ROBOT LOCOMOTION

Real-world humanoid locomotion with reinforcement learning

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

Humanoid robots that can autonomously operate in diverse environments have the potential to help address labor shortages in factories, assist elderly at home, and colonize new planets. Although 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 hypothesized 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 trained our model with large-scale model-free reinforcement learning on an ensemble of randomized environments in simulation and deployed it to the real-world zero-shot. Our controller could walk over various outdoor terrains, was robust to external disturbances, and could adapt in context.

INTRODUCTION

The dream of robotics has always been 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.

Roboticists designed the first full-sized 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*). However, the control problem remains a considerable challenge. Classical control methods can achieve stable and robust locomotion (*6*–*9*), and optimization-based strategies have shown the advantage of simultaneously authoring dynamic behaviors and obeying constraints (*10*–*12*). The most well-known examples are the Boston Dynamics Atlas robot doing back flips, jumping over obstacles, and dancing.

Although these approaches have made great progress, learning-based methods have become of increasing interest because of 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.

Here, we propose a learning-based approach for real-world humanoid locomotion (Movie 1). Our controller is a causal transformer that takes the history of proprioceptive observations and actions as input and predicts the next action. Our model is trained with large-scale reinforcement learning (RL) on thousands 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 used 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 hypothesized 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 versus 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 evaluated our model on a full-sized humanoid robot through a series of real-world and simulated experiments. We show that our policy enabled reliable outdoor walking without falls, was robust to external disturbances, could traverse different terrains, and carried payloads of varying mass. Moreover, we found that our approach compared favorably with the state-of-the-art model-based controller. Our policy exhibited natural walking behaviors, including following different commands, high-speed locomotion, and an emergent arm-swing motion. Our policy was adaptive and could change its behavior based on context, including gradual gait changes based on slowly varying terrains and rapid adaptation to sudden obstacles. To understand different design choices, we analyzed our method in controlled experiments and found that the transformer architecture outperformed other neural network architectures, that the model benefited from larger context, and that joint training with teacher imitation and RL was beneficial.

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 will encourage future research on scalable learning-based approaches for humanoid robots.

RESULTS

Digit humanoid robot

Digit is a general-purpose humanoid robot developed by Agility Robotics, standing at approximately 1.6 m tall with a total weight of 45 kg. The robot's floating-base model is equipped with 30 degrees

University of California, Berkeley CA, USA.
*Corresponding author. Email: ilija@berkeley.edu (I.R.); txiao@berkeley.edu (T.X.); bikezhang@berkeley.edu (B.Z.)
†These authors contributed equally to this work.
‡These authors contributed equally to this work.

Movie 1. Outdoor deployment. A reinforcement learning-based controller enables real-world humanoid locomotion.

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, whereas the toe joint is actuated by means of rods attached at the tarsus joint. Digit has been used as a humanoid platform for mechanical design (*33*), locomotion control (*27*, *34*, *35*), state estimation (*36*), and planning (*37*–*39*).

Outdoor deployment

We begin by reporting the results of deploying our controller to a number of outdoor environments. Examples are shown in Movie 1 and Fig. 1, including everyday human environments, plazas, walkways, sidewalks, running tracks, and grass fields. The terrains varied considerably in terms of material properties, like concrete, rubber, and grass, as well as conditions, like dry under the afternoon sun and damp in the early morning. Our controller was 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 we were comfortable deploying it without a safety gantry. Over the course of 1 week of full-day testing in outdoor environments, we did not observe any falls. Nevertheless, because our controller acted on the basis of the history of observations and actions and did not include any additional sensors like cameras, it could bump and get trapped by obstacles, like steps, but managed to adapt its behavior to avoid falling.

Indoor experiments and simulation benchmark

We conducted a series of experiments in the laboratory environment to test the performance of the proposed approach in controlled settings (Fig. 2 and movie S1).

External forces

Robustness to external forces is a critical requirement for real-world deployment of humanoid robots. We tested whether our controller could handle sudden external forces while walking. These experiments included throwing a large yoga ball at the robot, pushing the robot with a wooden stick, and pulling the robot from the back while it was walking forward (Fig. 2A). We found that our controller was able to stabilize the robot in each of these scenarios. Given that the humanoid is a highly unstable system and that the disturbances we applied were 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 conducted a series of experiments on different terrains in the laboratory (Fig. 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: rubber, cloth, cables, and bubble wrap, which altered the roughness of the terrain and could potentially lead to challenging entanglement and slipping situations, because the robot did not use exteroceptive sensing. Despite these impediments, our controller traversed all these terrain types. Last, 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 were 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 evaluated the robot's ability to carry loads of varying mass, shape, and center of mass while walking forward (Fig. 2C). We conducted 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 was able to successfully complete its walking route while carrying each of these loads. Our learning-based controller adapted 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 was able to adapt its behavior according to the context.

Comparison with the state of the art

We compared our controller with the native controller provided by Agility Robotics, which is the state of the art for this robot. To quantify the performance across many runs, we used the high-fidelity simulator by Agility Robotics. We chose three different scenarios: walking over slopes, steps, and unstable ground (Fig. 2D). We commanded the robot to walk forward and considered a trial as successful if the robot could cross the terrain without falling. Crossing a portion of the terrain obtained partial success. We report the mean success rate with 95% confidence interval (CI) per terrain across 10 runs (Fig. 2D). We found that both ours and the native controller walked well on slopes. Next, we observed that our controller outperformed the native controller on steps. The native controller struggled to correct itself from a trapped foot and shut off. We replicated this scenario in the real world and have observed consistent behavior, shown in movie S2. In contrast, our controller recovered successfully. Note that our controller was not trained on steps in simulation and that the foot-trapping recovery behaviors were emergent. Last, we compared the two controllers on a simulated terrain with unstable planks. This setting is challenging because the terrain can dislodge under the robot feet. We found that our controller considerably outperformed the native controller. We did not evaluate the controllers on this terrain in the real world because of concerns for potential hardware damage.

