[TINY PAPER] LIFTING EGO WORLD MODELS FOR PLANNING AND CONTROL

Alex N. Wang[1] **Trevor Darrell**[2†] **Pavel Izmailov**[1†] **Yutong Bai**[2‡] **Amir Bar**[2‡]

ABSTRACT

As world models develop and simulate more complex embodiments, action dimensionality tends to increase. This increase in dimensionality makes these models more difficult to control, less interpretable, and exponentially costly when used for search-based planning. In our work, we demonstrate *lifting* a low-level egocentric, embodied world model to a higher level of abstraction using a new, high-level action space. To implement this lifting, we train a lightweight policy to predict low-level world model inputs conditioned on the high-level action space. Using this policy, we lift the world model and perform search-based planning in the high-level action space. In our results, we show that high-level waypoint actions are more effective than goal observations for conditioning a egocentric embodied agent, and that planning with a lifted world model (LWM) in high-level action space reduces joint error by $4.7\times$ more than planning in low-level joint space.

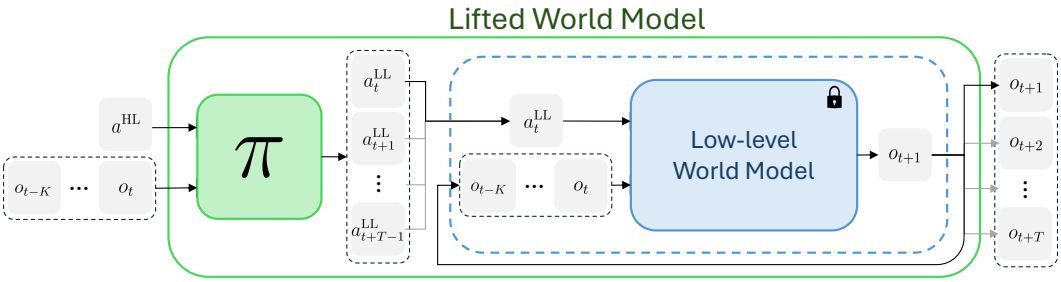

Figure 1: **Lifted World Model.** The lifted world model predicts a sequence of observations $o_{t+1:t+T}$ given a high-level action a^{HL}. The policy π predicts a sequence of T low-level actions $a^{\text{LL}}_{t:t+T-1}$ while the low-level world model autoregressively samples a sequence of new observations.

1 INTRODUCTION

As environments and world models become more complex, so do the action spaces. For human-like embodiments (Roetenberg et al., 2009) actions can be high-dimensional per-joint parameters. This can make a world model difficult to interpret, control and use in planning. In particular, search-based planning like Cross-Entropy Method (CEM) (Rubinstein, 1997) scales poorly and is exponentially expensive with action dimensionality (Bharadhwaj et al., 2020; Psenka et al., 2026).

To overcome this, we lift a world model to a higher-level of abstraction using a high-level action space. First, we define a new *waypoint* action space suitable for specifying goals for human-like embodied agents by projecting future joint positions onto the current egocentric image observation. Then we train a lightweight policy to translate these waypoints into low-level joint movements and find that this approach improves goal-conditioned policy performance by $5.8\times$ compared to using goal observations like in past work (Sridhar et al., 2024; Chi et al., 2025).

Next, we use the waypoint-conditioned policy to lift a PEVA (Bai et al., 2025) world model to form a lifted world model (LWM) that is conditioned on high-level waypoints. Using this LWM, we perform using CEM by searching over the high-level waypoint space rather than the low-level, high-dimensional joint space that PEVA was trained on. We find that planning in waypoint space performs $4.7\times$ better than in joint space, finds better solutions with over different CEM budgets and over varying distances and time horizons.

[1]New York University [2]BAIR, UC Berkeley
[†]Advising authors. [‡]Equal contribution advising authors.

Figure 2: Visualization of o_t, o_g, goal pose p_g and waypoints a^{HL} in o_t. Goal pose p_g is not seen in o_g making it unsuitable as a goal.

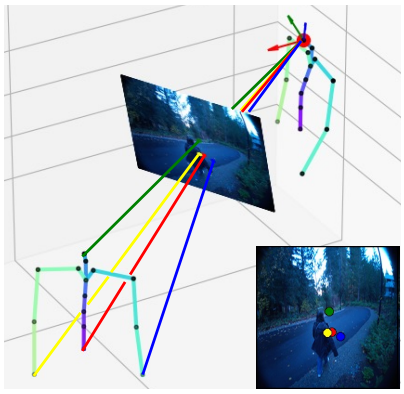

Figure 3: Computing waypoints from future poses during training. Goal pose **(front)** is projected onto the observation o_g of current pose **(back)**.

2 Lifting an Ego World Model

In this section, we introduce the three components of our method: (1) high-level waypoint action space for controlling egocentric, embodied agents, (2) the waypoint-conditioned controller policy that predicts sequences of low-level joint actions, and lastly, (3) the combination of the policy with a low-level world model to form a lifted world model (LWM) and it's use in planning. For background on world models and definitions for pose, and actions see Appendix A.

2.1 High-level Action Space

The high-level action space is designed to be intuitive and low-dimensional. We construct these actions a^{HL} as annotations in the current observation o_t. Each high level action consists of a set of 2D points called *waypoints* that are directly annotated on o_t. In conjunction, the set of points 2D points consitute a short-term goal pose in 3D space.

Waypoints for a Human Embodiment. For the PEVA embodiment, we define a^{HL} as four waypoints representing the pelvis, head, left and right hands,

$$a^{\text{HL}} = \{w_{\text{pelvis}}, w_{\text{head}}, w_{\text{left\_hand}}, w_{\text{right\_hand}}\} \tag{1}$$

affording navigation, interaction and camera control while abstracting away the spine, shoulder-elbow-wrist, etc (see Figure 2 (right)). Each waypoint is assigned a unique color: pelvis ●, head ●, left hand ●, right hand ●. While 2D image points do not exactly specify a 3D goal, the base image o_t and the relative spacing of the waypoints provides substantial context for inferring the goal pose.

