ANIMAL LOCOMOTION

Multi-expert learning of adaptive legged locomotion

Chuanyu Yang[1]*, Kai Yuan[1]*, Qiuguo Zhu[2], Wanming Yu[1], Zhibin Li[1†]

Achieving versatile robot locomotion requires motor skills that can adapt to previously unseen situations. We propose a multi-expert learning architecture (MELA) that learns to generate adaptive skills from a group of representative expert skills. During training, MELA is first initialized by a distinct set of pretrained experts, each in a separate deep neural network (DNN). Then, by learning the combination of these DNNs using a gating neural network (GNN), MELA can acquire more specialized experts and transitional skills across various locomotion modes. During runtime, MELA constantly blends multiple DNNs and dynamically synthesizes a new DNN to produce adaptive behaviors in response to changing situations. This approach leverages the advantages of trained expert skills and the fast online synthesis of adaptive policies to generate responsive motor skills during the changing tasks. Using one unified MELA framework, we demonstrated successful multiskill locomotion on a real quadruped robot that performed coherent trotting, steering, and fall recovery autonomously and showed the merit of multi-expert learning generating behaviors that can adapt to unseen scenarios.

INTRODUCTION

Adaptive motor skills enable living organisms to accomplish sophisticated motor tasks and offer them better chances to survive in nature. Among vast sensorimotor skills, locomotion is essential for most animals to move in the environment. Therefore, to understand and create adaptive locomotion behaviors is a long-standing scientific theme for biologists and roboticists. From a neurological perspective, it is worth understanding how the sensorimotor control system in animals processes various sensory information and produces adaptive reactions in unseen situations (*1*–*4*). From a robotics perspective, it is interesting to take a bioinspired approach and transfer biological principles, such as primitive neural circuits, to produce robot behaviors similar to that of animals (*1*). Because the underlying mechanisms of the motor cortex cannot yet be fully replicated (*4*, *5*), we take the latter approach by drawing inspiration from biological motor control to develop learning algorithms that can achieve skill adaptation for robot locomotion (*6*).

In this study, we investigate how an artificial agent can learn to generate multiple motor skills from a set of existing skills, particularly for critical tasks that require immediate responses. As an example of learning a complex physical task, playing soccer consists of several subskills, such as dribbling, passing, and shooting. During training, players first practice the most important subskills separately. Once mastered, all different subskills are used in a flexible combination to improve all these techniques. Our research in multiskill learning has studied skill-adaptive capabilities such as these, and we draw inspiration from animal motor control when designing the learning and control architecture. Using a quadruped robot as a test bed, we aim to produce adaptive robot behaviors to succeed in unexpected situations in a responsive manner.

Background

The DARPA Robotics Challenge (DRC) from 2012 to 2015 fostered the development of semiautonomous robots for dangerous missions such as disaster response in unstructured environments (*7*). Most DRC robots had different forms of legged design for the dexterity to traverse irregular surfaces. Despite the tremendous engineering efforts, no robot could recover autonomously from falls (*8*). To date, most legged robots still lack such an autonomous ability to generate adaptive actions to deal with unexpected situations.

Because of the uncertainties in unforeseen situations, locomotion failures are likely to happen. We illustrate these challenges in robot locomotion using field tests (Fig. 1A) and the adaptive behaviors that are robust to uncertainties (Fig. 1, B and C). Typically, falling occurs within a second of the robot losing balance, and the time window for fall prevention is around 0.2 to 0.5 s. Therefore, it is critical to immediately coordinate different locomotion modes to mitigate perturbations and prevent or recover from failures. In comparison with robot systems, biological systems (e.g., cats, dogs, and humans) exhibit higher versatility (*9*), and the key to the performance gap is the difference in motion intelligence, which allows biological systems to handle changing and complex situations (*9*–*11*).

Our paper studies a machine learning approach that learns reactive locomotion skills and generates adaptive behaviors by reusing and recombining trained skills. Here, we investigate the motor skills in the form of feedback control policies to address the reactive adaptation to multimodalities during robot locomotion, leading to increased robustness against failures.

Related work

When a robot interacts with its environment, it can be difficult to determine which part of the robot is in contact; this is challenging for classical control solutions because careful modeling of contacts is often needed. The main approach in the legged locomotion community uses model-based mathematical optimization to solve these multicontact problems, such as model predictive control (MPC), whole-body quadratic programming (QP), and trajectory optimization (TO). To achieve fast online computation, MPC uses simplified models and short predictive horizons to plan task-space motions for walking (*12*), running (*13*), and pacing (*14*). QP methods are used for whole-body control to map task-space motions (e.g., those from the MPC) to joint-space actions while considering the whole robot model and physical constraints (*15*, *16*). A unified but computationally expensive technique is to optimize all models and constraints together (whole robot model, contact model, environment constraints), i.e., through nonlinear MPC (NMPC) (*17*) and whole-body optimization (*18*–*20*).

[1]School of Informatics, University of Edinburgh, Edinburgh, UK. [2]Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou, China.
*These authors contributed equally to this work.
†Corresponding author. Email: zhibin.li@ed.ac.uk

Fig. 1. Challenging locomotion scenarios and agile maneuvers of a quadruped robot. (**A**) Three challenging scenarios of the Jueying robot during various tests: unexpected body contacts with the environment and unpredictable robot states. The white circled regions highlight unusual contact that can occur at any part of the body. (**B**) Different adaptive behaviors from our proposed learning framework that generated dynamic motions and complex coordination of legs for immediate recovery from failures. Time in snapshots is measured in seconds. (**C**) Resilient locomotion using adaptive skills in the presence of unexpected disturbances.

In the optimization scheme, all physical contacts between the robot and the environment need to be defined as constraints in the formulation. The contact sequence—such as the contact location, timing, and duration—needs to be specified either by manual design or by an additional planner (*21*, *22*). Furthermore, the explicit properties of the robot and the environment need to be modeled (*23*) but are expensive to compute (*24*) and thus difficult to run in real time in complex settings even with exhaustive computing (*25*). This fundamental principle suffers from the curse of dimensionality and therefore limits the scalability to real-time solutions in more complex and challenging problems (*26*).

To this end, deep reinforcement learning (DRL) becomes attractive for acquiring task-level skills: Through rewarding intended outcomes and penalizing undesired ones, an artificial agent can learn desirable behaviors (*21*, *22*). Using DRL offers several advantages: The training process can be realized by using physics engines to perform a large number of iterations in simulations without risks of hardware damage; the agent can explore freely and learn effective policies that are difficult for humans to manually design; and the computation of readily trained neural networks can be real time. For legged locomotion, many DRL results have been achieved in simulation (*27*, *28*) and, in recent studies, on hardware (*29*–*31*), e.g., demonstration of learning-based control on a real robot using separate policies for fall recovery and walking (*32*). However, similar to other DRL approaches, the learning policies in (*32*) were specialized in separate tasks instead of being a unified policy across different tasks. This is a common feature due to the learning structure that only trains a single DRL agent for solving one specific task, which results in a narrowly skilled policy.

Hierarchical reinforcement learning (HRL) solves complex tasks at different levels of temporal abstraction using multiple experts' existing knowledge (*33*): Experts are trained to encode low-level motor primitives, whereas a high-level policy selects the appropriate expert (*34*, *35*). However, generating new skills cannot be achieved in the standard HRL framework because only one expert is selected at a time. This problem can be addressed by learning to continuously blend high-dimensional variables of all experts (*36*). One related approach is the mixture of experts (MoE) that synthesizes the outputs of individual experts specialized on subproblems using a gating function (*37*), which has been used in robotics (*38*), computer vision (*39*), and computer graphics (*40*). However, compared with blending the experts' high-dimensional variables, MoE has known limitations

in scaling to high degree-of-freedom (DoF) systems (*40*) and exhibits expert imbalance problems, i.e., favoring certain experts while degrading others (*41*).

Our approach

We draw inspiration from biological motor control to design our control and learning framework. Biological studies suggest that motor behavior is controlled by the central nervous system (CNS) that resets the reference position of body segments, and the difference between the reference and actual position excites the muscular activities for generating appropriate forces (*42*). This precludes the need to compute the inverse dynamics, simplifies control, and minimizes the computation (*43*). Because the spring-damper property provided by the impedance control resembles the elasticity of biological muscles, we applied the equilibrium-point (EP) control hypothesis to generate joint torques by offsetting the equilibrium point.

Inspired by the biomechanical control of muscular systems and the EP hypothesis, we distribute the robot control in two layers: (i) At the bottom layer, we use torque control to configure the joint impedance for the robot, and (ii) at the top layer, we designate deep neural networks (DNNs) to produce set points for all joints to modulate posture and joint torques, establishing force interactions with the environment (see Materials and Methods). By doing so, we can focus on developing the learning algorithms at the top layer to achieve motor intelligence.

Contributions

This work aims to demonstrate how hierarchical motor control using DRL can achieve a breadth of adaptive behaviors for contact-rich locomotion. We propose a multi-expert learning architecture (MELA) that contains multiple expert neural networks, each with a unique motor skill, and a gating neural network (GNN) that fuses expert networks dynamically into a more versatile and adaptive neural network.

Compared with the approach of using kinematic primitives (*44*, *45*), the proposed MELA policy indirectly modulates the joint torques by changing the reference joint angles, where the resulting motions are the natural outcomes of the dynamic interactions with the environment. In contrast to other hierarchical learning approaches that select one policy representing one skill at a time (*34*, *35*), MELA continuously combines network parameters of all experts seamlessly and is therefore more responsive than other methods because there is no wait time due to disjointed switching of experts. In addition, because MELA synthesizes the experts in a high-dimensional feature space, i.e., weighted average of the network parameters (weights and biases of the neural networks), it does not suffer from the same expert imbalance problems as MoE (*41*). Similar multi-expert structures that blend experts in high-dimensional feature space were studied in computer graphics for kinematic animation (*41*, *46*, *47*) but have not yet been developed as feedback policies for the control of dynamical systems such as robots.

