})). \tag{16}$$

A forward kinematics procedure (Appendix J) computes 3D positions for every joint $\{x_{\text{pelvis}}, \ldots, x_{\text{left\_hand}}\} = \text{forward\_kinematics}(p)$ for both p_g and $\hat{p}_g$. The per-joint distances are then computed and averaged over joints for MJE:

$$\text{MJE} = \frac{1}{|\text{joints}|} \sum_{j \in \text{joints}} |\hat{x}_j - x_j|. \tag{17}$$

4.1 Goal-conditioned policy

Here we assess the actions generated using our policy trained on Nymeria. We ablate each addition made to a diffusion policy in the quantitative results and report *leaf*, *intermediate*, and *all*-joint MJE on the evaluation set. We also include additional experiments on waypoint visibility and on using depth to specify an exact 3D goal. Qualitative results show that the policy generalizes well to counterfactual waypoints that differ from the waypoints in the data. In addition, the policy can use image context to infer sensible actions from the same waypoints. Finally, we show that goal pose p_g cannot be identified from observation o_g, limiting o_g's usefulness as a goal for egocentric, embodied agents. See Appendix E for implementation details.

Quantitative Results. We report policy performance in Table 1. "Unconditioned" uses no goal information; "goal-conditioned" is when either o_g or a^{HL} is used to condition the policy. "Initial distance" is the MJE between starting pose p_t and goal pose p_g while "random weights" refers to actions sampled from an untrained policy. While the NoMaD baseline outperforms an untrained network,

Table 1: Policy Ablations. We evaluate each change applied to the Base Policy. Results are reported on the validation set. Mean joint error (MJE) is reported in meters.

	Unconditioned MJE			Goal-Conditioned MJE		
Model	**Leaf ↓**	**Int. ↓**	**All ↓**	**Leaf ↓**	**Int. ↓**	**All ↓**
Initial Distance	0.445	0.419	0.426	0.445	0.419	0.426
Random Weights	0.749	0.707	0.718	0.735	0.692	0.703
Base Policy	0.427	0.397	0.405	0.414	0.384	0.392
+ architecture changes	0.406	0.375	0.384	0.388	0.359	0.367
+ pose context	0.359	0.329	0.337	0.343	0.316	0.323
+ waypoint conditioning	**0.353**	**0.323**	**0.331**	0.262	0.236	0.243
+ waypoint masking	0.437	0.408	0.415	**0.243**	**0.219**	**0.226**
+ 3d conditioning	0.360	0.330	0.338	0.243	0.219	0.226
+ waypoint masking	0.415	0.391	0.398	0.223	0.202	0.208

it reduces MJE by only 2.1cm from the initial distance and adding o_g as goal-conditioning reduces it by only another 1.3cm. Adding architectural changes and pose context $\boldsymbol{p}_t$ improve overall performance, but not goal conditioning. The policy learns coherent action sequences, but the goal observation o_g does not meaningfully steer them toward the goal.

Adding waypoint conditioning improves goal-conditioned MJE by 8.8cm over unconditioned generation. This improvement indicates that conditioning on a^{HL} shifts the low-level action distribution towards actions that better reach goal pose p_g. Furthermore, waypoint masking improves goal-conditioned performance at the cost of unconditioned MJE. This tradeoff is acceptable since we prioritize goal-conditioned action generation to lift the world model in later experiments. Overall, we find that waypoints are an effective and interpretable high-level action space for goal-conditioned generation.

Qualitative Action Generations. We visualize generated actions in the "Predicted actions" rows of Figure 4. Each action in the sequence is shown by projecting the resulting 3D pose onto o_t. The policy controls the agent to both navigate the environment and interact using its hands. In example 2, it reaches to grasp the doorknob, and in example 3, it raises its hand while holding a camera.

In Figure 6, we evaluate the policy on counterfactual waypoints that do not match the ground truth. In counterfactual 1, the policy is given a sparse high-level action of only a head ● waypoint. Using this waypoint, the policy moves the agent to the right while facing left, inferring reasonable positions for the unconditioned joints. In Figure 7 we evaluate the influence of scene context on the generated actions. Contexts 1 and 2 use the same waypoints yet result in different action sequences. We see that in context 1—in front of a stove—the left- ● and right-hand ● waypoints prompt the agent to grasp the pot, while in context 2 the agent walks forward in the open room.

