DEEP NEUROETHOLOGY OF A VIRTUAL RODENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Parallel developments in neuroscience and deep learning have led to mutually productive exchanges, pushing our understanding of real and artificial neural networks in sensory and cognitive systems. However, this interaction between fields is less developed in the study of motor control. Existing experimental research and neural network models have been focused on the production of individual behaviors, yielding little insight into how intelligent systems can produce a rich and varied set of motor behaviors. In this work we develop a virtual rodent that learns to flexibly apply a broad motor repertoire, including righting, running, leaping and rearing, to solve multiple tasks in a simulated world. We analyze the artificial neural mechanisms underlying the virtual rodent's motor capabilities using a neuroethological approach, where we characterize neural activity patterns relative to the rodent's behavior and goals. We show that the rodent solves tasks by using a shared set of force patterns that are orchestrated into task-specific behaviors over longer timescales. Through methods familiar to neuroscientists, including representational similarity analysis, dimensionality reduction techniques, and targeted perturbations, we show that the networks produce these behaviors using at least two classes of behavioral representations, one that explicitly encodes behavioral kinematics in a task-invariant manner, and a second that encodes task-specific behavioral strategies. Overall, the virtual rat promises to facilitate grounded collaborations between deep reinforcement learning and motor neuroscience.

1 INTRODUCTION

Animals have nervous systems that allow them to coordinate their movement and perform a diverse set of complex behaviors. While complex behavior exists across the animal kingdom, mammals, in particular, are generalists, capable of flexibly selecting and adaptively refining behaviors towards varied goals. To date, the complexity and adaptability observed in mammals far surpasses what is possible to produce in robots or artificial motor control systems. This suggests that studies of animal behavior may yield basic insights into classes of algorithms capable of flexibly producing complex movement. And while neuroscientists aim to understand how animal nervous systems are capable of solving motor challenges, AI researchers can leverage those insights to design algorithms to flexibly control artificial systems.

Recently, there has been renewed interest in a constructive approach in which artificial models that solve tasks that parallel those solved by real animals serve as normative models of biological intelligence. Researchers have attempted to leverage these models to gain insights into the algorithmic objectives implemented by neurobiological circuits, prominently in vision (Khaligh-Razavi & Kriegeskorte, 2014; Yamins et al., 2014; Kar et al., 2019), but also increasingly in other areas, including audition (Kell et al., 2018) and navigation (Banino et al., 2018; Cueva & Wei, 2018). However these constructive efforts have been of a more limited scope in studies of the motor system. Existing approaches for building artificial models of motor or behavioral production, largely focus on the production of single behaviors. In the context of locomotion, especially for research focused on reproducing observed gait patterns of animals, efforts constructing biologically inspired controllers have been scientifically informative (Grillner et al., 2007; Ijspeert et al., 2007; Ramdya et al., 2017), and similar ideas have been used for animation (Coros et al., 2011) and even paleobiology (Nyakatura et al., 2019). In addition, reaching movements have been studied using similar methods, though again in the context of relatively constrained behavioral paradigms (Lillicrap & Scott, 2013), and often with supervised data available (Sussillo et al., 2015; Michaels et al., 2019).

Together, while these approaches model the interactions between animals and their physical environments (Chiel & Beer, 1997), they do not deal with the fundamental challenges and opportunities of motor and behavioral control in mammals, namely using the same body and behavioral repertoire to solve multiple distinct cognitive challenges. While some efforts have been made to analyze neural activity in reduced systems trained to solve multiple tasks (Song et al., 2017; Yang et al., 2019), those studies have lacked the essential element of motor control in a physical environment.

Emerging deep reinforcement learning algorithms are now allowing artificial agents to perform complex and adaptive movement in physical environments with egocentric sensory information that is increasingly similar to that available to animals (Peng et al., 2016; 2017; Heess et al., 2017; Merel et al., 2019a;b). While these artificial networks could be analyzed directly, we are not able to directly compare the findings to biological data from animal models such as rodents, which offer a broad behavioral repertoire and far greater experimental tractability for manipulating and recording from the brain. To facilitate grounded investigation into circuits capable of generating multiple behaviors with some shared motor elements, we introduce a virtual model of a rodent, and demonstrate its promise as a tool for understanding the computations underlying movement generation. Our virtual rodent exists in a physical world and is equipped with a set of actuators that must be coordinated for it to run, jump, spin, and rear. It also possesses a sensory system that allows it to use visual input from an egocentric camera located on its head as well as proprioceptive information to sense the configuration of its body in space. To control its body, the rodent is episodically presented with a series of motor and cognitive tasks from which it learns end-to-end to produce effective movements and solve tasks with high reliability.

We then ask "Can a neuroscientist understand a virtual rodent?" – a more grounded take on the originally satirical "Can a biologist fix a radio?" (Lazebnik, 2002) or the more recent "Could a neuroscientist understand a microprocessor?" (Jonas & Kording, 2017). We take a more sanguine view of the tremendous advances that have been made in computational neuroscience in the past decade, and posit that the supposed 'failure' of these approaches in synthetic systems is partly a misdirection. Analysis approaches in neuroscience were developed with the explicit purpose of understanding sensation and action in real brains, and often implicitly rooted in the types of architectures and processing that are thought relevant in biological control systems. With this philosophy, we use analysis approaches common in neuroscience to explore the types of representations and dynamics that the virtual rodent's neural network employs to allow it to coordinate multiple complex movements in the service of solving motor and cognitive tasks.

2 Approach

2.1 Virtual rodent body

We implemented a virtual rodent body in MuJoCo (Todorov et al., 2012), based on measurements of laboratory rats (see Appendix A.1). The rodent body has 38 controllable degrees of freedom. The tail, spine, and neck consist of multiple segments with joints, but are controlled by tendons that co-activate multiple joints (spatial tendons in MuJoCo). Upon publication, we will release the rodent model.

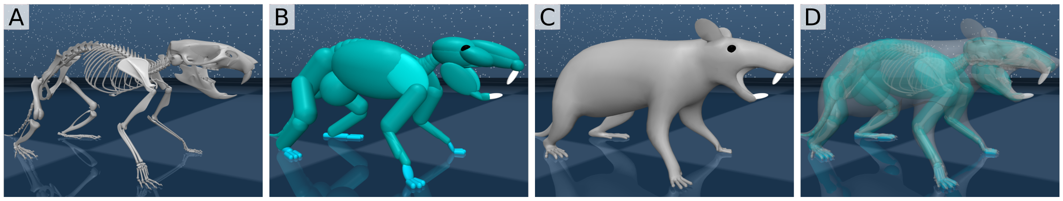

Figure 1: (A) Anatomical skeleton of a rodent (as reference; not part of physical simulation). (B) A body designed around the skeleton to match the anatomy and model collisions with the environment. (C) Purely cosmetic skin to cover the body. (D) Semi-transparent visualization of (A)-(C) overlain.