Our presented work contributes to a learning framework that (i) generates multiple distinctive skills effectively, (ii) diversifies expert skills using cotraining, and (iii) synthesizes multiskill policies with adaptive behaviors in unseen situations. The adaptive behaviors are verified by proof-of-concept experiments on a real robot and various extreme test scenarios in a physics simulation. By synthesizing multiple expert skills, the collective expertise of MELA is more versatile than each single expert, because of the dynamic structure of MELA, which integrates all experts based on the online state feedback. Such multi-expert learning allows each expert to specialize in unique locomotive skills, i.e., some prioritize postural control and failure recovery, while others acquire strategies for maximizing task performance. As a result, MELA can perform a broad range of adaptive motor skills in a holistic manner and is more versatile because of the diversification among experts.

The proposed framework is effective in achieving reactive and adaptive motor behaviors to changing situations consisting of unforeseen scenarios, unexpected disturbances, and different locomotion modes, which have not been addressed well by other unified frameworks in the literature. We will show the advantages of our proposed work, which can produce a variety of strategies to ensure task success. For example, the robot can produce quick responses to recover balance in different unexpected postures and stabilize dynamic transitions. This is an indicative level of machine intelligence—the ability to autonomously produce locomotive skills that would require substantial intelligence to design if they needed to be programmed by humans.

RESULTS

We summarize the results and the learning method in Movie 1. In the following sections, we report the results of adaptive locomotion behaviors, followed by technical details of the learning framework.

Multi-expert learning framework

We first define key terminology to help explain the concepts in this article: motor skill, expert, and locomotion mode. Motor skill (or "skill" for short): a feedback policy that generates coordinated actions to complete a specific type of task; this serves as a building block for constructing more complex maneuvers. Expert: a DNN with specialized motor skills. Locomotion mode: a pattern of coordinated limb movement, such as standing, turning on the spot, trotting forward/backward, steering left/right, and fall recovery.

To complete tasks in unseen scenarios, an artificial agent needs the ability to adapt relevant skills during runtime, which is why we propose the use of MELA: an HRL structure that consists of a collection of DNNs and a GNN. As shown in Fig. 2, the GNN continuously fuses expert DNNs into a single synthesized neural network at every time step by computing the weighted average of all experts' network parameters. The synthesized policy fully encodes the motor skills of the experts in a high-dimensional feature space (see Materials and Methods). Through repurposing and combining existing skills, MELA acquires a wide array of adaptive behaviors and achieves versatile locomotion in unseen scenarios.

The multi-expert policy of MELA is trained in a two-step process. In the first stage, we train a set of policies, where each one accomplishes a distinct task. Specifically, we use the trained experts, such as fall recovery and trotting experts, to initialize all experts by two subgroups. In the second stage, we use a GNN as the merging mechanism to fuse all network parameters and train all experts together so that their collective specializations can be fully used by the gating network. Meanwhile, the gating network is trained to learn the continuous and variable activation of different experts to generate optimal policies at each control loop (see Materials and Methods). The MELA policy was trained in the physics simulation and evaluated on a real robot system.

For constructing MELA, the number of experts is determined by the relation between the desired locomotion tasks and the required

Movie 1. Versatile locomotion. Adaptive behaviors learned by a MELA policy.

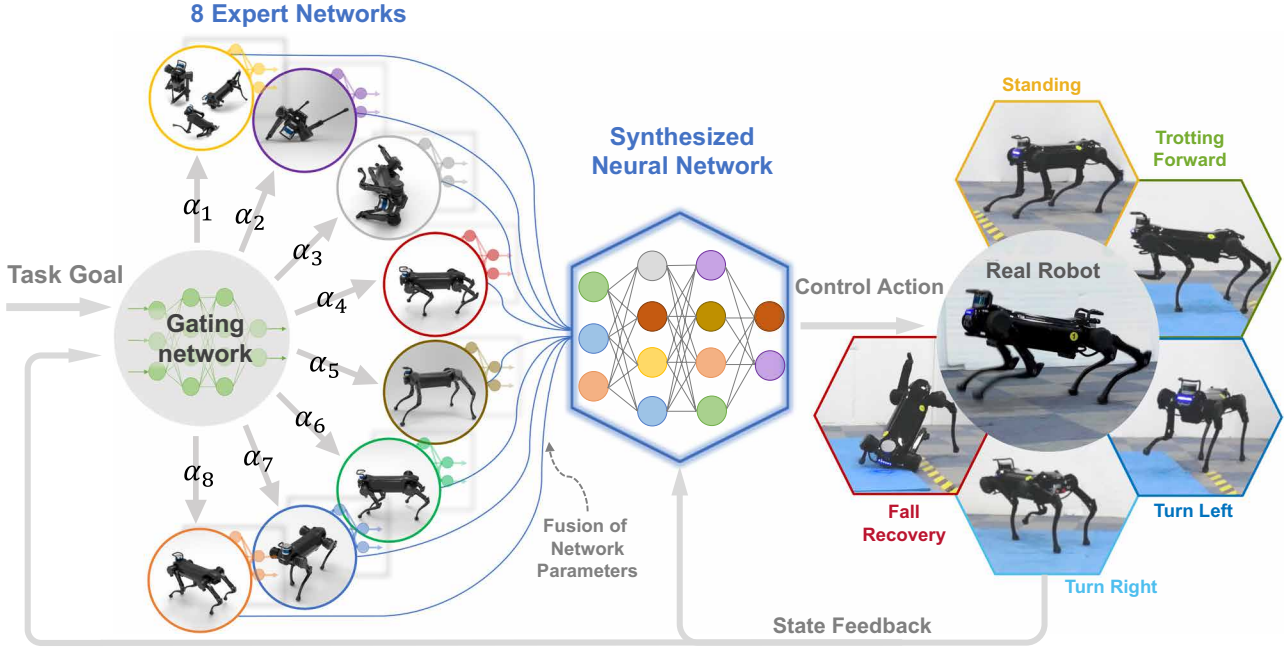

Fig. 2. MELA: A hierarchical DRL framework that generates adaptive behavior by combining multiple DNNs together to produce versatile locomotive skills. The GNN generates variable weights (α) to fuse the parameters of all eight expert networks (each expert is illustrated by its primary motor skill) such that newly synthesized motor skills are adapted to different locomotion modes by blending useful learned behaviors collectively from the consortium of experts.

skills. A particular motor skill corresponds to a distinct control strategy (e.g., to roll the body by pushing the ground), and thus, one locomotion task would require a set of different skills. In our study, we focus on five locomotion tasks (Fig. 2): fall recovery, standing, turning left, turning right, and trotting. All these movements need a variety of motor skills for interacting with the environment.

According to the configurations that the robot would encounter during these five tasks, a basic number of experts can be determined. There are seven distinct situations each requiring at least one motor skill, namely, (i) fall recovery from a supine pose (lying on the back), (ii) fall recovery from lateral decubitus poses (lying on the left/right side), (iii) balance control during stance, (iv) body postural control,

(v) trotting forward, (vi) left steering, and (vii) right steering. Hence, a minimum of seven DNNs can be used to represent these skills. We also introduced a redundant expert that can represent nonlinear features and additional skills that are difficult to anticipate. As a result, we constructed the MELA using eight experts in total. Further comparison found that using more than eight experts had no further performance but more training time (fig. S1). It shall be noted that the seven situations are only used as a guide to determine the number of experts at the initial design process and do not need to match the order of numbering of MELA's learned motor skills in the latter section.

Learning individual motor skills—Fall recovery and trotting

In challenging physical tasks, it is difficult to directly train control policies where a variety of skills are needed, such as recovering from falls and resuming walk gaits. Previous studies show that prelearning skills allow the experts to learn task representations; otherwise, the policies were not able to learn to solve more difficult composite tasks (*36*, *40*). Like training for sports, it is essential to practice individual skills that are distinctly different from each other. Similarly, MELA has a two-step training procedure for experts, in which two distinct, separate modes are specialized first: fall recovery and trotting. In the following, we will show the experimental results of fall recovery and trotting using individually trained neural networks and then present the multimodal locomotion experiments using MELA.

For quadrupeds, a canonically stable configuration is a standing posture with four feet forming a support polygon close to the body's length and width. For fall recovery, the DRL agent is rewarded for feedback policies that restore such stable postures from various failure states. We applied random initial configurations to explore diverse robot states and facilitated the agent's ability to generalize policies for various fall poses (see the Supplementary Materials).

We evaluated the robustness of recovery policies and categorized the learned reactive behaviors into four strategies (Fig. 3, A to C): (i) natural rolling exploiting semipassive movements, (ii) active righting and flipping, (iii) standing up from prone positions, and (iv) stepping. Natural rolling describes the behavior where the robot exploits its natural dynamics and gravity to roll over. This is activated when the robot is in a prone and/or lateral decubitus position, as shown in Fig. 3A. Active righting is the strategy where the robot pushes itself using the leg and elbow, creating momentum to actively flip itself to a prone position, as shown in Fig. 3B. Stepping emerges when necessary during standing to regain balance, involving coordination and switching of support legs. An example of stepping is shown in Fig. 3C, and such multicontact switching was all generated naturally using the learned policy based on the online state feedback.

From all fall recovery experiments (Fig. 3, A to C, and movies S1 and S2), we can see the responsive and versatile reactions, including the emerged stepping behavior. Compared with baseline fall recovery that is manually engineered with a fixed pattern (fig. S2), our learning policy is able to recover from various fall scenarios because it can respond to dynamic changes using online feedback, while the handcraft controller only addresses a narrow range of situations.

