# Real-World Humanoid Locomotion with Reinforcement Learning

Ilija Radosavovic*, Tete Xiao*, Bike Zhang*,
Trevor Darrell†, Jitendra Malik†, Koushil Sreenath†

University of California, Berkeley

Project Page

**Humanoid robots that can autonomously operate in diverse environments have the potential to help address labour shortages in factories, assist elderly at homes, and colonize new planets. While classical controllers for humanoid robots have shown impressive results in a number of settings, they are challenging to generalize and adapt to new environments. Here, we present a fully learning-based approach for real-world humanoid locomotion. Our controller is a causal transformer that takes the history of proprioceptive observations and actions as input and predicts the next action. We hypothesize that the observation-action history contains useful information about the world that a powerful transformer model can use to adapt its behavior in-context, without updating its weights. We train our model with large-scale model-free reinforcement learning on an ensemble of randomized environments in simulation and deploy it to the real world zero-shot. Our controller can walk over various outdoor terrains, is robust to external disturbances, and can adapt in context.**

# Introduction

The dream of robotics has always been that of general purpose machines that can perform many tasks in diverse, unstructured environments. Examples include moving boxes, changing tires, ironing shirts, and baking cakes. This grand goal calls for a general purpose embodiment and a general purpose controller. A humanoid robot could, in principle, deliver on this goal.

Indeed, roboticists designed the first full-sized real-world humanoid robot (*1*) in the 1970s. Since then, researchers have developed a variety of humanoid robots to push the limits of robot locomotion research (*2–5*). The control problem, however, remains a considerable challenge. While classical control methods can achieve stable and robust locomotion (*6–9*), optimization-based strategies have shown the advantage of simultaneously authoring dynamic behaviors and obeying constraints (*10–12*). The most well-known are the examples of the Boston Dynamics Atlas robot doing back flips, jumping over obstacles, and dancing.

While these approaches have made great progress, learning-based methods have become of increasing interest due to their ability to learn from diverse simulations or real environments. For example, learning-based approaches have proven very effective in dexterous manipulation (*13–15*), quadrupedal locomotion (*16–18*), and bipedal locomotion (*19–23*). Moreover, learning-based approaches have been explored for small-sized humanoids (*24, 25*) and combined with model-based controllers for full-sized humanoids (*26, 27*) as well.

In this paper, we propose an end-to-end learning-based approach for full-sized humanoid locomotion (Figure 1). We present a transformer-based controller that predicts future actions autoregressively from the history of past observations and actions (Figure 7). Our model is trained with large-scale reinforcement learning (order of 10 Billion samples) on an ensemble of randomized environments in simulation and deployed to the real world in a zero-shot fashion.

Our approach falls in the general family of techniques for sim-to-real transfer with domain randomization (*28–31*). Among these, the recent approaches for learning legged locomotion have employed either memory-based networks like Long Short-Term Memory (LSTM) (*14, 23*) or trained an explicit estimator to regress environment properties from Temporal Convolutional Network (TCN) features (*17, 18*).

We hypothesize that the history of observations and actions implicitly encodes the information about the world that a powerful transformer model can use to adapt its behavior dynamically at test time. For example, the model can use the history of desired vs actual states to figure out how to adjust its actions to better achieve future states. This can be seen as a form of in-context learning often found in large transformer models like GPT-3 (*32*).

We evaluate our model on a full-sized humanoid robot through a series of real-world and simulated experiments. We show that our policy enables reliable outdoor walking without falls (Figure 1), is robust to external disturbances, can traverse different terrains, and carry payloads of varying mass (Figure 2A-C). Moreover, we find that our approach compares favorably to the state-of-the-art model-based controller (Figure 2D). Our policy exhibits natural walking behaviors, including following different commands (Figure 3), high-speed locomotion, and an emergent arm swing motion (Figure 4). Importantly, our policy is adaptive and can change its behavior based on context, including gradual gait changes based on slowly varying terrains (Figure 5) and rapid adaptation to sudden obstacles (Figure 6). To understand different design choices, we analyze our method in controlled experiments and find that the transformer architecture outperforms other neural network architectures, the model benefits from larger context, and that joint training with teacher imitation and reinforcement learning is beneficial (Figure 8).

Our results suggest that simple and general learning-based controllers are capable of complex, high-dimensional humanoid control in the physical world. We hope that our work may encourage future exploration of scalable learning-based approaches for humanoid robotics.

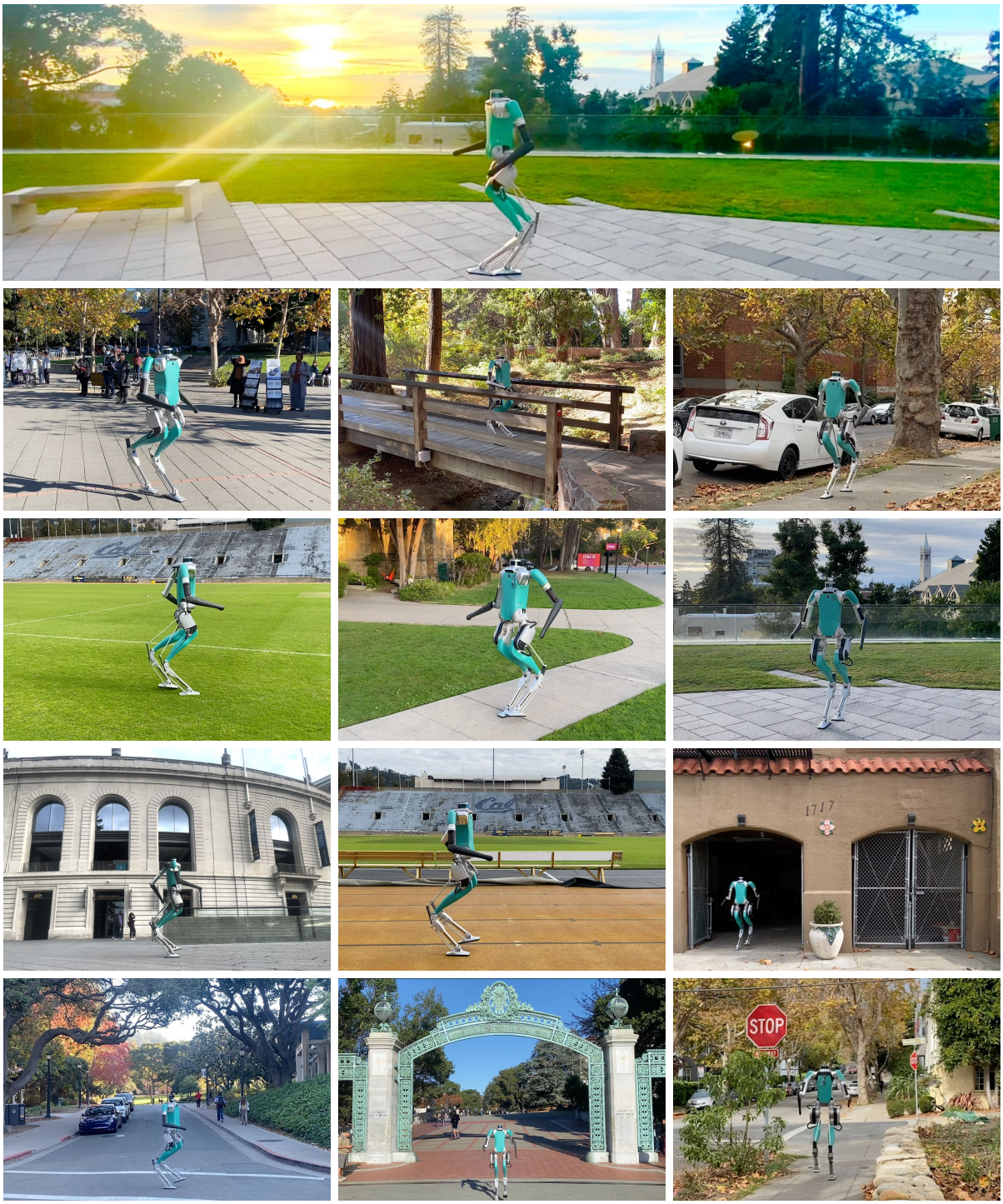


Figure 1: **Deployment to outdoor environments.** We deploy our model to a number of outdoor environments. Example videos are shown in Movie 1. We find that our controller is able to traverse a range of everyday environments including plazas, side walks, tracks, and grass fields.