2.2 Tasks the virtual rodent

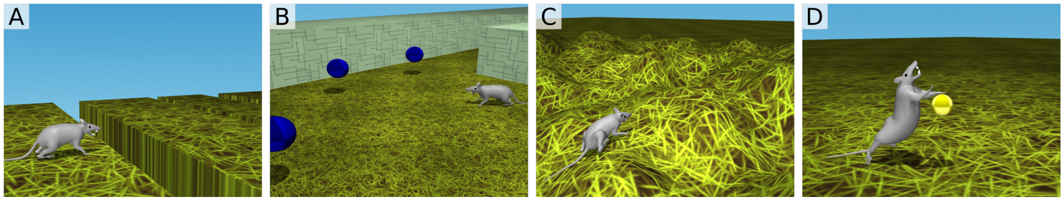

Figure 2: Visualizations of four tasks the virtual rodent is trained to solve: (A) jumping over gaps ("gaps run"), (B) foraging in a maze ("maze forage"), (C) escaping from a hilly region ("bowl escape"), and (D) touching a ball twice with a forepaw with a precise timing interval between touches ("two-tap").

We implemented four tasks (adapted from Merel et al. (2019a); Tassa et al. (2018) as well as (Kawai et al., 2015)) to encourage diverse motor behaviors in the rodent. The tasks are as follow: (1) Run along a corridor, over "gaps", with a reward for traveling along the corridor at a target velocity (Figure 2A). (2) Collect all the blue orbs in a maze, with a sparse reward for each orb collected (Figure 2B). (3) Escape a bowl-shaped region by traversing hilly terrain, with a reward proportional to distance from the center of the bowl (Figure 2C). (4) Approach orbs in an open field, activate them by touching them with a forepaw, and touch them a second time after a precise interval of 800ms with a tolerance of ±100ms; there is a time-out period if the touch is not within the tolerated window and rewards are provided sparsely on the first and second touch (Figure 2D).

For it to be possible for all tasks to be performed by a single agent, they must share an observation and action space. The observation space for all tasks consists of proprioceptive inputs as well as "raw" egocentric RGB-camera (64×64 pixels) input from a head-mounted camera. The proprioceptive inputs include internal joint angles and angular velocities, the positions and velocities of the tendons that provide actuation, egocentric vectors from the root (pelvis) of the body to the positions of the head and paws, a vestibular-like upright orientation vector, touch or contact sensors in the paws, as well as egocentric acceleration, velocity, and 3D angular velocity of the root. We do not provide the agent with a cue or context indicating which task it is in. Rather, the agent must infer which task it is in from the visual input and behave appropriately.

2.3 Training a multi-task policy

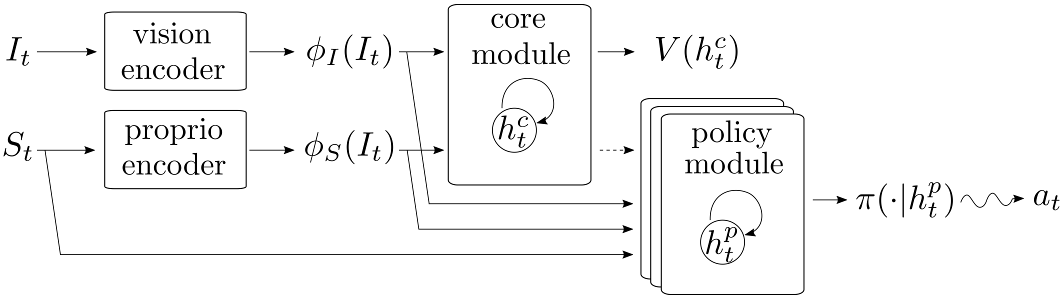

Figure 3: The virtual rodent agent architecture. Egocentric visual image inputs are encoded into features via a small residual network (He et al., 2016) and proprioceptive state observations are encoded via a small multi-layer perceptron. The features are passed into a recurrent LSTM module (Hochreiter & Schmidhuber, 1997). The core module is trained by backpropogation during training of the value function. The outputs of the core are also passed as features to the policy module (with the dashed arrow indicating no backpropogation along this path during training) along with shortcut paths from the proprioceptive observations as well as encoded features. The policy module consists of one or more stacked LSTMs (with or without skip connections) which then produce the actions via a stochastic policy.

Emboldened by recent results in which end-to-end RL produces a single terrain-adaptive policy (Peng et al., 2016; 2017; Heess et al., 2017), we decided to train a single model on the multiple motor-control-reliant tasks. To train a single policy to perform all four tasks, we used an IMPALA-style setup for actor-critic DeepRL (Espeholt et al., 2018); parallel workers collected rollouts, logged them to a replay, from which a central learner sampled data to perform updates. The value-function critic was trained using off-policy correction via V-trace. To update the actor, we used a variant of MPO (Abdolmaleki et al., 2018) where the E-step is performed using advantages determined from the empirical returns and the value-function, instead of the Q-function (Anonymous, 2019). Empirically, we found that the "escape" task was more challenging to learn during interleaved training relative to the other tasks. Consequently, for the results we present, we trained a single-task expert on the escape task and trained the multi-task policies using kickstarting for that task (Schmitt et al., 2018), with a weak coefficient (.001 or .005). Kickstarting on this task made the seeds more reliably solve all four tasks, facilitating comparison of the multi-task policies with different architectures (i.e. the policy having 1, 2, or 3 layers, with or without skip connections across those layers). The result of this training procedure is a single neural network that is capable of using visual inputs to determine how to behave as well as coordinating its body to move in ways required to solve the tasks. See video examples of a single policy solving episodes of each task: gaps, forage, escape, and two-tap.

3 Analysis

To characterize the principles and mechanisms by which the networks control the virtual rodent, we logged the animal's kinematics, joint angles, and computed forces, sensory inputs, and neural activity during 25 trials per task from each network architecture. We borrow analyses and perturbation techniques from neuroscience, where a range of techniques have been developed to analyze the properties of real neural networks, albeit often in the context of more reduced experiments.

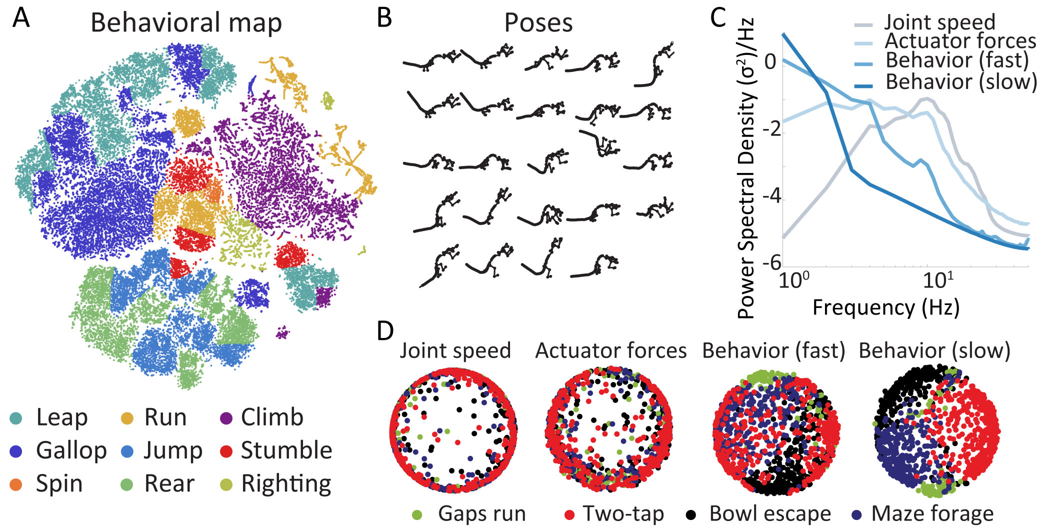

Figure 4: Virtual rats learn to use a broad set of shared and task-specific locomotor behaviors. (A) t-SNE embedding of 60 behavioral features describing the pose and kinematics of the virtual rat, colored by hand-labeling of coarse behavioral categories. t-SNEs were made using the Barnes-Hut approximation with a perplexity of 30. (B) Examples of the animal pose over 25 different regions of the map. (C) Power spectral density of four different behavioral feature categories, computed by averaging the spectral density of the top ten principal components of each motor feature, weighted by the variance they explain. (D) We embedded the feature matrices of the same feature categories into two-dimensions using multidimensional scaling, which revealed the emergence of task-specificity in behaviors at longer timescales.