Natural walking

Omnidirectional walking

Our controller performed omnidirectional locomotion by following velocity commands. Specifically, it was 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 sampled commands randomly

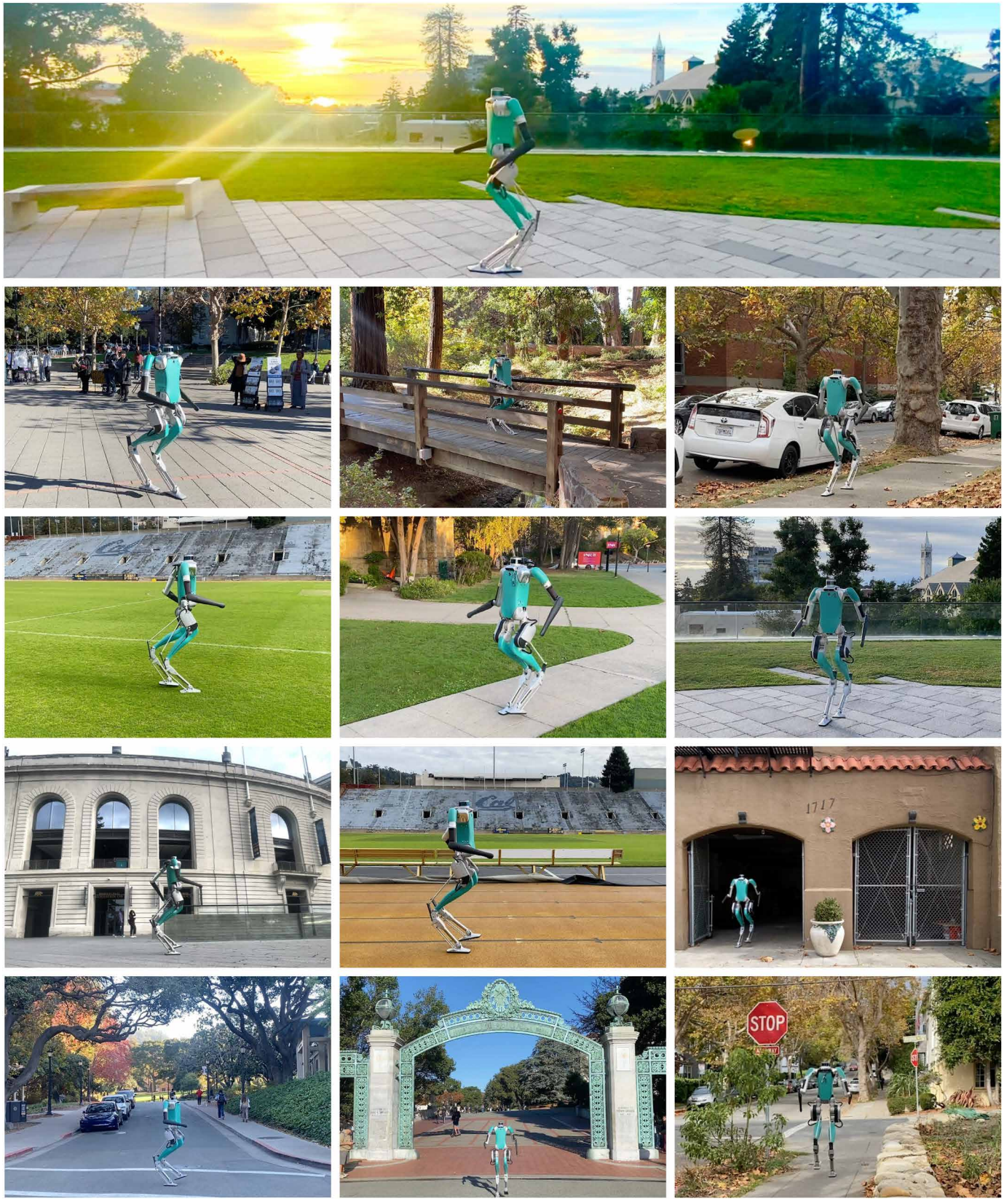

Fig. 1. Deployment in outdoor environments. We deployed our model in a number of outdoor environments. Example videos are shown in Movie 1. We found that our controller was able to traverse a range of everyday environments including plazas, sidewalks, tracks, and grass fields.