# Results

**Digit humanoid robot.** Digit is a general-purpose humanoid robot developed by Agility Robotics, standing at approximately 1.6 meters tall with a total weight of 45 kilograms. The robot's floating-base model is equipped with 30 degrees of freedom, including four actuated joints in each arm and eight joints in each leg, of which six are actuated. The passive joints, the shin and tarsus, are designed to be connected through the use of leaf springs and a four-bar linkage mechanism, while the toe joint is actuated by means of rods attached at the tarsus joint. Digit robot has been used as a humanoid platform for mechanical design (*33*), locomotion control (*27, 34, 35*), state estimation (*36*), planning (*37–39*), etc.

## Outdoor deployment

We begin by reporting the results of deploying our controller to a number of outdoor environments. Examples are shown in Figure 1 and Movie 1. These include everyday human environments, plazas, walkways, sidewalks, running tracks, and grass fields. The terrains vary considerably in terms of material properties, like concrete, rubber, and grass, as well as conditions, like dry in a sunny afternoon or wet in the early morning. Our controller is trained entirely in simulation and deployed to the real world zero-shot. The terrain properties found in the outdoor environments were not encountered during training. We found that our controller was able to walk over all of the tested terrains reliably and were comfortable deploying it without a safety gantry. Indeed, over the course of one week of full-day testing in outdoor environments we did not observe any falls. Nevertheless, since our controller acts based on the history of observations and actions and does not include any additional sensors like cameras, it can bump and get trapped by obstacles like steps, but manage to adapt its behavior to avoid falling (see Section 6 for additional discussion and analysis of adaptation).

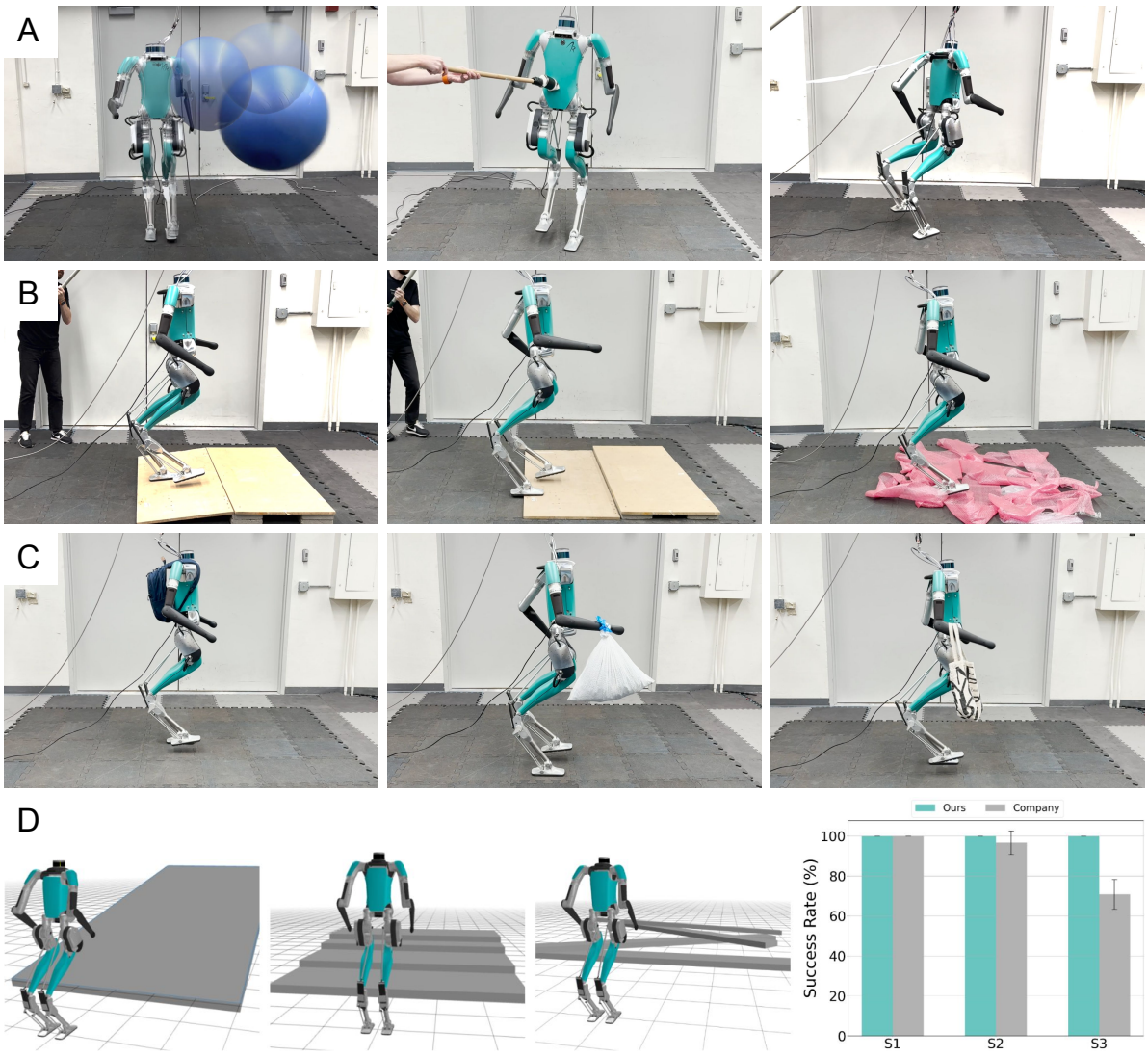


Figure 2: **Indoor and simulation experiments.** We test the robustness of our controller to (**A**) external disturbances, (**B**) different terrains, and (**C**) payloads. Videos are shown in Movie 7. We find that our controller is able to tackle of the scenarios successfully, including those that are considerably out of the training distribution. (**D**) We find that our controller outperforms the state-of-the-art company controller across three different settings in simulation. The gains are larger for harder terrains, like steps and unstable ground. We replicate a subset of the scenarios on hardware and observe consistent behaviors, which can be seen in examples from Movie 2.

## Indoor and simulation experiments

We conduct a series of experiments in the laboratory environment to test the performance of the proposed approach in controlled settings (Figure 2).

**External forces.** Robustness to external forces is a critical requirement for real-world deployment of humanoid robots. We test if our controller is able to handle sudden external forces while walking. These experiments include throwing a large yoga ball at the robot, pushing the robot with a wooden stick, and pulling the robot from the back while it is walking forward (Figure 2A). We find that our controller is able to stabilize the robot in each of these scenarios. Given that the humanoid is a highly unstable system and that the disturbances we apply are sudden, the robot must react in fractions of a second and adjust its actions to avoid falling.

**Rough terrain.** In addition to handling external disturbances, a humanoid robot must also be able to locomote over different terrains. To assess the capabilities of our controller in this regard, we conduct a series of experiments on different terrains in the laboratory (Figure 2B). Each experiment involved commanding the robot to walk forward at a constant velocity of 0.15 m/s. Next, we covered the floor with four different types of items: rubbers, cloths, cables, and bubble wraps, which altered the roughness of the terrain and could potentially lead to challenging entanglement and slipping situations, as the robot does not utilize exteroceptive sensing. Despite these impediments, our controller was able to traverse all these terrain types. Finally, we evaluated the controller's performance on two different slopes. Our simulations during training time included slopes up to 10% grade, and our testing slopes are up to 8.7% grade. Our results demonstrate that the robot was able to successfully traverse both slopes, with more robustness at higher velocity (0.2 m/s) on steeper slopes.