3.1 Ethology of the virtual rodent

Taking inspiration from ethological approaches, we begin our analysis with a careful evaluation of the virtual rodent's behavior. The rodent locomotes using a set of actuators that produce joint specific forces and kinematics on short timescales. On longer timescales, the force patterns across the body are organized into coordinated, re-used behaviors, such as running, jumping, and turning, many of which are challenging from the standpoint of motor control. To systematically examine the classes of behaviors these networks learn to generate, and how they are differentially deployed across tasks, we adopted a behavioral mapping approach, where we generated a set of features describing the pose and kinematics of animals and embedded them in 2D using t-SNE (Berman et al., 2014). Features were computed by taking the top 15 principal components of the virtual rat's joint angles and marker positions to yield a set of eigenpostures (Stephens et al., 2008). We then computed the top 15 principal components of the Morlet wavelet transform of these two sets of eigenpostures, on 25 scales between either 1 to 25 Hz timescales (behavioral features) , 0.3 to 5 Hz (slow behavioral features) or 5-25 Hz (fast behavioral features). In all cases, this yielded a 60 dimensional feature set describing the animals pose and kinematics. Embedding these features using t-SNE (Maaten & Hinton, 2008) produced a behavioral map in which virtual rodent behaviors are segregated to different regions of the map. The map captures some of the relationship structure among the broad range of locomotor behaviors, including rearing, jumping, running, climbing and spinning (see video). These behavioral features, which vary over a longer timescale than forces or velocity, provided a complementary basis for characterizing the animal's behavior (Figure 4C). Interestingly, while patterns of activation of low level motor features were largely shared across tasks (Figure 4D). In contrast, longer timescale behaviors, especially slow behavioral features, were more differentiated across tasks, suggesting that the network learned to adapt similar movements in a selective manner for different tasks (Figure 4D).

3.2 Representational structure of the virtual rodent's neural network

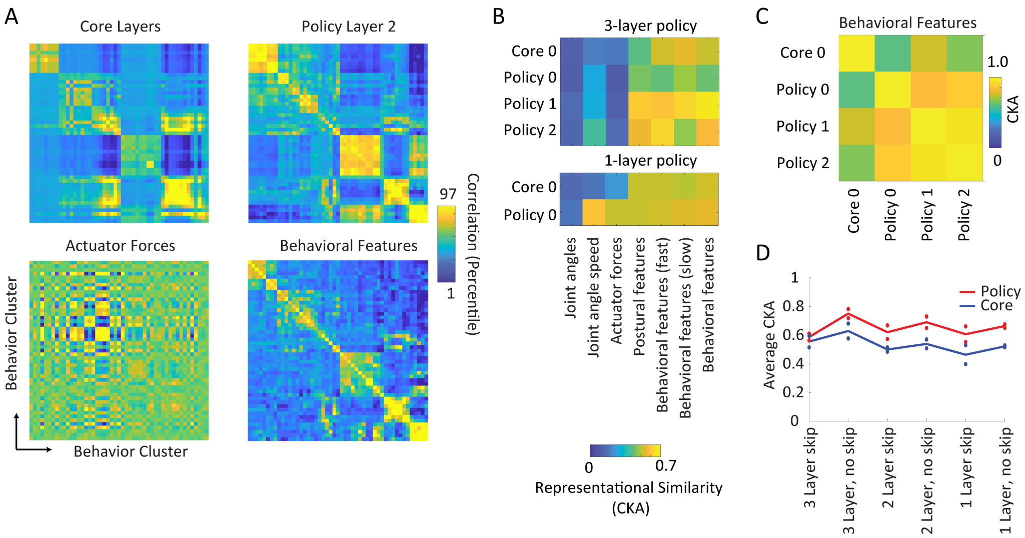

Figure 5: Representational structure of the rodent's neural network. (A) Example dissimilarity matrices of the core and policy networks, behavioral features, and actuator forces, computed across 50 different behavioral categories. (B) CKA index of neural and behavioral dissimilarity matrices for 3 and 1 policy layer architectures. (C) CKA index of neural dissimilarity matrices for all network layers. (D) Average CKA index between policy layers and behavioral features, compared across architectures. Points show values from individual network seeds.

In addition to behavioral parameters, we concurrently measured and analyzed the neural activity of LSTM cell units in each layer. Inspection of neural activity during task performance revealed that core and policy units operate on distinct timescales (See Appendix A.2, Figure 9). Units in the core are typically active over timescales of 1-10 seconds, likely representing variables associated with context and reward. In contrast, units in policy layers are more active over subsecond timescales, likely encoding motor and behavioral features.

To quantify which aspects of behavior were encoded in the core and policy layers, we used representational similarity analysis, which can provide a global measure of how well behavioral parameters are encoded in networks, and how this encoding varies across layers and architectures (Kriegeskorte & Diedrichsen, 2019; Kornblith et al., 2019). To compute how well a population encodes behavioral parameters, such as forces or behaviors, we can compute the dissimilarity of the neural representations between neurons in a given layer across different categories of force patterns or behaviors (Figure 5A). We balanced these categories by dividing each behavioral feature class into 50 clusters using k-means. Following computation of the dissimilarity matricies, one can compare the correlation, dot product, or other similarity metrics such as the centered kernel alignment (CKA) index between the behavioral dissimilarity matrix and neural dissimilarity matrix, or between different neural dissimilarity matricies. The former provides a measure of how well a neural population encodes a behavioral feature, while the later provides information regarding how similar this encoding pattern is over neural populations.

Representational analyses revealed that core and policy layers encoded distinct behavioral features. Policy layers contained greater information about behaviors, especially over fast timescales while core layers showed more moderate encoding that was stronger for slow behavioral features (Figure 5B). In networks trained with three policy layers, representations were distributed in timescales across layers, with the last layer (Policy 2) showing stronger encoding of fast behavioral features, and the first layer (Policy 0) instead showing stronger encoding of slow behavioral features. The pattern of behavioral encoding was shared across policy layers, but highly distinct from the core (Figure 5C). In networks with a single policy layer, a broader set of behavioral representations were present in both core and policy layers. The strength of behavioral encoding was similar across all architectures tested (Figure 5D).