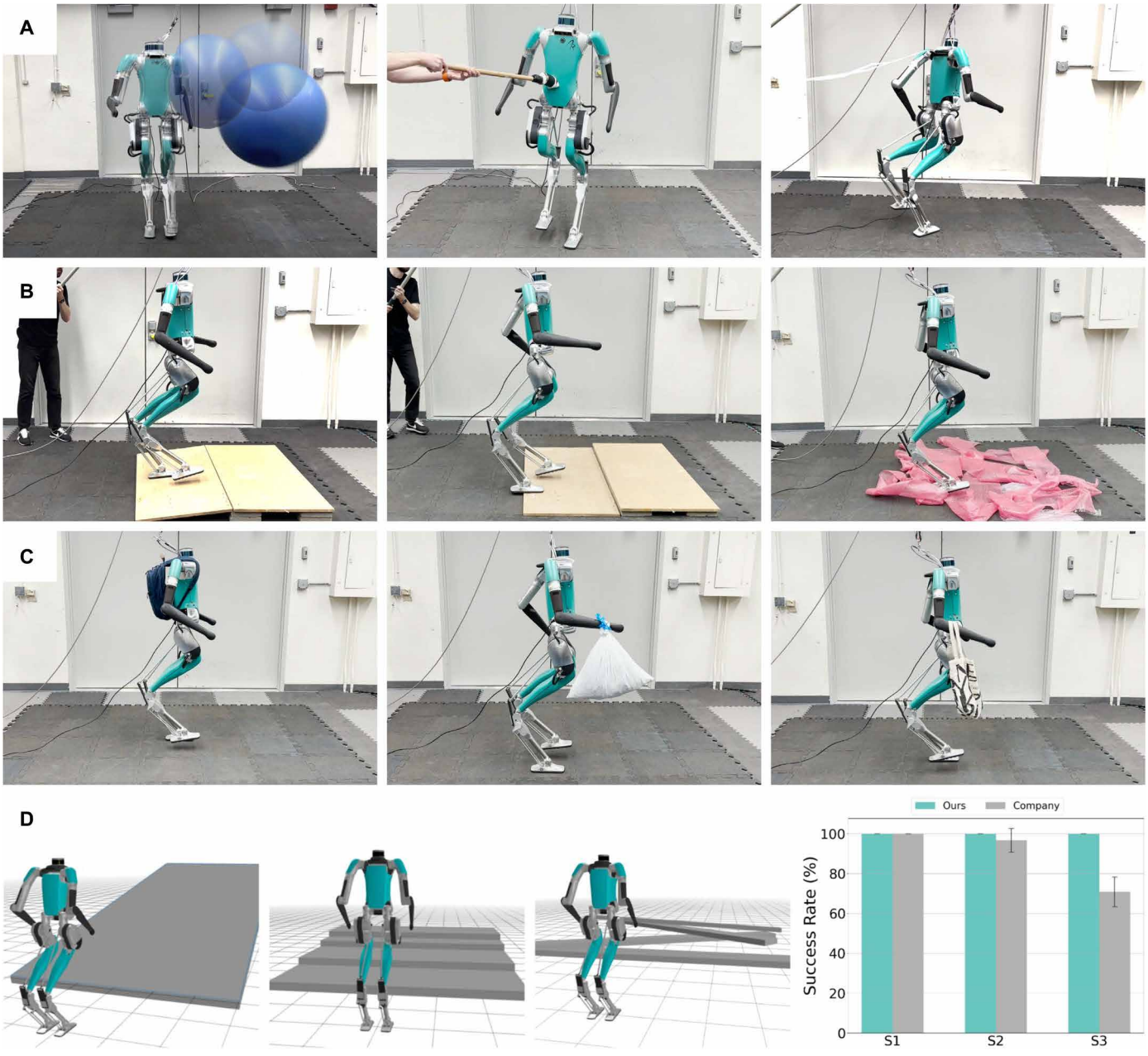

Fig. 2. Indoor experiments and simulation benchmark. We test the robustness of our controller to (**A**) external disturbances, (**B**) different terrains, and (**C**) payloads. Videos are shown in movie S1. We found that our controller was able to tackle all of the scenarios successfully, including those that were considerably out of the training distribution. (**D**) We found that our controller outperformed the state-of-the-art native controller across three different settings in simulation. The improvements in stability are larger for harder terrains, like steps and unstable ground. We replicated a subset of the scenarios on hardware and observed consistent behaviors, which can be seen in examples from movie S2.

every 10 s. At deployment, we found that our controller followed commands accurately. In addition, it generalized to continuously changing commands, supplied via a joystick in real time, which was different from training. We show examples of walking forward, backward, and turning in Fig. 3 and in movie S3.

Dynamic arm swing

A distinct feature of natural human walking is the arm swing. Studying the arm-swing behavior in humans has a long history in biomechanics (*40*–*42*). There are a number of existing hypotheses for why humans might swing their arms while walking. Examples include arm swinging leading to dynamic stability (*43*), reducing the metabolic energy cost of walking (*44*), and being 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.

When training our neural network controller, we did not impose explicit constraints on the arm-swing motion in the reward function or use any reference trajectories for the arms. After training, we observed

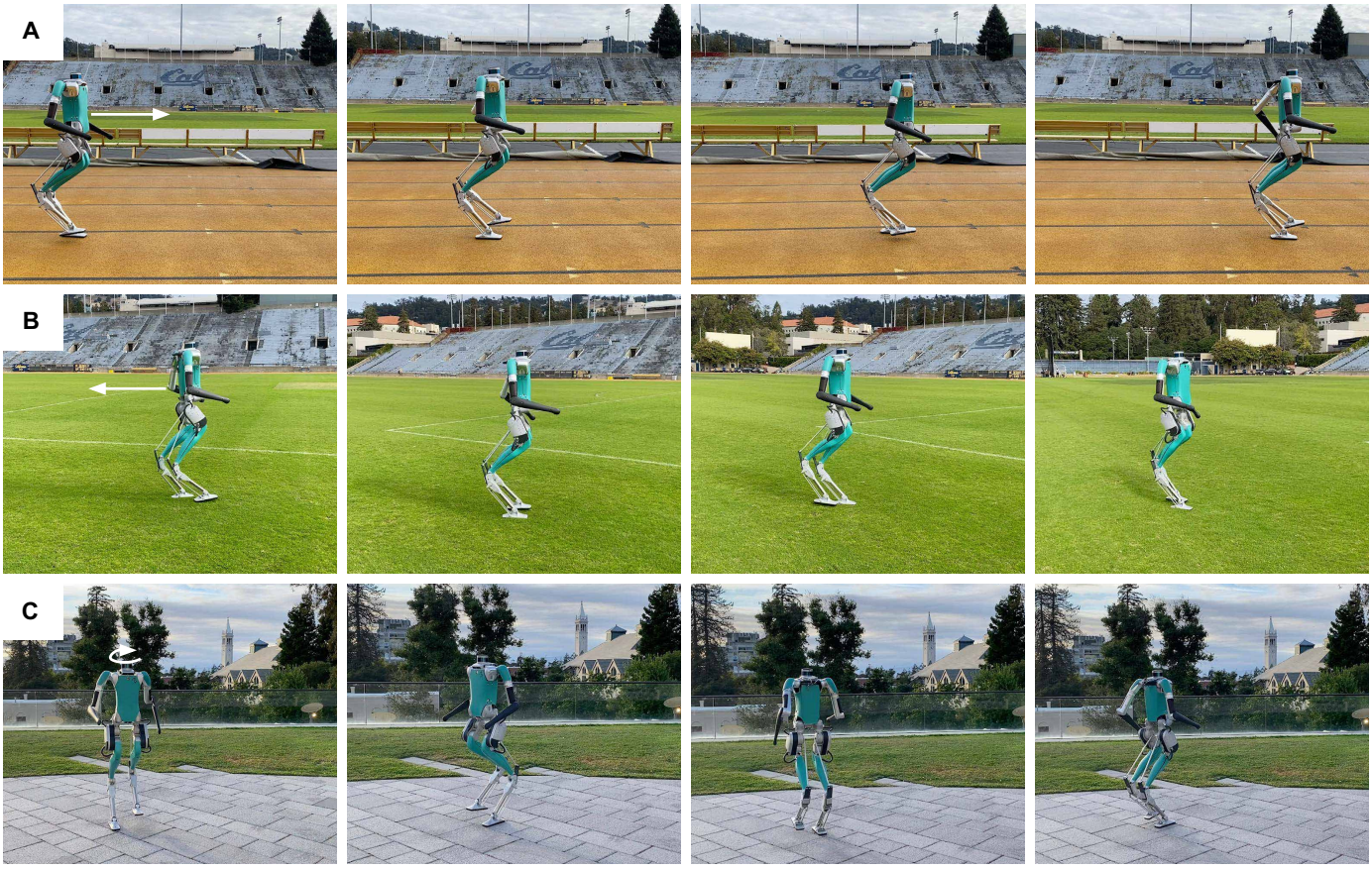

Fig. 3. Omnidirectional walking. Our learning-based controller is able to accurately follow a range of velocity commands to perform omnidirectional locomotion, including (**A**) walking forward, (**B**) backward, and (**C**) turning. Video examples are shown in movie S3.

emergent arm-swing motions with phase opposite to the legs, as shown in Fig. 4A. We note that our reward function included energy minimization terms, which might suggest a relationship between the observed motions and energy expenditure.

Fast walking

There is considerable difference between walking at low and high speeds. We analyzed the performance of our controller when walking fast in the real world. Figure 4B shows the velocity-tracking performance given a commanded step velocity at 1 m/s. The corresponding video is in movie S4. We observed that the robot achieved the commanded velocity from rest within 1 s and tracked it accurately for the duration of the course.

In-context adaptation

Emergent gait changes based on terrain

We commanded the robot to walk forward over a terrain consisting of three sections in order: flat ground, downward slope, and flat ground again, shown in Fig. 5A. We found that our controller changed its walking behavior entirely based on the terrain. Specifically, it started by normal walking on flat ground, transitioned to using small steps without lifting its legs to the normal height on downward slope, and then returned to normal walking on flat ground again. These behavior changes were emergent and not prespecified.

