NIL: No-data Imitation Learning Supplementary Materials

Mert Albaba[1,2] Chenhao Li[1] Markos Diomataris[1,2] Omid Taheri[2]
Andreas Krause[1] Michael J. Black[2]

[1]ETH Zürich [2]Max Planck Institute for Intelligent Systems
nil.is.tue.mpg.de

1. Additional Experiments

1.1. Video Selection Protocol

We generate three videos from the same initial frame and prompt, and select the best of 3 generations based on optical flow variance and Pixel mean squared error (MSE). To test robustness to video selection, we trained NIL on the *rejected* videos (Vid. 1, 2). Table 1 shows only mild degradation in the performance across all tasks, indicating NIL is robust to imperfect generations.

Table 1. **Robustness to Video Selection:** We evaluate NIL on learning from generated sub-optimal videos, which are rejected by our video selection protocol.

		NIL	Vid. 1	Vid. 2
Unitree H1	Pixel MSE	**114.8**	131.3	167.3
	Flow Var.	0.37	**0.35**	0.55
	Env. Reward	**396.1**	392.5	388.7
Talos	Pixel MSE	**49.9**	85.6	96.1
	Flow Var.	**0.06**	0.15	0.20
	Env. Reward	**352.8**	348.4	312.5
Unitree G1	Pixel MSE	**54.0**	63.6	78.8
	Flow Var.	**0.08**	0.09	0.14
	Env. Reward	356.9	**362.5**	358.4
Unitree A1	Pixel MSE	**283.1**	349.9	394.9
	Flow Var.	**1.07**	1.69	1.90
	Env. Reward	**290.3**	285.8	240.8

1.2. Open-Source Video Diffusion Models

We use Kling AI to generate reference videos in Section 4.4. In order to evaluate the reliance of NIL on proprietary video diffusion models, we train NIL using only videos generated by two open-source video diffusion models: WAN [4] and LTX [2].

Table 2 confirms that NIL is reproducible and performs strongly without proprietary models.

Table 2. **Open Source Models:** We evaluate reliability of NIL with open-source video diffusion models.

	Environment Reward ↑			
	Unitree H1	Talos	Unitree G1	Unitree A1
NIL	**396.1**	352.8	356.9	**290.3**
NIL-WAN	392.1	382.0	381.9	268.5
NIL-LTX	373.7	**393.6**	**394.3**	286.8
Expert	400	400	400	300

1.3. Sensitivity to Camera

To evaluate NIL's robustness to different camera settings, we train NIL with following camera settings: static, 45° azimuth, and field-of-view (FoV) jitter (5%/15%, applied during simulation rendering). We also test a multi-view setup where rewards are averaged across two views (follow and 45° azimuth).

Table 3 shows NIL remains strong with different camera settings and mild perturbations, and multi-view setup even improves the performance of NIL on Unitree G1.

Table 3. **Camera Sensitivity**: We train NIL with different camera settings to evaluate sensitivity to a specific camera setting.

	Default (Follow)	Static	45°	5% FoV	15% FoV	Multi View
H1	**396.1**	393.7	394.4	395.8	368.2	393.1
G1	356.9	396.8	394.8	360.1	374.5	**397.4**

1.4. Robustness to Frame Interpolation

For temporal alignment between simulation renderings (100FPS) and generated videos (24FPS), we upsample generated videos 4x (Section 3.4). We evaluate NIL's dependence to frame interpolation and we test NIL *without* up-

sampling, instead using raw generated videos by *downsampling* simulator renderings to 25FPS (taking 1 frame per 4 steps). We also test NIL with upsampling generated videos 2x and downsampling simulator renderings to 50FPS (1 frame per 2 steps).
Table 4 shows that NIL's performance is stable without upsampling generated videos.

Table 4. **Robustness to Frame Interpolation**: We evaluate NIL's dependence to frame interpolation and train NIL with different interpolation settings.

	H1	Talos	G1	A1
NIL	396.1	352.8	356.9	290.3
Raw (24 FPS)	392.5	349.2	382.1	296.0
Upsampled 2x	388.2	352.9	369.3	291.5

2. Extended Tables

We report standard errors for Table 2 and Figure 6-d in this section.

2.1. Continuous Control of Various Robots

Table 5. Mean and standard errors in continuous control tasks.