Waypoint visibility. Since waypoints are defined in observation o_t, we investigate whether performance decreases when the joint is not visible. Table 2

Fig. 6: Generating actions from waypoints not seen in data. **Ground truth:** the agent walks down the path. **Ex 1:** head ● waypoint on the right → agent moves right, faces left. **Ex 2:** four waypoints above the bench → agent climbs onto the bench.

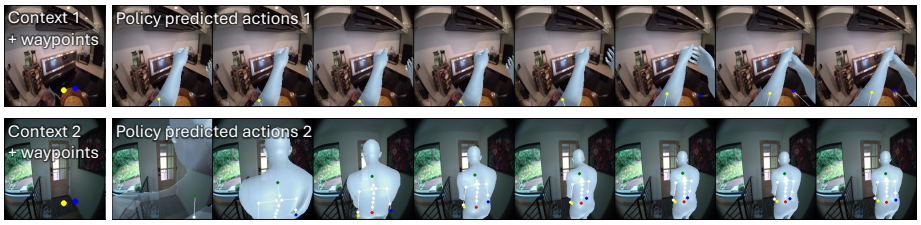

Fig. 7: Contextually aware action generation. The policy predicts different actions from the same waypoints depending on the scene. **Context 1:** agent grasps the pot. **Context 2:** agent walks right and turns to the left.

separates goal-conditioned MJE by the visibility of the joint in o_t. We see a substantial 26.5cm increase in MJE for observation-conditioned policies when the joint is not visible in the current frame, compared to a modest 12.7cm increase for waypoint conditioning. This suggests that a waypoint-conditioned policy learns better whole-body motion patterns that generalize even when joints are not visible at test time. Furthermore, waypoint masking improves visible performance while reducing the not-visible performance drop to only 8.8cm. This shows that learning to infer the position of the masked joint generalizes to non-visible joints.

3D Waypoints. We assess the efficacy of 2D waypoints by comparing with a policy trained using waypoints augmented with depth values. This uses privileged depth information to specify an exact 3D goal, increasing waypoint dimensionality by 50% (8 to 12 dimensions). Results are shown in the last two rows of Table 1, where 3D conditioning improves goal-conditioned *all* MJE by at most 2cm. We further assess 3D waypoints for CEM planning in Section 4.2.

Decomposing Action Generation. We further explore goal-conditioning for egocentric action generation by decomposing it into two separate tasks: motion generation and goal pose prediction. For *motion generation*, we train a policy to predict joint action sequences with the goal pose p_g as input. This is a privileged setting where the policy is given the goal pose and must learn to predict the actions to reach it. For *goal pose prediction*, we replace the denoising UNet with

Table 2: Goal-conditioned action generation with metrics separated based on the visibility of each joint in the current observation o_t. MJE in meters.

	Visible MJE			Not Visible MJE		
Model	**Leaf ↓**	**Int. ↓**	**All ↓**	**Leaf ↓**	**Int. ↓**	**All ↓**
Base Policy	0.360	0.297	0.314	0.605	0.705	0.678
+ architecture changes	0.340	0.278	0.295	0.566	0.659	0.634
+ pose context	0.305	0.254	0.268	0.483	0.551	0.533
+ waypoint conditioning	**0.260**	**0.207**	**0.222**	0.327	0.358	0.349
+ waypoint masking	**0.251**	**0.200**	**0.213**	**0.284**	**0.307**	**0.301**

Table 3: Decomposing goal-conditioned action generation into motion generation given a goal pose and predicting the goal pose given the goal observation. MJE in meters.

Task	**Leaf MJE ↓**	**Int. MJE ↓**	**All MJE ↓**
Motion generation	0.115	0.101	0.105
Goal pose prediction	0.299	0.272	0.279

an MLP to directly predict the goal pose p_g conditioned on the goal observation o_g. See Appendix K for policy architectures.