To understand this behavior better, we studied the patterns of neural activity of our transformer model over time. First, we examined the responses of individual neurons and found that certain neurons correlated with gait. Namely, they had high amplitude during walking on flat ground and low amplitude on the downward slope. Two such neurons are shown in Fig. 5B. Moreover, some neurons correlated with terrain types. Their responses were high on flat terrain and low on slope, as shown in Fig. 5C. We also analyzed the neural responses in aggregate by performing dimensionality reduction. We projected the 192-dimensional hidden state from each time step into a two-dimensional vector using principal components analysis (PCA) and *t*-distributed stochastic neighbor embedding (t-SNE). In Fig. 5D, we show the results color-coded by terrain types (terrain labels only used for visualization) and clusters based on terrain. These suggest that our representations capture important terrain and gait-related properties.

Emergent recovery from foot-trapping

Next, we studied the ability of our controller to recover from foot-trapping that occurred when one of the robot legs hit a discrete step obstacle. Note that steps or other forms of discrete obstacles were not seen during training. This setting is relevant because our robot is blind and may find itself in such situations during deployment. We found that our controller was still able to detect and react to

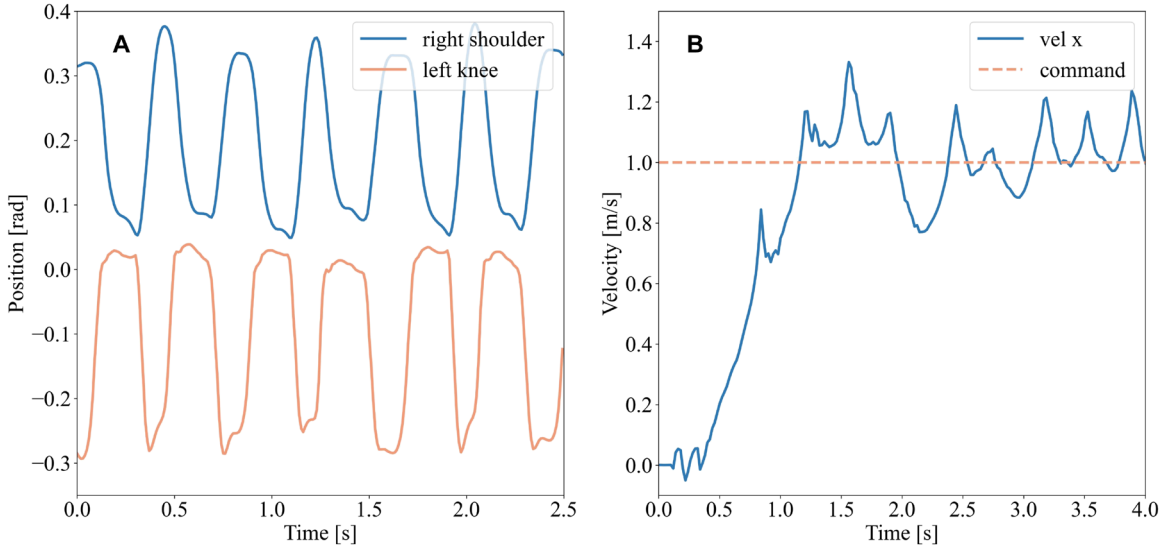

Fig. 4. Arm swing and fast walking. (**A**) The learned humanoid locomotion in our experiments exhibits human-like arm swing behaviors in coordination with leg movements, which is a contralateral relationship between the arms and the legs. (**B**) Our controller is able to perform fast walking on hardware. The video is shown in movie S4.

foot-trapping events on the basis of the history of observations and actions. Specifically, after hitting the step with its leg, the robot attempted to lift its legs higher and faster on subsequent attempts. Figure 6A shows an example episode. We also show a representative example for one of each of the two legs in movie S6. We found that our controller recovered from different variations of such scenarios consistently. This behavior was emergent and not preprogrammed or encouraged during training.

To further investigate this behavior, we studied the pattern of neural activity during an episode that contains foot trapping and recovery, shown in Fig. 6. Figure 6B shows 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 observed a change in the pattern of activity, highlighted with a rectangle, that occurred during the foot-trapping event. Figure 6C shows the mean neuron response over time, and there is a deviation from normal activity during the foot-trapping event. These suggest that our transformer model was able to implicitly detect such events on the basis of 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 the next action. We trained our model using large-scale simulation and deployed it to the real world in a zero-shot fashion. We show that our policy enabled reliable outdoor walking without falls, was robust to external disturbances, and could traverse different terrains and carry payloads of varying mass. Our policy exhibited natural walking behaviors, including following different commands, high-speed locomotion, and an emergent arm-swing motion. Moreover, we found that our controller could 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.

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 was not perfectly symmetrical, because the motors on two sides did not produce identical trajectories. This resulted in a slight asymmetry in movement, with the controller being better at lateral movements to the left compared with the right. In addition, our policy was not perfect at tracking the commanded velocity. Last, under excessive external disturbances, like a very strong pull of a cable attached to the robot, the robot could fall.

Our neural network controller is a general transformer model. Compared with 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

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 $P(s_{t+1} \mid s_t, a_t)$ 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 $R(s_{t+1} \mid s_t, a_t)$, 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 RL, which aims to find an optimal policy that maximizes the expected cumulative reward over a finite or infinite horizon.