Computing Ground-truth Actions. During training, high-level action labels are computed using the ground-truth goal pose projected back onto the current image. 3D positions can be obtained from the pose representation using forward kinematics (See Appendix H) which are then projected into the image plane using the camera function $\mathbf{P}$ (Figure 3)

$$\{\boldsymbol{x}_{\text{pelvis}}, \ldots, \boldsymbol{x}_{\text{left\_hand}}\} = \text{forward\_kinematics}(p_g) \tag{2}$$

$$w_{(\cdot)} = \mathbf{P}\boldsymbol{x}_{(\cdot)}. \tag{3}$$

2.2 Action Conditioned Policy

We train a lightweight policy to map high-level actions into the low-level space. At time t the policy predicts a sequence of low-level actions $a^{\text{LL}}_{t:t+T}$ conditioned on context observations $\boldsymbol{o}_t = \{o_{t-K_\pi}, \ldots, o_t\}$, context poses $\boldsymbol{p}_t = \{p_{t-K_\pi}, \ldots, p_t\}$ and a high-level waypoints a^{HL}_t

$$a^{\text{LL}}_{t:t+T} = \pi_\theta(\boldsymbol{o}_t, \boldsymbol{p}_t, a^{\text{HL}}). \tag{4}$$

Waypoints may also be masked, leaving those joints unconditioned. Having all waypoints masked equates to unconditioned action generation. The policy context size K_π may not match the world model context K. The policy is able to learn short-term motion patterns and uses image and pose context to infer 3D goals from 2D annotations. For masking and architectural details, see Appendix B.

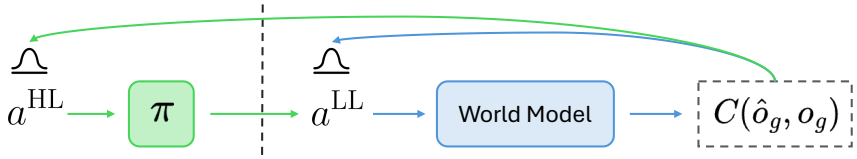

Figure 4: Planning using the Lifted World Model. Samples are drawn from the high-level (waypoint) action space compared to the low-level world model which uses the low-level space.

2.3 Lifted World Model

Now, using the policy, we can lift the low-level world model to a higher level of abstraction in the high-level action space (Figure 1). The lifted world model (LWM) predicts a sequence of observations given image and pose context and a single, high-level action

$$o_{t:t+T} = f^{\text{HL}}(\boldsymbol{o}_t, \boldsymbol{p_t}, a^{\text{HL}}). \tag{5}$$

Internally, the policy is used to predict a sequence of low-level actions that are used to autoregressively sample new observations from the low-level world model.

$$a^{\text{LL}}_{t:t+T} = \pi_\theta(\boldsymbol{o}_t, \boldsymbol{p}_t, a^{\text{HL}}) \tag{6}$$

$$o_{\tau+1} = f_\phi(\boldsymbol{o}_\tau, a^{\text{LL}}_\tau) \tag{7}$$

for $\tau \in \{t, \ldots, t+T\}$. Generated observations are included in the context $\boldsymbol{o}_\tau$ for $\tau > t$.

Search-based Planning. Using the LWM, we can plan in the high-level action space using the Cross-Entropy Method. Given a starting state and goal, we sample actions, simulate new observations using the world model and select the best trajectories to update the action distribution. Planning with the LWM samples actions in the waypoint space, while PEVA planning samples actions in low-level joint space. See Figure 4 for a visualization.

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

$$\boldsymbol{a}^{\text{LL}*} = \underset{\boldsymbol{a}^{\text{LL}}=a_t,\ldots,a_{t+T}}{\arg\min} \mathbb{E}_{\hat{o}_g \sim f_\theta(\boldsymbol{a}^{\text{LL}}), \boldsymbol{a}^{\text{LL}} \sim p_{\text{LL}}(\cdot)}[C(\hat{o}_g, o_g)], \tag{8}$$

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

$$\boldsymbol{a}^{\text{HL}*} = \underset{\boldsymbol{a}^{\text{HL}}=a^{\text{HL}}_1,\ldots,a^{\text{HL}}_{L-1}}{\arg\min} \mathbb{E}_{\hat{o}_g \sim f_\phi(\boldsymbol{a}^{\text{LL}}), \boldsymbol{a}^{\text{LL}} \sim \pi_\theta(\boldsymbol{a}^{\text{HL}}), \boldsymbol{a}^{\text{HL}} \sim p_{\text{HL}}(\cdot)}[C(\hat{o}_g, o_g)]. \tag{9}$$

The image inputs and the autoregressive sampling for the world model and policy are omitted for clarity. Optimization proceeds iteratively, the first N action samples are simulated and assigned a cost based on the last frame perceptual distance. The best K samples with the lowest cost are used to update the prior distribution from the next iteration of the algorithm.

3 Experiments

In this section we present results on training the controller policy using waypoint conditioning and planning using the lifted world model. For metrics and implementation details see Appendix C.

Base World Model and Nymeria Dataset. We use PEVA (Bai et al., 2025) trained on the Nymeria (Ma et al., 2024) dataset as our base low-level world model. Nymeria consists of videos collected by participants wearing ProjectAria (Engel et al., 2023) glasses and XSens (Roetenberg et al., 2009) motion capture suits performing continuous 15 minute skits in 50 different indoor and outdoor environments. The mocap suits provide 3D body motion and joint angle data that is used as low-level, high-dimensional joint angle actions. Nymeria presents the unique combination of a flexible human-like embodiment that is put to use in hybrid navigation + interaction tasks.