The Jueying robot can robustly trot under three ground conditions with different stiffness, friction, and obstacles (Fig. 3, D to F, and movie S3). In Fig. 3D, the concrete ground was covered by thin carpets with high friction, whereas in Fig. 3E, 2-cm-thick foam mats were laid, creating a softer and more slippery surface. In Fig. 3F, 5-cm-thick bricks were scattered as small obstacles. The learned motor skills were robust under different ground conditions, and the Jueying robot was able to continue trotting steadily in all three scenarios.

All these trained policies have exhibited behaviors of compliant interaction to handle physical interactions and impacts. The joint impedance mode offers the ability to indirectly regulate joint torques using the deviation between the desired joint position q^d and actual joint position q^m via the principle similar to the series elastic actuators, i.e., $\tau = K_p(q^d - q^m)$ (*48*). The expert has indirectly learned active compliance control by regulating the references based on feedback of the current joint positions to minimize joint torques.

Multi-expert skills for multimodal locomotion

Analysis of learned MELA policy

After the first stage of the two-stage training process, the network parameters of fall recovery and trotting policies are transferred into the expert networks in MELA. In the second stage, all experts are cotrained with the gating network, and MELA repurposes the initial experts to learn adaptive behaviors necessary for multimodal locomotion, while the gating network learns how to blend the acquired skills to respond to the changing tasks. As a result, MELA is able to achieve noncyclic and asymmetric motions (fall recovery), rhythmic movements (trotting), goal-oriented tasks (target-following), as well as all dynamical transitions between different modes. These key adaptive behaviors of the MELA policy were tested on a real robot, which demonstrated the capabilities of achieving a diversity of locomotion tasks, adapting to external environmental changes responsively, while also following user commands.

To analyze the inner workings of MELA, we performed systematic tests of the trained MELA networks in the physics simulation so as to fully cover the wide range of sensory inputs. Without hardware risks, we operated the robot in extremely dynamic motions to activate different experts, allowing analysis of all distinct motor skills. This provides data of variable weights that reflect the activation level of all experts over each motor skill. Figure 4A shows the correlations and patterns between the activation of experts and the motor skills and reveals that each motor skill has a dominant expert, suggesting the primary specialization of each MELA expert.

As shown in Fig. 4A, eight fundamental motor skills are acquired: (i) back righting, e.g., push elbow to roll over from supine positions (lying on the back); (ii) lateral rolling, e.g., retrieve legs and roll the body to a prone position (lying on the abdomen); (iii) postural control, e.g., maintain a nominal body posture; (iv) standing balancing, e.g., maintain stable stance and take steps when necessary; (v) turning left; (vi) turning right; (vii) trotting at small steps; and (viii) trotting at larger steps. These eight fundamental motor skills are the building blocks for MELA to compose variable skills and produce adaptive behaviors.

The skill specialization and distribution among experts emerge naturally through the MELA cotraining. Therefore, the order of the experts does not need to follow the numeration of the specialized motor skills. The primary motor skill specialization of experts 1 to 8 is turning right (skill vi), standing balance (skill iv), large-step trotting (skill viii), turning left (skill v), body posture control (skill iii), back righting (skill i), small-step trotting (skill vii), and lateral rolling (skill ii). As the result of introduced redundancy, expert 7 was exploited and trained as a complementary role in conjunction with expert 3 for trotting forward, i.e., experts 7 and 3 were specialized in trotting at small and large steps, respectively. An alternative

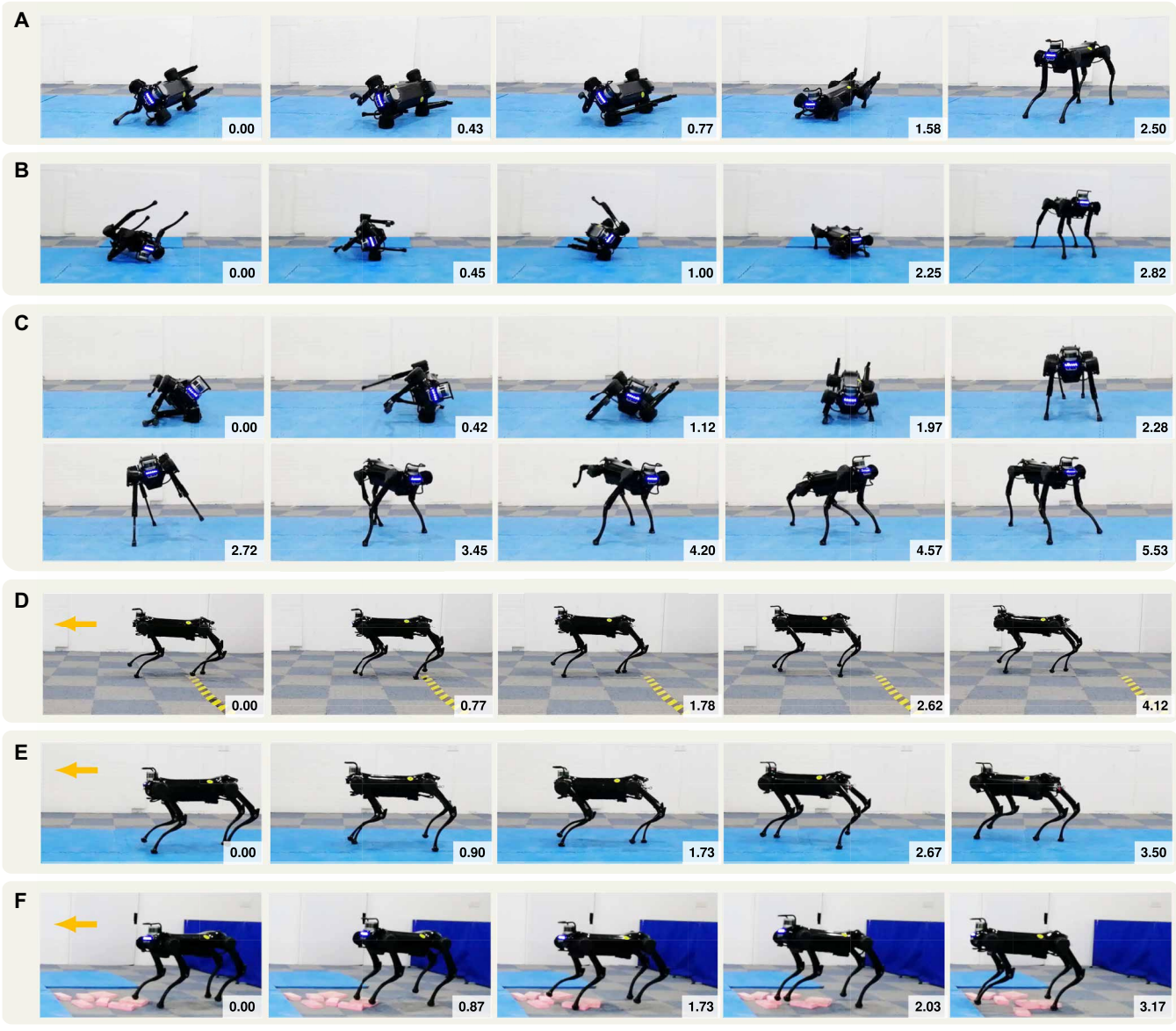

Fig. 3. Individual motor skills for the fall recovery and trotting. (**A**) A configuration between prone and lateral decubitus positions where legs were stuck underneath the body: The robot first pushed the ground to lift up the body for ground clearance and then retrieved legs to a prone posture for standing up. (**B**) The robot actively used elbow-push to generate a large momentum to self-right to a prone position. (**C**) A stepping behavior was learned and performed naturally to keep balance. (**D**) Stable trotting on a hard floor. (**E**) Stable trotting on soft slippery foam mats. (**F**) Stable trotting over scattered obstacles, showing the compliant interaction and robustness learned by the trotting expert. Time in snapshots is measured in seconds.

visualization can be found in fig. S3, where experts are sorted by activation patterns across different motor skills.

Analysis of skill adaptation and transfer

Apart from the skill specialization (Fig. 4A), we studied how skills are adapted and transferred in the MELA networks by using t-distributed stochastic neighbor embedding (t-SNE) to analyze coactions of the gating network and the expert networks, respectively. The t-SNE algorithm is a dimensionality reduction technique to embed and visualize high-dimensional data in a low-dimensional space. It first computes a conditional probability distribution, representing the similarities of samples in the original high-dimensional space based on a distance metric, and then projects samples to a low-dimensional space in a probabilistic manner. Therefore, similar output actions from the networks will appear with high probability in the same neighborhood as clustered points (Fig. 4, B to E), and vice versa.