Both models are evaluated using MJE between the predicted final pose $\hat{p}_g$ and the ground-truth goal pose p_g. Table 3 shows that predicting an action sequence given the goal pose is easier than predicting the goal pose p_g from o_g. This indicates that egocentric observations are insufficient to identify the goal pose p_g since the embodiment is seldom visible in observation o_g — the hands may be visible, but the rest of the body is rarely seen.

4.2 Lifted World Model Planning

Using the Lifted World Model, we perform planning for hybrid navigation + interaction tasks sampled from the Nymeria dataset. Each task consists of observations $\boldsymbol{o}_t$, poses $\boldsymbol{p}_t$, and a goal observation o_g. Planning outputs a sequence of actions, and performance is measured by the MJE between the true goal pose p_g and the final pose reached by the planning output. Note that waypoints are not provided, and the task is to search for waypoints that reach the goal pose.

Task Generation. We randomly sample start and goal observation pairs from the Nymeria evaluation set to generate tasks. Tasks are constrained to have ≥ 1 joint from the goal pose p_g visible in observation o_t to avoid goals in unseen areas and reduce world model hallucinations. We also filter tasks for *leaf* MJE ≥ 0.1m between p_t and p_g to avoid trivial stationary tasks.

Quantitative Results. We report planning results in Table 4. As before, initial distance represents the MJE between p_t and p_g. We also include a 1-NN copy baseline, selecting the action based on nearest $[p_t; p_g]$. PEVA CEM searches directly in low-level joint-action space, while Lifted CEM searches for waypoints that generate actions to reach the goal. PEVA CEM reduces all MJE by only

Table 4: Search-based planning results using Cross-Entropy Method on 128 tasks. 6 CEM iterations, 64 samples/iteration. Mean joint error shown in meters (m).

Method	Leaf MJE ↓	Int. MJE ↓	All MJE ↓
Initial Distance	0.724	0.697	0.704
Copy Baseline (1-NN)	0.696	0.663	0.672
Uncond. Policy	0.677	0.641	0.650
Diffusion Policy	0.605	0.578	0.585
HDP	0.540	0.502	0.512
PEVA CEM	0.637	0.608	0.616
Lifted CEM (3D)	0.453	0.407	0.420
Lifted CEM (2D, ours)	**0.411**	**0.359**	**0.374**

8.8cm, while Lifted CEM reduces it by 33cm. We sample 64 joint action sequences from each set of waypoints and report the average MJE. To compare with a policy-only approach, we also evaluate the unconditioned policy, an image-conditioned Diffusion Policy [7] and a Hierarchical Diffusion Policy (HDP) [24]. Lifted CEM outperforms all policy-based methods, even ones trained on this goal-observation conditioned planning task. See Appendix A for dexterous manipulation experiments and Appendix F for waypoint joint ablations.

Qualitative Planning Visualizations. We visualize action and observation sequences using waypoints from CEM planning in Figure 8. Planning with Lifted CEM effectively reproduces the ground truth actions. In task 1, the agent moves to the right following the pelvis ● waypoint and faces left, while in task 2, it moves its hands over the counter to set down the plastic bag. We find that there are key waypoints that drive the action: (right hand ●, pelvis ●) in task 1 and (left hand ●, right hand ●) in task 2. Similar to how the policy can handle missing waypoints (Section 4.1), it can ignore outlier waypoints found by search: head ● in task 1 and pelvis ● in task 2. The policy in Lifted CEM is trained to produce realistic movements, while PEVA CEM may produce unnatural joint angles during planning. See Appendix M for PEVA planning visualizations.

Planning Efficiency. Planning using CEM with a large world model can be computationally expensive. Thus, we compare the performance of searching over high-level (Lifted CEM) vs. low-level actions (PEVA CEM) for different levels of compute. We measure the MJE of both methods for varying CEM iterations and samples per iteration. For compute measurements see Appendix G.

Results are shown in Figure 9. We show the cumulative min over CEM iterations, see Appendix I for without. Planning in high-level space outperforms planning in low-level joint action space for all numbers of samples and CEM iterations. Furthermore, we find that PEVA CEM worsens MJE after one step, which highlights the difficulty of naive search. In summary, lifting the abstraction of the world model yields better solutions with less compute.