3.1 Goal-conditioned Policy

We present results on waypoint conditioned action generation using our policy trained on nymeria. Model performance measured by mean joint error (MJE) is in Table 1. "Unconditioned" uses no goal information; "goal conditioned" is when either either o_g or a^{HL} are provided. "Initial distance" is the MJE between p_t and p_g and "random weights" is an untrained policy. Models conditioned

Table 1: Unconditioned and goal-conditioned action generation for a egocentric, embodied agent.

	Unconditioned			Goal Conditioned		
Model	**Leaf MJE**	**Int. MJE**	**All MJE**	**Leaf MJE**	**Int. MJE**	**All MJE**
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	0.439	0.410	0.418	**0.244**	**0.220**	**0.227**

Table 2: Search-based planning results using CEM. 6 iterations, 64 samples/iteration.

Method	**Leaf MJE**	**Int. MJE**	**All MJE**
Initial Distance	0.803	0.776	0.784
PEVA CEM	0.713	0.688	0.694
Lifted CEM	**0.390**	**0.348**	**0.360**
Lifted CEM (min)	0.242	0.193	0.210
Unconditioned π	0.712	0.683	0.690

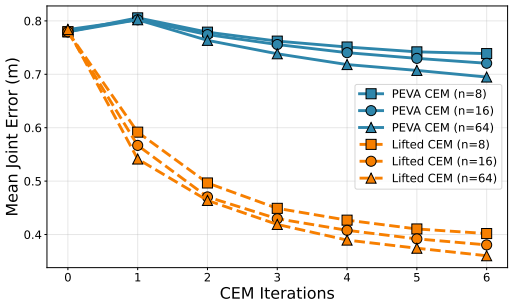

Figure 5: Planning with different CEM budgets.

on goal observation o_t base policy, +architecture, +pose context all have similar conditioned and unconditioned MJE. These policies learn coherent short-term motion, but o_g does not meaningfully affect the action distribution. However, with waypoint conditioning, we see that goal-conditioned generation improves over uncoditioned by 8.8cm, and adding waypoint masking further improves performance. This indicates that a^{HL} provides signal to egocentric action generation and is a suitable high-level action space. Additional policy experiments based on visibility, and decomposing the action generation task are in Appendix D.

3.2 Lifted World Model Planning

Using the lifted world model, we perform planning on hybrid navigation + interaction tasks sampled from the Nymeria dataset. Each task provides input observations $\boldsymbol{o}_t$, poses $\boldsymbol{p}_t$ and a goal observation o_g used to compute the cost. Planning outputs a sequence of joint actions, and performance is measured by the MJE between the true goal pose p_g and the final pose obtained from the predicted actions. Note that no goal information is provided and that searching with LWM must recover suitable waypoints to reach the goal. For results on unseen environments see Appendix E.

Task Generation. We randomly sample 200 tasks from the Nymeria evaluation set. Tasks are constrained to have ≥ 1 joint visible from p_g in o_t to reduce world model hallucinations and are filtered for MJE≥ 0.1m between p_t and p_g to avoid stationary tasks.

Quantitative Results. We report planning results in Table 2. Initial distance is the MJE between p_t and p_g. Planning in low-level joint only moves 0.09m closer to the goal while high-level waypoint space (Lifted CEM) reduces MJE by 0.42m. Sampling actions from an unconditioned policy is equivalent to not performing search and slightly outperforms searching in joint space showing the benefit of capturing short-term motion patterns. Nevertheless, it is not enough to reach a goal.

Planning Efficiency. Test-time planning is compute intensive so we study the performance of planning using the LWM for varying CEM compute budgets with results shown in Figure 5. Planning using LWM performs substantially better than with PEVA, from as low as 8 samples and 1 iteration through to the full 6 iterations and 64 samples. In contrast, planning with PEVA worsens after 1 iteration, highlighting the difficulty of searching in high dimensional joint space. In summary, lifting the abstraction of the world model not only improves performance, but also finds effective solutions more efficiently using much less compute.

4 Impact Statement

This paper presents work that aims to advance the field of world modelling, planning, navigation and interaction. There are many potential societal consequences for our work, none of which we feel are obvious, certain or consequential enough to be specifically highlighted here.

References

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, et al. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Yutong Bai, Danny Tran, Amir Bar, Yann LeCun, Trevor Darrell, and Jitendra Malik. Whole-body conditioned egocentric video prediction. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=XDTTwmjhAg.

Amir Bar, Gaoyue Zhou, Danny Tran, Trevor Darrell, and Yann LeCun. Navigation world models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15791–15801, 2025.

Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In *Learning for Dynamics and Control*, pp. 277–286. PMLR, 2020.

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.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10-11):1684–1704, 2025.

Maximilian Du and Shuran Song. Dynaguide: Steering diffusion policies with active dynamic guidance. In *Proceedings of the 39th Conference on Neural Information Processing Systems (NeurIPS)*, 2025.

Jakob Engel, Kiran Somasundaram, Michael Goesele, Albert Sun, Alexander Gamino, Andrew Turner, Arjang Talattof, Arnie Yuan, Bilal Souti, Brighid Meredith, et al. Project aria: A new tool for egocentric multi-modal ai research. *arXiv preprint arXiv:2308.13561*, 2023.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan P Foster, Pannag R Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. Openvla: An open-source vision-language-action model. In Pulkit Agrawal, Oliver Kroemer, and Wolfram Burgard (eds.), *Proceedings of The 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pp. 2679–2713. PMLR, 06–09 Nov 2025. URL https://proceedings.mlr.press/v270/kim25c.html.

Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, 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.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

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

Arjun Parthasarathy, Nimit Kalra, Rohun Agrawal, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning. *arXiv preprint arXiv:2512.09929*, 2025.

Michael Psenka, Michael Rabbat, Aditi Krishnapriyan, Yann LeCun, and Amir Bar. Parallel stochastic gradient-based planning for world models. *arXiv preprint arXiv:2602.00475*, 2026.

Daniel Roetenberg, Henk Luinge, Per Slycke, et al. Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors. *Xsens Motion Technologies BV, Tech. Rep*, 1(2009):1–7, 2009.

Reuven Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operational Research*, 99(1):89–112, 1997. ISSN 0377-2217. doi: https://doi.org/10.1016/S0377-2217(96)00385-2. URL https://www.sciencedirect.com/science/article/pii/S0377221796003852.

Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al. Dinov3. *arXiv preprint arXiv:2508.10104*, 2025.

Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 63–70. IEEE, 2024.

Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2): 344–357, 2017.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=D5RNACOZEI.

A PRELIMINARIES

In this section we state the concept of a world model and the pose and action representations used for embodied, human-like actions in our work.

World Models. A world model predicts a future observation o_{t+1} given an action a_t and a current observation o_t. Models also often condition on K previous observations, $\boldsymbol{o}_t = \{o_{t-K}, \dots, o_t\}$, giving us:

$$o_{t+1} = f_\phi(\boldsymbol{o}_t, a_t). \tag{10}$$

Pose representation. To represent the human-like embodiment, we use the XSens (Roetenberg et al., 2009) model following PEVA (Bai et al., 2025). A pose p consists of the 3D position $\boldsymbol{x}_{\text{pelvis}}$ of the pelvis and Euler angles $\boldsymbol{\phi}_{(\cdot)}$ for each joint. We use the upper-body PEVA checkpoint with 15 joints and the pose representation:

$$p = [\boldsymbol{x}_{\text{pelvis}}^\top | \boldsymbol{\phi}_{\text{pelvis}}^\top | \boldsymbol{\phi}_{\text{Lumbar5}}^\top | \dots | \boldsymbol{\phi}_{\text{left\_hand}}^\top] \in \mathbb{R}^{48}. \tag{11}$$

The pelvis position localizes the pose in a reference frame while the angles describe the orientation of each joint. This model assumes rigid bones, i.e. constant distance between joints.

A single pose p_t is defined with respect to its own pelvis frame, meaning that at the starting time t the position and Euler angle of the pelvis are both zero: $\boldsymbol{x}_{\text{pelvis}}, \boldsymbol{\phi}_{\text{pelvis}} = \mathbf{0}$. 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. Action a_t is the change between poses p_t, p_{t+1}

$$a_t = [\delta\boldsymbol{x}_{\text{pelvis}}^\top | \delta\boldsymbol{\phi}_{\text{pelvis}}^\top | \delta\boldsymbol{\phi}_{\text{Lumbar5}}^\top | \dots | \delta\boldsymbol{\phi}_{\text{left\_hand}}^\top] \in \mathbb{R}^{48}. \tag{12}$$

Non-pelvis joints are defined in the frame of their parent joint so angular displacement can be computed directly:

$$\delta\boldsymbol{x} = \boldsymbol{x}_{t+1} - \boldsymbol{x}_t, \tag{13}$$

$$\delta\boldsymbol{\phi}_{(\cdot)} = \mathrm{R}^{-1}(\mathrm{R}(\boldsymbol{\phi}_{t,(\cdot)})\mathrm{R}(\boldsymbol{\phi}_{t+1,(\cdot)})^\top). \tag{14}$$

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

B POLICY DETAILS

2D Goal Conditioning. Defining actions as 2D points in o_t makes them visually interpretable and practical to manually specify at test time improving policy and thus world model control. Additionally, waypoint actions are low dimensional and finite avoiding the curse of dimensionality during search-based planning. This is in contrast to the usual approach of using a goal-time observation o_g in methods like Sridhar et al. (2024) and Chi et al. (2025) which are difficult to obtain, high dimensional and cannot be searched.

Waypoint Masking. Waypoint masking is employed during training to support sparse waypoints during inference. For example, the pelvis can be controlled independently for navigation while the policy unconditionally controls the remaining joints. During training, half the time, each point is masked independently at $p(\text{mask}) = 0.5$, and the other half of the time, no points are masked. This masking procedure is also helpful to make use of waypoints that are not visible in o_t. Different head orientation may obscure otherwise visible waypoints that represent sensible motion sequences and using masking allows for training on these examples without associating all missing waypoints with them being out-of-frame.

Architecture. The policy $\pi(\cdot)$ is a diffusion policy that generates actions by directly denoising in joint angle space. First, a goal image o_g is created by plotting each point in a_t^{hl} onto observation o_t, each with a different color. Then, images $\boldsymbol{o}_t$ and o_g are encoded using a DINOv3-S (Siméoni et al., 2025) encoder. The tokens are not pooled to preserve spatial information, producing embeddings $z_{t-K_\pi:t}, z_g \in \mathcal{R}^{L\times D}$. Context poses $\boldsymbol{p}_t$ are linearly projected into D dimensions and are added to

the corresponding image embeddings $z'_{(\cdot)} = z_{(\cdot)} + \text{proj}(p_{(\cdot)})$. 3D positional embeddings are added to the vision tokens that are then processed using a vision transformer. The output tokens are then pooled to form context vector $c_t \in \mathcal{R}^D$ which is used as input to the denoising UNet to generate low-level actions.

The base policy is similar to Sridhar et al. (2024) with added pose context, a DINOv3 encoder, later pooling, 3D positional embeddings and a larger denoising network. See Appendix I for an architecture figure. These changes improve performance but any effective base architecture is suitable for lifting a world model.