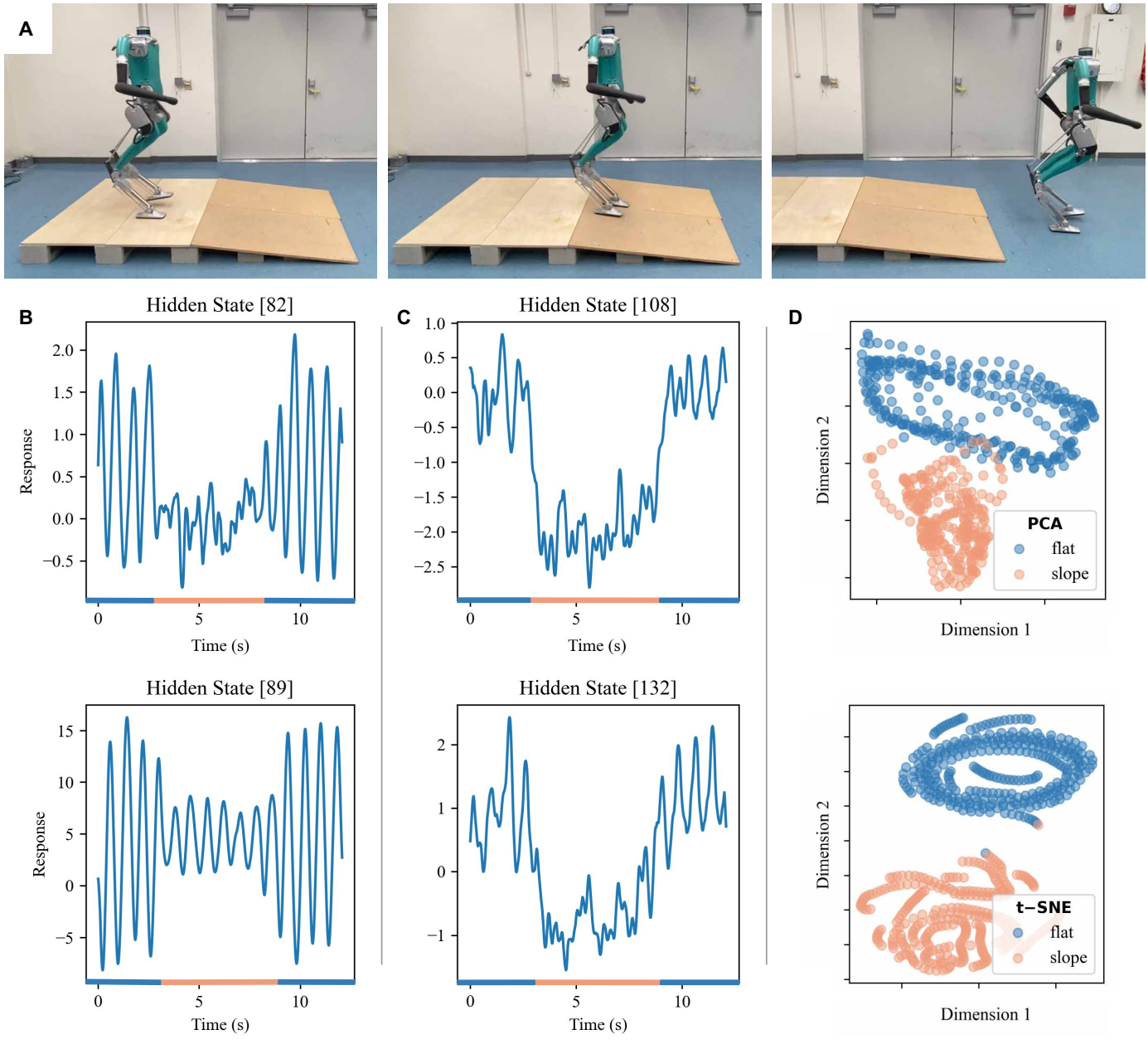

Fig. 5. Gait changes based on terrain type. (**A**) We commanded the robot to walk forward over a course consisting of three sections: flat, downward slope, and flat again. We observed that our controller adapts its behavior based on terrain, changing the gait from natural walking on flat terrain to small steps on downward slope, then to natural walking on flat terrain again. Video is shown in movie S5. This type of adaptation based on context was emergent and was not prespecified during training. (**B**) We analyzed the hidden state of the last layer of our neural network controller and found that certain neuron responses correlate with the gait patterns observed over different terrain sections. (**C**) In addition, some of the neuron responses correlate with changes in the terrain and are high for flat sections and low for the slope section. Numbers of neurons are included in square brackets. (**D**) To analyze the neural responses in aggregate, we projected the 192-dimensional hidden states to two dimensions using PCA and t-SNE. Each data point corresponds to one time step and is color-coded by the terrain section. We see that the hidden states get grouped into clusters on the basis of the terrain type.

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 $Z(o_t \mid s_t)$, which determines the probability of observing state s_t as o_t. The MDP now becomes a partially observable MDP (POMDP), where the agent must make decisions on the basis of 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 Fig. 7 and provide a comprehensive description of the method below.

Model architecture

Our aim is to find a policy π_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,

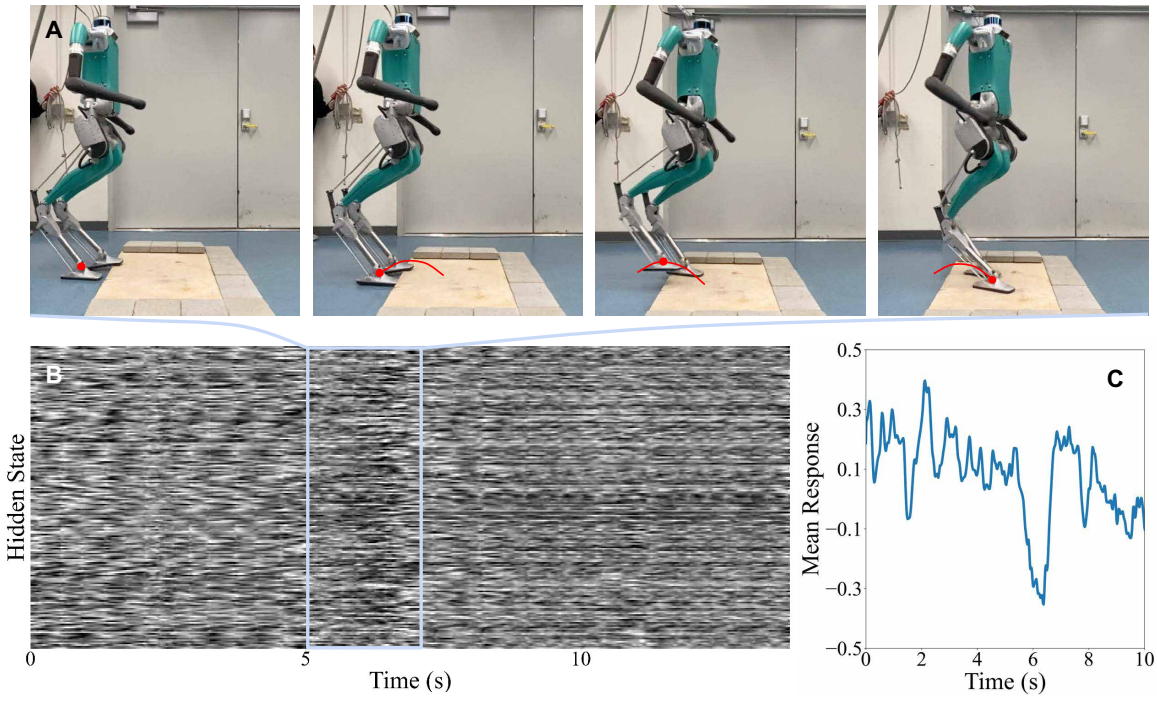

Fig. 6. Emergent recovery from foot-trapping. (**A**) Our controller was able to adapt to discrete obstacles not seen during training and recovered from foot-trapping by lifting its legs higher and faster on subsequent attempts. This behavior is consistent, and representative examples are shown in movie S6. (**B**) We analyzed the hidden state of the last layer of our transformer model and found that there is a change in the pattern of activity that correlates with the foot-trapping events. (**C**) Mean activation responses contain spikes during foot-trapping events as well.

