2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Ego-Vision World Model for Humanoid Contact Planning

Hang Liu[2], Yuman Gao[1], Sangli Teng[1], Yufeng Chi[1],
Yakun Sophia Shao[1], Zhongyu Li[3], Maani Ghaffari[2], and Koushil Sreenath[1]

***Abstract*— Enabling humanoid robots to exploit physical contact, rather than simply avoid collisions, is crucial for autonomy in unstructured environments. Traditional optimization-based planners struggle with contact complexity, while on-policy reinforcement learning (RL) is sample-inefficient and has limited multi-task ability. We propose a framework combining a learned world model with sampling-based Model Predictive Control (MPC), trained on a demonstration-free offline dataset to predict future outcomes in a compressed latent space. To address sparse contact rewards and sensor noise, the MPC uses a learned surrogate value function for dense, robust planning. Our single, scalable model supports contact-aware tasks, including wall support after perturbation, blocking incoming objects, and traversing height-limited arches, with improved sample efficiency and multi-task capability over on-policy RL. Deployed on a physical humanoid, our system achieves robust, real-time contact planning from proprioception and ego-centric depth images. Code and dataset are available at our website: https://ego-vcp.github.io/.**

I. Introduction

Humanoids are expected to advance from dynamic locomotion [1, 2] to intelligent interaction in complex, unstructured environments. Achieving this requires purposeful contact exploitation rather than simple avoidance. Effective humanoids must use their bodies to interact with the world as humans do, such as bracing against a wall for balance, blocking objects for safety, or ducking under obstacles. These contact-aware skills are essential for greater autonomy, robustness, and intelligence in robots.

Reasoning about contact remains challenging for humanoids [3–5]. Traditional optimization-based methods [6, 7] struggle with the complexity of real-time contact scheduling and are sensitive to model inaccuracies, reducing adaptability to unforeseen situations.

Parallelized simulation [8] has enabled on-policy RL to succeed in robot control for quadrupeds [9], bipeds [10], and humanoids [11]. However, these methods are sample-inefficient, especially with visual inputs [9], and struggle with multi-task learning.

We address this by integrating a learned world model with sampling-based Model Predictive Control (MPC). Our approach trains a scalable world model from a random, demonstration-free offline dataset, predicting future outcomes in the compressed latent space rather than raw pixels, and understanding action consequences. We introduce a surrogate value function to guide planning, allowing MPC to evaluate candidate action sequences efficiently. This synergy enables agile, vision-based contact planning for humanoids across tasks with improved sample efficiency and performance.

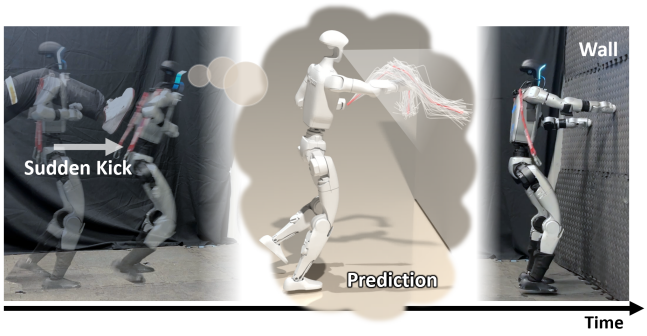

Fig. 1: An illustration of our framework in the "Support the Wall" task. When subjected to a sudden perturbation (left), the robot uses its learned world model to predict and plan a stabilizing action within its planning horizon (center). This allows it to successfully execute the plan and brace its hands against the wall to make contact and maintain balance (right).

The main contributions of this work are as follows:

1) **A Scalable Visual World Model for Dynamic Robots**: We learn a visual world model that scalably captures the dynamics of diverse contact tasks, trained entirely on a demonstration-free offline dataset.
2) **Planning from Pixels with Value-Guidance**: We introduce a sampling-based MPC framework using a learned surrogate value function to guide the planning process.
3) **Agile and Robust Real-world Visual Contact Planning**: We validate the proposed framework on a physical humanoid robot, demonstrating multiple novel agile and robust contact planning capabilities solely from ego-centric depth images and proprioceptive feedback.

II. Related Work

A. Model-Based Contact Planning

For both locomotion and manipulation, robotics is replete with contact-rich problems, made challenging by the non-smooth dynamics induced by the impact [1]. Optimization-based approaches address this by explicitly modeling these physical interactions, like linearizing the complex friction model into a Linear Complementarity Problem (LCP) [12], or relaxing it into a Cone Complementarity Problem (CCP) [13]. These formulations can then be embedded within a trajectory optimization framework [6, 7]. Another prominent paradigm is Hybrid Zero Dynamics (HZD) [14, 15], which addresses the non-smooth contact dynamics of legged locomotion by enforcing virtual constraints whose associated zero dynamics surface remains invariant through impacts. However, such model-based approaches are often hindered by model inaccuracies and high computational costs [16],

[1]University of California, Berkeley
[2]University of Michigan, Ann Arbor
[3]The Chinese University of Hong Kong