In Fig. 4 (B and C), the t-SNE analysis on the outputs of the gating network (the variable weights for all experts) reveals the relationship between experts and the resulting motor skills and how the gating network synthesizes experts after the MELA learning process. There are two maps of clusters in Fig. 4 (B and C), which are grouped by dashed lines corresponding to locomotion (green) and fall recovery (bronze), suggesting that the gating network perceives these two as different modes. The t-SNE analysis is labeled according to the experts (Fig. 4B) and motor skills (Fig. 4C), respectively, where the clustered

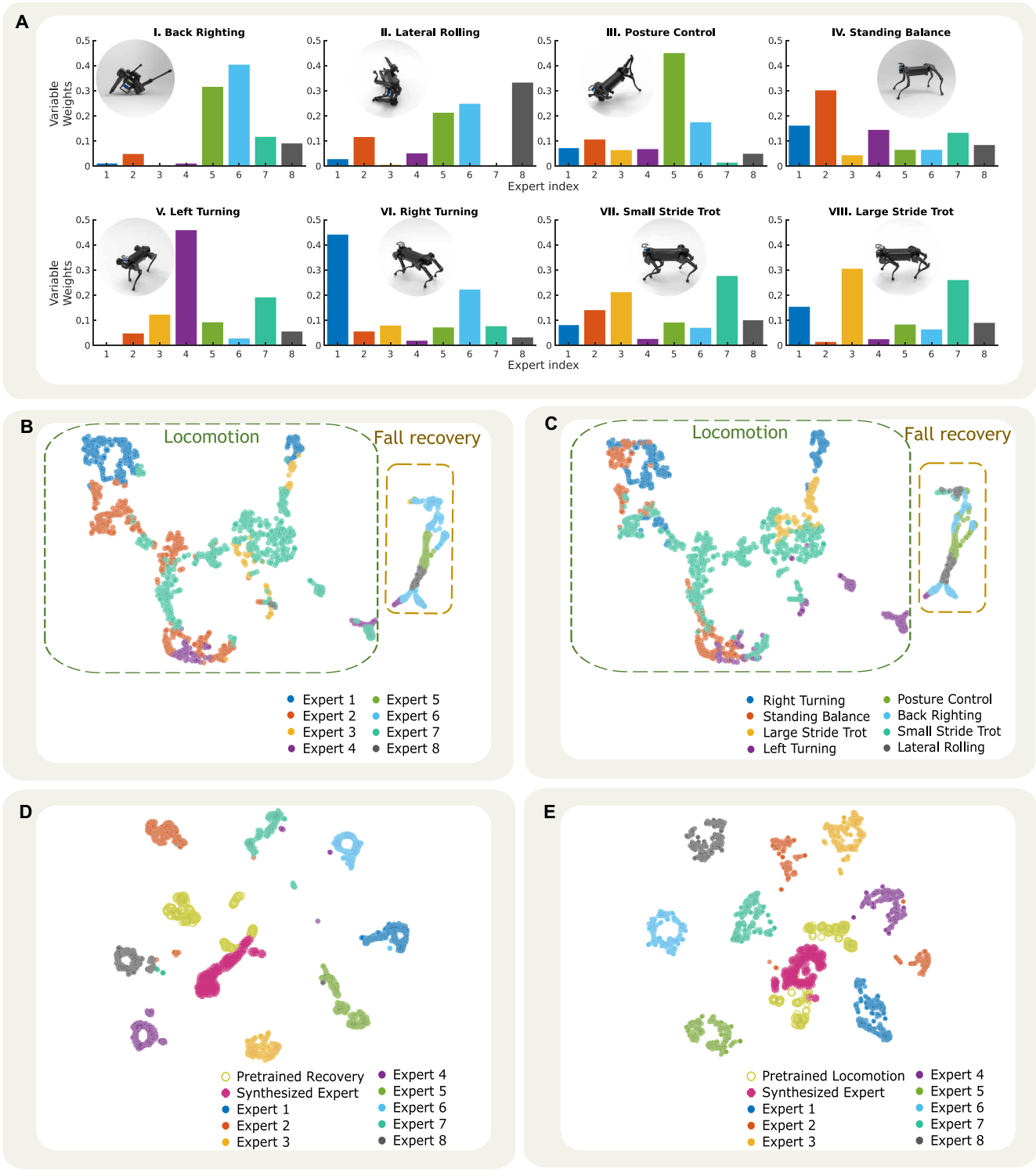

Fig. 4. Analysis of the specialization of experts across different motor skills, and the patterns of the gating network and the expert networks using the t-SNE. The analysis was based on the simulation tests using the policy from a single training run, which was representative because the training process can consistently reproduce policies with very similar characteristics of skill specialization. (**A**) Specialized activation of eight experts across different motor skills, where the distinct activation patterns indicate a unique specialization (table S1). (**B** and **C**) 2D projection of the gating network's activation pattern by t-SNE, where the neighborhoods and clusters of the samples are visualized. Samples representing similar activation appear in close proximity, whereas the different ones are distant from each other. (B) Output samples of the gating network, which are classified by the index of the dominant expert that has the highest activation (A). (C) Output samples of the gating network, which are classified by the physical states during a distinct locomotion mode, e.g., trotting, balancing, and turning left/right. (**D** and **E**) 2D projection of the actions from the pretrained, cotrained, and synthesized expert policies using t-SNE analysis: (D) and (E) are the target actions classified during fall recovery and trotting tasks, respectively.

samples have matching distributions mostly between these two maps, indicating that each expert's primary motor skill is in agreement with the activation patterns shown in Fig. 4A.

We also compared actions generated by all experts using t-SNE and revealed how multiple skills evolved and diversified after cotraining. As shown in Fig. 4 (D and E), the actions generated from eight experts are distant from each other, meaning that experts have been specialized toward more unique skills. The limited intersection between the clusters of the trained experts in stage 2 and the initial experts from stage 1 also implies that the trained experts have diversified from the original skills and further acquired more profound and newly emerged skills during MELA's cotraining stage. The data in Fig. 4 (D and E) show interesting results: The cluster of actions from the synthesized network intersects with those from the pretrained expert policies, meaning newly emerged behaviors of fall recovery and locomotion share some similarities with the original ones; the dynamically synthesized expert partially preserves the original skills, which are reconstructed by fusing eight distinctive experts.

Multiskill locomotion

To validate the performance of the MELA policy, we designed experiments that were safe to execute on the real robot with an increasing number of locomotion modes: (i) a single mode of fall recovery (Fig. 5A); (ii) a double mode of left-right steering on the spot (Fig. 5B); (iii) a triple mode of simultaneous left-right steering and trotting (Fig. 5C); and (iv) target-following locomotion involving all modes, i.e., standing, left-right steering, trotting, and fall recovery (movies S4 and S5).

In our study, adaptive behaviors refer to the online synthesized skills that adapt reactively to unseen situations. We summarize the adaptive behaviors achieved by MELA in two categories: (i) emerged skills that are newly acquired during training in stage 2 of MELA, i.e., skills for steering and turning (Fig. 5 and fig. S4A) and variable-speed trotting (fig. S5), and (ii) transitional skills that coordinate dynamical transitions smoothly between different locomotion modes, e.g., transition from various failure poses to trotting (Fig. 5 and fig. S4, B to E). Five representative cases of the adaptive behaviors from the MELA policy can also be found in fig. S4.

Figure 5A shows successful fall recovery performed by the MELA policy, and the similarity with those in Fig. 3 (A to C) indicates that the MELA policy has reused some pretrained skills. Figure 5B shows that the MELA policy was able to infer the heading direction from the target location and learned how to perform swift turning to track the changing target. During the left and right steering experiments (Fig. 5B), the average turning velocities were 1.6 rad/s (92.0 deg/s) and −1.1 rad/s (−61.7 deg/s) with peak values at 2.7 rad/s (156.8 deg/s) and −2.7 rad/s (−156.9 deg/s), respectively (fig. S6). Although the experts initialized in MELA were only for trotting and fall recovery, MELA was able to reshape the existing experts for the steering tasks as one of the newly emerged skills.

Figure 5 (C and D) shows target-following tasks requiring simultaneous trotting and steering on the real robot. The task was to chase a virtual target given by a user command, i.e., a variable position vector with respect to the robot. In Fig. 5C, a small target position ahead of the robot was provided, e.g., 0.28 m in the heading direction (fig. S7A), and the robot performed left/right turning while trotting forward. In the experiment shown in Fig. 5D, a farther target position of 0.48 m was commanded (fig. S7B), and the robot was chasing a distant target and trotting at larger steps. Torque saturation of motors occurred more often on the laboratory floor (fig. S8), and the robot had three successive tripping and recovery incidents. Key snapshots around falling and reactive responses are in Fig. 5D. Once the states of gait failures were sensed, MELA was able to produce immediate reactions to restore balance within one second (fig. S9), and the robot recovered from tripping and continued trotting without human intervention. Figure 5E shows an outdoor experiment where the robot first recovered from a falling posture and was later knocked down during a long walk on the grass. MELA was able to recover the robot from fall and resume trotting successfully (fig. S10). From all experiments, the synthesized MELA expert demonstrated flexible motor skills, coordinated movements, and smooth transitions, showing how crucial it is to have such reactive responses and feedback control for robust and autonomous locomotion.

As shown in Fig. 5, MELA enabled the robot to complete all validation successfully and demonstrated dynamic fall-resilient locomotion (see more in fig. S11). The gating network in MELA has learned how to generate variable weights for all experts in response to the state feedback and to provide smooth transitions across all modes; see data analysis of Fig. 5 (D and E) cases in figs. S12 and S13. Meanwhile, all the trained experts were activated coherently to collaborate with each other under the regulation of the gating network to synthesize an optimal skill suited for the situation. Additional analysis of the gating pattern and the relationship between experts is presented in the Supplementary Materials.

To further evaluate the performance, the MELA policy was validated by additional test scenarios in simulation that were not encountered during training, including gravel, inclined surfaces, a moving slope, rough terrain (fig. S14), as well as robustness tests with variations of masses and motor failures (fig. S15). During successful locomotion in these unseen scenarios, the MELA policy performed versatile adaptations to unexpected situations (movies S6 and S7). We note that the MELA framework has learned to deal with various transitions between different locomotion modes (figs. S16 to S20), and the synthesized policy is different from the eight basic motor skills, which indicates a nonlinear interpolated behavior among expert skills (fig. S21). All these experiments and simulations validate MELA's capability of producing flexible behaviors in a variety of unseen scenarios.

DISCUSSION

This study aims to achieve versatile robot motor skills in contact-rich multimodal locomotion. In contrast to most solutions that are dedicated to separate narrow-skilled tasks, we approached this challenge with a hierarchical control architecture of multi-expert learning—MELA—which is able to generate adaptive motor skills and achieve a breadth of locomotion expertise. In particular, MELA learns to generate adaptive behaviors from trained expert skills by dynamically fusing a new synthesized neural network, i.e., a feedback policy that reacts quickly to unforeseen situations. This is essential for autonomous robots to respond rapidly in critical conditions and is more useful for mission success in real-world applications.