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 used transformers (*50*) for sequential trajectory modeling and action prediction.

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:

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

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 feature spaces, effectively highlighting relevant observations and actions and thus enabling the inference of important information, such as gait and contact states. We used multilayer perceptrons (MLPs) to embed each observation-action pair into a feature space. To capture the positional information of each token in the sequence, we added sinusoidal positional encodings to the features. We leveraged 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 on the basis of 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 uses a multihead attention mechanism with four 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 1.4 million 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 RL, an agent must continuously gather experience through trial-and-error and update its policy to optimize the decision-making process. However, this process can be challenging, in particular in complex and high-dimensional environments, where obtaining a useful reward signal may require a substantial number of interactions and simulation steps. Through our investigation, we found that directly optimizing a policy using RL in observation space is slow

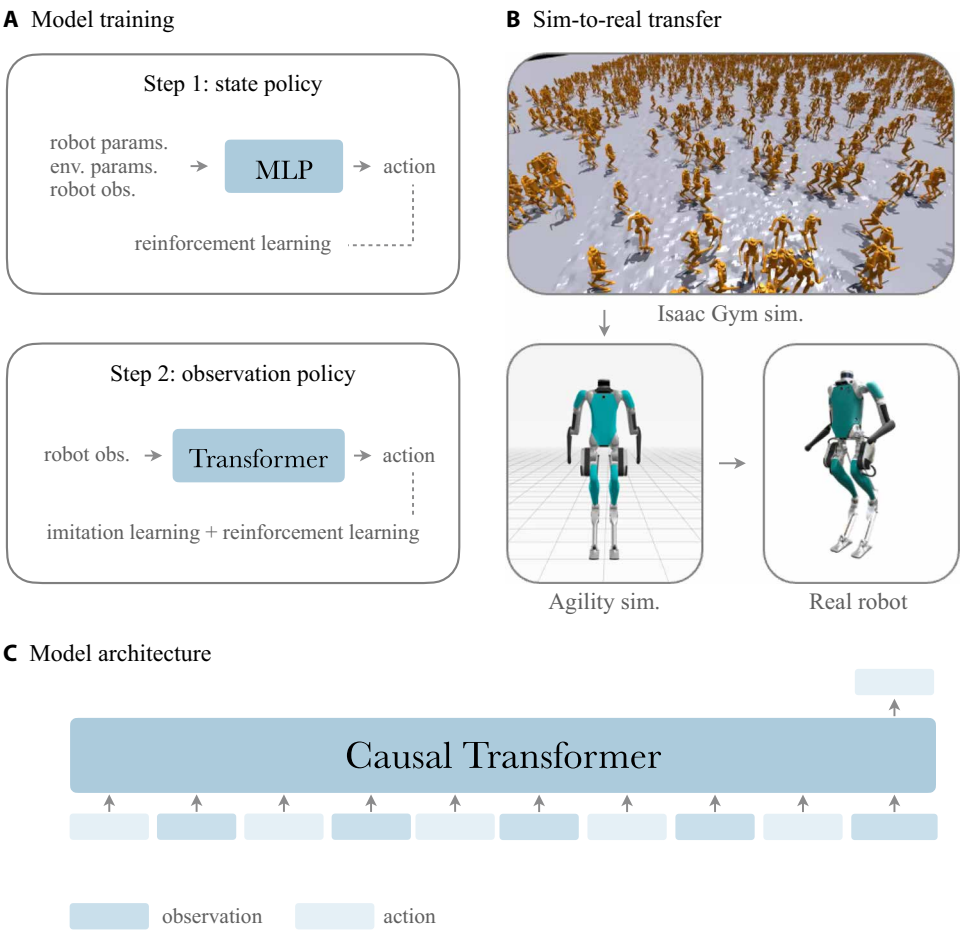

Fig. 7. Overview of the method. (**A**) Our training consisted of two steps. First, we assumed that the environment is fully observable and trained a teacher state policy $\pi_s(a_t \mid s_t)$. Second, we trained a student observation policy using a combination of teacher imitation and RL. (**B**) We leveraged fast GPU simulation powered by Isaac Gym and parallelized training across four A100 GPUs and thousands of randomized environments. Once a policy was trained in Isaac Gym, we validated it in the high-fidelity simulator provided by the robot manufacturer. Last, we transferred it to the real robot. (**C**) Our neural network controller is a causal transformer model trained to predict the next action from the history of observations and actions. We hypothesized that the observation-action history contains useful information about the world that a powerful transformer model can leverage to adjust its actions in context.

and resource intensive because of limited sample efficiency, which impairs our iteration cycles.

To overcome these limitations, we adopted a two-step approach. First, we assumed that the environment was fully observable and trained a teacher state policy $\pi_s(a_t \mid s_t)$ using simulation. This training was fast and resource efficient, and we tuned the reward functions, such as gait parameters, until an optimal state policy was obtained in simulation. Next, we distilled 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, because policies based on these separate spaces may have different reward manifolds with respect to the state and observation representations. To overcome this issue, we used a joint optimization approach combining RL loss with state-policy supervision. The objective function is defined as.

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

where λ 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 π_o and the state policy π_s. The weighting factor λ is gradually annealed to zero over the course of the training process, typically reaching zero at the midpoint of the training horizon, which enables the observation policy to benefit from the teacher early on and learn to surpass it eventually. Our approach does not require any precomputed trajectories or offline datasets, because both the state-policy supervision and RL supervision are optimized through on-policy learning.

We used the proximal policy optimization (PPO) algorithm (*54*) for training RL policies. The hyperparameters used in our experiments are shown in the Supplementary Materials. We used the actor-critic method and did not share weights. The Supplementary Materials lists the composition of the state and observation spaces. The action space consists of the PD set points for 16 actuated joints and the predicted PD gains for eight actuated leg joints. We did 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 was inspired by biomechanics study of human walking and tuned through trial and error. We did not have a precomputed gait library in our reward design. The detailed composition of our reward function can be found in the Supplementary Materials.

Simulation

Closed kinematic chain

In our simulation environment, we used 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 was unable to effectively model these dynamics. To address this limitation, we introduced a "virtual spring" model with high stiffness to represent the rods. We applied forces calculated from the spring's deviation from its nominal length to the rigid bodies. In addition, we used an alternating simulation substep method to quickly correct the length of the virtual springs to their nominal values. We found that these efforts collectively made sim-to-real transfer feasible.