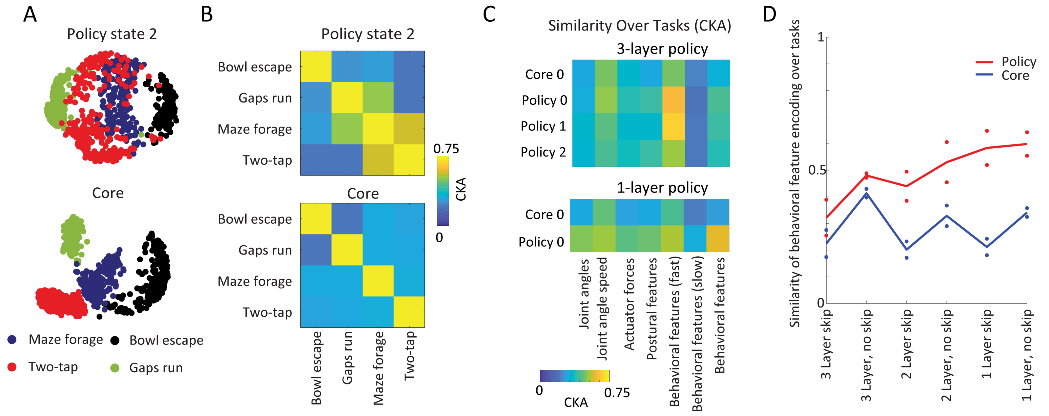

Figure 6: Task Selectivity in the core and policy networks. (A) We embedded the feature matrices of core and policy networks into two-dimensions using multidimensional scaling, which revealed task-specificity in core representations. (B) CKA index of the policy and core across different task conditions. (C) The CKA index between the dissimilarity matrices of each activation layer across tasks for different behavioral features. (D) The similarity of behavioral feature encoding (CKA index) across different architectures. Points show values from individual seeds.

We then investigated how these networks are able to differentiate movements in a task-specific manner, by characterizing how the representational structure varied across tasks. Embedding population activity into two-dimensions using multidimensional scaling revealed that core neuron representations were highly distinct across tasks, while policy layers contained more overlap (Figure 6A). To quantify this task-dependent coding, we compared the representations of the same behaviors over tasks. First, we computed the dissimilarity matricies of behaviors within each task. Second, we detected the shared set of behaviors across tasks, and computed the correlation (CKA) between the task-specific dissimilarity matricies across this shared set of tasks (Figure 6B). This analysis revealed that policy layers tended to possess a relatively similar encoding of behavioral features, especially fast behavioral features, over tasks (Figure 6C; Appendix A.3, Figure 10). Core layer representations across almost all behavioral categories were more variable across tasks. Notably, slow behavioral feature encodings were also more variable across tasks. Overall, this is consistent with the interpretation that policy networks possess a fixed, short timescale representation of behaviors, while core networks contain a longer timescale representation of behavioral sequences or abstract encodings of task variables.

Interestingly, when comparing this cross-task encoding similarity across architectures, we found that one layer networks showed a marked increase in the similarity of behavioral encoding across tasks (Figure 6D). This suggests that in networks with lower computational capacity, animals must rely on a smaller, shared behavioral representation across tasks.

3.3 Dynamical Properties of virtual rodent neural network activity

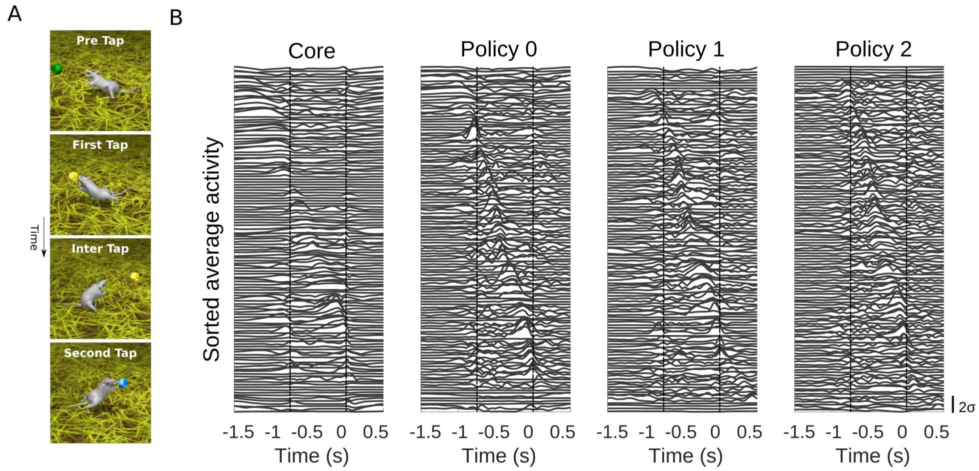

Figure 7: Neurons in core and policy networks show sequential activity during stereotyped behavior. (A) Example video stills showing the virtual rat engaged in the two-tap task (B) Average absolute z-scored activity traces of all 128 neurons in each layer during performance of the two-tap sequence. Traces are sorted by the time of peak average firing rate. Dashed lines indicate the times of first and second taps. Sequential neural activity is present during the two-tap sequence.

While representational similarity analysis assessed the overall representational structure of the population, it failed to provide insight into dynamical network mechanisms that may underlie the production of behavior, as well as more abstract features that reflect the reward structure of the task. For instance, in the two-tap task, peak activity in core and policy units was sequentially organized (Figure 7), uniformly tiling time between both taps of the two-tap sequence. This sequential activation in policy networks was observed across tasks and behaviors, including during jumping (see video), and notably during running, where, consistent with policy networks encoding short-timescale kinematic features in a task-invariant manner, neural activity sequences during running were largely conserved across tasks (See Appendix A.3, Figure 10). Interestingly, we also observed that neural variability is reduced during the inter-trial interval when more stereotyped behavior is demanded by the two-tap task, which may have bearing on classic results in motor control (See Appendix A.4, Figure 11).

To systematically examine the types of sequential or dynamical activity present in core and policy networks, we examined the dynamics of the networks' population vector over time. We first applied principal components analysis (PCA) to the activity during the performance of single tasks, and visualized the gradient of the population vector as a vector field. Figure 8A shows such a vector field representation of the first two principal components of the core and final policy layer during the two-tap task. We generated vector fields by discretizing the PC space into a two-dimensional grid and calculating the average neural activity gradient with respect to time for each bin. The vector fields showed strong signatures of rotational dynamics across all layers. To quantify the frequency of these dynamical patterns, we turned to jPCA, a dimensionality reduction method that extracts rotational dynamics in neural activity by fitting a rotational linear dynamical system to neural data (Churchland et al., 2012). The resulting jPCs form an orthonormal basis that spans the same space as the first six traditional PCs, while maximally emphasizing rotational dynamics. Figure 8B shows the vector fields of the first two jPC planes for the core and final policy layers along with their characteristic frequency. Consistent with our previous findings, jPC planes in the core have lower characteristic frequencies than those in policy layers across tasks (Figure 8C). The jPC planes also individually explained a large amount of total neural variability, suggesting that they are informative dimensions for describing population activity (Figure 8D).

