Human-AI Shared Control via Frequency-based Policy Dissection

Anonymous Author(s)
Affiliation
Address
email

Abstract

Human-AI shared control allows human to interact and collaborate with AI to accomplish control tasks in complex environments. Previous Reinforcement Learning (RL) methods attempt the goal-conditioned design to achieve human-controllable policies at the cost of redesigning the reward function and training paradigm. Inspired by the neuroscience approach to investigate the motor cortex in primates, we develop a simple yet effective frequency-based approach called *Policy Dissection* to align the intermediate representation of the learned neural controller with the kinematic attributes of the agent behavior. Without modifying the neural controller or retraining the model, the proposed approach can convert a given RL-trained policy into a human-interactive policy. We evaluate the proposed approach on the RL tasks of autonomous driving and locomotion. The experiments show that human-AI shared control achieved by Policy Dissection in driving task can substantially improve the performance and safety in unseen traffic scenes. With human in the loop, the locomotion robots also exhibit versatile controllable motion skills even though they are only trained to move forward. Our results suggest the promising direction of implementing human-AI shared autonomy through interpreting the learned representation of the autonomous agents. Code will be made available.

1 Introduction

In recent years, autonomous agents trained from deep Reinforcement Learning (RL) achieve huge success in a range of applications from robot control [61, 50], autonomous driving [7, 29], to the power system in smart building [40]. Despite the capability to effectively discover feasible policy under unknown system dynamics in a model-free setting [59], the lack of generalizability [14] in unseen environments hinders the real-world deployment of learning-based neural policy. Furthermore, it remains difficult to understand the internal mechanism and the decision making of the neural network [30, 70], especially when abnormal behaviors happen [10, 69, 45]. The lack of transparency and controllability greatly limit the application of the end-to-end neural controllers in safety-critical systems.

One potential direction to make the neural controller safer and more trustworthy is incorporating human into the control loop, which we refer as *human-AI shared control*. With human involvement, the performance and the safety of the shared autonomy systems can be substantially improved [48, 15, 49], even on new tasks and unseen environments. Human-AI shared control has been implemented in various forms, such as enforcing intervention [28, 46] or providing high-level commands [15] to the AI models. Previous work also explores goal-conditioned reinforcement learning as a form of human-AI shared control, where a high-level goal is fed as an input to the policy during training [53, 2, 15, 25]. Thus the behavior of the learned agent can be controlled by varying the input goal [43, 36, 71]. However, training goal-conditioned policy requires additional modification of the

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

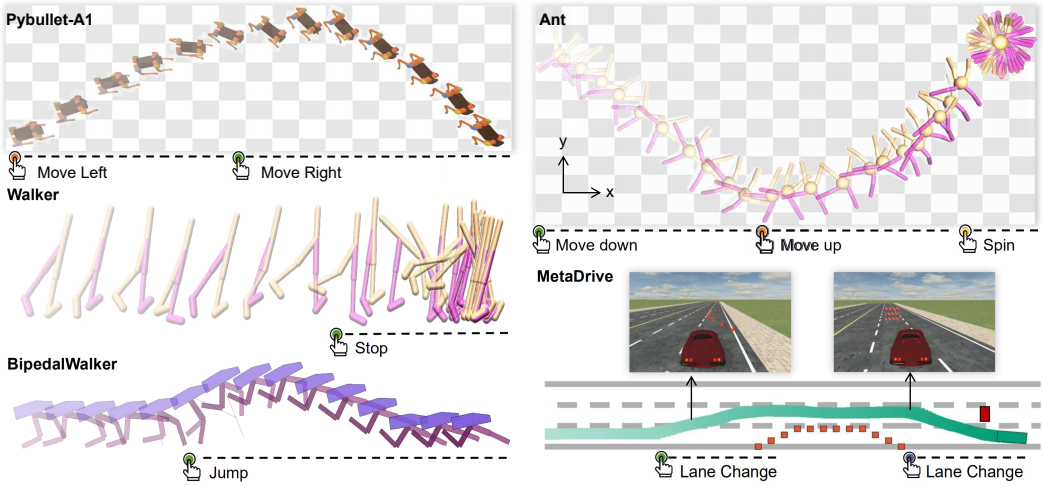

Figure 1: Examples of human-AI shared control enabled by Policy Dissection. Each autonomous agent is trained to solve its primal task. After training, the stimulation-evoked map is then generated by Policy Dissection as the interface for human to steer the agent. Various behaviors such as jumping, spinning, redirection, and lane changing can be triggered by human subject.

neural controller, algorithm, and the reward function, while the human control only happens at the goal level, still lacking interpretability and control of fine-grained movement.