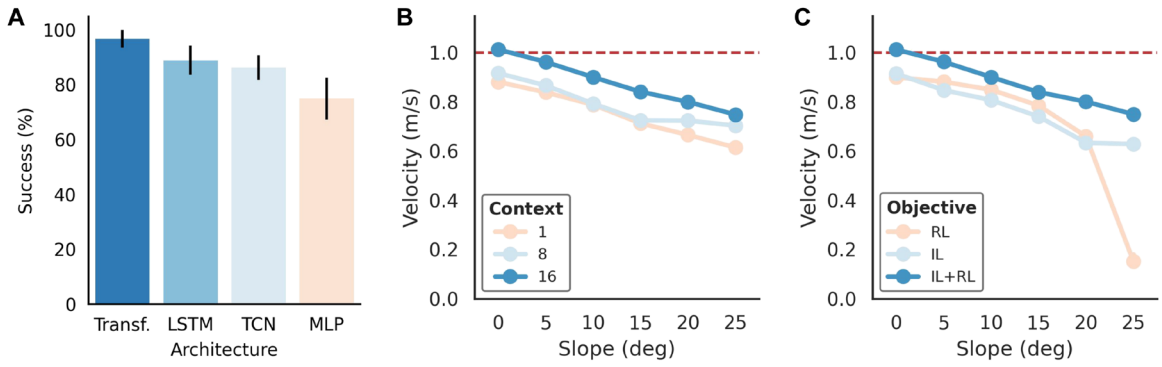

Fig. 8. Ablation studies. We performed ablation studies to understand the effects of key design choices. For fair comparisons, we kept everything fixed except for the varied component and followed the same hyperparameter tuning procedure. (**A**) We found that the transformer models outperformed the alternate neural network choices. (**B**) Our transformer-based controller benefited from larger context lengths. (**C**) Training with the joint objective consisting of both the imitation and RL terms outperformed training with either of the two alone.

Domain randomization

We randomized 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 Supplementary Materials summarizes the domain randomization items and the corresponding ranges and distributions. For the robot's walking environment, we randomized the terrain types, which included smooth planes, rough planes, and smooth slopes. The robot executed a variety of walking commands, such as walking forward, sideward, turning, or a combination thereof, which were randomly resampled at a fixed interval. We set the commands below a small cut-off threshold to zero. The Supplementary Materials lists the ranges of the commands used in our training.

Sim-to-real transfer

The sim-to-real transfer pipeline is shown in Fig. 7. We began by evaluating our approach in the high-fidelity Agility simulator developed by Agility Robotics. This enabled us to evaluate unsafe controllers and control for factors of variations. Unlike the Isaac Gym simulator that was used for training, Agility simulator accurately simulated 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 simulated sensor noise characterized for the real Digit robot. Note that the policy evaluation in Agility simulator did not make any change to the neural network parameters. This step only served to filter out unsafe policies.

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

Last, because the Isaac Gym simulator does not support accurate simulation of underactuated systems, it poses additional challenges for sim-to-real transfer. In this study, we used 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 perform ablation studies to analyze the key design choices in the 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: an MLP, a TCN (*59*), an LSTM (*60*), and 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 very influential in natural language processing (*32*). For fair comparisons, we used the same training framework for all neural network architectures and varied only the architecture of the student policy (Fig. 7). We optimized the hyperparameters for each of the models separately, controlled for different network sizes, and picked the settings that performed the best for each model choice.

In Fig. 8A, we report the mean success rate and the 95% CI computed across 30 trials from three different scenarios from Fig. 2D. We found that the transformer model outperforms other neural network choices by a considerable margin. Given the scaling properties of transformer models in neural language processing (*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 adaptation of its behavior implicitly based on the context of observations and actions. In Fig. 8B, we study the performance of our approach for different context lengths. We commanded the robot to walk forward at 1 m/s over two different slopes. We found that our model benefits from a larger context length in both settings.

Training objective

Our training objective from Eq. 2 consists of two terms, an imitation learning term based on teacher policy supervision and an RL term based on rewards. We studied the effects of both terms. Using only the imitation term is common in quadrupedal locomotion (*17*), whereas using only the RL term corresponds to learning without a teacher (*13*, *22*). In Fig. 8C, we report the results on the same slope setting as in the previous context length ablation. We found that the joint imitation and RL objective outperformed using either of the two terms alone.

Supplementary Materials

The PDF file includes:
Text S1 and S2
Tables S1 to S4

Other Supplementary Material for this manuscript includes the following:
Movies S1 to S6