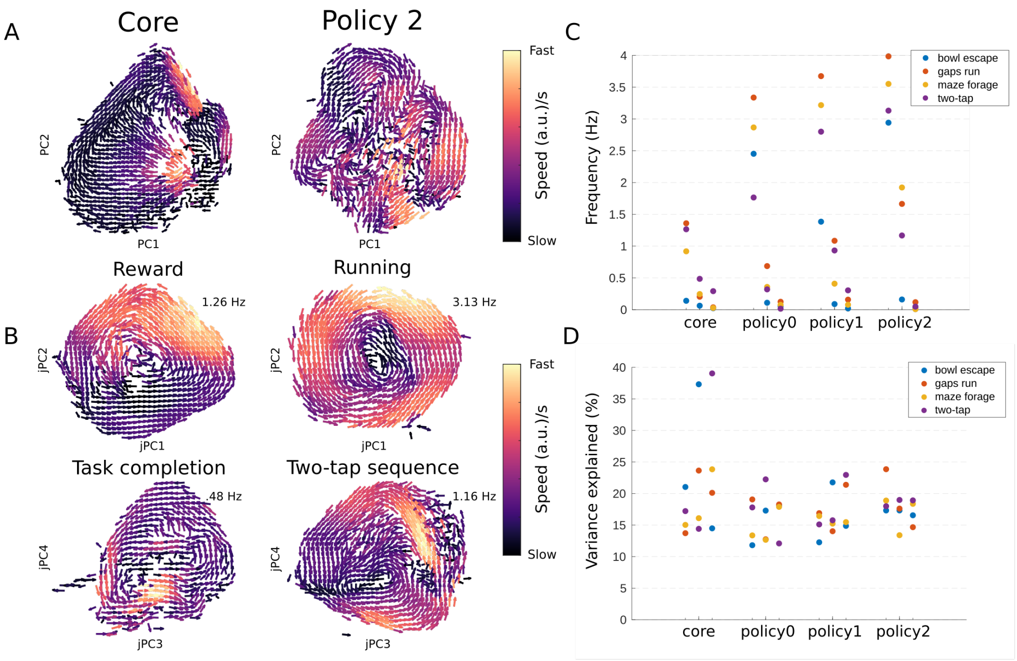

Figure 8: Latent network dynamics within tasks reflect rodent behavior on different timescales. (A) Vector field representation of the first two principal components of neural activity in the core and final policy layers during the two-tap task. PC spaces show signatures of rotational dynamics. (B) Vector field representation of first two jPC planes for the core and final policy layers during the two-tap task. Apparent rotations within the different planes are associated with behaviors and behavioral features of different timescales, labeled above. Columns denote layer (as in (A)), while rows denote jPC plane. (C) Characteristic frequency of rotations within each jPC plane. Groups of three points respectively indicate the first, second, and third jPC planes for a given layer. Rotations in the core are slower than those in the policy. (D) Variance explained by each jPC plane.

These rotational dynamics in the policy and core jPC planes were respectively associated with the production of behaviors and the reward structure of the task, respectively. For example, in the two-tap task, rotations in the fastest jPC plane in the core were concurrent with the approach to reward, while rotations in the second fastest jPC were concurrent with long timescale transitions between running to the orb and performing the two-tap sequence. Similarly, the fastest jPC in policy layers was associated with the phase of running, while the second fastest was associated with the phase of the two-tap sequence, (see video). This trend of core and policy neural dynamics respectively reflecting behavioral and task-related features was also present in other tasks. For example, in the maze forage task, the first two jPC planes in the core respectively correlated with reaching the target orb and discovering the location of new orbs, while those in the policy were correlated with low-level locomotor features such as running phase (see video). Along with our representational similarity analysis, these findings further support a model in which the core layer transforms sensory information into a sensorimotor contextual signal in a task-specific manner while the policy uses this signal and sensory inputs to generate appropriate behaviors in a more task-independent fashion. For a more complete set of behaviors with neural dynamics visualizations overlain, see Appendix A.5.

3.4 Consequences of neural perturbations

To demonstrate these differing roles of core and policy units in respectively controlling movement and task-relevant features, we performed causal perturbations of different neuronal subsets in the two-tap task. We identified two stereotyped behaviors (rears and spinning jumps) that were reliably used in two different seeds of the agent to reach the orb in the two-tap task. We ranked neurons according to their degree of modulation of their z-scored activity during the performance of these behaviors. We then inactivated subsets of neurons by clamping their activity to their mean values between the first and second taps and observed the effects of inactivation on trial success and behavior.

In both seeds analyzed, inactivation of policy units had a stronger effect on motor behavior than the inactivation of core units (Figure 12A). This can be seen in Appendix A.6 Figure 12B, where ablation of 64 modulated neurons in the final policy layer disrupts the performance of the spinning jump (video). In contrast, ablation of behavior-modulated core units did not prevent the production of the behavior, but rather affected the way in which the behavior is directed toward objects in the environment. For example, ablation of a subset of core units during the performance of a spinning jump often had a limited affect, but sometimes resulted in jumps that missed the target orbs (video; See Appendix A.6, Figure 12C).

We also performed a complementary perturbation aimed to elicit behaviors by overwriting the cell state of neurons in each layer with a time-varying average trajectory of neural activity measured during natural performance of a target behavior. The efficacy of stimulation was found to depend on the gross body posture and behavioral state of an animal, but was nevertheless successful in some cases. For example, during the two-tap sequence, we were able to elicit spinning movements common to searching behaviors in the forage task (video; See Appendix A.6, Figure 12D, E). The efficacy of this activation was more reliable in layers closer to the motor output (Figure 12D). In fact, activation of core units rarely elicited spins, but rather elicited sporadic dashes reminiscent of the searching strategy of many models during the forage task (video).

4 Discussion

Neuroscientists seek multiple levels of insight into animal behavior as well as the structure and function of the nervous systems. For many computational neuroscientists and artificial intelligence researchers alike, an aim is to reverse-engineer the nervous system at an appropriate level of abstraction. In the motor system, such an effort requires that we build models of animals and produce artificial nervous systems capable of controlling their synthetic bodies across a range of behavior. Here we introduced a virtual rodent capable of performing a variety of complex of locomotor behaviors. By training a single policy to solve multiple tasks with distinct, but overlapping, motor and behavioral demands, we have produced a simplified virtual nervous system that allows us to practice the techniques applied by neuroscientists and verify the extent to which they produce sensible and interpretable results. An assumption underlying our approach is that the morphology of the virtual rodent body combined with the tasks we trained the neural architecture to solve will result in basic functional similarities between real and virtual rodent nervous systems. However, the key advantage of this approach is that while in real animals it is often difficult to measure actuator forces and kinematics, let alone sensory inputs across many behaviors, in the virtual rodent we can fully observe sensory inputs, neural activity, and behavior, which facilitates more comprehensive testing of theories related to how behavior is generated.

We demonstrated that while the virtual rat used a similar set of short-timescale movements across tasks, while task-specific behaviors emerged on longer timescales. These behaviors were controlled by a hierarchically organized neural network, in which upper layers reflect longer-timescale representations of behavioral context, and lower levels contain shorter-timescale information about forces and production of individual behaviors.

We took two perspectives to characterizing neural activity in the virtual rodent: representational similarity and dynamical systems. While these are sometimes thought of as competing conceptual frameworks (Shenoy et al., 2013), we found them to address structure in neural dynamics at complementary temporal scales. During individual behaviors, networks appeared to follow fixed dynamical trajectories. Yet across multiple behaviors, the subspaces used to encode these dynamics were similar in accordance to the similarity of each behavior. While representational methods may in some cases show greater variation across network instances (Maheswaranathan et al., 2019), we found that these analysis perspectives provided distinct, but aligned insights.