In this work we develop a minimalist approach called *Policy Dissection*, to enable human-AI shared control on autonomous agents trained on a wide range of tasks. Policy Dissection does not impose assumption on the agent's training scheme, and it requires neither retraining the controller's network nor modifying the environment. Policy Dissection achieves human-AI shared control by firstly dissecting the internal representations of the learned policy and aligning them with specific kinematic attributes, and then modulating the activation of the internal representations to elicit the desired motor skills. Policy Dissection takes the inspiration from neuroscience studies on motor cortex of primates: exerting electrical stimulation on different areas of motor cortex can elicit meaningful body movements. A stimulation-evoked map can be built to reveal the relationship between the evoked body movement and the motor neurons located in different areas of motor cortex [21, 20, 33]. In order to replicate the stimulation-evoked map of a neural controller, the proposed Policy Dissection follows a frequency analysis and matching strategy, associating the kinematic attribute and the unit activation. The procedure of building stimulation-evoked map by Policy Dissection is shown in Fig. 2. Concretely, we first roll out the given policy for several episodes. After collecting a set of time series of the activities of all units, kinematics and dynamics, we identify the units whose activities align with the kinematic and dynamic attributes in a way that there is the lowest frequency discrepancy between the specific units and the kinematic attributes in the frequency domain. The stimulation-evoked map between units and attributes is thus built and used to modulate the activation intensity of identified units so that the desired motor skills can be elicited. Therefore, Policy Dissection constructs an interpretable control interface on top of the trained agent for human to interact with and evoke certain behaviors of the agent, thus achieving human-AI shared control.

We evaluate the proposed method on several RL agents ranging from locomotion robots to autonomous driving agents in simulation environments. Experimental results suggest that meaningful motor primitives emerge in the internal representation. As shown in Fig. 1, novel behaviors can be deliberately evoked by activating units related to certain motor primitives, even though they are not the necessary behaviors to solve the primal task. In the quantitative evaluation, we use the human-AI shared control system enabled by Policy Dissection to improve the generalization in test-time unseen environment and achieve task transfer. In an autonomous driving task, we first train baseline agents under mild traffic condition and evaluate the driving policy on new test environments containing unseen near-accidental scenes. Different to the poor performance of the baseline agent in these new environments, the human-AI shared control system achieves superior performance and safety guarantee, with 95% success rate and almost zero cost. In quadrupedal locomotion task, a robot with only proprioceptive state input can avoid obstacles with shared control, even though its primal task is moving forward. These results show that the proposed Policy Dissection not only helps understand

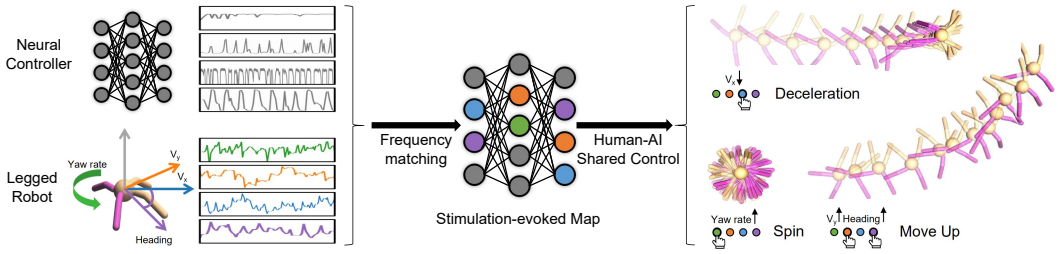

Figure 2: Overview of the proposed Policy Dissection. Taking Ant robot as an example, it first identifies the connection between unit activation and kinematic attributes (painted in the same colors). After that human can activate units to evoke desired behaviors by stimulating one or more motor primitives. For example, stimulating motor primitive related to yaw rate makes the Ant spin, while activating motor primitive related to velocities in different axes brings moving-up or deceleration.

the learned representations of the neural controller, but also brings the intriguing potential application of human-AI shared control.