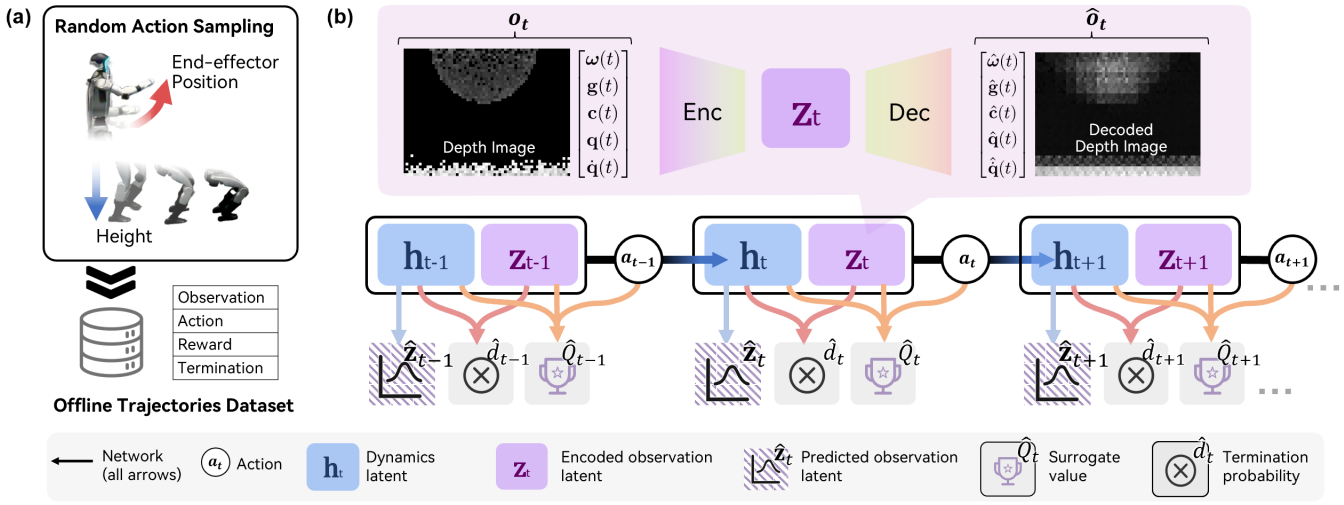

Fig. 2: **World Model Training Pipeline.** The pipeline begins with the offline data collection process shown in **(a)**, where a dataset $\mathcal{D}$ of trajectories is generated by applying randomly sampled high-level actions (end-effector position $p_{ee}^{\top}$ and body height h_{body}) to a simulated humanoid equipped with a trained low-level policy. This dataset is then used to train the world model, as depicted in **(b)**. At each timestep t, an observation o_t, consisting of a depth image and proprioception, is encoded into an observation latent z_t, which is then decoded to produce a reconstruction $\hat{o}_t$. Concurrently, a recurrent network updates its latent h_t based on the previous state and action. The model predicts (i) $\hat{z}_t$, a prior sample of the observation latent; (ii) $\hat{d}_t$, the termination probability; and (iii) $\hat{Q}_t$, a surrogate action-value guiding the planner in evaluating different actions. All of these predictions are optimized against the ground-truth data from the offline trajectories, enabling the model to learn both the environment's dynamics and a robust value function for planning.

which complicate real-time deployment. Furthermore, their reliance on predefined structures, such as periodic gaits [17] or reference foot-end trajectories [14], makes it difficult to scale them to more general, aperiodic whole-body contact scenarios.

B. Learning-Based Contact Planning

Learning-based contact planning has demonstrated potential for dynamic skills [3, 5, 9, 10, 18, 19], yet three key challenges persist. First, interaction capabilities are hindered by simplified 2.5D elevation maps [4], which fail to represent dynamic or overhanging obstacles like moving objects or archways. Second, sample efficiency remains a bottleneck; heavy reliance on synthetic data [8] makes vision-based on-policy RL computationally challenging [9], while the sparse, discontinuous nature of contact complicates exploration for model-free methods [5, 20]. Finally, limited multi-task adaptability prevents policies from generalizing across diverse object interactions or varying task definitions [21].

C. Planning with Robotic World Models

A world model [22] provides a learned internal model of an environment, enabling robots to predict future outcomes within an abstract latent space. Integrating these models into model-based planning offers a promising trajectory for enhancing generalization and sample efficiency in contact planning [23, 24]. While early frameworks focused on accelerating policy learning [23, 24], contemporary world models have evolved into generative systems [25, 26] capable of simulating complex physical causalities and dynamics. Parallel efforts in robotics leverage neural dynamics models to represent intricate systems [27–30], which can be integrated into sampling-based MPC [16, 31, 32] for adaptive control [4, 33, 34].

Nonetheless, enabling robotic world models to fully generalize remains an open problem. This is especially the case for contact planning, as the underlying whole-body contact state is not directly observable and is difficult to infer and predict from partial, noisy sensory data.

III. Methods

Section III-A first describes the **low-level controller** that executes motor control and defines the high-level planning interface used throughout the paper. We then present our **data collection** procedure for training the planner in Section III-B. Finally, we detail the two core components of our high-level planner: the **architecture and training of the world model** (Section III-C), and the **value-guided sampling MPC** framework that utilizes the world model for test-time planning (Section III-D).

A. Low-Level Controller

Our framework utilizes a low-level controller capable of tracking diverse locomotion and manipulation commands. The controller tracks the command vector $c = [v^{\top}, p_{ee}^{\top}, h_{body}]^{\top}$, where v, p_{ee}, and h_{body} denote the desired linear velocity, end-effector position, and body height, respectively. Its observation space is purely proprioceptive, comprising angular velocity (ω), the projected gravity vector (g), the command (c), joint positions (q), and joint velocities ($\dot{q}$). The controller is trained in simulation via PPO, following established approaches [8, 35].

B. Data Collection

Once a reliable low-level controller is obtained, we collect an offline object-interaction dataset for the subsequent vision-based world model training. Our offline dataset $\mathcal{D}$ is generated by collecting trajectories τ in simulation across three object types: a ball, a wall, and an arch. At each