In comparison with MoE, MELA's approach of fusing network parameters prevents the expert imbalance problem and provides diversity among expert skills. As a result, all experts are required to have the same neural network structure for implementing MELA. The training of MELA is a two-stage process, with an initialization of fall recovery and trotting policies at the first stage and

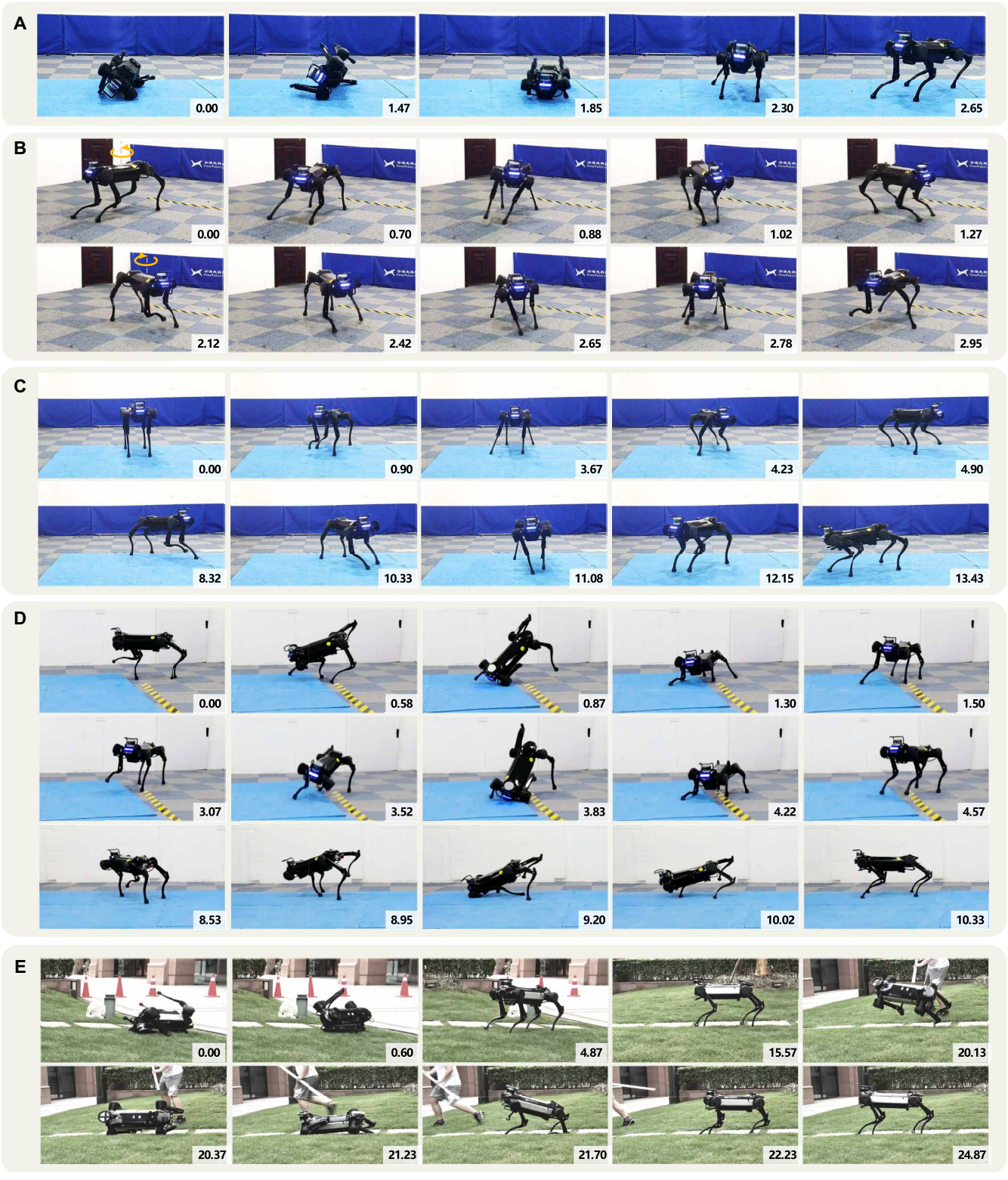

Fig. 5. Dynamically synthesized MELA policy running on a real quadruped robot and demonstrating adaptive and agile locomotion behaviors. (**A**) Successful fall recovery performed by the MELA expert, inheriting original skills from the pretrained experts. (**B**) Newly emerged skills, dynamic steering on the spot, naturally learned through the MELA framework. (**C**) Target-following experiment with simultaneous trotting and steering using online synthesized skills. (**D**) Target-following experiment showing the capability of failure-resilient trotting and critical recovery within one second (average of 0.5 s for restoring body posture and 0.4 s for returning to the trotting mode). (**E**) Adaptive trotting and fall recovery experiment on a sloped grass field, showing the capability of robust traversal in an unstructured environment. Time in snapshots is measured in seconds.

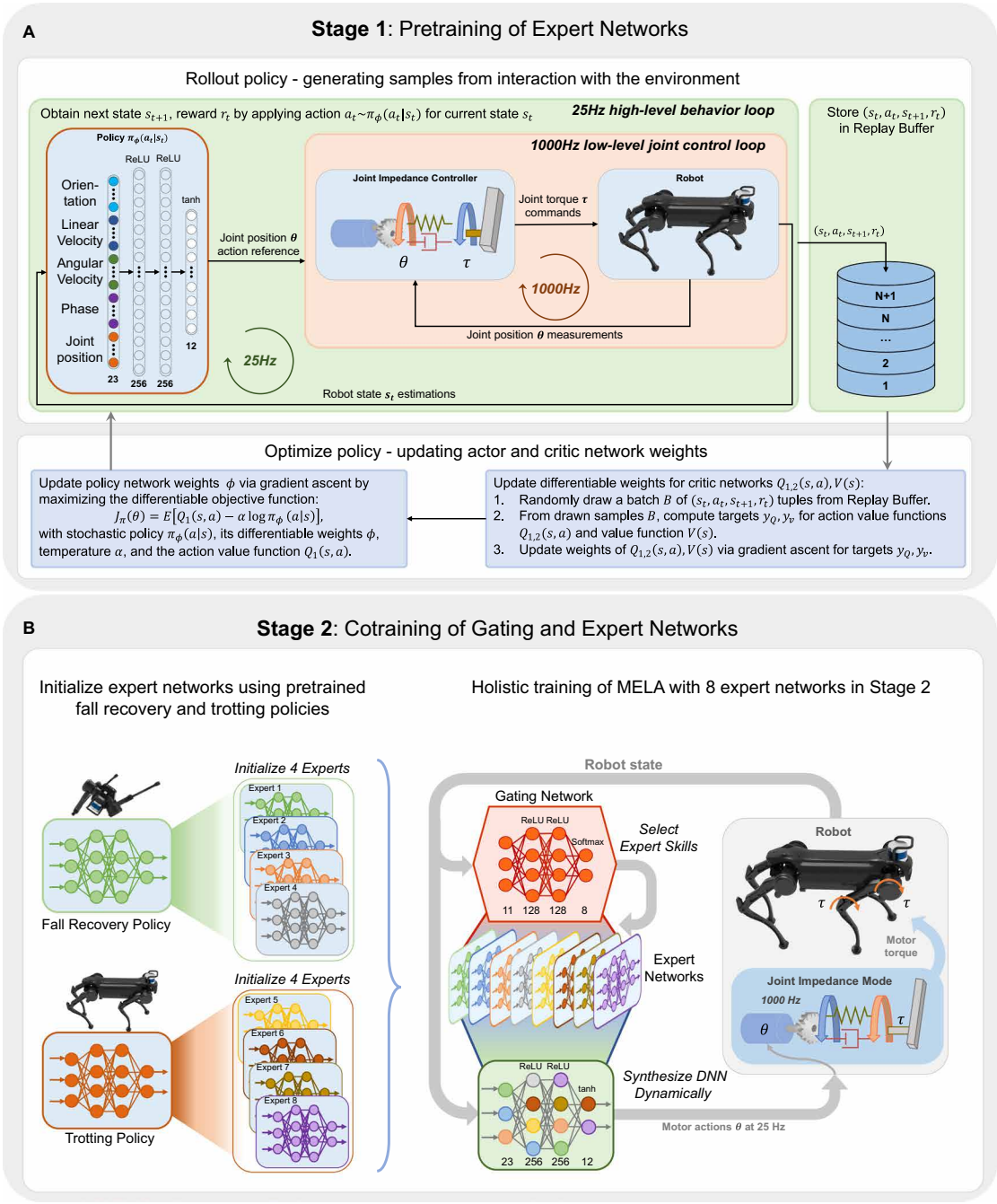

Fig. 6. Two-stage training of MELA. (**A**) In stage 1, the fall recovery and trotting policies are individually trained. (**B**) In stage 2, the pretrained trotting and fall recovery policies from stage 1 are used to initialize two evenly distributed groups of experts, each containing four experts. All these expert networks are cotrained together with the gating network.

the multi-expert cotraining with the gating network at the second stage. The t-SNE analysis of all expert networks and the gating network suggests that the consortium of multiple experts have expanded the initial pretrained expert skills and acquired more distinct and diverse skillsets; the high-level gating network has learned to distinguish each expert and blend the weights of each specialization according to different conditions; and the composed synthesized MELA expert partially preserves some of the original skills. We implemented the MoE approach using the same pretrained experts and training procedure, and the MoE policy exhibited degenerated experts in steering skills, which led to asymmetric gait and turning behavior (note S3 and fig. S22).

The experimental and simulation results outline MELA's key contributions in learning a variety of adaptive behaviors from specialized experts, the adaptation to changing environments, and the robustness against uncertainties. The experimental results show that MELA achieved multimodal locomotion with agile adaptation and fast responses to different situations and perturbations, i.e., smooth transitions between standing balancing, trotting, turning, and fall recovery. As a learning-based approach, MELA leverages computational intelligence and shows the advantage of generating adaptive behaviors compared with traditional approaches that purely rely on explicit manual programming.