2 Related Work

Human-AI Shared Control. Existing methods on human-AI shared control can be roughly divided into two categories. The first category is to have human in the control loop to train learning systems and then execute without human in test time [72]. By incorporating human into the training, previous works successfully improve the performance on visual input control tasks such as Atari game [1, 51, 63]. Robotic control tasks also benefit from human feedback [28, 39, 57, 46, 64]. The other category is to have human in both training and test time to accurately accomplish human-assistive tasks. A series of works study this problem based on the Atari Game [48, 52, 8, 31]. In addition, the human-AI shared control system is built on autonomous vehicle [19], robotics arm [27] and in multi-agent setting [31]. The reported results on various tasks reflect the effectiveness and efficiency of training and coordination with human-assistive AI.

Neural Network Interpretability. There is a growing interest in AI community to understand what is learned inside the neural network. Previous interpretability methods mostly focus on explaining the networks trained for visual tasks [73, 55, 11, 74, 38]. Various empirical studies have found that meaningful and interpretable structures emerge in the internal representation when trained for the primal tasks. For example, object detectors emerge in scene classifier [6], and disentangled layout and attribute controllers emerge from image generation model [66]. Manipulating the output image of deep generative models is achieved by editing the disentangled hidden features [5, 56, 65]. On the other hand, visuomotor control tasks can also acquire visual understanding to explain the behavior chosen by agent with visual input through saliency map [58, 3, 47, 12, 26, 22].

Neuroscience Inspired Control. The complex behaviors shown by recent autonomous agents prompt researchers to develop new analysis methods. As a source of inspiration, neuroscience provides useful tools [44, 33], and effective procedures [24, 18, 42] for understanding animal brains which are also like black-box systems. Recent works try to develop neural network-based policies from the perspective of neuroscience [17, 4, 68, 13]. In a closely related work on understanding learned agent, Merel et al. [42] use RSA [34] with CKA [32] similarity index to reveal that behaviors spanning multiple timescales are encoded in distinct neural layers. Apart from providing the neuroscientific tools, many concepts from neuroscience, such as motor primitive also inspire researchers to design generalizable controllers [62, 60, 41].

3 Policy Dissection Method

Policy Dissection aims at deriving human-controllable policy from a well-trained RL agent. It has two unique features to enable human-AI shared control: First, it operates on a well-trained RL agent without re-training or modifying the original reward/environment and the policy architecture. We only need to roll out the agent for several episodes and log its neural activity and kinematic attributes. Based on the collected records, certain units and kinematic attributes will be aligned. Thus our

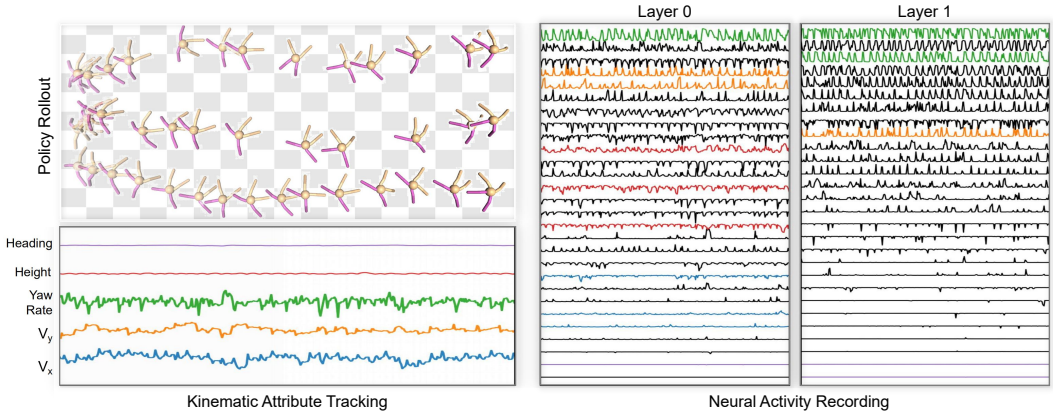

Figure 3: We first rollout the trained policy for several episodes and record the neural activities and track kinematic attributes, such as velocity and yaw rate. After applying the frequency matching to identify motor primitives, we can align units with many kinematic attributes. The curves of kinematic attributes and the unit activation in the same motor primitive are painted in the same colors. For clarity, we only show the result of one recorded episode and a proportion of units, and the curves of units are sorted by their amplitude.

method is non-intrusive to the primal task and trained model, and applicable to a wide range of environments. Second, Policy Dissection can discover generalizable motor skills that even the agent is not trained to perform. As shown in Fig. 1, the Ant robot can be steered to move up/down even it is only trained to move forward as fast as possible in the primal task. We will introduce the three essential steps of the Policy Dissection method in detail.