Lifting Using 3D Waypoints. We evaluate a world model lifted to use 3D waypoints in search-based planning. The results are plotted alongside Lifted

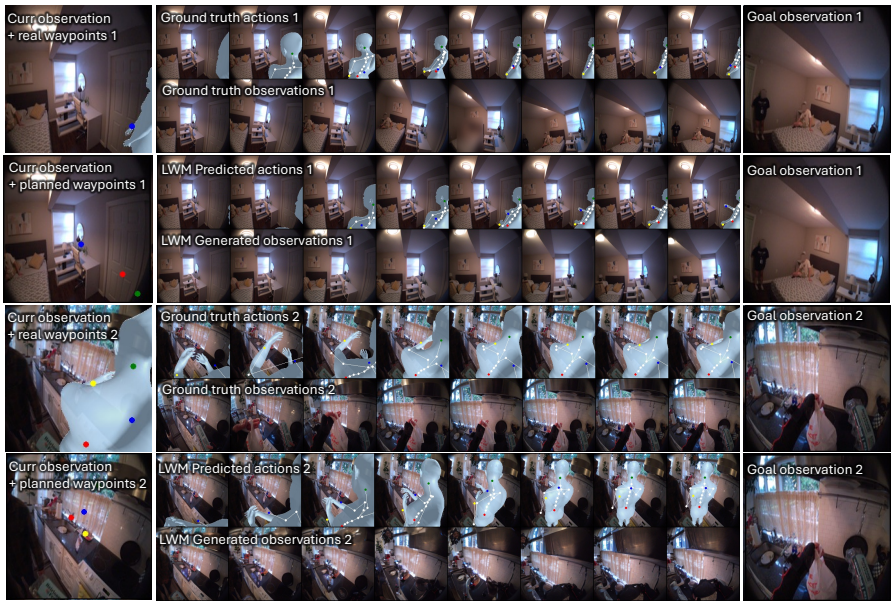

Fig. 8: Visualizations of planning solutions. Each row shows from left→right: the current observation o_t with waypoints, visualized actions (upper row), the resulting observations (lower row), and the goal observation o_g. **Task 1:** navigating around the room. **Task 2:** raising hands to set down the plastic bag on the counter.

CEM in Figure 9. 3D waypoints perform similarly after one CEM step, but lag behind with further iterations due to the larger action dimensionality (12 vs. 8) and the difficulty of finding a correct depth value.

Varying Goal Horizons. We also report planning results for goals sampled from different timesteps into the future and different initial MJE distances. Figure 10 shows performance when goals are sampled from varying time horizons. In this setting, PEVA CEM plans for actions of sequence length equal to the goal horizon whereas Lifted CEM is fixed at 8 due to fixed policy length. Despite this, we find that Lifted CEM outperforms PEVA CEM over all goal horizons.

Figure 11 plots performance against initial MJE, with tasks grouped into 20 quantile buckets. The $y = x$ line is a baseline where no action is taken, where the initial and final MJE are equal. Lifted CEM outperforms PEVA CEM at most initial distances while both methods perform worse when initial MJE $\leq$ 0.3m.

Unseen Environments. To test the generalization of our method, we train a policy on a new training set excluding videos from Nymeria locations 6, 19, and 34Then, we use this policy to lift the same PEVA model and evaluate the lifted model on tasks from the held-out locations.

Results are presented in Table 5. Lifted CEM using the held-out policy outperforms PEVA CEM and is only slightly worse than using the base policy trained on all environments. Therefore, the short-term lightweight policy design generalizes well to unseen environments.