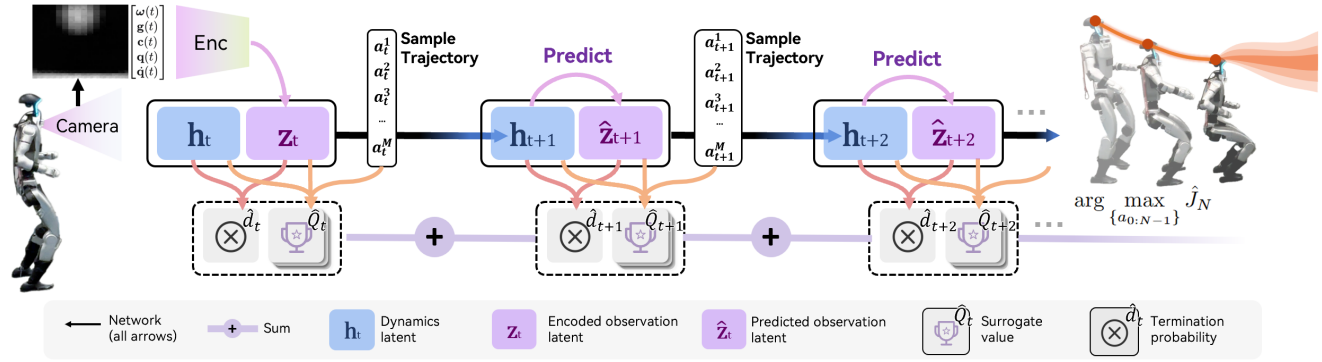

Fig. 3: **Value-Guided Sampling MPC.** This figure illustrates how the trained world model is used for planning via value-guided sampling MPC. This process performs open-loop prediction to find the best action sequence starting from a single real observation. At inference time, this process begins by encoding the current observation o_t into its latent state z_t, after which the planner samples a batch of $M = 1024$ candidate action sequences over a planning horizon of $N = 4$ steps. The world model predicts the future latent state $(h_{t+k}, \hat{z}_{t+k})$ by recursively applying its learned dynamics model. At each prediction step, the surrogate value $(\hat{Q}_{t+k})$ evaluates the sampled actions, while the termination signal, $\hat{d}_{t+k}$, predicts the probability of robot failure, such as falling; if this probability exceeds a threshold of 0.9, all subsequent values for that trajectory are set to zero. The planner evaluates M candidate trajectories, where the score for each trajectory is calculated by the objective function $\hat{J}_N$ in Eq. (17). This set of scored trajectories is then optimized using the Cross-Entropy Method (CEM) to find the optimal action sequence.

timestep, the robot receives an observation o_t including a downsampled 64×48 ego-centric depth image and proprioceptive signals, executes a randomly sampled action $a_t = \left[p_{ee}^\top, h_{body}\right]^\top$. We exclude the desired linear velocity v from the planner's action space to force the robot to solve contact-rich problems through postural manipulation. In return, the robot receives the reward r_t and the termination signal d_t from the environment. These collected transition tuples $\{o_t, a_t, r_t, d_t\}$ are then stored in a final trajectory dataset structured as `[Batch, Time, Data]`.

At each step, we sample the finite differences of planner actions $a_t = a_{t-1} + \eta \cdot \delta$ from $\delta \sim \mathcal{U}(-1, 1)$, where η is a scalar step that controls the magnitude of the update. This step is performed after normalizing the task space $\left[p_{ee}^\top, h_{body}\right]^\top$ of the low-level controller. η is set to 0.32. The purpose of using such a method to sample the actions is to (1) avoid using any demonstration, which is expensive to obtain for the whole-body commands of a humanoid, and (2) avoid ineffective and jittery behavior data from random sampling. For a detailed description of the process, see the illustration video on our website and code.

C. Ego-Vision Humanoid World Model

Prior auto-regressive models learn system dynamics by mimicking existing controllers for continuous tasks such as velocity tracking [29]. However, when applied to high-dimensional image observations, this pixel-prediction approach suffers from compounding errors over long horizons. This issue is exacerbated in **contact-aware** scenarios where defining a goal trajectory in the pixel space is often intractable.

To address this, we draw inspiration from general world models such as Dreamer [36] and JEPA [25]. We focus on predicting the abstract latent states of future observations, enabling the model to capture more fundamental structures within the data. As illustrated in Fig. 2, our world model is composed of several key components detailed below.

First, the world model leverages a recurrent neural network (RNN) to maintain a deterministic dynamics latent state h_t. At each step, a stochastic latent state z_t, which extracts the abstract latent of the current observation, is inferred from the current observation o_t and the latent state h_t. Similar to an autoencoder, the model is trained to reconstruct the observation o_t as $\hat{o}_t$ after passing it through a bottleneck, which compels the latent state z_t to encode the most salient and abstract features. For notational simplicity, we let ϕ denote the parameters of all world model components, with q_ϕ and p_ϕ representing the encoder and decoder, respectively. The overall process can then be expressed as:

$$h_t := f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) \tag{1}$$

$$z_t \sim q_\phi(z_t \mid h_t, o_t) \tag{2}$$

$$\hat{o}_t \sim p_\phi(\hat{o}_t \mid h_t, z_t). \tag{3}$$

We model both z_t and $\hat{o}_t$ as Gaussian distributions.

Furthermore, we introduce a model that estimates the stochastic latent without using the current observation: given h_t, it predicts a latent $\hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t)$ that closely approximates z_t, thereby enabling rollouts in latent space.

Different from Dreamer [36], we need to consider an architecture that better addresses the challenges unique to robotics in the real world, such as (1) significant partial observability, (2) high sensor noise, and (3) sparse contact. These factors make it difficult to predict contact-aware rewards from the observation. Therefore, we design specialized heads that, conditioned on the latent (h_t, z_t) and a candidate action a_t, directly estimate the expected long-term outcome. Specifically, we predict a termination probability $\hat{d}_t$ and a surrogate value $\hat{Q}_t$, which represents the expected cumulative return. This allows the robot to evaluate the potential response to different actions directly from its learned latent.

$$\hat{d}_t \sim D_\phi(\hat{d}_t \mid h_t, z_t), \tag{4}$$

$$\hat{Q}_t := Q_\phi(h_t, z_t, a_t). \tag{5}$$

Our surrogate value could condition on the latent state z_t, allowing the robot to infer the current task context from