3.1 Monitoring Neural Activity and Kinematics

Since multi-layer perceptron (MLP) is the most common network architecture used in many RL tasks, we assume the neural controller is a MLP, which has L hidden layers and all layers have the same number of I hidden units. We denote a neuron by l, i as $z^{l,i}$, where $l = 1, ..., L$ is the index of layer and $i = 1, ..., I$ is the neuron index in each layer, and its output at time step t is $z_t^{l,i}$. The agent's kinematic attributes are represented by s_t^j, $j = 1, ..., J$, where J is the total number of attributes. We define the motor primitive as the unit $m^j \in \{z\}$ that steers the agent and causes transition of the kinematics. We assume that each kinematic attribute s^j has a corresponding motor primitive m^j to drive the attribute to desired state, and thus there are also J motor primitives. Our goal is to discover motor primitive m^j by associating the units with certain kinematic attribute.

As shown in Fig. 3, we take the Ant robot as an illustrative example. Policy Dissection starts by rolling out the pretrained agent for several episodes and tracking the activation of all units as well as the the kinematics, such as yaw rate and velocity. We record the $I \cdot L$ time series $\{\mathbf{z}^{l,i} = [z_1^{l,i}, z_2^{l,i}, ...]\}^{I \cdot L}$ measuring the neural activities and J time series of kinematic attributes $\{\mathbf{s}^j = [s_1^j, s_2^j, ...]\}^J$ for further analysis.

3.2 Associating Units with Kinematic Attributes

Observing the neural activity and kinematics recording in Fig. 3, it can be found that several repeated patterns of changing kinematics appear in different frequency when executing the policy, which is reminiscent of animal behavioral research [21, 20, 33]. It suggests that the motor skills has its predominant frequency component [42], controlling fast kinematics and slow kinematics. Since neural activation signals also encode distinct frequency information, this resemblance inspires us to process time series of both neural activity and kinematic attributes in frequency domain, where the difference of predominant frequency between two signals can be clearly revealed and the phase discrepancy of two time series are eliminated. As a consequence, this allows us to associate kinematic attributes with units whose activity best matches the corresponding motor primitives.

In Policy Dissection, all time series are transformed into the frequency domain through Discrete-time *Short-time Fourier transform* (STFT):

$$\text{STFT}(d, \omega|\mathbf{x}) \equiv \sum_{t=-\infty}^{\infty} x_t \text{W}(t-d) e^{-j\omega t}. \tag{1}$$

STFT splits the time series $\mathbf{x} = [x_1, x_2, ...]$ into a sequence of temporal window and $d \in [1, +\infty]$ denotes the time step of the center of the temporal window. STFT then conducts Fourier transform in each temporal window. $\omega \in [0, +\infty]$ denotes the frequency. To reduce spectral leakage and reserve the frequency information as much as possible, the window function $\text{W}(t)$ used in Policy Dissection is *blackman window*. Moreover, the window length is 64 with hop length 16. We apply STFT instead of naive Fourier transform since it can retain temporal information by breaking the time series into several overlapping chunks of frames. This feature is important since in the following frequency matching we focus on the similarity of resulting spectrograms as well as the temporal correlation of two signals. We use the spectrogram to characterize a measurement:

$$\text{SG}(d, \omega|\mathbf{x}) \equiv |\text{STFT}(d, \omega|\mathbf{x})|^2. \tag{2}$$

After obtaining the spectrograms for all kinematic attributes and neural activities, *frequency discrepancy* can be computed for each neuron-kinematics pair $(\mathbf{z}^{l,i}, \mathbf{s}^j)$:

$$\text{Dis}(\mathbf{z}^{l,i}, \mathbf{s}^j) = \sum_d \sum_\omega \|\text{SG}(d, \omega|\mathbf{z}^{l,i}) - \text{SG}(d, \omega|\mathbf{s}^j)\|_2. \tag{3}$$

For each kinematic attribute, *frequency matching* is applied to select the unit with minimal average discrepancy as the motor primitive responsible for the variation of kinematics:

$$m^j = \underset{z^{l,i}}{\text{argmin}} \sum_N \frac{\text{Dis}(\mathbf{z}^{l,i}, \mathbf{s}^j)}{N}, \tag{4}$$

where N is the number of collected episodes. We build $\{(m^j, s^j)\}, j = 1, ..., J$ as the stimulation-evoked map, which further enables human to evoke certain behaviors by activating one or more motor primitives.