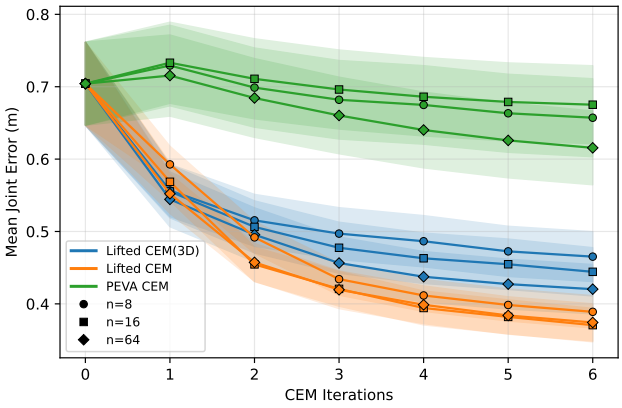

Fig. 9: Planning performance for different CEM budgets. Searching in our lifted action space scales better than in joint space. 3D waypoints are more difficult to search.

5 Related Work

World Models and Planning. Many world model works explore planning: PlaNet [15], TD-MPC2 [18], DINO-WM [48] and VJEPA-2 [2,27] plan in latent space, while UniSim [45] and NWM [4] are used downstream to plan in pixel space. Among these works, cross-entropy method [36] is the *de facto* planning approach, while MPPI [43] also sees use. DINO-WM [48] and Parthasarathy et al. [32] explore using gradient-based methods. DynaGuide [8] guides a diffusion policy via gradients through a latent world model. The Dreamer [14,16,17] series learns a world model for policy training. PEVA [3] trains a whole-body conditioned world model, and DexWM [12] conditions on hands. To our knowledge, our work is the first to use a world model to plan for complex, egocentric, human-like embodiments, and the first to lift a low-level world model to a higher level of abstraction to make this challenging task tractable.

Embodied, Egocentric, and Hierarchical Policies. NoMaD [38] tackles navigation from an egocentric point of view with a point-robot embodiment while Diffusion Policy [7] uses a similar architecture for manipulation with articulated robots. Nachum et al. [28] and later HIQL [31] are works that learn hierarchical policies. iDP3 [46] controls a humanoid at a joint level from an egocentric 3D camera. Large-scale models include GR00T-N1 [6], which controls an articulated humanoid and VLA approaches like OpenVLA [21] and Octo [30] for robotic manipulators tend to focus on exocentric manipulation tasks. No prior works address navigation and manipulation for embodiments from an egocentric view, which is a key affordance of human-like agents, nor do they consider the use of a low-dimensional, finite, and searchable input space for goal conditioning.

Motion Generation. Motion generation aims to learn a realistic distribution of human motion controlled by text, action labels, or keypoint constraints. MDM [42] and MotionDiffuse [47] use diffusion for text-conditioned motion generation and PriorMDM [37] extends this with dense end-effector and keypoint

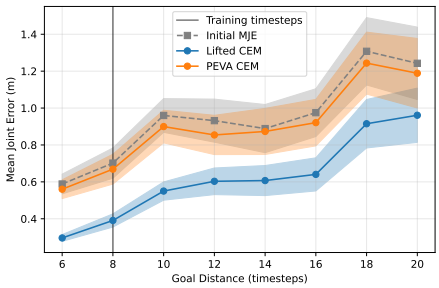

Fig. 10: Planning results for varying goal horizons.

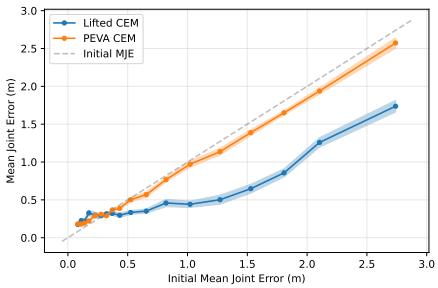

Fig. 11: Planning tasks by initial MJE grouped into 20 quantile buckets

Table 5: Lifted CEM planning on tasks in unseen environments. MJE in meters.

Method	Leaf MJE ↓	Int. MJE ↓	All MJE ↓
Initial Distance	0.618	0.589	0.597
PEVA CEM	0.572	0.546	0.553
Lifted CEM (ours)	0.373	0.318	0.333
Lifted CEM (held-out)	0.404	0.346	0.362

conditioning. OmniControl [44] supports sparse 3D keypoint conditioning over arbitrary joints and timesteps. CLoSD [41] and MaskedMimic [40] both utilize a physics simulator: the former closes the generation loop with simulation, while the latter performs motion inpainting. GOAL [39] focuses on hand-object grasping for a target object in an open-loop setting. All of these works control a human embodiment, but do not address reaching a precise goal state in an environment from egocentric observations. Our waypoint approach shares some similarity with keypoint conditioning, but specifies sparse 2D joint positions for the final goal pose, relying on the policy to capture short-term motion patterns instead of a dense sequence of 3D joint positions.