C Metrics and Implementation details

Metrics. The policy and planning tasks are evaluated by how close a sequence of joint actions starting at initial pose p_t reach the goal pose p_g. Specifically, we report the mean joint error (MJE) in meters of the *leaf* joints (pelvis, head and hands), the *intermediate* joints between them and *all* joints altogether.

First, we apply the sequence of predicted actions starting from initial pose p_t to obtain a predicted final pose $\hat{p}_g$.

$$\boldsymbol{x}_{t+1,\text{pelvis}} = \text{R}(\boldsymbol{\phi}_{t:t+1,\text{pelvis}})\boldsymbol{x}_{t,\text{pelvis}} + \boldsymbol{x}_{t:t+1,\text{pelvis}} \tag{15}$$

$$\boldsymbol{\phi}_{t+1,(\cdot)} = \text{R}^{-1}(\text{R}(\boldsymbol{\phi}_{t:t+1,(\cdot)})\text{R}(\boldsymbol{\phi}_{t,(\cdot)})). \tag{16}$$

Then forward kinematics is used to obtain the actual 3D positions for every joint $\{\boldsymbol{x}_{\text{pelvis}}, \ldots, \boldsymbol{x}_{\text{left\_hand}}\} = \text{forward\_kinematics}(p_g)$. Finally, MJE computed for every joint between the predicted and ground truth goal pose

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

Implementation Details. The policy is trained for 10 epochs with a learning rate of 5×10^{-4} using AdamW and 256 batch size. Action sequence length is $T = 8$ and context length is $K_\pi = 3$. The denoising diffusion network UNet dimensions are increased from $[64, 128, 256]$ in NoMaD to $[256, 384, 512]$ for all experiments. Nymeria data preparation matches PEVA (Bai et al., 2025), sampling at 4fps.

Our experiments use the upper-body PEVA checkpoint using 15 joints and an action dimensionality of $D = 48$. World model planning uses 6 CEM iterations and 64 samples per iteration. The world model context length is $K = 8$ and 64 denoising iterations are used. The action prior is a Gaussian $\mathcal{N}(\mathbf{0}, \sigma^2 I)$ where $\sigma = 0.05$ for a^{LL} and $\sigma = 0.3$ for a^{HL}. The boundaries of image are $[-0.5, 0.5]$.

D Additional Policy Experiments

Qualitative Action Generations. We present visualizations of actions generated by the waypoint-conditioned policy. Action sequences are visualized by projecting the 3D pose at each timestep onto the starting observation o_t, preserving environmental context. Figure 7 shows the controllability of our policy, generating actions conditioned on new, user-specified waypoints. Figure 7 shows examples where waypoints change the action sequence based on context.

Waypoint Visibility. Because waypoints are defined in observation o_t, we investigate if performance decreases when the goal pose is not visible at current time o_t. Table 3 separates the goal-conditioned MJE by goal joint visibility in o_t. We see a modest increase in MJE when the joints are not visible for waypoint-conditioned models, but we see an even more substantial drop for observation-conditioned models. This suggests that while waypoints inherently require visibility, waypoint-conditioned training learns better motion patterns that generalize to invisible goals at test time.

Decomposing Action Generation. To better understand goal-conditioning for egocentric action generation, we decompose it into two separate tasks: motion generation and goal pose prediction.

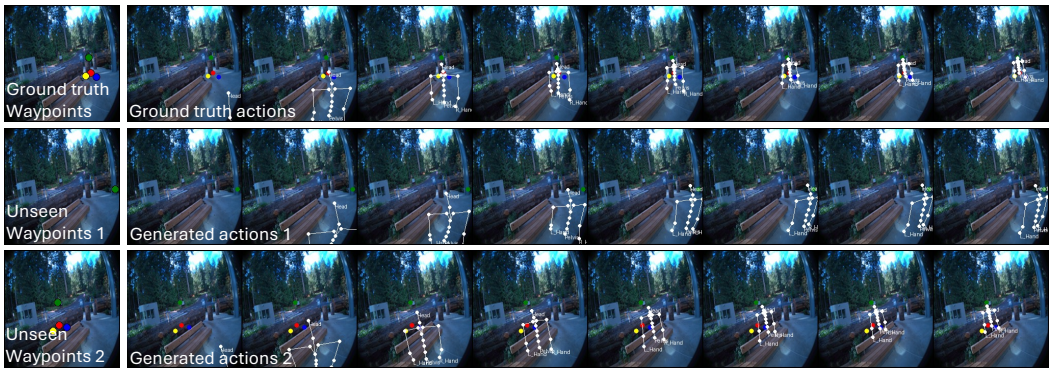

Figure 6: Generating actions from waypoints not in data. **Top:** ground truth waypoints and actions; walking down the path. **Middle:** • head, on the right; agent shifts to the right, facing left. **Bottom:** 4 waypoints above bench; agent climbs up onto the bench.

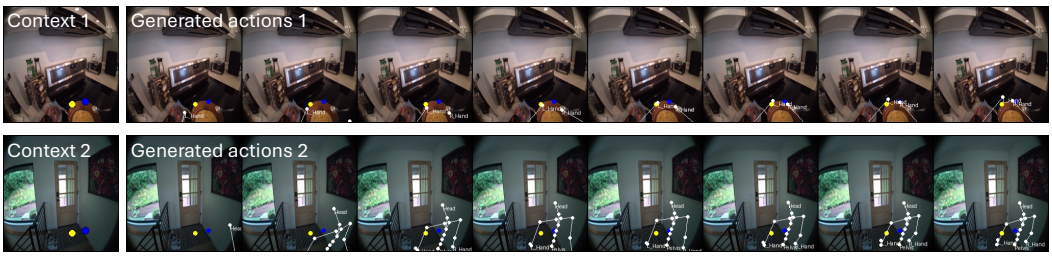

Figure 7: Contextually aware action generation. The policy generates different, contextually appropriate actions from the same waypoints.