REFERENCES AND NOTES

1. I. Kato, Development of WABOT 1, in *Biomechanism* (University of Tokyo Press, 1973).
2. K. Hirai, M. Hirose, Y. Haikawa, T. Takenaka, The development of Honda humanoid robot, in *IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 1998), vol. 2, pp. 1321–1326.
3. G. Nelson, A. Saunders, N. Neville, B. Swilling, J. Bondaryk, D. Billings, C. Lee, R. Playter, M. Raibert, Petman: A humanoid robot for testing chemical protective clothing. *J. Robot. Soc. Jpn.* **30**, 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, J. P. Laumond, L. Marchionni, H. Tome, F. Ferro, 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, 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, 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, D. E. Koditschek, Hybrid zero dynamics of planar biped walkers. *IEEE Trans Automat Contr* **48**, 42–56 (2003).
9. S. Collins, A. Ruina, R. Tedrake, M. Wisse, Efficient bipedal robots based on passive-dynamic walkers. *Science* **307**, 1082–1085 (2005).
10. Y. Tassa, T. Erez, 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, R. Tedrake, Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. *Auton Robots* **40**, 429–455 (2016).
12. J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, 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. 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, W. Zaremba, Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **39**, 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, L. Zhang, Solving Rubik's cube with a robot hand. 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, Y. Narang, DeXtreme: Transfer of agile in-hand manipulation from simulation to reality, in *Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2023), pp. 5977–5984.
16. J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning agile and dynamic motor skills for legged robots. *Sci. Robot.* **4**, eaau5872 (2019).
17. J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning quadrupedal locomotion over challenging terrain. *Sci. Robot.* **5**, eabc5986 (2020).
18. A. Kumar, Z. Fu, D. Pathak, J. Malik, RMA: Rapid motor adaptation for legged robots, *Proceedings of the Robotics: Science and Systems (RSS)*; Virtual Event, 12 to 16 July 2021.
19. H. Benbrahim, J. A. Franklin, Biped dynamic walking using reinforcement learning. *Rob. Auton. Syst.* **22**, 283–302 (1997).
20. R. Tedrake, T. W. Zhang, 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)* (IEEE, 2004), vol. 3, pp. 2849–2854.
21. Z. Xie, G. Berseth, P. Clary, J. Hurst, 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, 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, J. Hurst, Blind bipedal stair traversal via sim-to-real reinforcement learning, in *Proceedings of the Robotics: Science and Systems (RSS)* (RSS, 2021).
24. S. Iida, S. Kato, K. Kuwayama, T. Kunitachi, M. Kanoh, 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, 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, A. Hereid, Reinforcement learning-based cascade motion policy design for robust 3D bipedal locomotion. *IEEE Access* **10**, 20135–20148 (2022).
27. L. Krishna, G. A. Castillo, U. A. Mishra, A. Hereid, S. Kolathaya, Linear policies are sufficient to realize robust bipedal walking on challenging terrains. *IEEE Robot. Autom. Lett.* **7**, 2047–2054 (2022).
28. R. Antonova, S. Cruciani, C. Smith, D. Kragic, Reinforcement learning for pivoting task, arXiv:1703.00472 (2017).
29. F. Sadeghi, S. Levine, Cad2rl: Real single-image flight without a single real image, in *Proceedings of the Robotics: Science and Systems (RSS)* (RSS, 2016).
30. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, 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, 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, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. M. Candlish, A. Radford, I. Sutskever, D. Amodei, Language models are few-shot learners, in *Proceedings of the 34th International Conference on Neural Information Processing Systems* (NeurIPS, 2020), pp. 1877–1901.
33. A. K. Han, A. Hajj-Ahmad, M. R. Cutkosky, Bimanual handling of deformable objects with hybrid adhesion. *IEEE Robot. Autom. Lett.* **7**, 5497–5503 (2022).
34. G. A. Castillo, B. Weng, W. Zhang, 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, Y. Gu, Time-varying alip model and robust foot-placement control for underactuated bipedal robotic walking on a swaying rigid surface, in *2023 American Control Conference (ACC)* (IEEE, 2023), pp. 3282–3287.
36. Y. Gao, C. Yuan, Y. Gu, Invariant filtering for legged humanoid locomotion on a dynamic rigid surface. *IEEE ASME Trans. Mechatron.* **27**, 1900–1909 (2022).
37. A. Adu-Bredu, N. Devraj, 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. 12029–12036.
38. K. S. Narkhede, A. M. Kulkarni, D. A. Thanki, I. Poulakakis, A sequential mpc approach to reactive planning for bipedal robots using safe corridors in highly cluttered environments. *IEEE Robot. Autom. Lett.* **7**, 11831–11838 (2022).
39. A. Shamsah, Z. Gu, J. Warnke, S. Hutchinson, Y. Zhao, Integrated task and motion planning for safe legged navigation in partially observable environments. *IEEE Trans. Robot.* **39**, 4913–4934 (2023).
40. D. J. Morton, D. D. Fuller, *Human Locomotion and Body Form: A Study of Gravity and Man* (Williams & Wilkins, 1952).
41. H. Herr, M. Popovic, Angular momentum in human walking. *J. Exp. Biol.* **211**, 467–481 (2008).
42. S. H. Collins, P. G. Adamczyk, A. D. Kuo, Dynamic arm swinging in human walking. *Proc. R. Soc. B Biol. Sci.* **276**, 3679–3688 (2009).
43. J. D. Ortega, L. A. Fehlman, C. T. Farley, Effects of aging and arm swing on the metabolic cost of stability in human walking. *J. Biomech.* **41**, 3303–3308 (2008).
44. B. R. Umberger, Effects of suppressing arm swing on kinematics, kinetics, and energetics of human walking. *J. Biomech.* **41**, 2575–2580 (2008).
45. M. Murray, S. Sepic, E. Barnard, Patterns of sagittal rotation of the upper limbs in walking. *Phys. Ther.* **47**, 272–284 (1967).
46. J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J.Wu, 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, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. L. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Bińkowski, R. Barreira, O. Vinyals, A. Zisserman, K. Simonyan, Flamingo: A visual language model for few-shot learning. *Adv. Neural Inf. Process. Syst.* **35**, 23716 (2022).

48. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in *International Conference on Learning Representations* (2021).
49. A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training (2018).
50. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, *Proceedings of the 31st International Conference on Neural Information Processing Systems* (NeurIPS, 2017), pp. 6000–6010.
51. J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, arXiv:1810.04805 (2018).
52. L. Dong, S. Xu, 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, S. Zagoruyko, End-to-end object detection with transformers, in *ECCV 2020: 16th European Conference*, vol. 12346 of *Lecture Notes in Computer Science* (Springer, 2020), pp. 213–229.
54. J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, arXiv:1707.06347 (2017).
55. X. Da, O. Harib, R. Hartley, B. Griffin, J. W. Grizzle, From 2d design of underactuated bipedal gaits to 3d implementation: Walking with speed tracking. *IEEE Access* **4**, 3469–3478 (2016).
56. Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang, 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, G. State, Isaac Gym: High performance GPU-based physics simulation for robot learning, arXiv:2108.10470 (2021).
58. N. Rudin, D. Hoeller, P. Reist, M. Hutter, Learning to walk in minutes using massively parallel deep reinforcement learning, in *Conference on Robot Learning* (MLResearchPress, 2022).
59. S. Bai, J. Z. Kolter, V. Koltun, An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, arXiv:1803.01271 (2018).
60. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**, 1735–1780 (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, in *Proceedings of the Robotics: Science and Systems (RSS)* (RSS, 2018).

Acknowledgments: We thank S. Kamat, B. Shi, and S. Cakir for help with experiments; Y. Liu for discussions of spring approximation in simulation; A. Srinivas, A. Gupta, A. Kumar, W. Peebles, T. Brooks, M. Tancik, S. Chen, Z. Li, B. McInroe, and R. Huang for helpful discussions; G. State, P. Reist, V. Makoviychuk, A. Handa, and the I. Gym team for simulation discussions; J. Thomas, J. Thompson, L. Allery, J. Hurst, and the Agility Robotics team for hardware discussions. **Funding:** 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. **Author contributions:** I.R., T.X., and B.Z. led the project. T.D., J.M., and K.S. advised the project. **Competing interests:** J.M. is also (part-time) affiliated with Meta Platforms Inc., but this research was not sponsored or supported by Meta. The other authors declare that they have no competing interests. **Data and materials availability:** Additional data and materials are present in the Supplementary Materials.

Submitted 31 May 2023
Accepted 26 March 2024
Published 17 April 2024
10.1126/scirobotics.adi9579

Science Robotics

Real-world humanoid locomotion with reinforcement learning

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

Sci. Robot. **9** (89), eadi9579. DOI: 10.1126/scirobotics.adi9579

Editor's summary
The ability of robots to navigate adaptively and robustly in varying terrain increases their chances of success when deployed in the real world. However, stable locomotion of full-size bipedal humanoid robots creates a challenge from a controls perspective. Radosavovic *et al*. developed a reinforcement learning approach for controlling locomotion of a humanoid robot, Digit. They trained their model in simulation and subsequently deployed it into the real-world zero-shot and showed the potential for robust locomotion on various indoor and outdoor environments. The robot could exhibit natural and adaptive walking behaviors, including an emergent arm-swing motion, and adapt to external perturbations.
—Amos Matsiko

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.