6 Conclusion

We present a method for lifting a low-level world model to a higher level of abstraction by training a goal-conditioned policy that maps high-level actions to sequences of low-level actions, and instantiate it for a human-like embodiment using waypoints as the high-level action space. We show that embodied, egocentric agents benefit from specifying goals as waypoints grounded in the present observation, and that search-based planning with the resulting Lifted World Model is 3.8× more effective than planning in low-level joint space, at lower compute cost, and generalizes to environments unseen by the policy. Notably, our approach is lightweight, scalable, and requires no additional data collection or annotation beyond what is already used to train the world model. Moving forward, we believe lifting is a promising direction for keeping world models controllable and plannable as they scale to richer embodiments and more complex tasks; exploring alternative high-level action spaces and embodiments is a natural next step.

References

1. Agarwal, N., Ali, A., Bala, M., Balaji, Y., Barker, E., Cai, T., Chattopadhyay, P., Chen, Y., Cui, Y., Ding, Y., et al.: Cosmos world foundation model platform for physical ai. arXiv preprint arXiv:2501.03575 (2025)
2. Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., et al.: V-jepa 2: Self-supervised video models enable understanding, prediction and planning. arXiv preprint arXiv:2506.09985 (2025)
3. Bai, Y., Tran, D., Bar, A., LeCun, Y., Darrell, T., Malik, J.: Whole-body conditioned egocentric video prediction. In: The Thirty-ninth Annual Conference on Neural Information Processing Systems (2025), https://openreview.net/forum?id=XDTTwmjhAg
4. Bar, A., Zhou, G., Tran, D., Darrell, T., LeCun, Y.: Navigation world models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 15791–15801 (2025)
5. Bharadhwaj, H., Xie, K., Shkurti, F.: Model-predictive control via cross-entropy and gradient-based optimization. In: Learning for Dynamics and Control. pp. 277–286. PMLR (2020)
6. Bjorck, J., Castañeda, F., Cherniadev, N., Da, X., Ding, R., Fan, L., Fang, Y., Fox, D., Hu, F., Huang, S., et al.: Gr00t n1: An open foundation model for generalist humanoid robots. arXiv preprint arXiv:2503.14734 (2025)
7. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research **44**(10-11), 1684–1704 (2025)
8. Du, M., Song, S.: Dynaguide: Steering diffusion policies with active dynamic guidance. In: Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS) (2025)
9. Engel, J., Somasundaram, K., Goesele, M., Sun, A., Gamino, A., Turner, A., Talattof, A., Yuan, A., Souti, B., Meredith, B., et al.: Project aria: A new tool for egocentric multi-modal ai research. arXiv preprint arXiv:2308.13561 (2023)
10. Fu, Z., Zhao, Q., Wu, Q., Wetzstein, G., Finn, C.: Humanplus: Humanoid shadowing and imitation from humans. In: Conference on Robot Learning. pp. 2828–2844. PMLR (2025)
11. Gao, S., Liang, W., Zheng, K., Malik, A., Ye, S., Yu, S., Tseng, W.C., Dong, Y., Mo, K., Lin, C.H., et al.: Dreamdojo: A generalist robot world model from large-scale human videos. arXiv preprint arXiv:2602.06949 (2026)
12. Goswami, R.G., Bar, A., Fan, D., Yang, T.Y., Zhou, G., Krishnamurthy, P., Rabbat, M., Khorrami, F., LeCun, Y.: World models can leverage human videos for dexterous manipulation. arXiv preprint arXiv:2512.13644 (2025)
13. Ha, D., Schmidhuber, J.: Recurrent world models facilitate policy evolution. In: Advances in Neural Information Processing Systems 31, pp. 2451–2463. Curran Associates, Inc. (2018), https://papers.nips.cc/paper/7512-recurrent-world-models-facilitate-policy-evolution, https://worldmodels.github.io
14. Hafner, D., Lillicrap, T., Ba, J., Norouzi, M.: Dream to control: Learning behaviors by latent imagination. In: International Conference on Learning Representations
15. Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H., Davidson, J.: Learning latent dynamics for planning from pixels. In: International conference on machine learning. pp. 2555–2565. PMLR (2019)