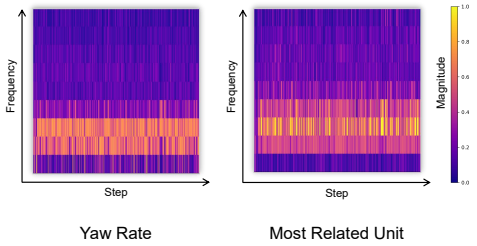

Figure 4: visualization of spectrograms of yaw rate and corresonding motor primitive.

3.3 Steering Agent via Stimulation-evoked Map

Human-AI shared can be fulfilled via the stimulation-evoked map. As shown in Fig. 2, we stimulate certain motor primitives that are tightly coupled with certain kinematics such as spinning (increasing yaw rate) and deceleration (decreasing linear velocity). In this section, we discuss the implementation details of the process to elicit the desired behaviors.

Identifying Correlation Coefficient. The correlation between activation value of motor primitives and the kinematics attributes needs to be determined. As shown in Ant robot of Fig. 1, activating the neurons controlling y-axis movement with *positive* value can drive the Ant move left, and a *negative* value makes the Ant move right.

We compute the correlation coefficient through *spectral phase discrepancy*. Since motor primitives have their own predominant frequency component responding to specific kinematic attributes responding in similar frequency, we first find the predominant frequency component:

$$\omega^* = \underset{\omega}{\text{argmax}} \sum_d \|\text{SG}(d, \omega|\mathbf{m}^j) - \text{SG}(d, \omega|\mathbf{s}^j)\|_2. \tag{5}$$

At each window of STFT, we can compute the phase discrepancy of the transforms of $\mathbf{m}^j$ and $\mathbf{s}^j$ at the predominant frequency ω^*. The average phase discrepancy at all temporal windows is:

$$p^j = \frac{1}{D} \sum_d [\Phi(\text{STFT}(d, \omega^*|\mathbf{m}^j)) - \Phi(\text{STFT}(d, \omega^*|\mathbf{s}^j))], \tag{6}$$

where D is the number of temporal windows and Φ is the function retrieving the phase. We then normalize p^j to get correlation coefficient ρ between m^j and s^j as

$$\rho^j = 1 - \frac{|2p^j|}{\pi}. \tag{7}$$

A positive ρ^j indicates we can amplify the kinematic attribute s^j by stimulating the motor primitive m^j with a positive value.

Identifying Unit Output. Suppose we want to amplify the scale of kinematic attribute s^j, the output value v^j for motor primitive m^j, namely the activation of m^j, can be determined by is:

$$v^j = \frac{k^j}{\rho^j}, \tag{8}$$

where k^j is a custom constant decided manually for each motor primitive.

Combining Motor Primitives. Some behavior requires stimulating one or more motor primitives to evoke. Taking the Ant robot shown in Fig. 2 as the example, moving towards y-axis can be implemented by two kinematic attributes including increasing V_y and heading towards y-axis. Therefore, we activate two corresponding motor primitives selected by Eq. 4 with value calculated by Eq. 8 to make the agent move towards y-axis.

4 Experiments

We first apply policy dissection to an autonomous driving task to study the learning dynamics in Sec. 4.2. We demonstrate that the emergent pivotal motor primitives are highly correlated with the success of policy learning. In Sec. 4.3, we show a novel application of the shared autonomy: improving the test-time generalization, where Human-AI shared control can be an alternative to overcoming the generalization gap between training and unseen test-time environment. The experiment demonstrates our method finds not only the correlation but causality between unit output and kinematics. In Sec. 4.4, we conduct experiments on quadrupedal robot and demonstrate that a quadrupedal robot trained to traverse complex terrain with only proprioceptive state can transfer to accomplish obstacle avoidance task with a bit human effort, even it does not have any information of the surrounding. Lastly, more qualitative demonstrations in Sec. 4.5 show that our method can be generalized to various environments and autonomy tasks.

4.1 Experimental Setting

The qualitative experiments are conducted on the Ant and Walker in MuJoCo [61] and the Bipedal-Walker [9] following the general setting. The quantitative experiments of human-AI shared control are conducted on MetaDrive and Pybullet-AI environments introduced as follows:

MetaDrive. We train agents to accomplish autonomous driving task in the safety environment of MetaDrive [37]. In this task, the goal is to maneuver the target vehicle to destination with low-level continuous control actions. An episode terminates only when the vehicle arrives the destination, drives out of road, or collides with the sidewalk. The observation of autonomous vehicle consists of map, navigation, lidar, and state information such as speed and heading.