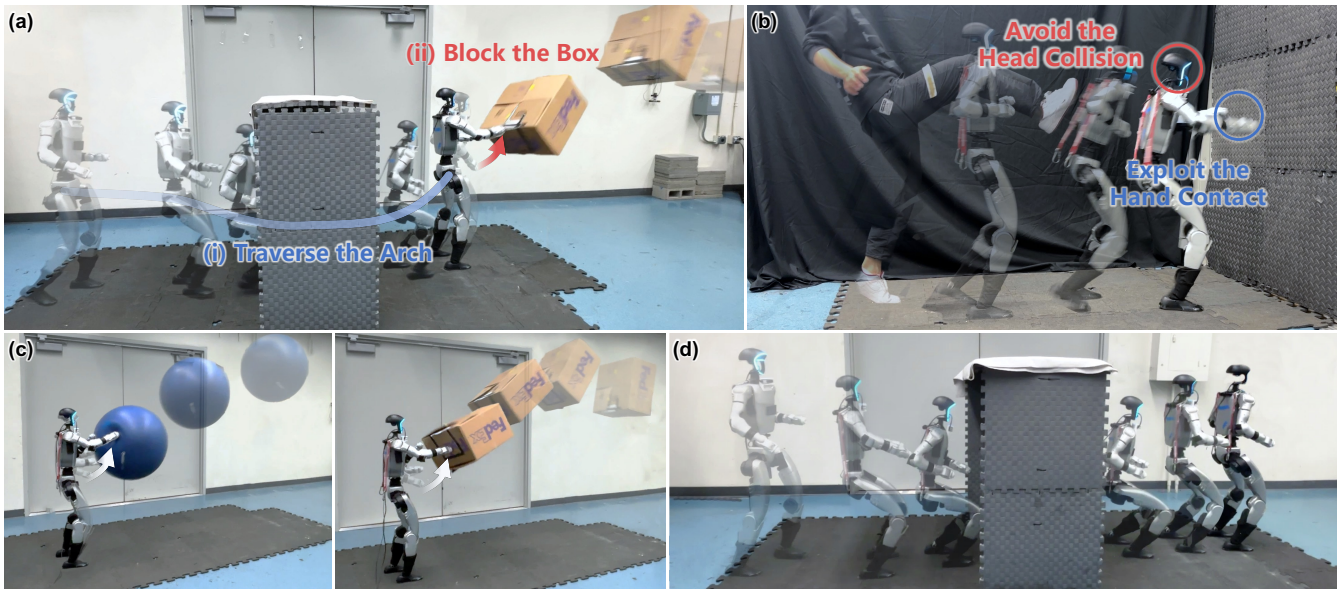

Fig. 4: Real-World experiments validating the proposed framework. (a) A demonstration of sequential task execution and generalization, where the robot traverses an arch (i) and then blocks a previously unseen box (ii). (b) Support the wall to maintain balance by bracing the wall with the hands when pushed towards the wall. (c) Blocking both an in-distribution ball (with a size consistent with the training data) and an unseen box; (d) Squat and traverse an arch.

its observations and dynamically adapt its objective. This enables us to train the model directly on a mixed dataset containing data from all tasks. We opt for a computationally efficient design consisting of a CNN for image feature extraction and MLPs for all other components.

The model is optimized by minimizing the loss as shown below. This total loss is a simple sum of three main components: a reconstruction loss ($\mathcal{L}_{\text{rec}}$), a joint-embedding predictive loss ($\mathcal{L}_{\text{jep}}$), and a Q-loss ($\mathcal{L}_{\hat{Q}}$):

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{jep}} + \mathcal{L}_{\hat{Q}}. \tag{6}$$

The reconstruction loss, $\mathcal{L}_{\text{rec}}$, ensures the world model can extract a tight latent space of the environment. It is defined as a combination of a Negative Log-Likelihood (NLL) for the observation reconstruction ($\mathcal{L}_{\text{obs}}$) and a Binary Cross-Entropy (BCE) loss for the termination signal ($\mathcal{L}_{\text{term}}$):

$$\mathcal{L}_{\text{rec}} = \mathbb{E}_{z_t \sim q_\phi(z_t|h_t,o_t)} \left[\mathcal{L}_{\text{obs}} + \mathcal{L}_{\text{term}} \right]. \tag{7}$$

The joint-embedding predictive loss, $\mathcal{L}_{\text{jep}}$, consists of two KL divergence terms that enforce a consistent and non-collapsing latent space [36, 37].

$$\begin{aligned} \mathcal{L}_{\text{jep}} = D_{\text{KL}}\Big(\text{sg}\big(q_\phi(z_t|h_t,o_t)\big) \;||\; p_\phi(\hat{z}_t|h_t) \Big) \\ + D_{\text{KL}}\Big(q_\phi(z_t|h_t,o_t) \;||\; \text{sg}\big(p_\phi(\hat{z}_t|h_t)\big) \Big), \end{aligned} \tag{8}$$

where sg is the stop-gradient operator.

The surrogate value loss, $\mathcal{L}_{\hat{Q}}$, is a mean-squared error term that trains the value function to estimate the target Q_{target}. We simply apply a Monte Carlo (MC) estimator here to get Q_{target}, and we empirically found that using an MC estimator yields more stable results in our scenario than TD-error:

$$\mathcal{L}_{\hat{Q}} = \mathbb{E}_{\tau \sim \mathcal{D}} \sum_t \mathbb{E}_{z_t \sim q_\phi(z_t|h_t,o_t)} \left[\left(Q_\phi(h_t, z_t, a_t) - Q_{\text{target}} \right)^2 \right]. \tag{9}$$

D. Value-Guided Sampling MPC

In practical applications involving complex robotics and perception, learning a perfect observation-action value function and greedily maximizing it remains a significant challenge. This difficulty stems from two primary sources:

1) **Challenges in Offline Learning:** Finite offline datasets provide incomplete coverage, leading to unreliable value estimates for out-of-distribution actions.
2) **Partial Observability and Physical Non-idealities:** Robotic systems suffer from partial observability, as the full state, such as contact forces, is not directly measured and is subject to sensor noise and action delays, both of which degrade value estimation.

To address this, we introduce a Value-Guided Sampling MPC framework. This approach explicitly treats the learned value function not as an optimal oracle, but as a powerful, albeit imperfect, heuristic to guide a robust, receding-horizon planning process.

Consider a standard MDP with state s_t and action a_t for the analysis of the variance reduction of our proposed method. When the perfect Q is available, we can leverage the Bellman principle to obtain the optimal control policy:

$$\pi(s_t) = \arg\max_{a_t} Q(s_t, a_t). \tag{10}$$

However, in practice, only the estimation $\hat{Q}$ is available, which may have a large variance. The imperfection of $\hat{Q}$ may lead to performance degradation. To mitigate this issue, we consider an N-step surrogate optimization:

$$\begin{aligned} \hat{\pi}(s_t) = \arg \max_{\{a_{t:t+N-1}\}} \hat{J}_N := \frac{1}{N} \sum_{k=0}^{N-1} \hat{Q}(s_{t+k}, a_{t+k}) \\ \text{s.t.} \quad s_{t+k+1} = f_\phi(s_{t+k}, a_{t+k}) \end{aligned} \tag{11}$$

By the definition of the Q function, $\hat{\pi}(\cdot)$ and π are obtained by solving two different optimal control problems, thus they