Incrementally and judiciously increasing the realism of the model's embodiment, behavioral repertoire, and neural architecture, is a natural path for future research. Our virtual rodent possesses fewer actuators and sensors than a real rodent, and we have yet to explore integration with olfactory, auditory, or whisker-based sensation (see Zhuang et al. (2017)). Our virtual rodent is capable of locomotor behaviors, but it will be productive to consider an increased diversity of behavioral tasks – involving decision making, memory-based navigation, and working memory, which could give insight into "cognitive" behaviors of which rodents are capable. Furthermore, biologically-inspired design of the neural architectures and training procedures should also facilitate comparisons to real neural recordings and manipulations. As progress is made in capturing broader capabilities of animal intelligence, we hope to more precisely isolate residual elements of animal behavior generation that are difficult to model and expect these will parallel those elements of intelligent control that are difficult to engineer for artificial systems.

Author Contributions

Contributions masked for anonymity.

Acknowledgments

The rodent skeleton that we used as a reference for designing the virtual rodent body was purchased from leo3Dmodels on TurboSquid. Other acknowledgements masked for anonymity.

References

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, and Martin Riedmiller. Maximum a posteriori policy optimisation. In *International Conference on Learning Representations*, 2018.

Anonymous. V-MPO: On-policy maximum a posteriori policy optimization for discrete and continuous control. *Under Review for International Conference on Learning Representations*, 2019.

Andrea Banino, Caswell Barry, Benigno Uria, Charles Blundell, Timothy Lillicrap, Piotr Mirowski, Alexander Pritzel, Martin J Chadwick, Thomas Degris, Joseph Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557(7705):429, 2018.

Gordon J Berman, Daniel M Choi, William Bialek, and Joshua W Shaevitz. Mapping the stereotyped behaviour of freely moving fruit flies. *Journal of The Royal Society Interface*, 11(99):20140672, 2014.

Hillel J Chiel and Randall D Beer. The brain has a body: adaptive behavior emerges from interactions of nervous system, body and environment. *Trends in neurosciences*, 20(12):553–557, 1997.

Mark M Churchland, M Yu Byron, John P Cunningham, Leo P Sugrue, Marlene R Cohen, Greg S Corrado, William T Newsome, Andrew M Clark, Paymon Hosseini, Benjamin B Scott, et al. Stimulus onset quenches neural variability: a widespread cortical phenomenon. *Nature neuroscience*, 13(3):369, 2010.

Mark M Churchland, John P Cunningham, Matthew T Kaufman, Justin D Foster, Paul Nuyujukian, Stephen I Ryu, and Krishna V Shenoy. Neural population dynamics during reaching. *Nature*, 487 (7405):51, 2012.

Stelian Coros, Andrej Karpathy, Ben Jones, Lionel Reveret, and Michiel Van De Panne. Locomotion skills for simulated quadrupeds. In *ACM Transactions on Graphics (TOG)*, volume 30, pp. 59. ACM, 2011.

Christopher J Cueva and Xue-Xin Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. *arXiv preprint arXiv:1803.07770*, 2018.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. IMPALA: Scalable distributed deep-rl with importance weighted actor-learner architectures. In *International Conference on Machine Learning*, pp. 1406–1415, 2018.

Sten Grillner, Alexander Kozlov, Paolo Dario, Cesare Stefanini, Arianna Menciassi, Anders Lansner, and Jeanette Hellgren Kotaleski. Modeling a vertebrate motor system: pattern generation, steering and control of body orientation. *Progress in brain research*, 165:221–234, 2007.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Nicolas Heess, TB Dhruva, Srinivasan Sriram, Jay Lemmon, Josh Merel, Greg Wayne, Yuval Tassa, Tom Erez, Ziyu Wang, Ali Eslami, et al. Emergence of locomotion behaviours in rich environments. *arXiv preprint arXiv:1707.02286*, 2017.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8): 1735–1780, 1997.

Auke Jan Ijspeert, Alessandro Crespi, Dimitri Ryczko, and Jean-Marie Cabelguen. From swimming to walking with a salamander robot driven by a spinal cord model. *Science*, 315(5817):1416–1420, 2007.

Eric Jonas and Konrad Paul Kording. Could a neuroscientist understand a microprocessor? *PLoS computational biology*, 13(1):e1005268, 2017.

Kohitij Kar, Jonas Kubilius, Kailyn Schmidt, Elias B Issa, and James J DiCarlo. Evidence that recurrent circuits are critical to the ventral streams execution of core object recognition behavior. *Nature neuroscience*, 22(6):974, 2019.

Risa Kawai, Timothy Markman, Rajesh Poddar, Raymond Ko, Antoniu L Fantana, Ashesh K Dhawale, Adam R Kampff, and Bence P Ölveczky. Motor cortex is required for learning but not for executing a motor skill. *Neuron*, 86(3):800–812, 2015.

Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915, 2014.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. *CoRR*, abs/1905.00414, 2019.

Nikolaus Kriegeskorte and Jrn Diedrichsen. Peeling the onion of brain representations. *Annual Review of Neuroscience*, 42(1):407–432, 2019.

Yuri Lazebnik. Can a biologist fix a radio? Or, what I learned while studying apoptosis. *Cancer cell*, 2(3):179–182, 2002.

Timothy P Lillicrap and Stephen H Scott. Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics. *Neuron*, 77(1):168–179, 2013.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Niru Maheswaranathan, Alex H. Williams, Matthew D. Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. *arXiv e-prints*, 2019.

Josh Merel, Arun Ahuja, Vu Pham, Saran Tunyasuvunakool, Siqi Liu, Dhruva Tirumala, Nicolas Heess, and Greg Wayne. Hierarchical visuomotor control of humanoids. In *International Conference on Learning Representations*, 2019a.

Josh Merel, Leonard Hasenclever, Alexandre Galashov, Arun Ahuja, Vu Pham, Greg Wayne, Yee Whye Teh, and Nicolas Heess. Neural probabilistic motor primitives for humanoid control. In *International Conference on Learning Representations*, 2019b.

Jonathan A Michaels, Stefan Schaffelhofer, Andres Agudelo-Toro, and Hansjörg Scherberger. A neural network model of flexible grasp movement generation. *bioRxiv*, pp. 742189, 2019.

John A Nyakatura, Kamilo Melo, Tomislav Horvat, Kostas Karakasiliotis, Vivian R Allen, Amir Andikfar, Emanuel Andrada, Patrick Arnold, Jonas Lauströer, John R Hutchinson, et al. Reverse-engineering the locomotion of a stem amniote. *Nature*, 565(7739):351, 2019.

Xue Bin Peng, Glen Berseth, and Michiel Van de Panne. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 35(4):81, 2016.

Xue Bin Peng, Glen Berseth, KangKang Yin, and Michiel Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 36(4):41, 2017.

Pavan Ramdya, Robin Thandiackal, Raphael Cherney, Thibault Asselborn, Richard Benton, Auke Jan Ijspeert, and Dario Floreano. Climbing favours the tripod gait over alternative faster insect gaits. *Nature communications*, 8:14494, 2017.

Alfonso Renart and Christian K Machens. Variability in neural activity and behavior. *Current opinion in neurobiology*, 25:211–220, 2014.

Simon Schmitt, Jonathan J Hudson, Augustin Zidek, Simon Osindero, Carl Doersch, Wojciech M Czarnecki, Joel Z Leibo, Heinrich Kuttler, Andrew Zisserman, Karen Simonyan, et al. Kickstarting deep reinforcement learning. *arXiv preprint arXiv:1803.03835*, 2018.

Krishna V Shenoy, Maneesh Sahani, and Mark M Churchland. Cortical control of arm movements: a dynamical systems perspective. *Annual review of neuroscience*, 36:337–359, 2013.