Pybullet-A1. The legged robots locomotion experiments are conducted on the Pybullet-Unitree-A1 [16] robot. We follow the same environment setting used by Locotransformer robot [67], including terrain shape, obstacle distributions, sensors, reward definition, and termination condition. Two agents are trained. One "insensible" agent with only proprioceptive state is trained to walk on uneven terrain, while the other agent with Locotransformer is directly trained to avoid obstacle. The shared control system will be built on the "insensible" agent, where human subjects help the agent sidestep obstacles.

Training Set and Test Set. As shown in Fig. 5, for both autonomous driving and robot locomotion tasks, we prepare training set and held-out test set. For driving task, the test set is used to benchmark the test-time generalizability of human-AI shared control system and AI-only control system. We train autonomous driving policies in 50 different environments where the traffic condition is mild and all traffic vehicles and target vehicle can drive smoothly towards their destinations. No obstacles are in these environments. In testing time, the trained agents will be evaluated in the other 20 unseen

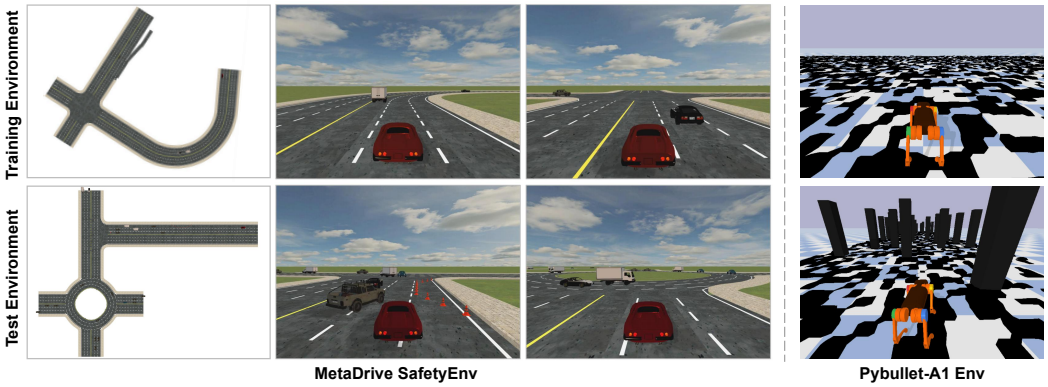

Figure 5: Examples of training and test scenes used in the generalization experiment. In MetaDrive SafetyEnv, the held-out 20 test scenes are more complex compared to 50 training scenarios. For quadruped robot locomotion task, the robot is trained to traverse open but bumpy terrains and tested on the terrain full of obstacles where obstacle avoidance ability is needed to reach the destination.

maps with *higher* traffic density. Besides, obstacles like traffic cones and breakdown vehicles will scatter randomly on the road. In this case, the target vehicle needs to be maneuvered delicately.We compare the performance and safety of the trained policy in two scenarios: runs in the test set solely, and runs with the human-AI shared control system built from Policy Dissection.

The primal task for legged robot is to move toward one direction as fast as possible. We design the test environment to be a forest-like environment where tree obstacles are randomly scattered. Therefore, though the overall goal is still to move forward, the agent needs extra effort to sidestep obstacles in the test environment.

Evaluation Metrics. For driving task, we evaluate agents on all the 20 test environments, and report the ratio of episodes where the agent arrives at the destination as the *success rate*. Also, *Out of Road rate* can be calculated if the termination is caused by driving out of road. For benchmarking the safety of agents, the collision to vehicles, obstacles, sidewalk and buildings raises a cost $+1$. In addition, human attention cost is a huge concern of human-AI shared control system. We quantify this using *human involvement*, which is measured by the number of environment steps when the human subject presses keys for activating units, or controls throttle, brake, and steering wheel. We average the sum of reward, cost and human involvement on 20 environments as *episodic return*, *episodic cost* and *human involvement*. The evaluation process above will be repeated 5 times for each agent, and we report the average value and std for all metrics. For the locomotion task, we use *moving distance* to measure the performance. Moving distance is the displacement of robot on specific direction, i.e. x-axis. The evaluation is also repeated for 5 times on the exclusive test environment.

All experiments without human are repeated 5 times with different random seeds. $N = 5$ episodes are used for applying Policy Dissection. For experiments involving human subject, we repeat each experiment with 5 human subjects. For driving task, all human subjects have driving license. Human subject can pause the experiments if any discomfort happens. No human subjects were harmed in the experiments. Human subjects have signed the experiment agreement and get compensation.