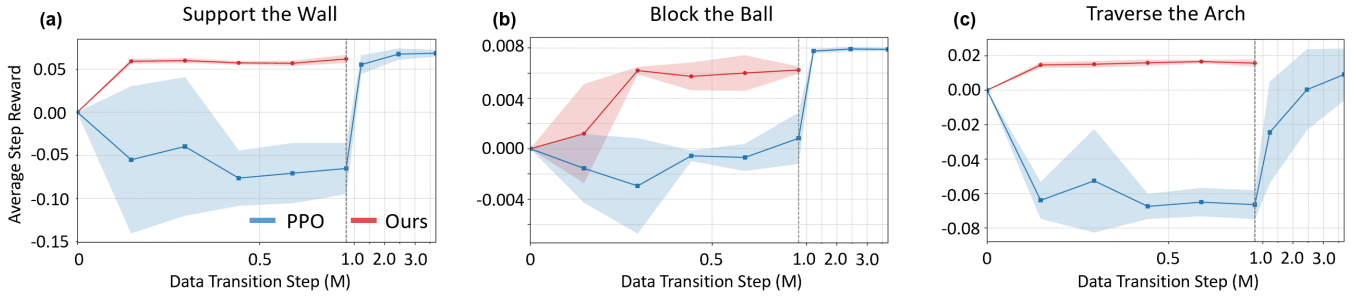

Fig. 5: Sample efficiency comparison: Our method uses an offline dataset collected from random actions, while PPO collects data from environments at every iteration. The x-axis represents the number of step transitions used, while the y-axis shows the reward for each task. A greater value on the x-axis indicates a larger amount of data used, and a higher value on the y-axis signifies better performance. While our method utilizes a dataset of at most 1M steps, we continued to train PPO for a greater number of steps to determine when it could achieve comparable performance.

are in general not identical. Despite the biases, we show that the surrogate objective can reduce the variance.

Consider the residual ϵ_{t+k} at time $(t+k)$ as:

$$\epsilon_k := \hat{Q}(s_{t+k}, a_{t+k}) - Q(s_{t+k}, a_{t+k}). \tag{12}$$

For shorthand notation, we use the subscript (k) to denote the quantity at time step $t+k$. Consider the surrogate objective:

$$\frac{1}{N}\sum_{k=0}^{N-1}\hat{Q}_k = \frac{1}{N}\sum_{k=0}^{N-1}(Q_k + \epsilon_k). \tag{13}$$

Let $J_N := \frac{1}{N}\sum_{k=0}^{N-1} Q_k$. As we use Monte Carlo returns to form Q_{target} and train $\hat{Q}$ via regression, we have:

$$\begin{aligned}\mathbb{E}[(J_N - \hat{J}_N)^2] &= \mathbb{E}[(\frac{1}{N}\sum_{k=0}^{N-1}\epsilon_k)^2] \\ &= \frac{1}{N^2}\sum_{(i,j)}\mathrm{Cov}[\epsilon_i, \epsilon_j] + \bar{\epsilon}^2.\end{aligned} \tag{14}$$

with $\bar{\epsilon} := \mathbb{E}[\epsilon_k]$ denoting the bias introduced by estimating Q_k corresponding to the infinite horizon MDP using truncated trajectories. We assume that the variance of $\epsilon_k, \forall k$, satisfies:

$$0 \le \sigma_{\min} \le \sqrt{\mathrm{Var}[\epsilon_k]} \le \sigma_{\max}, \qquad \text{(Bounded Variance)}$$

and the correlation is bounded by $\rho < 1$, such that $\forall i, j$

$$|\mathrm{Cov}[\epsilon_i, \epsilon_j]| \le \rho\sqrt{\mathrm{Var}[\epsilon_i]\,\mathrm{Var}[\epsilon_j]}. \quad \text{(Bounded Correlation)}$$

The upper bound on the variance term is achieved if all steps are positively correlated:

$$\begin{aligned}\mathbb{E}[(J_N - \hat{J}_N)^2] - \bar{\epsilon}^2 &\le \frac{N + \rho(N^2 - N)}{N^2}\sigma_{\max}^2 \\ &=: V_{\mathrm{ub}}.\end{aligned} \tag{15}$$

For the lower bound, similarly, we have:

$$\begin{aligned}\mathbb{E}[(J_N - \hat{J}_N)^2] - \bar{\epsilon}^2 &\ge \max\{\frac{N\sigma_{\min}^2 - \rho(N^2 - N)\sigma_{\max}^2}{N^2}, 0\} \\ &=: V_{\mathrm{lb}}.\end{aligned} \tag{16}$$

Thus, we have the limit when $\rho \to 0 : V_{\mathrm{ub}} \to \frac{\sigma_{\max}^2}{N}, V_{\mathrm{lb}} \to \frac{\sigma_{\min}^2}{N}$. As we compute $\hat{Q}$ from an offline dataset generated by random actions in each step, the correlation between $\hat{Q}_i$ from different time steps is weaker than by sampling using a state feedback policy. Thus, it is possible that ρ is small and $\hat{J}_N$ has a substantially lower variance than $\hat{Q}$. Although the surrogate objective does not preserve a local optimum, if the variance of $\hat{Q}$ dominates, this strategy can significantly improve the performance.

Based on the above analysis, we apply the surrogate **objective function** $\hat{J}_N$ from (11) using the latent h_t and $\hat{z}_t$ as the representation of the robot states. The optimal sequence, denoted $A_t^* = \{a_t^*, a_{t+1}^*, ..., a_{t+N-1}^*\}$, is the one that maximizes our surrogate objective $\hat{J}_N$:

$$\begin{aligned}A_t^* = \arg\max_{A_t}\ & \frac{1}{N}\sum_{k=0}^{N-1} Q_\phi(h_{t+k}, \hat{z}_{t+k}, a_{t+k}), \\ \text{s.t.}\quad & h_{t+k+1} = f_\phi(h_{t+k}, \hat{z}_{t+k}, a_{t+k}), \\ & \hat{z}_{t+k} \sim p_\phi(\hat{z}_{t+k} \mid h_{t+k}).\end{aligned} \tag{17}$$

As shown in Fig. 3, the framework's memory is maintained in two latent states: a dynamics latent state h_t and a current observation latent state z_t. The h_t is computed by the RNN from the previous latent (h_{t-1}, z_{t-1}) and the last action (a_{t-1}). The z_t is then generated in one of two ways: when an observation o_t is available, z_t is inferred using both the latent state h_t and the observation o_t; for future predictions, it is generated from the latent state h_t alone. Our world model also predicts the probability of robot failure, $\hat{d}_t$, such as falling; if this probability exceeds a threshold of 0.9, all subsequent value estimates for that trajectory are set to zero.