**Payloads.** Next, we evaluate the robot's ability to carry loads of varying mass, shape, and center-of-mass while walking forward (Figure 2C). We conduct five experiments, each with the robot carrying a different type of load: an empty backpack, a loaded backpack, a cloth handbag, a loaded trash bag, and a paper bag. Our results demonstrate that the robot is able to successfully complete its walking route while carrying each of these loads. Notably, our learning-based controller is able to adapt to the presence of a loaded trash bag attached to its arm, despite the reliance of our policy on arm swing movements for balancing. This suggests that our controller is able to adapt its behavior according to the context.

**Comparison to the state of the art.** We compare our controller to the company controller provided by Agility Robotics, which is the state of the art for this robot. To quantify the performance across many runs, we use the high-fidelity simulator by Agility Robotics. We consider three different scenarios: walking over slopes, steps, and unstable ground (Figure 2D). We command the robot to walk forward and consider a trial as successful if the robot crosses the terrain without falling. Crossing a portion of the terrain obtains partial success. We report the mean success rate with 95% CI per terrain across 10 runs (Figure 2D). We find that both ours and the company controller work well on slopes. Next, we see that our controller outperforms the company controller on steps. The company controller struggles to correct itself from foot trapping and shuts off. We replicated this scenario in the real world and have observed consistent behavior, shown in Movie 2. In contrast, our controller is able to recover successfully. Note that our controller was *not* trained on steps in simulation and that the foot-trapping recovery behaviors are emergent (see also Section 6). Finally, we compare the two controllers on a terrain with unstable planks. This setting is challenging as the terrain can dislodge under the robot feet. We find that our controller considerably outperforms the company controller. We did not evaluate the controllers on this terrain in the real world due to concerns for potential hardware damage.

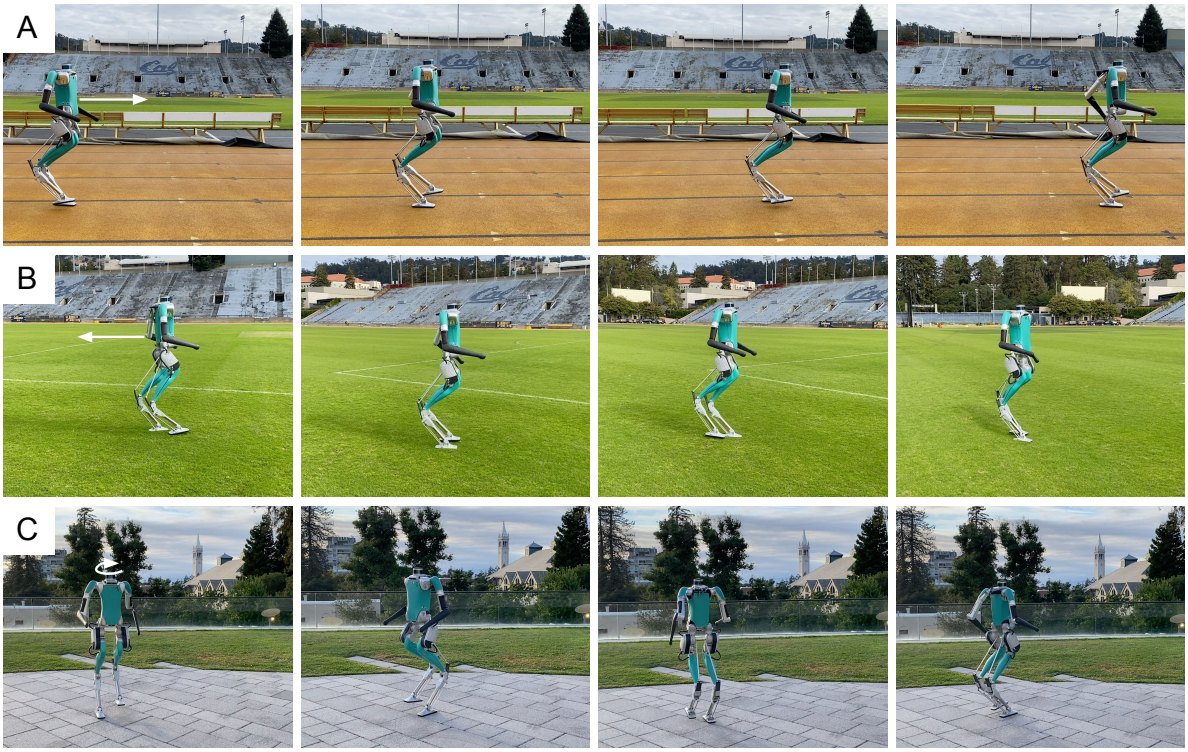


Figure 3: **Omnidirectional walking.** Our learning-based controller is able to accurately follow a range of velocity commands to perform omni-directional locomotion, including (**A**) walking forward, (**B**) backward, and (**C**) turning. Video examples are shown in Movie 3.

## Natural walking

**Omnidirectional walking.** Our controller performs omnidirectional locomotion by following velocity commands. Specifically, it is conditioned on linear velocity on the x-axis, linear velocity on the y-axis, and angular velocity around the z-axis. At training time, we sample commands randomly every 10 seconds (see the Appendix for details). At deployment, we find that our controller is able to follow commands accurately. In addition, it generalizes to continuously changing commands, supplied via a joystick in real time, which is different from training. We show examples of walking forward, backward, and turning in Figure 3, and in Movie 3.

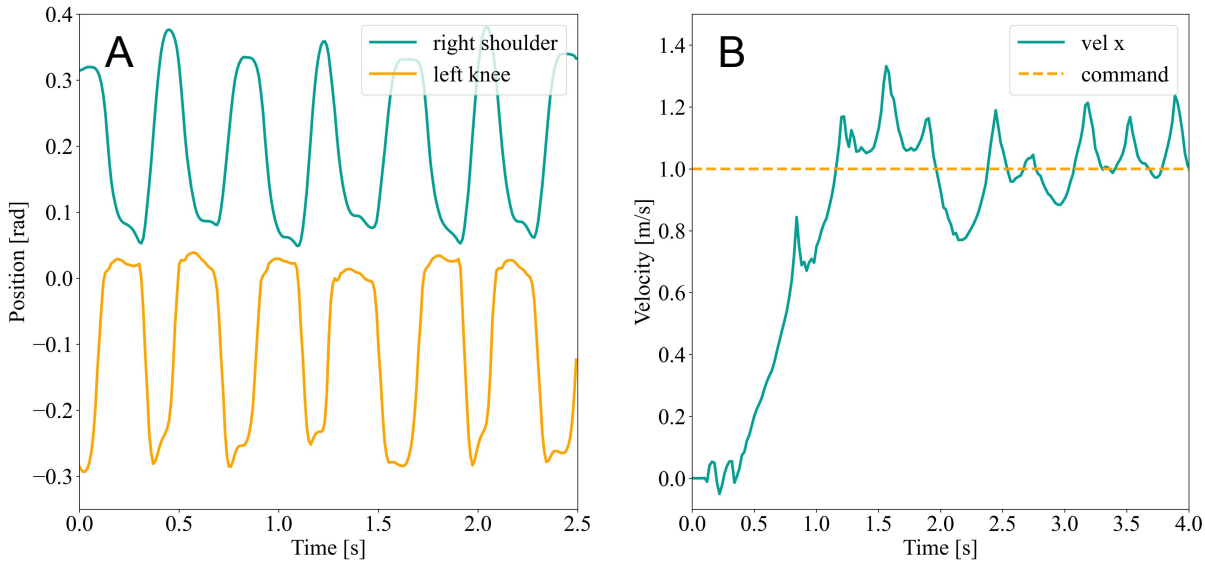


Figure 4: **Arm swing and fast walking.** (**A**) The learned humanoid locomotion in our experiments exhibits human-like arm swing behaviors in coordination with leg movements, i.e., a contralateral relationship between the arms and the legs. (**B**) Our controller is able to perform fast walking on hardware. The video is shown in Movie 4.