16. Hafner, D., Lillicrap, T.P., Norouzi, M., Ba, J.: Mastering atari with discrete world models. In: International Conference on Learning Representations
17. Hafner, D., Pasukonis, J., Ba, J., Lillicrap, T.: Mastering diverse control tasks through world models. Nature **640**(8059), 647–653 (2025). https://doi.org/10.1038/s41586-025-08744-2, https://doi.org/10.1038/s41586-025-08744-2
18. Hansen, N., Su, H., Wang, X.: Td-mpc2: Scalable, robust world models for continuous control. In: The Twelfth International Conference on Learning Representations
19. Hu, A., Russell, L., Yeo, H., Murez, Z., Fedoseev, G., Kendall, A., Shotton, J., Corrado, G.: Gaia-1: A generative world model for autonomous driving. arXiv preprint arXiv:2309.17080 (2023)
20. Ionescu, C., Papava, D., Olaru, V., Sminchisescu, C.: Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(7), 1325–1339 (2014). https://doi.org/10.1109/TPAMI.2013.248
21. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E.P., Sanketi, P.R., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: Openvla: An open-source vision-language-action model. In: Agrawal, P., Kroemer, O., Burgard, W. (eds.) Proceedings of The 8th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 270, pp. 2679–2713. PMLR (06–09 Nov 2025), https://proceedings.mlr.press/v270/kim25c.html
22. LeCun, Y., et al.: A path towards autonomous machine intelligence version 0.9. 2, 2022-06-27. Open Review **62**(1), 1–62 (2022)
23. Ma, L., Ye, Y., Hong, F., Guzov, V., Jiang, Y., Postyeni, R., Pesqueira, L., Gamino, A., Baiyya, V., Kim, H.J., et al.: Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In: European Conference on Computer Vision. pp. 445–465. Springer (2024)
24. Ma, X., Patidar, S., Haughton, I., James, S.: Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18081–18090 (2024)
25. Merel, J., Botvinick, M., Wayne, G.: Hierarchical motor control in mammals and machines. Nature Communications **10**(1), 5489 (2019). https://doi.org/10.1038/s41467-019-13239-6, https://doi.org/10.1038/s41467-019-13239-6
26. Mereu, R., Scannell, A., Hou, Y., Zhao, Y., Jitta, A., Dominguez, A., Acerbi, L., Storkey, A., Chang, P.: Generative world modelling for humanoids: 1x world model challenge technical report. arXiv preprint arXiv:2510.07092 (2025)
27. Mur-Labadia, L., Muckley, M., Bar, A., Assran, M., Sinha, K., Rabbat, M., LeCun, Y., Ballas, N., Bardes, A.: V-jepa 2.1: Unlocking dense features in video self-supervised learning. arXiv preprint arXiv:2603.14482 (2026)
28. Nachum, O., Gu, S.S., Lee, H., Levine, S.: Data-efficient hierarchical reinforcement learning. Advances in neural information processing systems **31** (2018)
29. Nicklas Hansen, Jyothir S V, V.S.Y.L.X.W.H.S.: Hierarchical world models as visual whole-body humanoid controllers (2025)
30. Octo Model Team, Ghosh, D., Walke, H., Pertsch, K., Black, K., Mees, O., Dasari, S., Hejna, J., Xu, C., Luo, J., Kreiman, T., Tan, Y., Chen, L.Y., Sanketi, P., Vuong, Q., Xiao, T., Sadigh, D., Finn, C., Levine, S.: Octo: An open-source generalist robot policy. In: Proceedings of Robotics: Science and Systems. Delft, Netherlands (2024)
31. Park, S., Ghosh, D., Eysenbach, B., Levine, S.: Hiql: offline goal-conditioned rl with latent states as actions. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. pp. 34866–34891 (2023)