We use the Cross-Entropy Method (CEM) to find the optimal action sequence A_t^*. Once identified, only the first action a_t^* is executed. This iterative re-planning allows the robot to continuously incorporate feedback from the environment, enabling it to react to disturbances and correct for model inaccuracies in real-time. From Table I, we select a planning horizon $N = 4$ as our default, as it provides the best overall performance across all tasks.

IV. Experiments

Our experimental platform is the Unitree G1 humanoid robot, equipped with a RealSense D435i camera. All quantitative analyses, including ablation studies and baseline comparisons, are conducted in a controlled simulation environment. All comparisons are conducted using a consistent set of training epochs and hyperparameters. The mean and

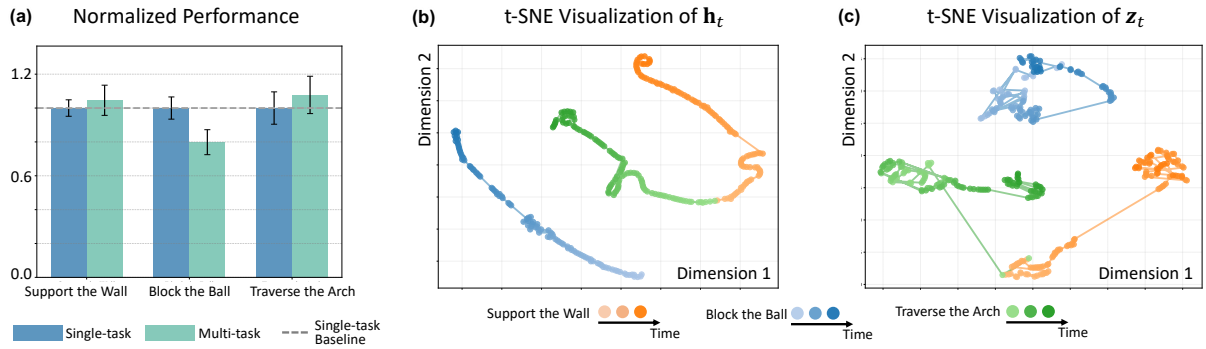

Fig. 6: Multi-task performance and latent space visualization. (a) A single model trained jointly on all tasks achieves comparable normalized performance to specialized single-task models. (b-c) The t-SNE visualizations reveal a clear separation of tasks in the latent spaces. As envisioned by our design, the latent h_t primarily represents dynamics, showing significant evolution over time, while the latent z_t provides a more compressed representation of the current environmental observation.

TABLE I: Single-Task Reward Evaluation of Our Method and Baselines: we analyze the influence of three key design choices on performance: the planning horizon N, the world model training methodology, and the objective function.

METHOD	Reward:Wall ↑	Reward:Ball ↑	Reward:Arch ↑
HORIZON N			
Ours, N=1	0.0557 ± 0.0047	-0.0066 ± 0.0050	-0.0396 ± 0.0121
Ours, N=2	0.0607 ± 0.0023	0.0056 ± 0.0003	0.0154 ± 0.0011
Ours, N=3	0.0611 ± 0.0025	0.0059 ± 0.0003	0.0156 ± 0.0011
Ours, N=4	0.0614 ± 0.0027	$\mathbf{0.0061} \pm 0.0003$	$\mathbf{0.0157} \pm 0.0015$
Ours, N=5	0.0598 ± 0.0049	0.0058 ± 0.0012	0.0144 ± 0.0062
Ours, N=6	$\mathbf{0.0617} \pm 0.0031$	0.0053 ± 0.0020	0.0115 ± 0.0099
WORLD MODEL			
ARWM	0.0609 ± 0.0047	0.0039 ± 0.0033	-0.0018 ± 0.0183
OBJECTIVE FUNCTION			
Rew-MPC	0.0302 ± 0.0204	-0.0033 ± 0.0044	-0.0211 ± 0.0092
TD-MPC	$\mathbf{0.0699} \pm 0.0035$	-0.0016 ± 0.0047	0.0145 ± 0.0005

standard deviation are then computed from ten independent trials across three different random seeds. We subsequently validate our approach with real-time experiments on a physical robot. We designed three core tasks to evaluate the model's ability to plan and execute diverse contact-rich behaviors, including **exploiting contact for stability** and **avoiding contact for safety**.

a) Task: (1) **Support the Wall**: the robot must resist external disturbances by stabilizing itself only through supportive hand contact; (2) **Block the Ball**: the robot must intercept a flying object only with defensive hand contact; (3) **Traverse the Arch**: the robot must pass through a low-clearance arch while avoiding unintended head contact.

b) Baselines: We compare our method with these baseline methods: (1) **PPO**: an implementation in [2]. (2) **ARWM**: replaces our framework with auto-regressive prediction training like [29, 30]. (3) **Rew-MPC**: replaces objective function Eq. (17) with $\sum_{k=0}^{N-1} \gamma^k \hat{r}_k$ from PlaNet [24]. (4) **TD-MPC**: replaces objective function Eq. (17) with $\sum_{k=0}^{N-1} \gamma^k \hat{r}_k + \gamma^N \hat{Q}$ from TD-MPC [23].

A. Advantages of the Use of Offline Data

Sample Efficiency In Single-Task: We first compare our method against PPO implemented in [2], an online on-policy RL algorithm that remains the dominant training method in the legged robotics domain. A key distinction is that PPO requires continuous interaction with the environment, whereas our approach is fully offline, trained from a fixed, demonstration-free dataset without any environment interaction. We do not compare against off-policy methods such as SAC and its variants, which, although more sample-efficient than on-policy approaches, are less commonly applied to humanoid robots in real-world settings due to their limited scalability to complex, high-DoF dynamic control.

As shown in Fig. 5, in three contact-reward-dominant tasks, our method completes the tasks using only 0.5M data steps. In contrast, PPO requires a significantly larger amount of data, especially in simulations that necessitate visual rendering, where it consumes considerably more time. While PPO can quickly match our efficiency in tasks with simple visual features and a stationary robot, such as "Block the Ball," our method achieves better performance on tasks with more complex visual representations and significant changes in the robot's viewpoint. For instance, in the "Traverse the Arch" task, where the robot's perspective changes dramatically between standing and squatting positions, our method substantially outperforms PPO.