For motion generation, we train a policy to predict joint action sequences with the goal pose p_g as input. This setup can be viewed as a "cheat" setting where the model is given the exact goal and has to replicate the joint action sequence present in the data to reach it. For goal pose prediction, we replace the denoising UNet with an MLP and train the model to predict the goal pose p_g conditioned on the goal observation o_g. Both models are evaluated using MJE between final predicted pose and ground truth goal pose.

Results in Table 4 indicate that predicting the joint sequence given the goal pose is easier than identifying the p_g from o_g. This indicates more broadly that egocentric observations are insufficient to infer the embodied pose as p_g is seldom visible from egocentric observations o_g — occasionally hands may be visible but the vast majority of the torso is never seen.

E Additional Planning Experiments

Unseen Environments. We evaluate the generalization of our lifting procedure to unseen environments. A new policy is trained on a new training set with all videos from Nymeria locations 6, 19 and 34 being removed and replaced with clips from other locations. This held-out policy is then used to lift the same pretrained PEVA world model.

The LWM using the held-out policy is evaluated on planning tasks sampled from the environments 6, 19 and 34. Results are presented in Table 5. We find that the held-out LWM generalizes well to unseen environments, greatly outperforming CEM in low-level space and only slightly underperforming the LWM using the in-distribution policy. Note all methods use the same PEVA checkpoint that has seen all 3 locations in world model training.

Table 3: Goal-conditioned action generation with metrics separated based on the visibility of each joint in the current observation o_t.

	Visible		Not Visible	
Model	**Leaf**	**Int.**	**Leaf**	**Int.**
Base Policy	0.360	0.297	0.605	0.705
+ architecture	0.340	0.278	0.566	0.659
+ proprioception	0.305	0.254	0.483	0.551
+ waypoints	**0.260**	**0.207**	0.327	0.358
+ waypoint mask	**0.252**	**0.201**	**0.284**	**0.307**

Table 4: Results decomposing goal-conditioned action generation into motion generation with known goal pose and predicting a goal pose given goal-time observation.

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

F Related Work

Embodied, Egocentric and Hierarchical Policies. Sridhar et al. (2024) tackles navigation from an egocentric point of view with a point-robot embodiment. Chi et al. (2025) explores manipulation, often with articulated robots. Nachum et al. (2018) is an earlier work that learns a hierarchy of policies. Large-scale foundation models like GR00T (Bjorck et al., 2025) can predict manipulation actions for many different embodiments from text directives. Other large models include OpenVLA (Kim et al., 2025), Octo (Octo Model Team et al., 2024) and are mostly trained for exocentric robotic manipulation tasks.

World Model Planning. Cross-entropy method (Rubinstein, 1997) is the *de-facto* planning method and is used alongside VJEPA-2 (Assran et al., 2025), NWM (Bar et al., 2025), DINO-WM (Zhou et al., 2025) for planning tasks. MPPI (Williams et al., 2017) is another method that sees use. Works have also explored gradient-based methods (Zhou et al., 2025; Parthasarathy et al., 2025) but often find they may perform worse than search-based methods and are not easily implemented with diffusion-based world models. Du & Song (2025) is uniquely relevant, and combines a transformer-based world model along with a policy to perform planning.

G Cost-function convergence during world model planning

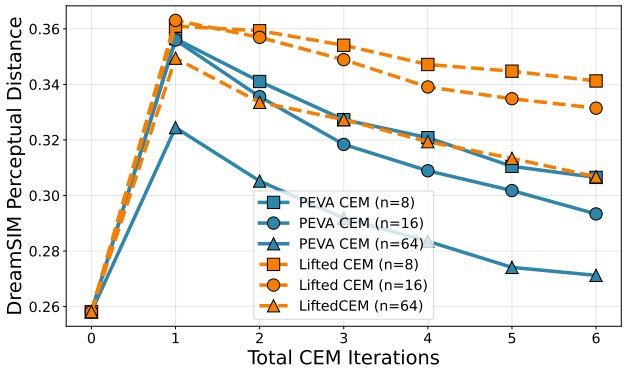

Figure 8: Cost function convergence with respect to CEM iterations and number of samples.

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

Method	**Leaf MJE**	**Int. MJE**	**All MJE**
Initial Distance	0.678	0.644	0.653
PEVA CEM	0.645	0.613	0.622
Lifted CEM	0.373	0.326	0.339
Lifted CEM (h/o)	0.389	0.341	0.355

H FORWARD KINEMATICS

We use a forward kinematics computation based on the XSens (Roetenberg et al., 2009) human model to compute joint positions in 3D space. The pose representation used in our paper directly specifies the pelvis position in 3D space as the root of our embodiment $\boldsymbol{x}_{\text{pelvis}}$. The 3D position of a child joint is computed relative to the parent joint.

$$\boldsymbol{x}_{\text{child}} = R(\boldsymbol{\phi}_{\text{child}})\boldsymbol{x}_{\text{parent}} + \Delta\boldsymbol{x}_{\text{parent,child}} \tag{18}$$

The euler angles $\boldsymbol{\phi}_{\text{child}}$ represent the frame change from the parent to the child. $\Delta\boldsymbol{x}_{\text{parent,child}}$ is the spatial offset representing the bone defined in the child frame. The individual bone lengths for each participant varies.

The kinematic parent-child relationships are shown in Table 6.