4.2 Policy Dissection for Understanding Learning Dynamics

In this section, we verify the correlation between motor primitives and relevant kinematic attributes. We first train PPO [54] agent on the MetaDrive's training environments and then conduct Policy Dissection on the trained policy and find two motor primitives in which **Primitive 1** is associated with **Speed** and **Primitive 2** associated with **Side Distance**. We then plot the evolution of the frequency discrepancy between the identified motor primitives and the kinematics alongside with the macroscopic measure of the learning progress in Fig. 6.

In the upper plot in Fig. 6, the decreasing discrepancy indicates that Primitive 1 to control speed gradually emerge. With strong negative correlation, the success rate rapidly increases when the discrepancy of speed primitive also rapidly decrease between epoch 20-30. Similarly, in the lower figure in Fig. 6, the decreasing discrepancy between Primitive 2 and side distance shows that the

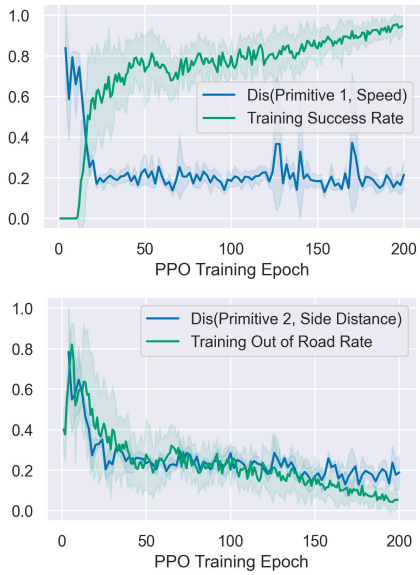

Figure 6: Learning Dynamics

Method	Human Involvement	Episodic Return	Episodic Cost	Success Rate
Human	405.57 ± 94.79	357.21 ± 38.3	0.47 ± 0.24	0.95 ± 0.03
SAC$^{w/oH}$	-	263.85 ± 34.74	5.75 ± 0.25	0.75 ± 0.12
SAC$^{H^n}$	26.52± 1.58	342.0± 6.25	0.42± 0.32	0.9± 0.0
SAC$^{H^*}$	2.85 ± 0.22	353.19 ± 13.62	0.80 ± 0.14	0.95 ± 0.03
PPO$^{w/oH}$	-	234.81 ± 23.45	3.37 ± 2.01	0.75 ± 0.03
PPO$^{H^n}$	30.58± 0.18	352.99± 6.59	0.32± 0.12	0.92± 0.02
PPO$^{H^*}$	**2.55** ± 0.35	**359.67** ± 11.20	**0.05** ± 0.08	**0.95** ± 0.02
Relu-PPO$^{H^*}$	4.65 ± 0.3	352.98 ± 7.35	0.35 ± 0.1	0.88 ± 0.07
Deep-PPO$^{H^*}$	4.88 ± 1.17	334.09 ± 2.91	0.48 ± 0.02	0.9 ± 0.02

Table 1: Improvements on the performance and the safety on test environments with human-AI shared control. H indicates "human". H^n and H^* indicates two takeover methods. H^n requires human to provide steering and throttle in the whole takeover period, while H^* only asks to activate motor primitives at the start of takeover. We also do ablation study on the activation function and number of hidden layers of MLP. The results show our method is general and robust to the policy architecture and RL methods.

internal unit of the agent is gradually specialized for controlling side distance. There exists strong correlation between the discrepancy and the out-of-road rate, which can only be reduced by improving the ability to keep side distance. We thus conclude that the pivotal primitives emerge during training.

4.3 Human-AI Shared Control for Test-time Generalization

Using the motor primitives identified above, we can determine the activation value of motor primitive for steering associated kinematics according to Eq. 8. We apply the same process to dissect the policy trained from SAC [23] with same network structure and build a stimulation evoked map for shared control. We then invite human subject to collaborate with the SAC and PPO agents on the test environments with different takeover methods. As shown in Tab. 1, the average human involvement, episodic cost, and success rate on the test environments are reported across policies with or without human involvement. The human involvement greatly improves the performance and safety when deploying the policy in unseen environments where delicate maneuvers such as side-passing and and yielding are evoked by human, if necessary, to help finish the task. Compared to naive takeover baseline indicating by H^n where human provide control signals in the whole takeover period, we find that the shared control mechanism enabled by our method only requires a small amount of human budget, since human only needs to activate the neuron at the start of takeover period.