Multi-Task Capability: In addition to sampling efficiency on the single task, we qualitatively analyze the challenges of applying online RL like PPO to a multi-task setting:

- **Reward Engineering:** online RL requires either (1) a unified reward function across all tasks, which is difficult to design, or (2) substantial engineering effort to implement complex logic for task switching and conditional rewards.
- **Catastrophic Forgetting & Curriculum Design:** online RL is prone to catastrophic forgetting. Mitigating this requires a carefully designed task sampling curriculum, the complexity of which grows as more tasks are added.

Our approach, which leverages offline data, circumvents these challenges by learning directly from a mixed dataset, enabling effective and scalable multi-task training. The result of our method in the multi-task setting is shown in Sec. IV-C.

B. Analysis of Key Design Choices

To demonstrate the necessity of our design, as shown in the Table I, we found that greedily maximizing the value (i.e.,

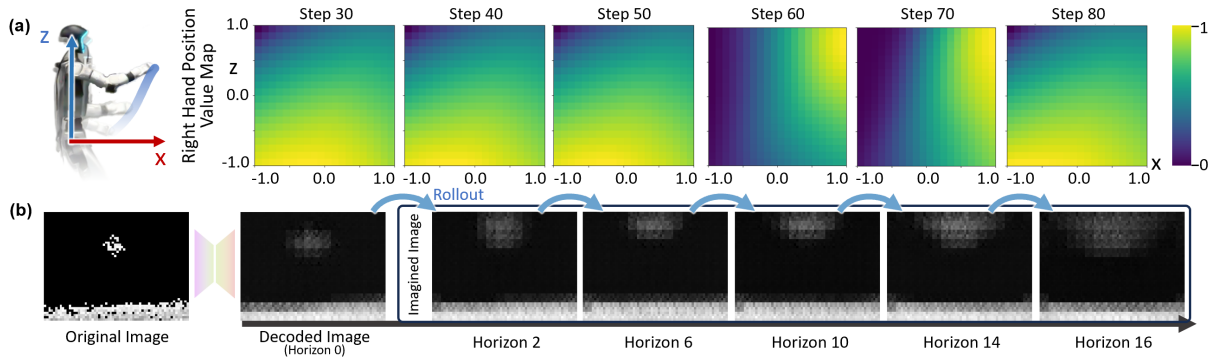

Fig. 7: Visualization of the world model's prediction and planning process during the "Block the Ball" task. (a) The evolution of the Q-value map for the hand's position in the X-Z plane. As the task unfolds (from Step 30 to 80, with the ball appearing at Step 60), the model dynamically updates its value estimates, with the high-value region (yellow) indicating the optimal location to intercept the object. (b) An open-loop prediction of future observations. Given an initial image, the model first reconstructs it (Horizon 0) and then generates a sequence of future frames (Horizon 2-16) by decoding its predicted latent states, visualizing its anticipation of the ball's trajectory. It is worth noting that while our planner uses a shorter **4-step horizon**, this visualization is for demonstrating the long-horizon physical intuition.

the horizon-1 case) is infeasible. This approach causes the robot to be myopic, favoring the maintenance of its default position and ignoring future contacts. We observe that longer horizons (e.g., N=6) degrade performance, likely because bias dominates from longer-term prediction (see Sec. III-D), whereas N=4 strikes a bias–variance sweet spot.

We also find that incorporating autoregressive prediction in our method (the ARWM baseline in Table I) is not necessary and can even be detrimental to value estimation in offline RL, as it overemphasizes precise prediction, which leads to value function overfitting.

As shown in the Table I, Rew-MPC, which uses reward as the objective function, yielded suboptimal results due to partial observability, noise, action lag, and sparse contact, which make rewards difficult to predict. And TD-MPC, which uses TD-target to evaluate trajectories, also produces unstable results. We attribute this to TD-error methods converging to deceptive solutions, where low TD error might mask highly inaccurate value estimates. As argued in [38], this phenomenon is caused by bias cancellation and the existence of infinitely many suboptimal solutions that satisfy the Bellman equation on an incomplete offline dataset.

C. Multi-Task Planning with Unified World Model

To evaluate the multi-task capability, we trained a single model on a combined dataset from all tasks and compared it to the single-task models from Table I. As shown in Fig. 6, the multi-task model achieves improved performance on two of the three tasks, with a minor drop in the "Block the Ball" task, which we attribute to its smaller reward scale.

To understand how this is achieved, we visualized the latent spaces using t-SNE. The visualizations reveal that our model learns to form distinct clusters for each task. The latent state h_t shows significant temporal evolution, confirming that the model learns to encode the unique latent dynamics for each task from the mixed data.

D. Model Interpretation and Visualization on Prediction

We provide visualizations in Fig. 7 that offer insight into the internal decision-making process of our framework. Specifically, we analyze this process on two levels: first, whether our model has learned a genuine understanding of the environment's dynamics, and second, how it leverages this understanding for its decision-making process.

Physical Intuition: To assess whether the learned model captures task-relevant physical properties, rather than collapsing to trivial solutions, we apply long-horizon open-loop rollouts for verification. As shown in Fig. 7 (b), the rollout preserves the parabolic motion of the ball over 16 steps, indicating that the latent states retain physically meaningful structure. Importantly, exhibiting long-horizon physical intuition in open-loop rollouts does not imply that increasing the MPC planning horizon will necessarily improve control performance. In sampling-based MPC, longer horizons typically induce a more challenging optimization landscape and exacerbate model bias, so a short 4-step receding horizon empirically provides a practical sweet spot, and we replan frequently.

Contact-Directed Planning: Fig. 7 (a) visualizes how the model leverages its predictions for planning by showing the evolution of the objective function value in Eq. (11) for the hand's target position. Early in the task (e.g., Step 30), the value map consistently encourages the hand to stay near a natural, energy-efficient default position. However, as the ball approaches and the plan solidifies (e.g., Step 60-70), a high-value region (yellow) emerges and sharpens, decisively guiding the robot's hand toward the optimal contact point. This dynamic evolution of the value map showcases the model's contact-directed reasoning, effectively forming an interpretable plan to achieve its objective.

E. Real-World Validation

We deployed our method for real-time experiments on Unitree G1 with 25 Hz real-time planning and evaluated a batch of 1024 action trajectories with a planning horizon of 4 steps at each timestep. The desired base velocity v was controlled by a human operator.