H Francis Song, Guangyu R Yang, and Xiao-Jing Wang. Reward-based training of recurrent neural networks for cognitive and value-based tasks. *Elife*, 6:e21492, 2017.

Greg J Stephens, Bethany Johnson-Kerner, William Bialek, and William S Ryu. Dimensionality and dynamics in the behavior of c. elegans. *PLoS computational biology*, 4(4):e1000028, 2008.

David Sussillo, Mark M Churchland, Matthew T Kaufman, and Krishna V Shenoy. A neural network that finds a naturalistic solution for the production of muscle activity. *Nature neuroscience*, 18(7): 1025, 2015.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Daniel LK Yamins, Ha Hong, Charles F Cadieu, Ethan A Solomon, Darren Seibert, and James J DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

Guangyu Robert Yang, Madhura R Joglekar, H Francis Song, William T Newsome, and Xiao-Jing Wang. Task representations in neural networks trained to perform many cognitive tasks. *Nature neuroscience*, 22(2):297, 2019.

Chengxu Zhuang, Jonas Kubilius, Mitra JZ Hartmann, and Daniel L Yamins. Toward goal-driven neural network models for the rodent whisker-trigeminal system. In *Advances in Neural Information Processing Systems*, pp. 2555–2565, 2017.

A Appendix

A.1 Rat measurements

In order to construct the virtual rodent model, we obtained the mass and lengths of the largest body segments that should affect the physical properties of the virtual rodent. First, we dissected cadavers of two female Long-Evans rats, and measured the mass of relevant limb segments and organs. Next, we measured the lengths of body segments over the skin of animals anesthetized with 2% v/v isoflurane. We confirmed that these skin based measurements approximated bone lengths by measuring bone lengths in a third cadaver. The care and experimental manipulation of all animals were reviewed and approved by the appropriate Institutional Animal Care and Use Committee.

	Animal (#)	
	63	**64**
Body part	**Mass (g)**	
Hindlimb L	21	26
Hindlimb R	21	26
Tail	8	10
Forelimb R	11	14
Forelimb L	12	13
Full torso	176	187
Head	26	26
Upper torso	78	71
Lower torso	98	114
Torso without organs	54	58
Intestines and stomach	22	32
Liver	26	17
Pelvis and kidneys	74	80
Jaw	2.43	4.7
Skull	23	21
Tail (base to mid)	5.92	7.2
Tail (mid to tip)	1.78	2.3
Scapula L	3.19	4.7
Humerus L	6.25	4.7
Radius/ulna L	2.61	2.8
Forepaw L	0.53	0.5
Scapula R	2.23	3.9
Humerus R	6.08	6.7
Radius/ulna R	2.17	3.3
Forepaw R	0.53	0.5
Hindpaw L	1.66	1.7
Tibia L	9	9
Femur L	13	16
Hindpaw R	1.81	1.6
Tibia R	5	6
Femur R	13	18
Total	281	301

Table 1: Before weighing, limb segments were divided at their respective joints. Mass of all segments includes all bones, skin, muscle, fascia and adipose layers. L and R refer to the left and right sides of the animal. Precision of measurements listed without decimal places is ± 0.5g

	Animal (#)						
	48	**62**	**55**	**56**	**64**	**63**	**62***
Age (days)	382	82	330	330	83	83	83
Mass (g)	325	273	389	348	283	269	273
Body part	**Length (mm)**						
Ankle to claw L	40.2	39.5	39.7	37.8	39.9	41.5	39.8
Ankle to toe L	38.4	38.12	37.7	35.6	36.6	39.3	38
Ankle to pad L	23.4	22.2	23	22.12	22.5	23.3	6.4
Ankle to claw R		38.2	40.4	38.3	39.3	39.6	38.3
Ankle to toe R		37	38.7	36.3	37.7	38.6	36.2
Ankle to pad R		22.4	23.3	21.9	21.8	23.1	24.1
Tibia L	50	36.3	38.5	49.2	35.8	38.7	34.1
Femur L	44.5	31.6	32.1	37.9	33.4	35.35	32.4
Tibia R		36.7	39.1	37.9	35.1	38.4	36.18
Femur R		32.9	32.1	38.7	31.9	32.1	32.6
Pelvis	25.8	32	31.7	30.2	26.7	27.2	
Wrist to claw L	15	18.8	17.6	18.6	16	19.02	19.2
Wrist to finger L		16	15.8	17.4	15.5	17.07	17.6
Wrist to pad L		6	6.4	8.34	4.9	6.1	6.4
Wrist to olecranon L	29.1	34	32.5	31.7	33.9	32.1	29.9
Humerus L	31.9	29.52	31	28.2	27	31.2	25.4
Scapula L	22.7	24	26.4	29.3	25.9	29.1	26.2
Wrist to claw R		16.8	17	17.8	15.9	16.3	18.1
Wrist to finger R		14.1	13	15.6	15.6	15.3	16.9
Wrist to pad R		5.6	5.8	6.55	5.2	5	5.8
Wrist to olecranon R		30.6	33.5	31.2	30.4	31.8	29.9
Humerus R		28.2	33.5	28.8	25	28.2	25.2
Scapula R		23.8	29.5	25.9	26.2	28.8	24.4
Headcap width	39						
Headcap length	30						
Skull width	38.8	23.35	23	21.8	22.8	23.9	22.2
Skull length	57	51.1	61	56.48	53.16	58.13	48
Skull height					21.59	21.5	21
Head to thoracic		48.6	71.4	68.68	65	60.4	71.2
Thoracic to sacral		73.1	73.6	62.9	65.04	64.7	68.8
Head to sacral	145	126	145.5	127.05	127.2	123.7	140.9
Head width	53.4						
Ear	18	17.55	19.3	17.9	19.2	18.8	
Eye	7.2	8.25	8.6	8.8	8.2	8.3	

Table 2: Length measurements of limb segments used to construct the virtual rodent model from 7 female Long-Evans rats. Measurements were performed using calipers either over the skin or over dissected bones (*). Thoracic and sacral refer to vertebral segments. L and R refer to the left and right sides of the animal's body.

A.2 Power spectral density of core and policy layers

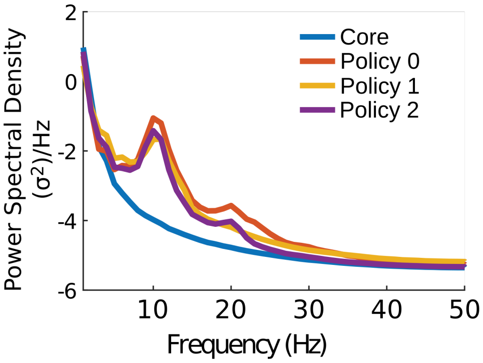

Figure 9: Power spectral density of four different network layers, computed by averaging the spectral density of the top ten principal components of each matrix of activations, weighted by the variance they explain. Notice that policy layers have more power in high frequency bands than core layers.