Kinematic parent	Child
N/A	Pelvis
Pelvis	L5
L5	L3
L3	T12
T12	T8
T8	Neck
Neck	Head
T8	R_Shoulder
R_Shoulder	R_UpperArm
R_UpperArm	R_Forearm
R_Forearm	R_Hand
T8	L_Shoulder
L_Shoulder	L_UpperArm
L_UpperArm	L_Forearm
L_Forearm	L_Hand

Table 6: Kinematic tree parent-child relationships

I Policy Architecture

Below are the policy architecture used in this paper. We also present the motion generation and goal prediction models used in Section D.

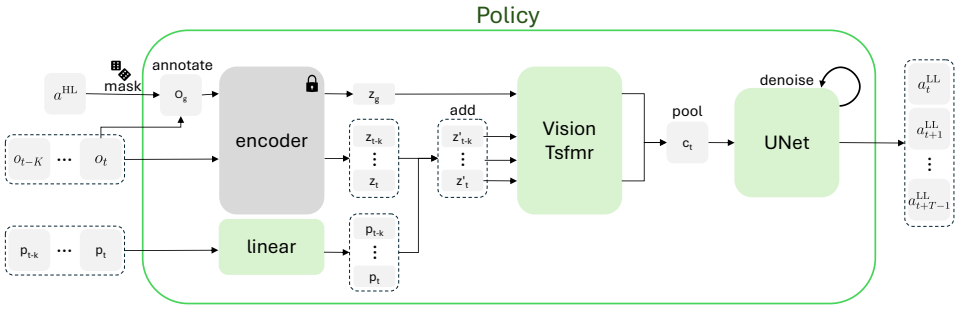

Figure 9: Policy architecture

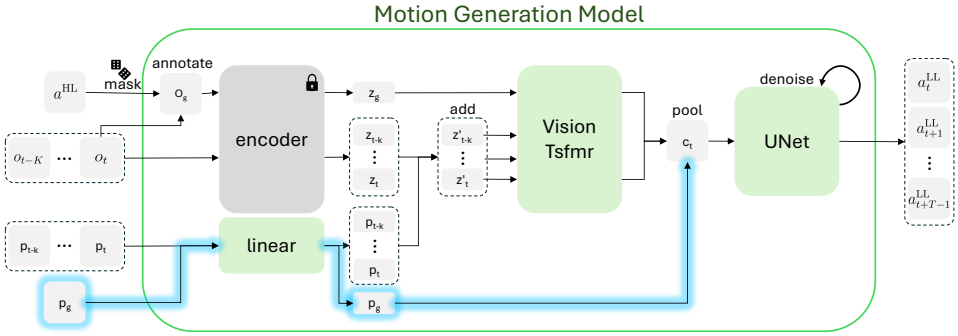

Figure 10: Motion Generation model architecture; changes highlighted in blue.

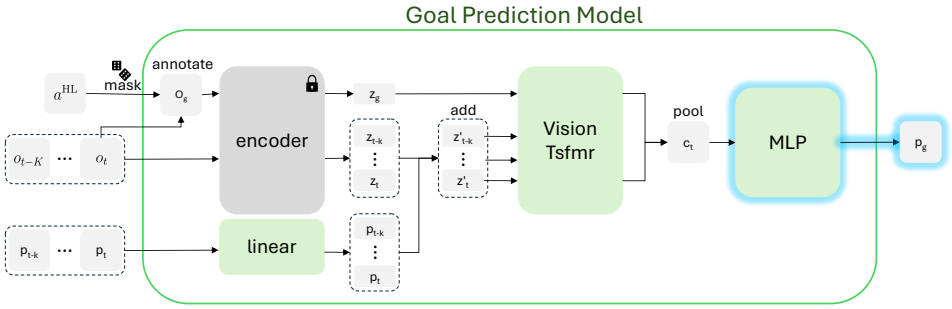

Figure 11: Goal prediction model architecture; changes highlighted in blue.

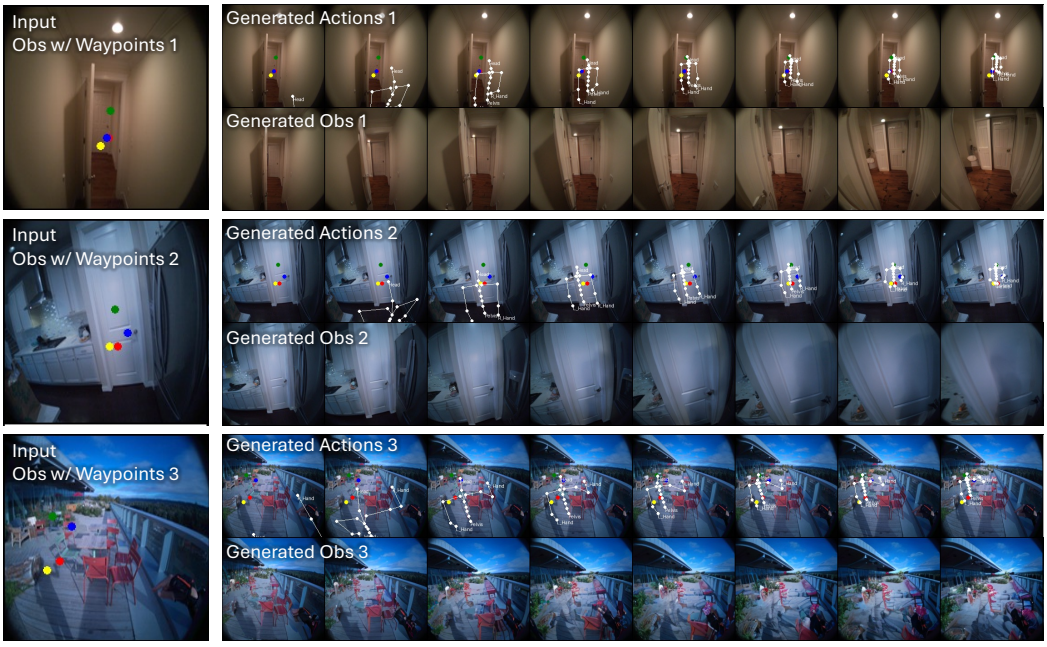

Figure 12: Visualization of generated actions and observations. The input is the current observation o_t with waypoint annotations. **Top:** walking through the doorway. **Middle:** grasping the doorknob. **Bottom:** Avoiding obstacles while filming with a camera.