Limitations and future work

Although our current MELA scheme is able to generate adaptive policies, it has no visual and haptic perception, which is critical for long-term motion planning (*49*), dynamic maneuvers (*50*), and the use of affordance to coordinate whole-body support poses (*51*). To acquire more advanced motion intelligence in unstructured environments, future research needs to integrate visual cues and haptic sensing to develop environment-aware locomotion.

While scaling up the number of locomotion modes, training in physics simulation may impose some limitations. Although all policies were validated by the Jueying robot in five locomotion modes,

the discrepancy between the simulation and the real world may accumulate and arise as an issue, when the number of tasks increases. Because the scope of this research is to achieve a diversity of reactive skills rather than sim-to-real transfer, we performed training in simulation and avoided potential damage to the real 40-kg robot during the exploration of the learning algorithm. On the basis of the results of MELA, the future work will be on the learning algorithms that can refine motor skills safely on real hardware for more complex multimodal tasks.

MATERIALS AND METHODS

In this section, we will first introduce the robot platform and then explain the core design of the MELA framework, including reward terms, state observations, and action space. Particularly, we will present an emulation of the frequency response of actuators and a loss function design for producing smooth and feasible actions. Last, we elaborate on the MELA framework and the two-stage MELA training procedure.

Robot platform

We implemented our learning algorithms on the Jueying quadruped robot (*52*) to validate the adaptive behaviors with feasible and safe tests on the real hardware. Jueying has 3 DoFs per leg (12 DoFs in total), which are actuated by brushless electric motors with low gear ratio (i.e., 7:1) and high-fidelity joint torque control (table S2).

DRL framework

The goal of the DRL is to train an artificial agent to infer optimal actions from the current state by learning from past experience. The experience samples are stored as tuples containing the current state s_t, the action generated from policy $a_t \sim \pi_t(s_t)$, the reward r_t, and the next transition state s_{t+1}. These samples are first collected and stored in a replay buffer, from which they are drawn later to train the policy. We used the Soft Actor Critic (SAC) algorithm (*53*), and below are the details of reward, state, action, training procedures, and loss function.

Reward design

For training individual tasks, such as fall recovery, trotting, and target-following, we designed a specific reward function with corresponding weights for reward terms that represent different physical quantities. The full list of reward terms is (i) base pose, (ii) base height, (iii) base velocity, (iv) joint torque regularization, (v) joint velocity regularization, (vi) body ground contact, (vii) foot ground contact, (viii) yaw velocity, (ix) swing and stance, (x) average foot placement, (xi) reference joint position, (xii) reference foot contact, (xiii) robot's heading to the goal, and (xiv) the goal position. Different tasks require a specific subset of reward components, i.e., fall recovery requires rewards (i) to (vii), trotting requires rewards (i) to (xii), and multimodal target-following locomotion requires all 14 reward terms. In summary, the first seven reward terms are common physical quantities across all tasks to ensure stable robot motion, whereas the other terms are task-specific. The mathematical formulations of all reward terms and the task-specific weights are in tables S3 and S4.

State observation

We used the following state observations that are essential and minimalistic to train successful policies: (i) base (robot body) orientation, (ii) angular velocity of the robot base, (iii) linear velocity of the robot base, (iv) joint positions, (v) phase vector, and (vi) goal position (*54*). The body orientation is represented as a normalized gravity vector projected in the robot local frame using the measurements from the inertial measurement unit (IMU). The angular velocities of the body and all the joint positions were measured by the IMU and joint encoders, respectively. The linear velocity was obtained from the state estimation by fusing the leg kinematics and the accelerations from IMU (as a strap-down inertial navigation system) and then transformed to the heading coordinate, so the resulting velocity is agnostic to the heading direction. The two-dimensional (2D) phase vector was designed to clock along the unit circle to describe the phase of the periodic trotting (fig. S23). Last, the target position is represented by a relative 3D vector with respect to the robot's local frame, and only the horizontal components are used as the state inputs. The detailed combination of the state observations for different tasks and networks is in table S5.

Action space

The benchmark of DRL-based locomotion (*32*, *55*) shows a suitable configuration for the action space that yields better performance and faster learning due to the compliant interaction: A DRL agent provides joint references and an impedance mode for controlling the joint. This setting suits the equilibrium-point control, and hence, we adopted such design of the action space here. To guarantee smooth and feasible actions for the real robot, we developed two important techniques: (i) the use of low-pass filters to emulate the characteristics of the frequency response of actuators, which enforces physically realizable reference motions, and (ii) the design of a particular loss function to generate smooth and nonjerky joint references and torques. We named these two techniques action filtering and smoothing loss, respectively.

Action filtering

Real actuators have limited control bandwidth, and hence, the references with frequencies higher than the bandwidth cannot be tracked. However, a common issue in simulation is that the learning policy takes advantage of the pure torque source with unlimited control bandwidth: exploitation of abrupt and jerky motions that are only possible in simulation to maximize the reward but infeasible on real systems. The difference between ideal actuators (pure torque source) in simulation and real actuators (restricted bandwidth, torque, speed, and power) needs to be addressed appropriately in the learning framework. In addition to the basic position and velocity limit (*56*), we performed action filtering using a first-order Butterworth filter to emulate the frequency response of real motors and to guide the policy to learn a smoother and more feasible behavior.

For the Jueying robot, we found that emulating the frequency response and setting the speed limit are sufficient to represent realistic characteristics of actuators. The properties of Jueying's actuators, such as good torque tracking control and low gear ratio that results in decoupled inertia and minimal gear friction, avoid the need for modeling detailed actuator properties and simplify the simulation setting. During the simulation training, we emulated the limited frequency response of actuators by applying action filtering on the output action with the cutoff frequency of 5 Hz, which was higher than the 1.67-Hz trotting gait (fig. S24). This provides realistic restriction of high-frequency actions and prevents the policy from overexploiting risky motions while still permitting necessary movements for dynamic tasks. For safety reasons in real experiments, we applied a more conservative cutoff frequency of 3 Hz in case of unexpected jerky references. As a result, all obtained policies exhibited

smooth motions within the bandwidth (fig. S24) that can be executed on the real robot directly and safely.

Smoothing loss

The action filtering alone may not always guarantee feasible motions because it only limits the frequency of the DNN output but not the magnitude, so the learning process may still explore and exploit low-frequency but large-amplitude motions regardless. Therefore, we further designed the smoothing loss based on the principle of minimal interaction to minimize the applied torques (*42*).

Biological studies show that when the CNS resets a new equilibrium, the displacement between the equilibrium and the actual position will activate a neuromuscular response that tries to reduce the muscular activity (torque). This principle of minimal interaction serves as a biological foundation of studying the proposed smoothing loss, which is effective for smoothing the exerted torque, i.e., $\tau = K_p(q^d - q^m)$. To guide the policy and generate actions following the minimal interaction principle, we designed a smoothing loss function $J_{smoothing}$ as

$$J_{smoothing}(\mu(s_t)) = \|\mu(s_t) - q\|_2 \quad (1)$$

where $\mu(s_t)$ values are the deterministic mean outputs of the stochastic policy used as the target joint references and the values of q are the measured joint positions. The smoothing loss $J_{smoothing}$ is the objective function that minimizes the differences between the target $\mu(s_t)$ and the measurement q. Because the joint references are the inputs for impedance control, this minimization leads to more gentle torque profiles, thus encouraging the learning of strategies with the least effort as possible.

The proposed smoothing loss $J_{smoothing}$ is incorporated into the SAC training loss $J_{SAC}(\pi)$ and is used for backpropagation of the neural networks, instead of being part of the reward function. Adding the smoothing loss term to $J_{SAC}(\pi)$ allows the information (the causality of actions) to backpropagate directly through the neural network and to bypass the process of reward bootstrap and Q-function approximation. Because, for training the policy, the Q-function requires iterations to obtain a valid and accurate enough approximation of the expected return, our approach of bypassing the Q-function avoids the wait time and permits the information to backpropagate within the first few iterations.

MELA training procedure

Figure 6 depicts both the network architecture and training procedure of our MELA framework. The MELA network consists of one gating network and eight expert networks (Fig. 6B). The gating network has two hidden layers with 128 neurons each, using a rectified linear unit (ReLU) as the activation function. All expert networks have two hidden layers with 256 neurons each and use a ReLU activation function, which has the same network structure as that of the pretrained expert policy networks shown in Fig. 6A.

Here, we elaborate on MELA's two-stage training procedure. In stage 1, successful trotting and fall recovery policies were pretrained using the scheme shown in Fig. 6A. In stage 2, MELA first initialized two groups of expert networks (four in each group) by copying the weights and bias from the two pretrained experts (Fig. 6B) and randomly initialized the weights and bias of the gating network. Then, MELA embedded all the experts together with the gating network and cotrained all of them with diverse samples. During the cotraining in stage 2, the gating network needed to learn how to compute correct activations for all experts and synthesize a new skill-adaptive network, as illustrated in Fig. 6B. The robot feedback states were the input of the synthesized network for generating motor actions.

Following the framework in Fig. 6, both stages were trained using the SAC algorithm (note S2), and the samples were collected at 25-Hz frequency while the actions were executed through the impedance control at 1000-Hz frequency. Both training stages initialized the robot in diverse configurations during the simulation episodes so as to increase the diversity of the collected samples, and additional details are described in note S2. The learning curves during stages 1 and 2 of the MELA training can be found in fig. S25.

All neuron connections within MELA are differentiable, including those between the gating network and expert networks. This allows every network weight and bias to update through backpropagation simultaneously. Thus, all the MELA networks can be trained with the same backpropagation techniques used for fully connected neural networks. The actor for SAC was encoded using a MELA network, whereas the critic consisting of two Q-functions was encoded as fully connected neural networks that were adopted from double Q-learning to prevent overestimation (*57*, *58*).