**Dynamic arm swing.** A distinct feature of natural human walking is the arm swing. Studying the arm-swing behavior in humans has a very long history in biomechanics (*40–42*). There are a number of existing hypothesis for why humans might be swinging their arms while walking. Examples include that arm-swinging leads to dynamic stability (*43*), that it reduces a metabolic energy cost of walking (*44*), and that it is an ancestral trait conserved from quadrupedal coordination (*45*). We are particularly inspired by the work of (*42*), which suggests that arm swinging may require little effort while providing substantial energy benefit. We test this hypothesis empirically during multiple types of arm motion including swinging, arms bound or held to the body, and arms swinging with phase opposite to normal.

When training our neural network controller, we do not impose explicit constraints on the arm swing motion in the reward function or use any reference trajectories to guide the arm motions. Interestingly, we observe that our trained policy exhibits an emergent arm swing behavior

similar to natural human walking, as shown in Figure 4A. The swinging arm is coordinated with the legs like humans. Specifically, when the left leg is lifting up, the right arm swings forward. We note that our reward function includes energy minimization terms which might suggest that the emergent arm swing motion might lead to energy savings in humanoid locomotion as well.

**Fast walking.** There is a considerable difference between walking at low and high speeds. We analyze the performance of our controller when walking fast in the real world. In Figure 4B, we show the velocity tracking performance given a commanded step velocity at 1 m/s. The corresponding video is in Movie 4. We see that the robot is able to achieve the commanded velocity from rest within 1 s and track it accurately for the duration of the course.

## In-context adaptation

**Emergent gait changes based on terrain.** We command the robot to walk forward over a terrain consisting of three sections in order: flat ground, downward slope, and flat ground, shown in Figure 5A. We find that our controller changes its walking behavior entirely based on the terrain. Specifically, it starts by normal walking on flat ground, transitions to using small steps without lifting its legs much on downward slope, and back to normal walking on flat ground again. These behavior changes are emergent and were not pre-specified.

To understand this behavior better, we study the patterns of neural activity of our transformer model over time. First, we look at the responses of individual neurons. We find that certain neurons correlate with gait. Namely, they have high amplitude during walking on flat and low amplitude on the downward slope. Two such neurons are shown in Figure 5B. Moreover, some neurons correlate with terrain types. Their responses are high on flat terrain and low on slope, as shown in Figure 5C. We also analyze the neural responses in aggregate by performing dimensionality reduction. We project the 192-dimensional hidden state from each timestep into

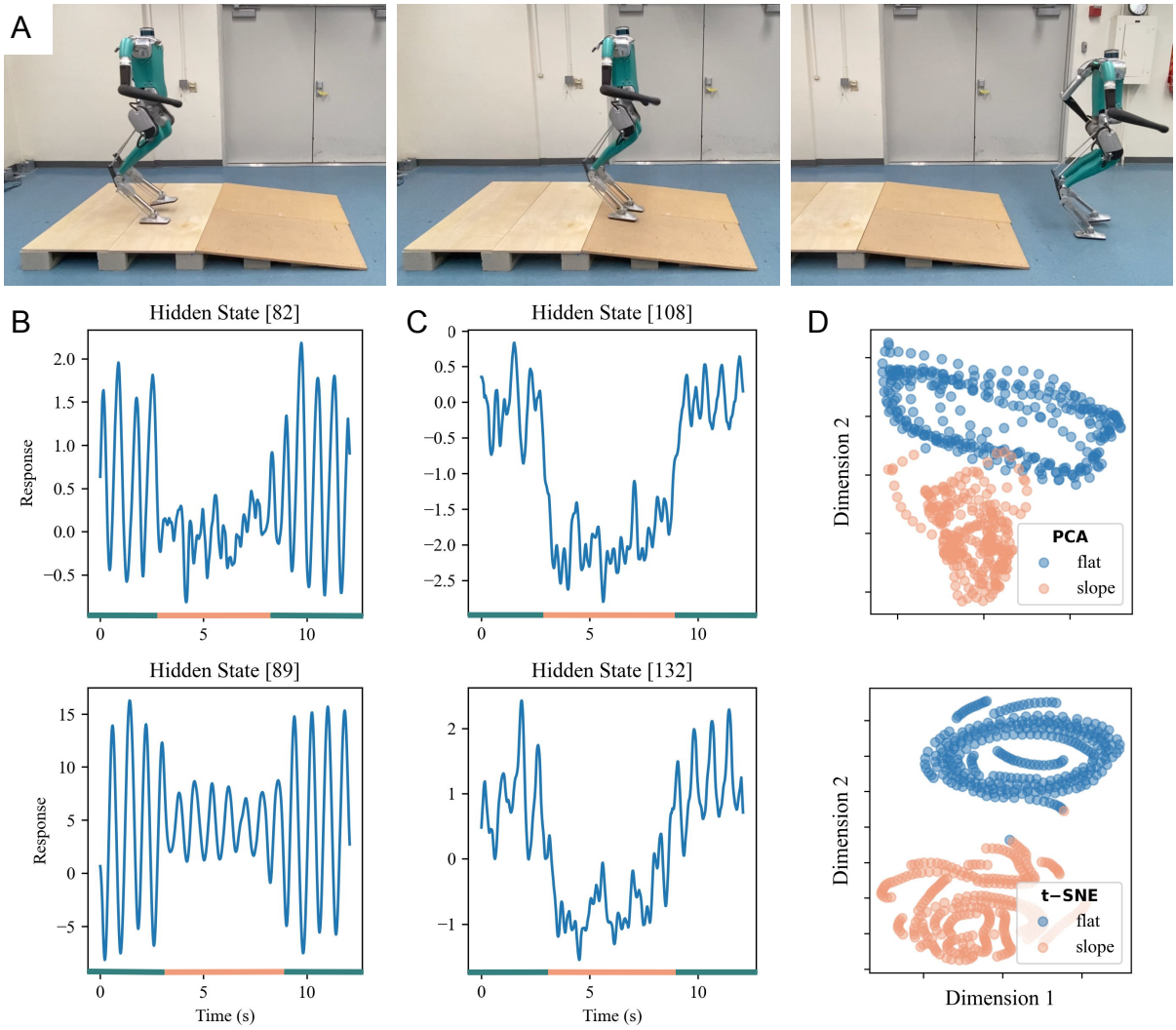


Figure 5: **Gait changes based on terrain type.** (**A**) We command the robot to walk forward over a course consisting of three sections: flat, downward slope, and flat again. We observe that our controller adapts its behavior based on terrain, changing the gait from natural walking on flat terrain, to small steps on downward slope, to natural walking on flat terrain again. Video is shown in Movie 5. This type of adaptation based on context is emergent and has not been pre-specified during training. (**B**) We analyze the hidden state of the last layer of our neural network controller and find that certain neuron responses correlate with the gait patterns observed over different terrain sections. (**C**) In addition, some of the neuron responses correlate changes in the terrain and are high for flat sections and low for the slope section. (**D**) To analyze the neural responses in aggregate, we project the 192-dimensional hidden states to two dimensions using PCA and t-SNE. Each data point corresponds to one timestep and is color-coded by the terrain section. We see that the hidden states get grouped into clear clusters based on the terrain type.