32. Parthasarathy, A., Kalra, N., Agrawal, R., LeCun, Y., Bounou, O., Izmailov, P., Goldblum, M.: Closing the train-test gap in world models for gradient-based planning. arXiv preprint arXiv:2512.09929 (2025)
33. Psenka, M., Rabbat, M., Krishnapriyan, A., LeCun, Y., Bar, A.: Parallel stochastic gradient-based planning for world models. arXiv preprint arXiv:2602.00475 (2026)
34. Robine, J., Höftmann, M., Uelwer, T., Harmeling, S.: Transformer-based world models are happy with 100k interactions. arXiv preprint arXiv:2303.07109 (2023)
35. Roetenberg, D., Luinge, H., Slycke, P., et al.: Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. Xsens Motion Technologies BV, Tech. Rep **1**(2009), 1–7 (2009)
36. Rubinstein, R.Y.: Optimization of computer simulation models with rare events. European Journal of Operational Research **99**(1), 89–112 (1997). https://doi.org/https://doi.org/10.1016/S0377-2217(96)00385-2, https://www.sciencedirect.com/science/article/pii/S0377221796003852
37. Shafir, Y., Tevet, G., Kapon, R., Bermano, A.H.: Human motion diffusion as a generative prior. In: The Twelfth International Conference on Learning Representations
38. Sridhar, A., Shah, D., Glossop, C., Levine, S.: Nomad: Goal masked diffusion policies for navigation and exploration. In: 2024 IEEE International Conference on Robotics and Automation (ICRA). pp. 63–70. IEEE (2024)
39. Taheri, O., Choutas, V., Black, M.J., Tzionas, D.: Goal: Generating 4d whole-body motion for hand-object grasping. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13263–13273 (2022)
40. Tessler, C., Guo, Y., Nabati, O., Chechik, G., Peng, X.B.: Maskedmimic: Unified physics-based character control through masked motion inpainting. ACM Transactions On Graphics (TOG) **43**(6), 1–21 (2024)
41. Tevet, G., Raab, S., Cohan, S., Reda, D., Luo, Z., Peng, X.B., Bermano, A.H., van de Panne, M.: CLoSD: Closing the loop between simulation and diffusion for multi-task character control. In: The Thirteenth International Conference on Learning Representations (2025), https://openreview.net/forum?id=pZISppZSTv
42. Tevet, G., Raab, S., Gordon, B., Shafir, Y., Cohen-Or, D., Bermano, A.H.: Human motion diffusion model. In: The Eleventh International Conference on Learning Representations
43. Williams, G., Aldrich, A., Theodorou, E.A.: Model predictive path integral control: From theory to parallel computation. Journal of Guidance, Control, and Dynamics **40**(2), 344–357 (2017)
44. Xie, Y., Jampani, V., Zhong, L., Sun, D., Jiang, H.: Omnicontrol: Control any joint at any time for human motion generation. In: The Twelfth International Conference on Learning Representations
45. Yang, S., Du, Y., Ghasemipour, S.K.S., Tompson, J., Kaelbling, L.P., Schuurmans, D., Abbeel, P.: Learning interactive real-world simulators. In: The Twelfth International Conference on Learning Representations (2024), https://openreview.net/forum?id=sFyTZEqmUY
46. Ze, Y., Chen, Z., Wang, W., Chen, T., He, X., Yuan, Y., Peng, X.B., Wu, J.: Generalizable humanoid manipulation with 3d diffusion policies. In: 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2873–2880. IEEE (2025)
47. Zhang, M., Cai, Z., Pan, L., Hong, F., Guo, X., Yang, L., Liu, Z.: Motiondiffuse: Text-driven human motion generation with diffusion model. IEEE transactions on pattern analysis and machine intelligence **46**(6), 4115–4128 (2024)

48. Zhou, G., Pan, H., LeCun, Y., Pinto, L.: DINO-WM: World models on pre-trained visual features enable zero-shot planning. In: Forty-second International Conference on Machine Learning (2025), https://openreview.net/forum?id=D5RNACOZEI