Let x, y, h denote the dimensions of the input, the output, and the hidden layer, respectively; let W and B be the network's weights and bias, respectively. The parameter set of the skill-adaptive network is

$$\psi_{synth} = \{W_0 \in \mathbb{R}^{x\times h}, W_1 \in \mathbb{R}^{h\times h}, W_2 \in \mathbb{R}^{h\times y}, B_0 \in \mathbb{R}^{x}, B_1 \in \mathbb{R}^{h}, B_2 \in \mathbb{R}^{y}\} \quad (2)$$

and the parameter set of each individual expert network is

$$\psi^n_{expert} = \{W^n_0 \in \mathbb{R}^{x\times h}, W^n_1 \in \mathbb{R}^{h\times h}, W^n_2 \in \mathbb{R}^{h\times y}, B^n_0 \in \mathbb{R}^{x}, B^n_1 \in \mathbb{R}^{h}, B^n_2 \in \mathbb{R}^{y}\} \quad (3)$$

During runtime, the weights W and bias B are fused by the weighted sum formulation as

$$W_i = \sum_{n=1}^{8} \alpha_n W^n_i, B_i = \sum_{n=1}^{8} \alpha_n B^n_i \quad (4)$$

where $n = 1, \ldots, 8$ is the index of experts, $i = 0, 1, 2$ is the index of the corresponding layer, and $\alpha_n \in [0,1]$ are the variable weights generated by the gating network.

The fused weights W and bias B are used to construct the synthesized network dynamically during runtime using the following equation:

$$\Phi_{synth} = Tanh(W_2 ReLU(W_1 ReLU(W_0 X + B_0) + B_1) + B_2) \quad (5)$$

where $X \in R^x$ is the input parameter and $Tanh(*)$ and $ReLU(*)$ are the nonlinear activation functions. The sum of the eight variable weights α_n, $(n = 1, \ldots, 8)$ is normalized to 1 using a Softmax function, also known as normalized exponential function. There are several nonlinear features in the blending process: Each expert DNN is a nonlinear control policy by nature, and each blending of weight is produced by a nonlinear rescaling of the output of the gating network with a Softmax function, which normalizes different values of the original sum. Therefore, the resulting synthesized expert is a highly nonlinear control policy—a nonlinear mapping between the feedback states and actions that is required to deal with challenging scenarios.

SUPPLEMENTARY MATERIALS

robotics.sciencemag.org/cgi/content/full/5/49/eabb2174/DC1
Note S1. Data analysis of MELA learning results.
Note S2. Additional Materials and Methods.
Note S3. Expert imbalance phenomenon.
Fig. S1. Comparison of MELA's learning curves using different numbers of expert networks.
Fig. S2. Baseline experiments of fall recovery and trotting from engineered controllers.
Fig. S3. Activation patterns of experts across all motor skills.
Fig. S4. Five representative cases showing adaptive behaviors of the MELA expert under unseen situations in simulation.
Fig. S5. Forward trotting velocity during the variable-speed trotting simulation.
Fig. S6. Heading angle and angular velocity during the steering experiment on the real robot.
Fig. S7. Relative target positions with respect to the robot from the user command as the input to the MELA networks during the real locomotion experiment.
Fig. S8. Measured torques of the front left leg during the real locomotion experiment (Fig. 5D).
Fig. S9. Roll and pitch angles during the real locomotion experiment.
Fig. S10. Target position, body orientation, and velocity during the real locomotion experiment on grass (Fig. 5E).
Fig. S11. Adaptive and agile locomotion behaviors on pebbles and grass demonstrated by MELA policy.
Fig. S12. Continuous and variable weights of all experts during the real MELA experiment (Fig. 5D).
Fig. S13. Continuous and variable weights of all experts during the real MELA experiment on grass (Fig. 5E).
Fig. S14. Four types of unseen terrains for testing the multiskill MELA policy in the simulation.
Fig. S15. Simulated test scenarios for evaluating the robustness of the MELA policy.
Fig. S16. Representative adaptive behavior from the simulated scenario of steering on spot (fig. S4A and movie S6).
Fig. S17. Representative adaptive behavior from the simulated scenario of steering while recovering to trotting (fig. S4B and movie S6).
Fig. S18. Representative adaptive behavior from the simulated scenario of tripping (fig. S4C and movie S6).
Fig. S19. Representative adaptive behavior from the simulated scenario of a large impact (fig. S4D and movie S6).
Fig. S20. Representative adaptive behavior from the simulated scenario of falling off a cliff (fig. S4E and movie S6).
Fig. S21. Analysis of responses from the MELA policy during the simulated scenario of a large external perturbation (fig. S4D and movie S6).
Fig. S22. Phenomenon of asymmetric gait and imbalanced experts from MoE.
Fig. S23. Illustration of the 2D phase vector for training the locomotion policy.
Fig. S24. Normalized power spectrum analysis of motions during the real locomotion experiment (without the DC component).
Fig. S25. Learning curves during the two-stage training of MELA and MoE.
Fig. S26. Nine distinct configurations used as the initialization for training fall recovery policies in simulation.
Fig. S27. Setting of the target location for training MELA policies in simulation.
Table S1. Distribution matrix of expert specializations over motor skills.
Table S2. Specification of the Jueying quadruped robot.
Table S3. Detailed descriptions of the individual reward terms.
Table S4. Weights of the reward terms for different tasks.
Table S5. Selection of state inputs for different tasks and neural networks.
Table S6. Proportional-derivative parameters for the joint-level PD controller.
Table S7. Hyperparameters for SAC.
Movie S1. Experiments of fall recovery.
Movie S2. Experiments of outdoor fall recovery and compliant interactions.
Movie S3. Experiments of trotting.
Movie S4. Experiments of adaptive locomotion and behaviors.
Movie S5. Experiments of outdoor fall-resilient locomotion on irregular terrains.
Movie S6. Simulation of representative adaptive behaviors from the multiskill MELA expert.
Movie S7. Simulation of extensive scenarios and crash tests.
Movie S8. Baseline experiments of fall recovery and trotting from default controllers.
References (*54*–*64*)

REFERENCES AND NOTES

1. A. J. Ijspeert, A. Crespi, D. Ryczko, J.-M. Cabelguen, From swimming to walking with a salamander robot driven by a spinal cord model. *Science* **315**, 1416–1420 (2007).
2. T. Drew, J. Kalaska, N. Krouchev, Muscle synergies during locomotion in the cat: A model for motor cortex control. *J. Physiol.* **586**, 1239–1245 (2008).
3. M. Mischiati, H.-T. Lin, P. Herold, E. Imler, R. Olberg, A. Leonardo, Internal models direct dragonfly interception steering. *Nature* **517**, 333–338 (2015).
4. S. K. Karadimas, K. Satkunendrarajah, A. M. Laliberte, D. Ringuette, I. Weisspapir, L. Li, S. Gosgnach, M. G. Fehlings, Sensory cortical control of movement. *Nat. Neurosci.* **23**, 75–84 (2020).
5. H. Markram, The blue brain project. *Nat. Rev. Neurosci.* **7**, 153–160 (2006).
6. S. Gay, J. Santos-Victor, A. Ijspeert, Learning robot gait stability using neural networks as sensory feedback function for central pattern generators, in *Proceedings of the 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2013), pp. 194–201.
7. DARPA Robotics Challenge (DRC); https://www.darpa.mil/program/darpa-robotics-challenge.
8. C. G. Atkeson, B. P. W. Babu, N. Banerjee, D. Berenson, C. P. Bove, X. Cui, M. DeDonato, R. Du, S. Feng, P. Franklin, M. Gennert, J. P. Graff, P. He, A. Jaeger, J. Kim, K. Knoedler, L. Li, C. Liu, X. Long, T. Padir, F. Polido, G. G. Tighe, X. Xinjilefu, No falls, no resets: Reliable humanoid behavior in the DARPA Robotics Challenge, in *Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2015), pp. 623–630.
9. N. A. Bemstein, *The Co-Ordination and Regulation of Movements* (Pergamon, 1967).
10. M. L. Latash, Stages in learning motor synergies: A view based on the equilibrium-point hypothesis. *Hum. Mov. Sci.* **29**, 642–654 (2010).
11. J. Ramos, S. Kim, Dynamic locomotion synchronization of bipedal robot and human operator via bilateral feedback teleoperation. *Sci. Robot.* **4**, eaav4282 (2019).
12. D. Dimitrov, A. Sherikov, P. Wieber, A sparse model predictive control formulation for walking motion generation, in *Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2011), pp. 2292–2299.
13. H.-W. Park, P. M. Wensing, S. Kim, Online planning for autonomous running jumps over obstacles in high-speed quadrupeds, in *Proceedings of Robotics: Science and Systems (RSS)* (2015).
14. 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 *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 1–9.
15. S. Feng, E. Whitman, X. Xinjilefu, C. G. Atkeson, Optimization-based full body control for the DARPA Robotics Challenge. *J. Field Robot.* **32**, 293–312 (2015).
16. M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, M. Hoepflinger, ANYmal—A highly mobile and dynamic quadrupedal robot, in *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2016), pp. 38–44.
17. M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, J. Buchli, Whole-body nonlinear model predictive control through contacts for quadrupeds. *IEEE Robot. Autom. Lett.* **3**, 1458–1465 (2018).
18. H. Dai, A. Valenzuela, R. Tedrake, Whole-body motion planning with centroidal dynamics and full kinematics, in *Proceedings of the 2014 IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (IEEE, 2014), pp. 295–302.
19. A. W. Winkler, C. D. Bellicoso, M. Hutter, J. Buchli, Gait and trajectory optimization for legged systems through phase-based end-effector parameterization. *IEEE Robot. Autom. Lett.* **3**, 1560–1567 (2018).
20. I. Chatzinikolaidis, Y. You, Z. Li, Contact-implicit trajectory optimization using an analytically solvable contact model for locomotion on variable ground. *IEEE Robot. Autom. Lett.* **5**, 6357–6364 (2020).
21. A. Cully, J. Clune, D. Tarapore, J.-B. Mouret, Robots that can adapt like animals. *Nature* **521**, 503–507 (2015).
22. E. O. Neftci, B. B. Averbeck, Reinforcement learning in artificial and biological systems. *Nat. Mach. Intell.* **1**, 133–143 (2019).
23. K. Bouyarmane, S. Caron, A. Escande, A. Kheddar, Multi-contact planning and control, in *Humanoid Robotics: A Reference* (Springer, 2019), pp. 1763–1804.
24. B. Siciliano, O. Khatib, *Springer Handbook of Robotics* (Springer, 2016).
25. M. Posa, thesis, Massachusetts Institute of Technology (2017).
26. R. Bellman, Dynamic programming. *Science* **153**, 34–37 (1966).
27. X. B. Peng, G. Berseth, K. Yin, M. Van De Panne, DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Trans. Graph.* **36**, 41:1–41:13 (2017).
28. X. B. Peng, P. Abbeel, S. Levine, M. van de Panne, DeepMimic: Example-guided deep reinforcement learning of physics-based character skills. *ACM Trans. Graph.* **37**, 143:1–143:14 (2018).
29. J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, V. Vanhoucke, Sim-to-Real: Learning agile locomotion for quadruped robots, in *Proceedings of Robotics: Science and Systems (RSS)* (2018).
30. T. Li, H. Geyer, C. G. Atkeson, A. Rai, Using deep reinforcement learning to learn high-level policies on the ATRIAS biped, in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)* (IEEE, 2019), pp. 263–269.
31. Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst, M. Panne, Learning locomotion skills for Cassie: Iterative design and sim-to-real, in *Proceedings of the Conference on Robot Learning (CoRL)* (PMLR, 2020), pp. 100:317–329.