Our real-world deployments, shown in Fig. 4, included both single-task and multi-task models, both of which proved capable of completing their assigned tasks. The policy also demonstrated the ability to generalize to out-of-distribution

(OOD) scenarios, such as blocking a previously unseen box. These experiments validate that our method can achieve agile and robust vision-based control. Crucially, the learned policy exhibits reactive, context-dependent behavior rather than overfitting to a single action pattern. For instance, in the "Support the Wall" task, the robot only braces its hands against the wall when actively disturbed and returns to a neutral stance once balance is recovered.

V. Conclusion

By integrating a scalable ego-centric visual world model with value-guided sampling-based MPC, we demonstrate that humanoid robots can efficiently and robustly learn agile, contact-rich behaviors from offline, demonstration-free data, advancing data-efficient, vision-based planning for real-world robotic interaction.

ACKNOWLEDGMENT

This work was supported in part by NSF Grant CMMI-1944722 and by the Robotics and AI Institute. M. Ghaffari receives support from AFOSR MURI FA9550-23-1-0400. Z. Li was funded in part by the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government via the Hong Kong Centre for Logistics Robotics. The authors would like to thank Jiaze Cai and Yen-Jen Wang for their help with experiments. We are also grateful to Bike Zhang, Fangchen Liu, Chaoyi Pan, Junfeng Long, and Yiyang Shao for their valuable discussions. This work was done during H. Liu's visit to UC Berkeley.

References

[1] M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

[2] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on robot learning*, PMLR, 2022, pp. 91–100.

[3] F. Jenelten, J. He, F. Farshidian, and M. Hutter, "Dtc: Deep tracking control," *Science Robotics*, vol. 9, no. 86, eadh5401, 2024.

[4] P. Roth, J. Frey, C. Cadena, and M. Hutter, "Learned perceptive forward dynamics model for safe and platform-aware robotic navigation," in *Robotics: Science and Systems Conference*, 2025.

[5] H. Liu, S. Teng, B. Liu, W. Zhang, and M. Ghaffari, "Discrete-time hybrid automata learning: Legged locomotion meets skateboarding," *arXiv preprint arXiv:2503.01842*, 2025.

[6] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A unified mpc framework for whole-body dynamic locomotion and manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4688–4695, 2021.

[7] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.

[8] NVIDIA, *Isaac Sim*, version 5.0.0.

[9] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 11 443–11 450.

[10] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 840–888, 2025.

[11] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Real-world humanoid locomotion with reinforcement learning," *Science Robotics*, vol. 9, no. 89, eadi9579, 2024.

[12] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[13] M. Anitescu, "Optimization-based simulation of nonsmooth rigid multibody dynamics," *Mathematical Programming*, vol. 105, no. 1, pp. 113–143, 2006.

[14] E. R. Westervelt, J. W. Grizzle, and D. E. Koditschek, "Hybrid zero dynamics of planar biped walkers," *IEEE transactions on automatic control*, vol. 48, no. 1, pp. 42–56, 2003.

[15] K. Sreenath, H.-W. Park, and I. Poulakakis, "A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1170–1193, 2011.

[16] H. Xue, C. Pan, Z. Yi, G. Qu, and G. Shi, "Full-order sampling-based mpc for torque-level locomotion control via diffusion-style annealing," *arXiv preprint arXiv:2409.15610*, 2024.

[17] Y. Gong *et al.*, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *2019 American control conference (ACC)*, IEEE, 2019, pp. 4559–4566.

[18] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *The International Journal of Robotics Research*, vol. 43, no. 4, pp. 572–587, 2024.

[19] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning*, PMLR, 2023, pp. 22–31.

[20] C. Zhang, W. Xiao, T. He, and G. Shi, "Wococo: Learning whole-body humanoid control with sequential contacts," *arXiv preprint arXiv:2406.06005*, 2024.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[22] D. Ha and J. Schmidhuber, "Recurrent world models facilitate policy evolution," *Advances in neural information processing systems*, vol. 31, 2018.

[23] N. Hansen, X. Wang, and H. Su, "Temporal difference learning for model predictive control," *arXiv preprint arXiv:2203.04955*, 2022.

[24] D. Hafner *et al.*, "Learning latent dynamics for planning from pixels," in *International conference on machine learning*, PMLR, 2019, pp. 2555–2565.

[25] M. Assran *et al.*, "V-jepa 2: Self-supervised video models enable understanding, prediction and planning," *arXiv preprint arXiv:2506.09985*, 2025.

[26] J. Bruce *et al.*, "Genie: Generative interactive environments," in *Forty-first International Conference on Machine Learning*, 2024.

[27] M. O'Connell *et al.*, "Neural-fly enables rapid learning for agile flight in strong winds," *Science Robotics*, vol. 7, no. 66, 2022.

[28] K. Zhang, B. Li, K. Hauser, and Y. Li, "Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation," in *Proceedings of Robotics: Science and Systems (RSS)*, 2024.

[29] C. Li, A. Krause, and M. Hutter, "Offline robotic world model: Learning robotic policies without a physics simulator," *arXiv preprint arXiv:2504.16680*, 2025.

[30] C. Li, A. Krause, and M. Hutter, "Robotic world model: A neural network simulator for robust policy optimization in robotics," *arXiv preprint arXiv:2501.10100*, 2025.

[31] G. Williams *et al.*, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2017, pp. 1714–1721.

[32] C. Pan, Z. Yi, G. Shi, and G. Qu, "Model-based diffusion for trajectory optimization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 57 914–57 943, 2024.

[33] G. B. Margolis *et al.*, "Learning to jump from pixels," in *5th Annual Conference on Robot Learning*.

[34] W. Xiao, H. Xue, T. Tao, D. Kalaria, J. M. Dolan, and G. Shi, "Anycar to anywhere: Learning universal dynamics model for agile and adaptive mobility," *arXiv preprint arXiv:2409.15783*, 2024.

[35] S. Wandong, *Legged lab: Direct isaaclab workflow for legged robots*, version 1.0.0, 2025.

[36] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap, "Mastering diverse domains through world models," *arXiv preprint arXiv:2301.04104*, 2023.

[37] X. Chen and K. He, "Exploring simple siamese representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 15 750–15 758.

[38] S. Fujimoto, D. Meger, D. Precup, O. Nachum, and S. S. Gu, "Why should i trust you, bellman? the bellman error is a poor replacement for value error," in *International Conference on Machine Learning*, PMLR, 2022, pp. 6918–6943.