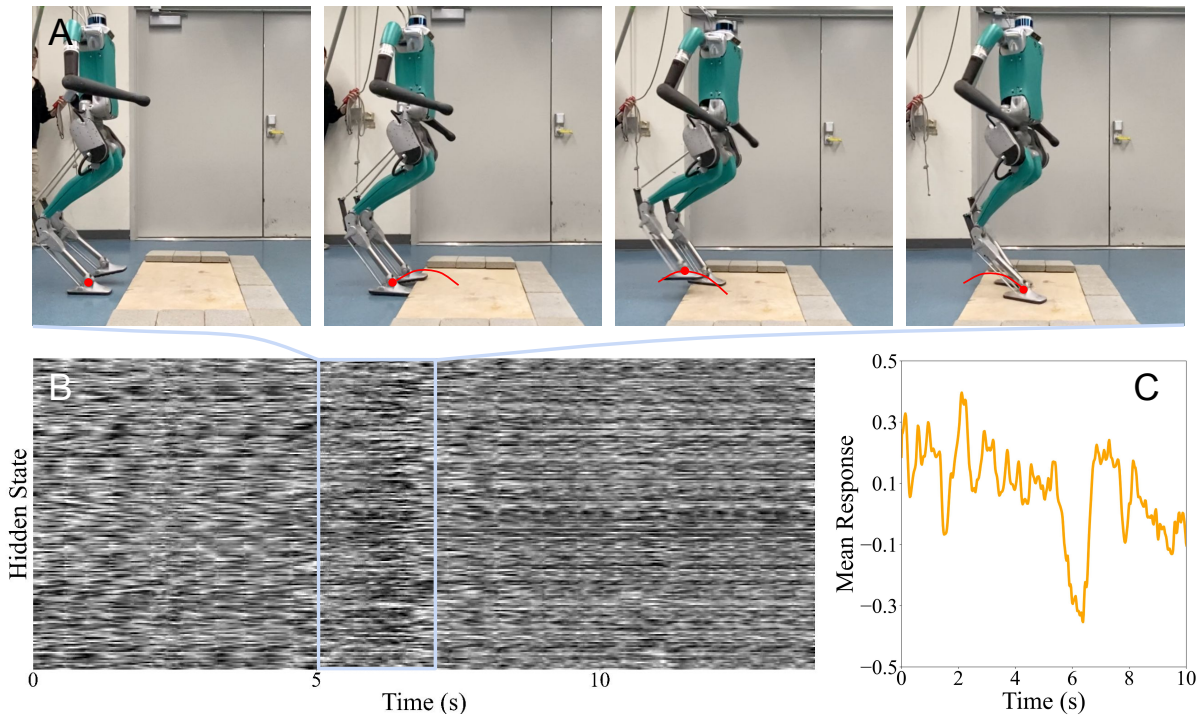


Figure 6: **Emergent recovery from foot-trapping.** (**A**) Our controller is able to adapt to discrete obstacles not seen during training and recovers from foot-trapping by lifting its legs higher and faster on subsequent attempts. This behavior is consistent and representative examples are shown in Movie 6. (**B**) We analyze the hidden state of the last layer of our transformer model and find that there is a change in the pattern of activity that correlates with the foot-trapping events. (**C**) Mean activation responses contain clear spikes during foot-trapping events as well.

a 2-dimensional vector using PCA and t-SNE. In Figure 5D, we show the results color-coded by terrain type (terrain labels only used for visualization) and see clear clusters based on terrain. These suggest that our representations capture important terrain and gait related properties.

**Emergent recovery from foot-trapping.** Next, we study the ability of our controller to recover from foot-trapping that occurs when one of the robot legs hits a discrete step obstacle. Note that steps or other form of discrete obstacles were not seen during training. This setting is relevant since our robot is blind and may find itself in such situations during deployment. We find that our controller is still able to detect and react to foot-trapping events based on the

history of observations and actions. Specifically, after hitting the step with its leg the robot will attempt to lift its legs higher and faster on subsequent attempts. Figure 6A, shows an example episode. We show a representative example for one of each of the two legs in Movie 6. We find that our controller is able to recover from different variations of such scenarios consistently. This behavior is emergent and was not pre-programmed or encouraged during training.

To understand this behavior better, we study the pattern of neural activity during an episode that contains foot-trapping and recovery, shown in Figure 6. In Figure 6B, we plot the neural activity over time. Each column is a 192-dimensional hidden state of the last layer of our transformer model and each row is the value of an individual neuron over time. We see a clear change in the pattern in activity, highlighted with a rectangle, that occurs during the foot-trapping event. In Figure 6C, we show the mean neuron response over time and see that there is a clear deviation from normal activity during the foot-trapping event. These suggest that our transformer model is able to implicitly detect such events based on neural activity.

# Discussion

We present a learning-based controller for full-sized humanoid locomotion. Our controller is a causal transformer that takes the history of past observations and actions as input and predicts future actions. We train our model using large-scale simulation and deploy it to the real world in a zero-shot fashion. We show that our policy enables reliable outdoor walking without falls, is robust to external disturbances, can traverse different terrains, and carry payloads of varying mass. Our policy exhibits natural walking behaviors, including following different commands, high-speed locomotion, and an emergent arm swing motion. Moreover, we find that our controller can adapt to novel scenarios at test time by changing its behavior based on context, including gait changes based on the terrain and recovery from foot-trapping.

**Limitations.** Our approach shows promising results in terms of adaptability and robustness to different terrains and external disturbances. However, it still has some limitations that need to be addressed in future work. One limitation is that our policy is not perfectly symmetrical, as the motors on two sides do not produce identical trajectories. This results in a slight asymmetry in movement, with the controller being better at lateral movements to the left compared to the right. Additionally, our policy is not perfect at tracking the commanded velocity. Finally, under excessive external disturbances, like a very strong pull of a cable attached to the robot, can cause the robot to fall.

**Possible extensions.** Our neural network controller is a general transformer model. Compared to alternate model choices, like TCN and LSTM, this has favorable properties that can be explored in future work. For example, it should be easier to scale with additional data and compute (*46*) and enable us to incorporate additional input modalities (*47*). Analogous to fields like vision (*48*) and language (*49*), we believe that transformers may facilitate our future progress in scaling learning approaches for real-world humanoid locomotion.

# Materials and Methods

This section describes in detail the policy learning procedure, the simulation process, the sim-to-real transfer deployment, and the analysis of the transformer-based controller. An overview of our method is shown in Figure 7. The policy learning includes two steps: teacher state policy training and student observation policy learning. We adopt a massively parallel simulation environment, where we introduce a simulation method that can simulate closed kinematic chains enabling us to simulate the underactuated Digit humanoid robot. We explain the procedure for sim-to-real transfer in detail. Finally, we provide analysis of our transformer policy.

## Policy learning

**Problem formulation.** We formulate the control problem as a Markov Decision Process (MDP), which provides a mathematical framework for modeling discrete-time decision-making processes. The MDP comprises the following elements: a state space $S$, an action space $A$, a transition function $P(s_{t+1}|s_t, a_t)$ that determines the probability of transitioning from state $s_t$ to $s_{t+1}$ after taking action $a_t$ at time step $t$, and a scalar reward function $R(s_{t+1}|s_t, a_t)$, which assigns a scalar value to each state-action-state transition, serving as feedback to the agent on the quality of its actions. Our approach to solving the MDP problem is through Reinforcement Learning (RL), which aims to find an optimal policy that maximizes the expected cumulative reward over a finite or infinite horizon.

In practice, estimating true underlying state of an environment is impossible for real-world applications. In the presence of a noisy observation space, the MDP framework needs to be modified to reflect the uncertainty in the observations. This can be done by introducing an observation space $O$ and an observation function $Z(o_t|s_t)$, which determines the probability of observing state $s_t$ as $o_t$. The MDP now becomes a Partially Observable Markov Decision Process (POMDP), where the agent must make decisions based on its noisy observations rather than the true state of the environment. The composition of the action, observation and state spaces is described in the following section. We illustrate our framework in Figure 7 and provide a comprehensive description of the method below.