32. 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).
33. A. G. Barto, S. Mahadevan, Recent advances in hierarchical reinforcement learning. *Discrete Event Dyn. Syst.* **13**, 41–77 (2003).
34. K. Frans, J. Ho, X. Chen, P. Abbeel, J. Schulman, Meta learning shared hierarchies, in *Proceedings of the 2018 International Conference on Learning Representations (ICLR)* (2018).
35. J. Merel, A. Ahuja, V. Pham, S. Tunyasuvunakool, S. Liu, D. Tirumala, N. Heess, G. Wayne, Hierarchical visuomotor control of humanoids, in *Proceedings of the 2018 International Conference on Learning Representations (ICLR)* (2018).
36. T. Haarnoja, K. Hartikainen, P. Abbeel, S. Levine, Latent space policies for hierarchical reinforcement learning, in *Proceedings of the 35th International Conference on Machine Learning* (PMLR, 2018), pp. 1851–1860.
37. R. A. Jacobs, M. I. Jordan, S. J. Nowlan, G. E. Hinton, Adaptive mixtures of local experts. *Neural Comput.* **3**, 79–87 (1991).
38. K. Mülling, J. Kober, O. Kroemer, J. Peters, Learning to select and generalize striking movements in robot table tennis. *Int. J. Robot. Res.* **32**, 263–279 (2013).
39. X. Chang, T. M. Hospedales, T. Xiang, Multi-level factorisation net for person re-identification, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2018), pp. 2109–2118.
40. X. B. Peng, M. Chang, G. Zhang, P. Abbeel, S. Levine, MCP: Learning composable hierarchical control with multiplicative compositional policies, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2019), pp. 3686–3697.
41. H. Zhang, S. Starke, T. Komura, J. Saito, Mode-adaptive neural networks for quadruped motion control. *ACM Trans. Graph.* **37**, 145:1–145:11 (2018).
42. A. G. Feldman, V. Goussev, A. Sangole, M. F. Levin, Threshold position control and the principle of minimal interaction in motor actions. *Prog. Brain Res.* **165**, 267–281 (2007).
43. E. Bizzi, N. Hogan, F. A. Mussa-Ivaldi, S. Giszter, Does the nervous system use equilibrium-point control to guide single and multiple joint movements? *Behav. Brain Sci.* **15**, 603–613 (1992).
44. F. L. Moro, N. G. Tsagarakis, D. G. Caldwell, A human-like walking for the COmpliant huMANoid COMAN based on CoM trajectory reconstruction from kinematic Motion Primitives, in *Proceedings of the 2011 11th IEEE-RAS International Conference on Humanoid Robots (Humanoids)* (IEEE, 2011), pp. 364–370.
45. A. T. Sprowitz, M. Ajallooeian, A. Tuleu, A. J. Ijspeert, Kinematic primitives for walking and trotting gaits of a quadruped robot with compliant legs. *Front. Comput. Neurosci.* **8**, 27 (2014).
46. D. Holden, T. Komura, J. Saito, Phase-functioned neural networks for character control. *ACM Trans. Graph.* **36**, 42:1–42:13 (2017).
47. S. Starke, H. Zhang, T. Komura, J. Saito, Neural state machine for character-scene interactions. *ACM Trans. Graph.* **38**, 178:1–178:14 (2019).
48. J. W. Hurst, A. A. Rizzi, Series compliance for an efficient running gait. *IEEE Robot. Autom. Mag.* **15**, 42–51 (2008).
49. C. Gilbert, Visual control of cursorial prey pursuit by tiger beetles (Cicindelidae). *J. Comp. Physiol. A.* **181**, 217–230 (1997).
50. J. M. Camhi, E. N. Johnson, High-frequency steering maneuvers mediated by tactile cues: Antennal wall-following in the cockroach. *J. Exp. Biol.* **202**, 631–643 (1999).
51. J. Borràs, C. Mandery, T. Asfour, A whole-body support pose taxonomy for multi-contact humanoid robot motions. *Sci. Robot.* **2**, eaaq0560 (2017).
52. Jueying® | DeepRobotics; http://www.deeprobotics.cn/default/details.
53. T. Haarnoja, A. Zhou, P. Abbeel, S. Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in *Proceedings of the 35th International Conference on Machine Learning (ICML)* (PMLR, 2018), pp. 1861–1870.
54. C. Yang, K. Yuan, S. Heng, T. Komura, Z. Li, Learning natural locomotion behaviors for humanoid robots using human bias. *IEEE Robot. Autom. Lett.* **5**, 2610–2617 (2020).
55. X. B. Peng, M. van de Panne, Learning locomotion skills using DeepRL: Does the choice of action space matter?, in *Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation* (ACM, 2017), pp. 12:1–12:13.
56. A. Rupam Mahmood, D. Korenkevych, B. J. Komer, J. Bergstra, Setting up a reinforcement learning task with a real-world robot, in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2018), pp. 4635–4640.
57. H. van Hasselt, Double Q-learning, in *Proceedings of the 23rd International Conference on Neural Information Processing Systems* (Curran Associates Inc., 2010), pp. 2613–2621.
58. H. Van Hasselt, A. Guez, D. Silver, Deep reinforcement learning with double Q-learning, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence* (AAAI, 2016), pp. 2094–2100.
59. N. Hogan, D. Sternad, Dynamic primitives of motor behavior. *Biol. Cybern.* **106**, 727–739 (2012).
60. E. Spyrakos-Papastavridis, N. Kashiri, J. Lee, N. G. Tsagarakis, D. G. Caldwell, Online impedance parameter tuning for compliant biped balancing, in *Proceedings of the 2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2015), pp. 210–216.
61. C. Yang, K. Yuan, W. Merkt, T. Komura, S. Vijayakumar, Z. Li, Learning whole-body motor skills for humanoids, in *Proceedings of the 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2018), pp. 270–276.
62. Y. Hashiguchi, K. Takaoka, M. Kanemaru, The development of a practical dexterous assembly robot system without the use of force sensor, in *Proceedings of the 2001 IEEE International Symposium on Assembly and Task Planning (ISATP2001). Assembly and Disassembly in the Twenty-first Century. (Cat. No.01TH8560)* (IEEE, 2001), pp. 470–475.
63. Y. Zhao, N. Paine, S. J. Jorgensen, L. Sentis, Impedance control and performance measure of series elastic actuators. *IEEE Trans. Ind. Electron.* **65**, 2817–2827 (2018).
64. E. Coumans, Y. Bai, PyBullet, a Python module for physics simulation for games, robotics and machine learning; http://pybullet.org.

Acknowledgments: We thank J. Ramos and K. Korobchevskaya, who read an early draft of this manuscript and offered valuable suggestions, and C. McGreavy for proofreading this manuscript and producing the artwork. Z.L. thanks A. Mukovskiy for guidance on the equilibrium-point theories since the AMARSi project. Sincere appreciation to DeepRobotics Co. Ltd. for permitting all crash tests, and those who helped with experiments: Z. Sun, J. Zhang, X. Mo, C. Li, Z. Chu, and C. Li. **Funding:** This research work is partly supported by the EPSRC CDT in Robotics and Autonomous Systems (EP/L016834/1) and the Open Project (ICT1900349, ICT20005) from Zhejiang University, and the rest is self-financed. **Author contributions:** C.Y. contributed to the multi-expert learning structure, simulation, data analysis, experiments, and the manuscript. K.Y. contributed to the hierarchical learning structure, C++ software on the robot, simulation, experiments, data collection, and the manuscript. Q.Z. contributed to the development of Jueying robots and hardware facilities and supported all robot experiments. W.Y. contributed to the writing, figures, and data analysis in the manuscript. Z.L. directed the research, designed and debugged key hardware experiments, and authored the manuscript. **Competing Interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials.

Submitted 10 April 2020
Accepted 13 November 2020
Published 9 December 2020
10.1126/scirobotics.abb2174

Citation: C. Yang, K. Yuan, Q. Zhu, W. Yu, Z. Li, Multi-expert learning of adaptive legged locomotion. *Sci Robot.* **5**, eabb2174 (2020).