A.3 Comparison for running neural population activity across tasks

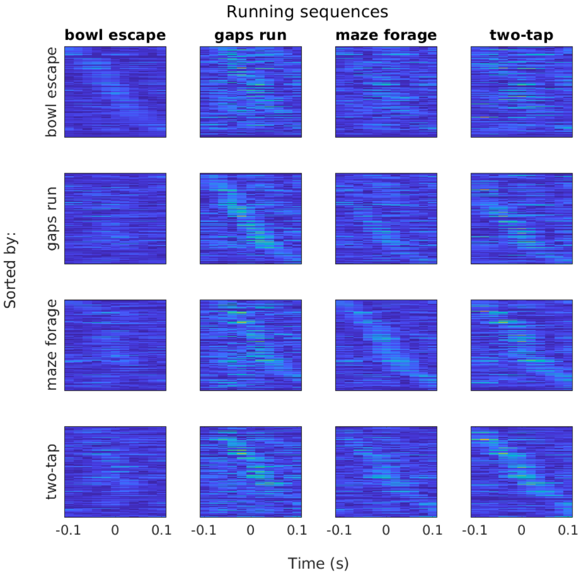

Figure 10: Average activity in the final policy layer during running cycles across different tasks. In each heatmap, rows correspond to the average absolute zscored activity for individual neurons, while columns denote time relative to the mid stance of the running phase. Across heatmaps, neurons are sorted by the time of peak activity in the tasks denoted on the left, such that each column of heatmaps contains the same average activity information with rearranged rows. Aligned running bouts were acquired by manually segmenting the the principal component space of policy 2 activity to find instances of mid-stance running and analyzing the surrounding 200 ms.

A.4 Stereotyped behavior initiation and neural variability

During the execution of stereotyped behaviors, neural variability is reduced (Figure 11). Recall that in our setting, neurons have no intrinsic noise, but inherit motor noise through observations of the state (i.e. via sensory reafference). This effect loosely resembles, and perhaps informs one line of interpretation of the widely reported phenomenon of neural variability reducing with stimulus or task onset (Churchland et al., 2010). Our reproduction of this effect, which simply emerges from training, suggests that variance modulation may partly arise from moments in a task that benefit from increased behavioral precision (Renart & Machens, 2014).

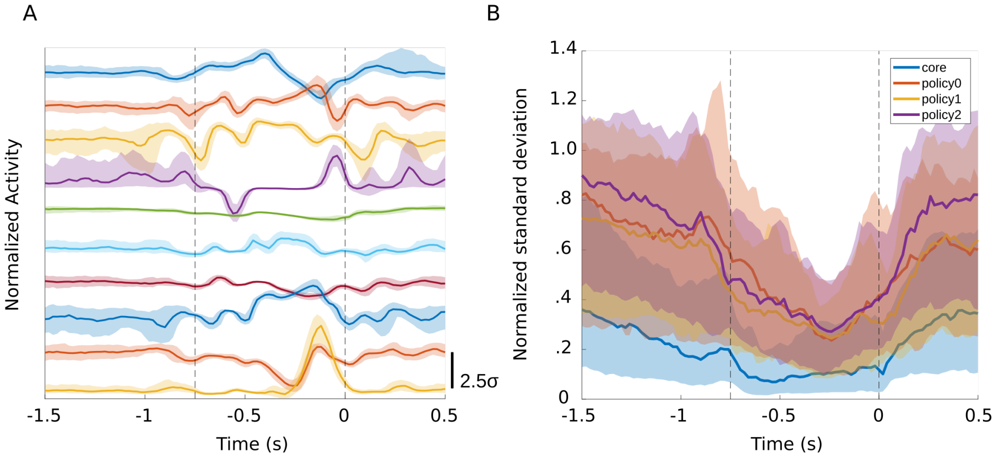

Figure 11: Quantification of neural variability in inter-tap interval of two-tap task relative to the second tap. (A) Example normalized activity traces of ten randomly selected neurons in the final policy layer. Lines indicate mean normalized activity whiles shaded regions range from the 20th percentile to the 80th percentile. Dashed lines indicate the times of first and second taps. (B) Standard deviation of normalized activity across all neurons in the final policy layer as a function of time relative to the second tap. Lines indicate the mean standard deviation while shaded regions range from the 20th percentile to the 80th percentile. Observe that variability is reduced during the two-tap interval.

A.5 Neural dynamics visualized during task behavior

For completeness, we provide links to videos of a few variants of neural dynamics for each task.

Network	Visualization	Task (link)
1-layer policy	PCA	gaps
	PCA	forage
	PCA	escape
	PCA	two-tap
3-layer policy	PCA	gaps
	PCA	forage
	PCA	escape
	PCA	two-tap
3-layer policy	jPCA	gaps
	jPCA	forage
	jPCA	escape
	jPCA	two-tap

Table 3: Links to representative visualizations of neural dynamics and behavior

A.6 Perturbation results

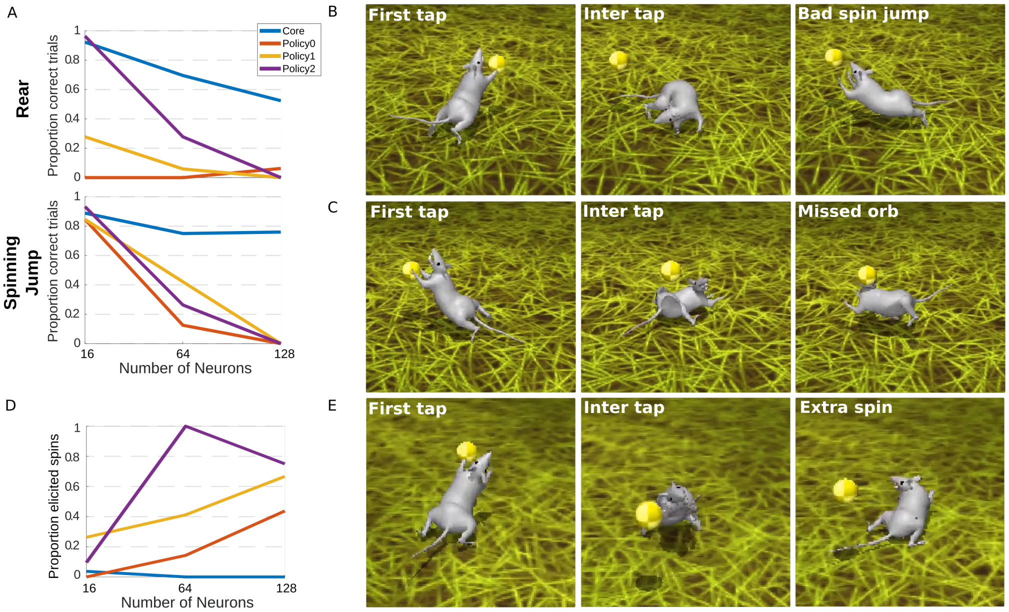

Figure 12: Causal manipulations reveal distinct roles for core and policy layers in the production of behavior. (A) Two-tap accuracy during the inactivation of units modulated by idiosyncratic behaviors within the two-tap sequence. Core inactivation has a weaker negative effect on trial success than policy inactivation for several levels of inactivation. (B) Representative example of a failed trial during core inactivation in a model that performs a spinning jump during the two-tap sequence. The model is still able to perform the spinning jump behavior, but misses the orb. (C) Representative example of a failed trial during inactivation of the final policy layer in a model that performs a spinning jump during the two-tap sequence. The model is incapable of producing the spinning jump behavior while inactivated. (D) Proportion of attempts at stimulation that successfully elicited spin behavior during the two-tap task. The efficacy of this activation was more reliable in layers closer to the motor output. (E) Representative example of a single trial in which an extra spin occurs.