**Model architecture.** Our aim is to find a policy $\pi_o$ for real-world deployment in the POMDP problem. Our policy takes as input a history trajectory of observation-action pairs over a context window of length $l$, represented as $o_t, a_{t-1}, o_{t-1}, a_{t-2}, ..., o_{t-l+1}, a_{t-l}$, and outputs the next action $a_t$. To achieve this, we utilize transformers (*50*) for sequential trajectory modeling and action prediction.

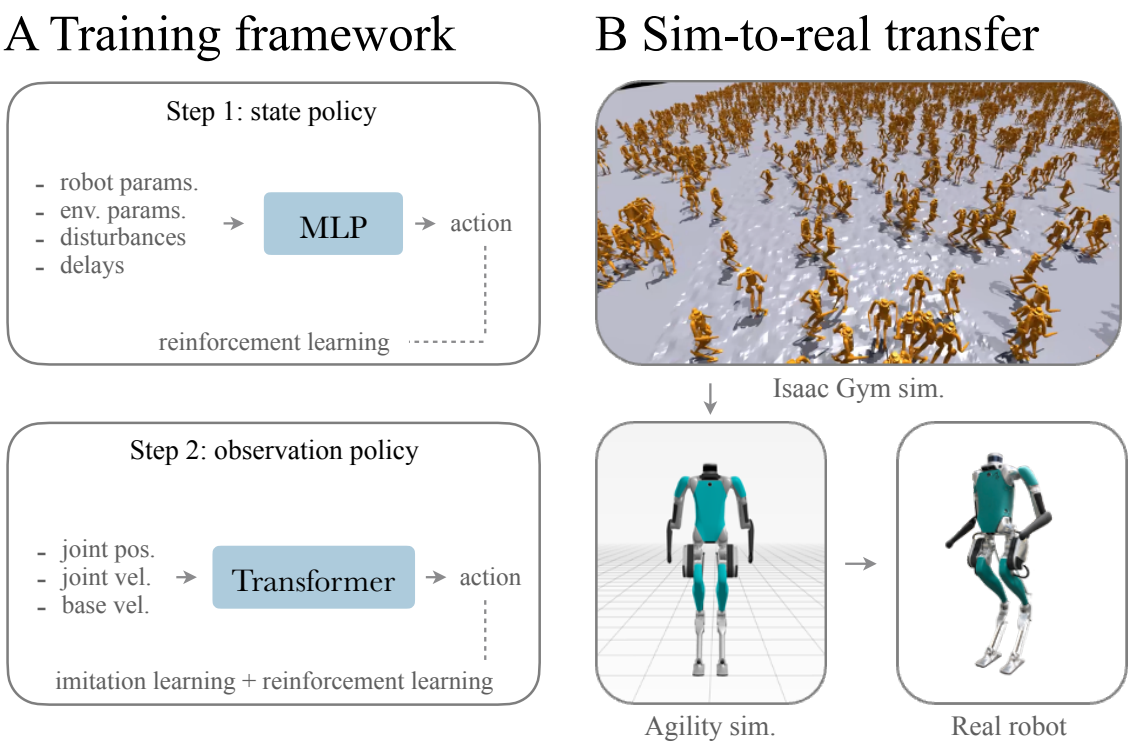


Figure 7: **Overview of the method.** (**A**) Our training consists of two steps. First, we assume that the environment is fully observable and train a teacher state policy $\pi_s(a_t|s_t)$. Second, we train a student observation policy using a combination of teacher imitation and reinforcement learning. (**B**) We leverage fast GPU simulation powered by Isaac Gym and parallelize training across multiple GPUs and thousands of randomized environments. Once a policy is trained in Isaac Gym, we validate it in the high-fidelity Agility simulator and then transfer it to the real robot. (**C**) Our neural network controller is a causal transformer model trained by autoregressive prediction of future actions from the history of observations and actions. We hypothesize that the observation-action history contains useful information about the world that a powerful transformer model can leverage to adjust its actions in-context.

Transformers are a type of neural network architecture that have been widely used in sequential modeling tasks, such as natural language processing (*32, 49, 51*), audio processing (*52*), and increasingly in computer vision (*48, 53*) as well. The key feature of transformers is the use of a self-attention mechanism, which allows the model to weigh the importance of each input element in computing the output. The self-attention mechanism is implemented through a self-attention function, which takes as input a set of queries $Q$, keys $K$, and values $V$ and outputs a weighted sum, computed as follows:

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V, \tag{1}$$

where $d_k$ is the dimensionality of the key. The self-attention mechanism enables the transformer to capture long-range dependencies between input elements.

We represent each observation-action pair in the locomotion trajectory as a token. Transformers are able to extract the structural information of these tokens through a repeated process of assigning weights to each token (softmax on $Q$ and $K$) in time, and mapping the tokens ($V$) into features spaces, effectively highlighting relevant observations and actions and thus enabling the inference of important information such as gait and contact states. We employ Multi-Layer Perceptrons (MLPs) to embed each observation-action pair into a feature space. To capture the positional information of each token in the sequence, we add sinusoidal positional encodings to the features. We leverage the temporal dependencies among the observations and actions by restricting the self-attention mechanism to only attend to preceding tokens, resulting in a causal transformer (*49*).

Transformers have proven to be effective in the realm of in-context learning, where a model's behavior can be dynamically adjusted based on the information present in its context window. Unlike gradient-based methods that require fine-tuning on task-specific data samples, transformers can learn in-context, providing them with the flexibility to handle diverse inputs.

The transformer model used in this study has four blocks, each of which has an embedding dimension of 192 and employs a multi-head attention mechanism with 4 heads. The MLP ratio of the transformer is set to 2.0. The hidden size of the MLP for projecting input observations is [512, 512]. The action prediction component of the model uses an MLP with hidden sizes of [256, 128]. Overall, the model contains 1.4M parameters. We use a context window of 16. The teacher state model is composed of an MLP with hidden sizes of [512, 512, 256, 128].

**Teacher state-policy supervision.** In Reinforcement Learning (RL), an agent must continuously gather experience through trial-and-error and update its policy in order to optimize the decision-making process. However, this process can be challenging, particularly in complex and high-dimensional environments, where obtaining a useful reward signal may require a significant number of interactions and simulation steps. Through our investigation, we found that directly optimizing a policy using RL in observation space is slow and resource-intensive, due to limited sample efficiency, which impairs our iteration cycles.

To overcome these limitations, we adopt a two-step approach. First, we assume that the environment is fully observable and train a teacher state policy $\pi_s(a_t|s_t)$ using simulation. This training is fast and resource-efficient, and we tune the reward functions, such as gait-parameters, until an optimal state policy is obtained in simulation. Next, we distill the learned state policy to an observation policy through Kullback-Leibler (KL) divergence.

**Joint optimization with reinforcement learning.** The discrepancy between the state space and the observation space can result in suboptimal decision-making if relying solely on state-policy supervision, as policies based on these separate spaces may have different reward manifolds with respect to the state and observation representations. To overcome this issue, we utilize a joint optimization approach combining RL loss with state-policy supervision. The

objective function is defined as:

$$L(\pi_o) = L_{RL}(\pi_o) + \lambda D_{KL}(\pi_o \parallel \pi_s), \tag{2}$$

where $\lambda$ is a weighting factor representing the state-policy supervision, $L_{RL}(\pi_o)$ is the RL loss, and $D_{KL}(\pi_o \parallel \pi_s)$ is the KL divergence between the observation policy $\pi_o$ and the state policy $\pi_s$. The weighting factor $\lambda$ is gradually annealed to zero over the course of the training process, typically reaching zero at the mid-point of the training horizon, which enables the observation policy to benefit from the teacher early on and learn to surpass it eventually. It is important to note that our approach does not require any pre-computed trajectories or offline datasets, as both the state-policy supervision and RL-supervision are optimized through on-policy learning.