	Environment Reward ↑			
	Unitree H1	Talos	Unitree G1	Unitree A1
NIL (ours)	396.1 ± 4.1	352.8 ± 40.2	356.9 ± 40.9	290.3 ± 6.5
AMP	393.5 ± 8.0	231.1 ± 43.4	393.4 ± 17.6	286.9 ± 9.7
GAIfO	347.8 ± 12.3	204.4 ± 51.2	353.1 ± 4.4	260.8 ± 6.7
BCO	72.0 ± 15.0	26.6 ± 2.4	21.2 ± 4.8	30.3 ± 6.7
Expert	400	400	400	300

2.2. Loco-Manipulation

Table 6. Scores and standard errors in Loco-Manipulation tasks.

	Sit	Hang	Balance
NIL	169.5 ± 1.92	89.7 ± 0.04	172.0 ± 1.79
RL	170.3 ± 1.48	89.8 ± 0.02	168.3 ± 0.85
Maximum	200	100	200

3. Experimental Details

In all experiments, a penalty of -200 is imposed if the agent falls before reaching maximum environment steps. This early termination penalty ensures that agents are incentivized to maintain stable and natural motion.

3.1. Continuous Control of Various Robots

We use default termination conditions defined in LocoMujoco [1]. For each task, training is limited to 400 timesteps (300 timesteps for Unitree A1). The maximum score in each environment (Unitree H1, Talos, Unitree G1) is 400 (300 for Unitree A1). Therefore, we refer to this upper limit as the expert score in Table 2.
The expert data used to train upper baselines is comprised of motion capture trajectories from the benchmark of the tasks we use: LocoMujoco [1]. The motion capture data available in this benchmark is re-targetted to specific robot embodiments, considering their unique body parameters.

3.2. Loco-Manipulation

We use default termination conditions defined in HumanoidBench [3]. For each task, training is limited to 200 timesteps. For *Sit* and *Balance* tasks, the maximum reward is 200. For the *Highbar* task, the maximum reward is 100. The *Highbar* task in our work expects agent to hang from the bar and stay still, which is different from HumanoidBench [3], where the agent is expected to move its feet above the bar and stand upside down in that pose. For Figure 6-d, we normalize scores by maximum scores.

4. Regularization Rewards

In addition to the imitation rewards, we incorporate regularization terms to ensure that the learned behavior is smooth and physically plausible. These regularization rewards penalize excessive or unrealistic control commands and deviations from desired configurations. In the following, we describe each component in detail.

4.1. Continuous Control of Various Robots

4.1.1. Angular Velocity Penalty (Lateral)

To discourage excessive lateral angular motion, we penalize the squared angular velocities about the x- and y-axes. Let ω_x and ω_y denote the x and y components of the body's angular velocity. The penalty is computed as:

$$r_{\text{ang}} = -0.05\Big(\omega_x^2 + \omega_y^2\Big). \tag{1}$$

This term limits undesired rotations that can lead to unstable behavior. For the quadruped (Unitree A1), we instead penalize deviations from a flat base orientation using the squared roll (ϕ) and pitch (θ) angles:

$$r_{\text{ang}} = -2.5\Big(\phi^2 + \theta^2\Big). \tag{2}$$

4.1.2. Joint Torques Penalty

To discourage aggressive actuation, the penalty for joint torques is defined as the negative L2 norm of the joint torques:

$$r_{\text{jt}} = -1.0 \times 10^{-5} \sum_j \tau_j^2, \tag{3}$$

where τ_j represents the torque applied at joint j.

4.1.3. Joint Acceleration Penalty

To promote smooth control, we penalize high joint accelerations. Let $q_{\text{acc},i}$ denote the acceleration of joint i. The corresponding penalty is:

$$r_{\text{ja}} = -2.5 \times 10^{-7} \sum_i q_{\text{acc},i}^2. \tag{4}$$

4.1.4. Action Rate Penalty

Large changes between consecutive actions are discouraged by penalizing the squared difference between the current action a_t and the previous action a_{t-1}:

$$r_{\text{ar}} = -0.01 \left\| a_t - a_{t-1} \right\|_2^2. \tag{5}$$

4.1.5. Angular Velocity Z Penalty

We also regulate the rotation around the z-axis by comparing the measured angular velocity ω_z with a desired command (set to 0.0). With a standard deviation $\sigma = 0.5$, the penalty is given by:

$$r_{\text{ang\_z}} = \exp\Big(-\frac{(0.0 - \omega_z)^2}{\sigma^2}\Big). \tag{6}$$

This term rewards the agent for maintaining a near-zero angular velocity about the z-axis.