J Policy Visualizations

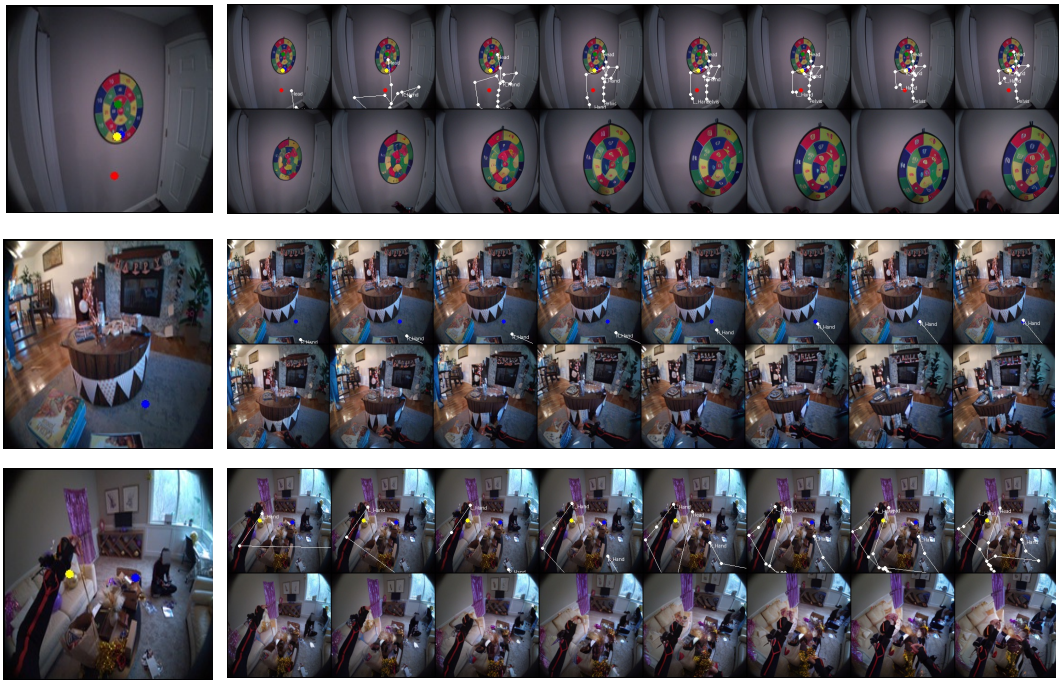

Figure 13: Additional visualizations of waypoint actions and rollout.

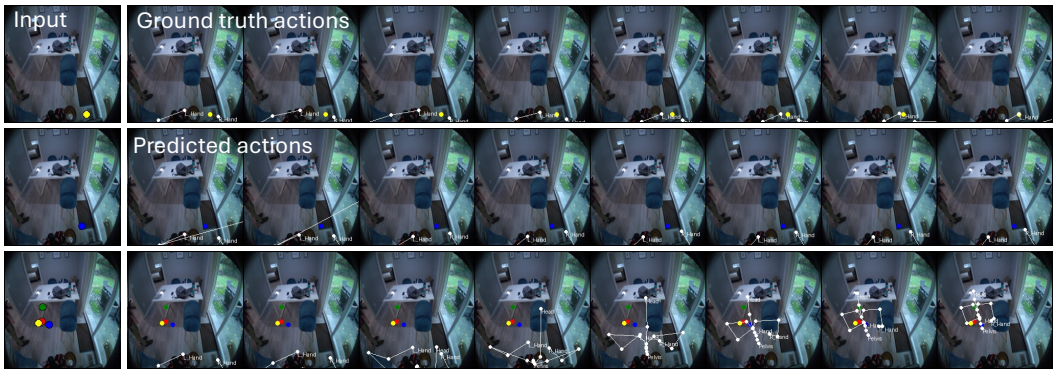

Figure 14: Extra Counterfactual Visualizations

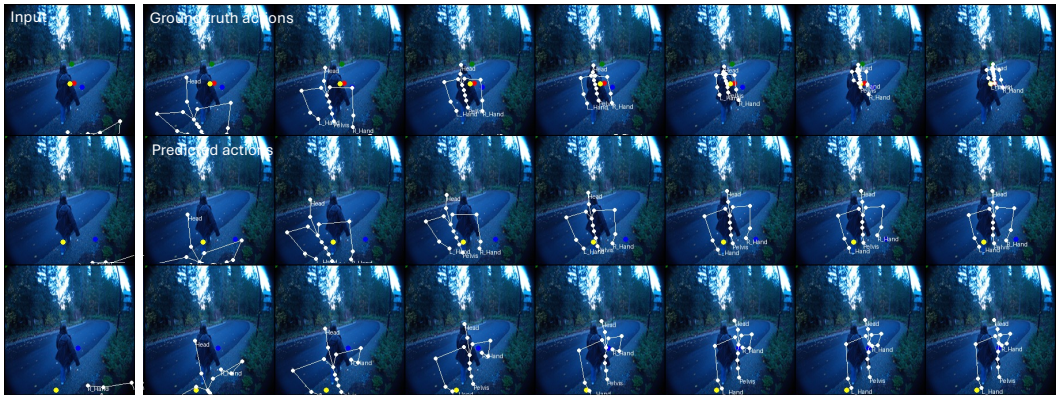

Figure 15: Extra Counterfactual Visualizations