We use the proximal policy optimization (PPO) algorithm (*54*) for training RL policies. The hyperparameters used in our experiments are shown in the supplement. We use the actor-critic method and do not share weights. The supplement lists the composition of the state and observation spaces. The action space consists of the PD setpoints for 16 actuated joints and the predicted PD gains for 8 actuated leg joints. We do not train the policy to control the four toe motors, and instead we set the motors as their default positions using fixed PD gains. This is a widely adopted approach in model-based control (*55, 56*).

Our reward function is inspired by biomechanics study of human walking and tuned through trial and error. We do not have pre-computed gait library in our reward design. The detailed composition of our reward function can be found in the supplement.

## Simulation

**Closed kinematic chain.** In our simulation environment, we use the Isaac Gym simulator (*57, 58*) to model the rigid-body and contact dynamics of the Digit humanoid robot. Given the closed kinematic chains and underactuated nature of the knee-shin-tarsus and tarsus-toe joints of the robot, Isaac Gym is unable to effectively model these dynamics. To address this

limitation, we introduce a "virtual spring" model with high stiffness to represent the rods. We apply forces calculated from the spring's deviation from its nominal length to the rigid bodies. Additionally, we employ an alternating simulation sub-step method to quickly correct the length of the virtual springs to their nominal values. We found that these efforts collectively make sim-to-real transfer feasible.

**Domain randomization.** We randomize various elements in the simulation, including dynamics properties of the robot, control parameters, and environment physics, as well as adding noise and delay to the observations. The supplement summarizes the domain randomization items and the corresponding ranges and distributions. For the robot's walking environment, we randomize the terrain types, which include smooth planes, rough planes, and smooth slopes. The robot executes a variety of walking commands such as walking forward, sideward, turning, or a combination thereof, which are randomly resampled at a fixed interval. We set the commands below a small cut-off threshold to zero. The supplement lists the ranges of the commands used in our training.

## Sim-to-real transfer

The sim-to-real transfer pipeline is shown in Figure 7. We begin by evaluating our approach in the high fidelity Agility simulator developed by Agility robotics. This enables us to evaluate unsafe controllers and control for factors of variations. Unlike the Isaac Gym simulator that was used for training, Agility simulator accurately simulates the dynamics and physical properties of the Digit robot, including the closed kinematic chain structure that is not supported by Isaac Gym. In addition, Agility simulator simulates sensor noises characterized for the real Digit robot. Note that the policy evaluation in Agility simulator does not make any change to the neural network parameters. This step only serves to filter out unsafe policies.

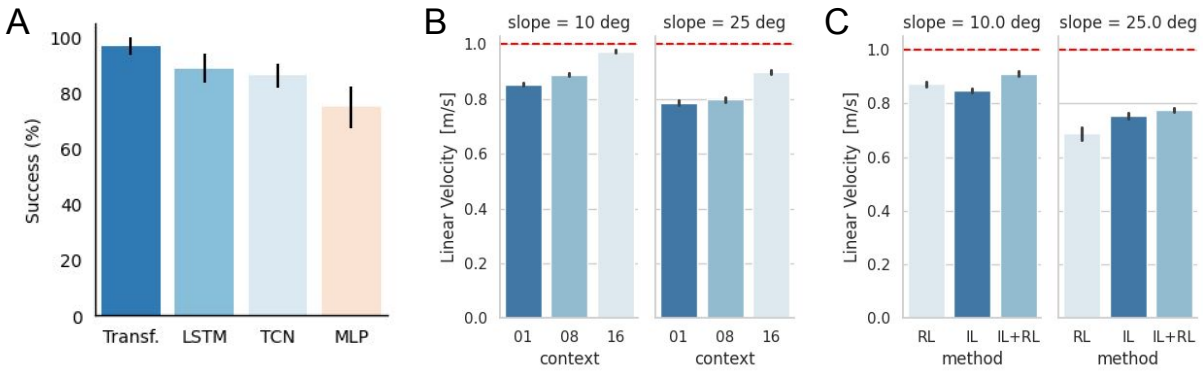


Figure 8: **Ablation studies.** We perform ablation studies to understand the impact of key design choices. For fair comparisons, we keep everything fixed except for the varied component and follow the same hyper-parameter tuning procedure. (**A**) We find that the transformer models outperform the alternate neural network choices. (**B**) Our transformer-based controller benefits from larger context lengths. (**C**) Training with the joint objective consisting of both the imitation and reinforcement learning terms outperforms training with either of the two alone.

For the deployment on hardware, we run the neural network policy at 50 Hz and the joint PD controller at 1 kHz. We can get access to joint encoders and IMU information through the API provided by Agility Robotics. We found that a combination of dynamics, terrain, and delay randomization leads to a high-quality sim-to-real transfer.

Finally, since the Isaac Gym simulator does not support accurate simulation of under-actuated systems, it poses additional challenges for sim-to-real transfer. In this study, we employed approximation methods to represent the closed kinematic chain structure. We believe that our framework will benefit from improving the simulator in the future.

## Ablation studies

In this section we preform ablation studies to analyze the key design choices in method. We compare different neural network architectures, context lengths, and training objective variants. Moreover, we analyze the attention maps of our transformer controller.

**Neural network comparisons.** We consider four different neural network architectures: 1) a Multi-Layer Perceptron (MLP), 2) a Temporal Convolutional Network (TCN) (*59*), 3) a Long Short-Term Memory (LSTM) (*60*) and 4) a Transformer model (*50*). The MLP is widely used for quadrupedal locomotion (*58, 61*). The TCN achieves state-of-the-art quadrupedal locomotion performance over challenging terrain (*17*). The LSTM shows the state-of-the-art performance for bipedal locomotion (*22, 23*). Transformer models have not been used for humanoid locomotion before but have been incredibly impactful in natural language processing (*32*). For fair comparisons, we use the same training framework for all neural network architectures and vary only the architecture of the student policy (Figure 7). We optimize the hyper parameters for each of the models separately, control for different network sizes, and pick the settings that performs the best for each model choice.

In Figure 8A, we report the mean success rate and the 95% confidence interval (CI) computed across 30 trials from 3 different scenarios from Figure 2D. We find that the transformer model outperforms other neural network choices by a considerable margin. Given the scaling properties of transformer models in NLP (*46*), this is a promising signal for using transformer models for scaling learning-based approaches for real-world humanoid locomotion in the future.

**Transformer context length.** A key property of our transformer-based controller is to adapt its behavior implicitly based on the context of observations and actions. In Figure 8B, we study the performance of our approach for different context lengths. We command the robot to work forward at 1 m/s over two different slopes. We randomize the initial positions and heading and report the mean linear velocity and 95% CI across 20 trials. We find that our model benefits from a larger context length in both settings.

**Training objective.** Our training objective from Equation 2 consists of two terms, an imitation learning term based on teacher policy supervision and a reinforcement learning term based

on rewards. We study the impact of both terms. Using only the imitation term is common in quadrupedal locomotion (*17*) while using only reinforcement learning term corresponds to learning without a teacher (*13, 22*). In Figure 8C, we report the results on the same slope setting as in the previous context length ablation. We find that the joint imitation and reinforcement learning objective outperforms using either of the two terms alone.

# Acknowledgments

This work was supported in part by DARPA Machine Common Sense program, ONR MURI program (N00014-21-1-2801), NVIDIA, InnoHK of the Government of the Hong Kong Special Administrative Region via the Hong Kong Centre for Logistics Robotics, The AI Institute, and BAIR's industrial alliance programs. We thank Sarthak Kamat, Baifeng Shi, and Saner Cakir for help with experiments; Aravind Srinivas, Agrim Gupta, Ashish Kumar, William Peebles, Tim Brooks, Matthew Tancik, Shuxiao Chen, Zhongyu Li, and Benjamin McInroe for helpful discussions; Gavriel State, Philipp Reist, Viktor Makoviychuk, Ankur Handa, and the Isaac Gym team for simulation discussions; Jack Thomas, Jake Thompson, Levi Allery, Jonathan Hurst, and the Agility Robotics team for hardware discussions.