4.1.6. Joint Deviation Penalty (Hip, Ankle and Torso)

To keep the joints close to a nominal configuration, we penalize deviations from default angles. For the hip joints the penalty is:

$$r_{\text{hip\_dev}} = -0.2 \sum_{j \in \text{hip joints}} \left| q_{\text{pos},j} - q_{\text{default},j} \right|, \tag{7}$$

for the torso joint:

$$r_{\text{torso\_dev}} = -0.1 \left| q_{\text{pos}} - q_{\text{default}} \right|. \tag{8}$$

and for the ankle joints we penalize violations of the mechanical joint limits:

$$r_{\text{ankle}} = -\sum_{j \in \text{ankles}} \left[q_{\min,j} - q_j \right]^+ + \left[q_j - q_{\max,j} \right]^+, \tag{9}$$

where $[\cdot]^+ = \max(\cdot, 0)$ and $[q_{\min,j},\, q_{\max,j}]$ is the admissible range for each ankle joint.
The total joint deviation penalty is:

$$r_{\text{joint\_dev}} = r_{\text{hip\_dev}} + r_{\text{torso\_dev}} + r_{\text{ankle}}. \tag{10}$$

4.1.7. Total Regularization Penalty

The overall regularization penalty applied at each time step is the sum of all individual penalties:

$$\mathcal{P}_t = r_{\text{ang\_xy}} + r_{\text{jt}} + r_{\text{ja}} + r_{\text{ar}} + r_{\text{ang\_z}} + r_{\text{joint\_dev}} \tag{11}$$

This aggregated term is incorporated into the overall reward function to promote smooth, efficient, and physically realistic motion.

4.2. Loco-Manipulation

For the loco-manipulation tasks (Sit, Balance, and Hang), we use the following terms defined in HumanoidBench [3]:

- $\mathbf{r}_{\text{upright}}$: Rewards the upright posture.
- $\mathbf{r}_{\text{stand}}$: Rewards maintaining the head height above a threshold.
- $\mathbf{r}_{\text{effort}}$: Penalizes control effort (denoted as e in [3]).
- $\mathbf{r}_{\text{still}}$: Rewards stationarity.
- $\mathbf{r}_{\text{posture}}$: Rewards proper neck posture.

Using these primitives, we define the regularization $\mathcal{P}_t$ for each task as follows for NIL.

4.2.1. Sit

The Sit task regularization prioritizes energy efficiency and upright posture:

$$\mathcal{P}_{\text{sit}} = r_{\text{effort}} + 0.1 \cdot (r_{\text{upright}} \cdot r_{\text{posture}}) + 0.1 \cdot r_{\text{still}}. \tag{12}$$

4.2.2. Balance

The Balance task regularization encourages the agent to spend minimal control effort and stay upright:

$$\mathcal{P}_{\text{balance}} = r_{\text{stand}} + r_{\text{upright}} + r_{\text{effort}}. \tag{13}$$

4.2.3. Hang (Highbar)

The original HumanoidBench high-bar task reward encourages inverted postures. We replace it with a regularization function that rewards upright torso orientation and penalizes excessive actuator effort:

$$\mathcal{P}_{\text{hang}} = r_{\text{upright}} + r_{\text{effort}}. \tag{14}$$

References

[1] Firas Al-Hafez, Guoping Zhao, Jan Peters, and Davide Tateo. Locomujoco: A comprehensive imitation learning benchmark for locomotion. *arXiv preprint arXiv:2311.02496*, 2023. 2

[2] Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. Ltx-2: Efficient joint audio-visual foundation model. *arXiv preprint arXiv:2601.03233*, 2026. 1

[3] Carmelo Sferrazza, Dun-Ming Huang, Xingyu Lin, Youngwoon Lee, and Pieter Abbeel. Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation. *arXiv preprint arXiv:2403.10506*, 2024. 2, 3

[4] Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 1