# References

1. I. Kato, "Development of wabot 1," *Biomechanism*, 1973.

2. K. Hirai, M. Hirose, Y. Haikawa, and T. Takenaka, "The development of honda humanoid robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2. IEEE, 1998, pp. 1321–1326.

3. G. Nelson, A. Saunders, N. Neville, B. Swilling, J. Bondaryk, D. Billings, C. Lee, R. Playter, and M. Raibert, "Petman: A humanoid robot for testing chemical protective clothing," *Journal of the Robotics Society of Japan*, vol. 30, no. 4, pp. 372–377, 2012.

4. O. Stasse, T. Flayols, R. Budhiraja, K. Giraud-Esclasse, J. Carpentier, J. Mirabel, A. Del Prete, P. Souères, N. Mansard, F. Lamiraux *et al.*, "Talos: A new humanoid research platform targeted for industrial applications," in *IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*. IEEE, 2017, pp. 689–695.

5. M. Chignoli, D. Kim, E. Stanger-Jones, and S. Kim, "The mit humanoid robot: Design, motion planning, and control for acrobatic behaviors," in *IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2021, pp. 1–8.

6. M. H. Raibert, *Legged robots that balance*. MIT press, 1986.

7. S. Kajita, F. Kanehiro, K. Kaneko, K. Yokoi, and H. Hirukawa, "The 3d linear inverted pendulum mode: A simple modeling for a biped walking pattern generation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2001.

8. 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.

9. S. Collins, A. Ruina, R. Tedrake, and M. Wisse, "Efficient bipedal robots based on passive-dynamic walkers," *Science*, vol. 307, no. 5712, pp. 1082–1085, 2005.

10. Y. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4906–4913.

11. S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous robots*, vol. 40, pp. 429–455, 2016.

12. J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and S. Kim, "Dynamic locomotion in the mit cheetah 3 through convex model-predictive control," in *IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2018, pp. 1–9.

13. OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

14. OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving rubik's cube with a robot hand," *arXiv preprint arXiv:1910.07113*, 2019.

15. A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam *et al.*, "Dextreme: Transfer of agile in-hand manipulation from simulation to reality," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5977–5984.

16. J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

17. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

18. A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *Robotics: Science and Systems (RSS)*, 2021.

19. H. Benbrahim and J. A. Franklin, "Biped dynamic walking using reinforcement learning," *Robotics and Autonomous Systems*, vol. 22, no. 3-4, pp. 283–302, 1997.

20. R. Tedrake, T. W. Zhang, and H. S. Seung, "Stochastic policy gradient reinforcement learning on a simple 3d biped," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3. IEEE, 2004, pp. 2849–2854.

21. Z. Xie, G. Berseth, P. Clary, J. Hurst, and M. van de Panne, "Feedback control for cassie with deep reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1241–1246.

22. J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 7309–7315.

23. J. Siekmann, K. Green, J. Warila, A. Fern, and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning," *Robotics: Science and Systems (RSS)*, 2021.

24. S. Iida, S. Kato, K. Kuwayama, T. Kunitachi, M. Kanoh, and H. Itoh, "Humanoid robot control based on reinforcement learning," in *Micro-Nanomechatronics and Human Science, 2004 and The Fourth Symposium Micro-Nanomechatronics for Information-Based Society, 2004.* IEEE, 2004, pp. 353–358.

25. D. Rodriguez and S. Behnke, "Deepwalk: Omnidirectional bipedal gait by deep reinforcement learning," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 3033–3039.

26. G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Reinforcement learning-based cascade motion policy design for robust 3d bipedal locomotion," *IEEE Access*, vol. 10, pp. 20 135–20 148, 2022.

27. L. Krishna, G. A. Castillo, U. A. Mishra, A. Hereid, and S. Kolathaya, "Linear policies are sufficient to realize robust bipedal walking on challenging terrains," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2047–2054, 2022.

28. R. Antonova, S. Cruciani, C. Smith, and D. Kragic, "Reinforcement learning for pivoting task," *arXiv preprint arXiv:1703.00472*, 2017.

29. F. Sadeghi and S. Levine, "Cad2rl: Real single-image flight without a single real image," *Robotics: Science and Systems (RSS)*, 2016.

30. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2017, pp. 23–30.

31. X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 3803–3810.

32. T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

33. A. K. Han, A. Hajj-Ahmad, and M. R. Cutkosky, "Bimanual handling of deformable objects with hybrid adhesion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5497–5503, 2022.

34. G. A. Castillo, B. Weng, W. Zhang, and A. Hereid, "Robust feedback motion policy design using reinforcement learning on a 3d digit bipedal robot," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5136–5143.

35. Y. Gao, Y. Gong, V. Paredes, A. Hereid, and Y. Gu, "Time-varying alip model and robust foot-placement control for underactuated bipedal robot walking on a swaying rigid surface," *arXiv preprint arXiv:2210.13371*, 2022.

36. Y. Gao, C. Yuan, and Y. Gu, "Invariant filtering for legged humanoid locomotion on a dynamic rigid surface," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 1900–1909, 2022.

37. A. Adu-Bredu, N. Devraj, and O. C. Jenkins, "Optimal constrained task planning as mixed integer programming," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 029–12 036.

38. K. S. Narkhede, A. M. Kulkarni, D. A. Thanki, and I. Poulakakis, "A sequential mpc approach to reactive planning for bipedal robots using safe corridors in highly cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 831–11 838, 2022.

39. A. Shamsah, Z. Gu, J. Warnke, S. Hutchinson, and Y. Zhao, "Integrated task and motion planning for safe legged navigation in partially observable environments," *IEEE Transactions on Robotics*, 2023.

40. D. J. Morton and D. D. Fuller, *Human locomotion and body form: a study of gravity and man*. Williams & Wilkins, 1952.

41. H. Herr and M. Popovic, "Angular momentum in human walking," *Journal of experimental biology*, vol. 211, no. 4, pp. 467–481, 2008.

42. S. H. Collins, P. G. Adamczyk, and A. D. Kuo, "Dynamic arm swinging in human walking," *Proceedings of the Royal Society B: Biological Sciences*, vol. 276, no. 1673, pp. 3679–3688, 2009.

43. J. D. Ortega, L. A. Fehlman, and C. T. Farley, "Effects of aging and arm swing on the metabolic cost of stability in human walking," *Journal of biomechanics*, vol. 41, no. 16, pp. 3303–3308, 2008.

44. B. R. Umberger, "Effects of suppressing arm swing on kinematics, kinetics, and energetics of human walking," *Journal of biomechanics*, vol. 41, no. 11, pp. 2575–2580, 2008.

45. M. Murray, S. Sepic, and E. Barnard, "Patterns of sagittal rotation of the upper limbs in walking," *Physical therapy*, vol. 47, no. 4, pp. 272–284, 1967.

46. J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, "Scaling laws for neural language models," *arXiv:2001.08361*, 2020.

47. J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23 716–23 736, 2022.

48. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.

49. A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

50. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

51. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

52. L. Dong, S. Xu, and B. Xu, "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition," in *IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 5884–5888.

53. N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV 2020: 16th European Conference*, 2020.

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

55. X. Da, O. Harib, R. Hartley, B. Griffin, and J. W. Grizzle, "From 2d design of underactuated bipedal gaits to 3d implementation: Walking with speed tracking," *IEEE Access*, 2016.

56. Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, and J. Grizzle, "Feedback control of a cassie bipedal robot: Walking, standing, and riding a segway," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 4559–4566.

57. V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

58. 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.

59. S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

60. S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, 1997.

61. J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *Robotics: Science and Systems (RSS)*, 2018.