Different from the original PPO policy which is a MLP with 2 hidden layers, 256 units per layer and "tanh" activation function, we also repeat the same experiment for Deep-PPO agent which has 6 hidden layers, and Relu-PPO which has Relu as activation function. The results reported in 1 show our method is invariant to the network architectures and RL algorithms.

4.4 Human-AI Shared Control for Task Transfer

In addition to the improved test-time generalization, the stimulation-evoked map enables RL agent to accomplish new task with the help of human. Recent quadrupedal robots trained by RL exhibit promising ability to walk on bumpy terrain via end-to-end control [35]. However, the agent only takes proprioceptive states as input without the observation of surroundings, and thus it can not avoid obstacles by its own. As shown in Fig. 5, we train the agent only to walk on uneven terrain but test it in a new environment full of obstacles. By applying our method, motor primitives related to yaw rate and speed control can be found and serve as the interface for human to steer the "insensible" quadrupedal

Method	Reward	Moving Distance
State$^{w/oH}$	18.23 ± 10.55	6.55 ± 0.04
State$^{H^*}$	413.93 ± 93.33	16.94 ± 3.08
State+Vision	445.47± 190.85	18.64 ± 6.45

Table 2: Test on Obstacle Avoidance Task.

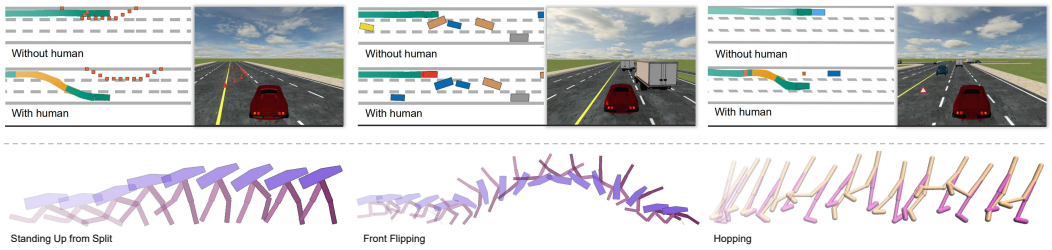

Figure 7: Qualitative demonstration of Policy Dissection. First row plots the driving results of the trained PPO agent with or without human-AI shared control in three driving cases. Each case plots the trajectories with human involvement, where different colors represent trajectory segments resulted from different motor primitives. Red and orange denote "braking" and "lane changing" respectively. The second row plots more examples of evoking behaviors in different locomotion environments.

robot to avoid obstacles. This forms the "State$^{w/oH}$" method. We also include "State+Vision" method [67], which trains a visuomotor policy for legged robot directly in the test environment, enabling it to sidestep obstacles. As shown in Table 2, with human involvement, Policy Dissection successfully transforms the "insensible" policy that is not fit to this task and greatly improves the performance, achieving comparable performance to the agent trained directly for this task.

4.5 Demonstration of Human-AI Shared Control in Diverse Environments

As shown in Fig. 7, in the top-down view we visualize the trajectories of the PPO agent with or without human shared control on the test environment of driving task. Since the PPO agent is trained in the mild traffic density without obstacles, it is incompatible to brake and side-pass. In contrast, with human in the control loop, the lane changing and braking can be deliberately evoked by human, helping the agent solve the near-accidental cases.

We provide more qualitative results of human-AI shared control on policies trained in MetaDrive, Pybullet-A1, Ant, Walker and BipedalWalker environments. Please find in the supplemental material for details about identifying motor primitives and evoking these behaviors. **We also include a demo video of human-AI shared control in the supplementary material.**

5 Conclusion

Inspired by the neuroscience approach to study the motor cortex in primates, we present a simple yet effective method called Policy Dissection to align internal neural representations and kinematic attributes of a given trained policy. The resulting stimulation-evoked map allows human to steer the agent and evoke its certain behaviors. Policy Dissection implements the human-AI shared control systems on top of the models trained from standard RL algorithms without re-training or modification. The human-AI shared control system is quantitatively evaluated in autonomous driving and locomotion tasks, and the result suggests the improvement on test-time generalizability and ability to achieve task transfer. We further provide intriguing demonstrations on various other environments to show the generality of our method and the wider applications of human-AI collaboration.

Limitations and societal impact. Our study finds that a small portion of units are not aligned with any motor primitive. The agent can still accomplish the task even if we set them as zero. More investigation is needed for understanding the role of all units and making the neural policy transparent. Another limitation of Policy Dissection is that we only evaluate this method in continuous control tasks with state vector as input. Wider range of tasks and neural network architecture such as discrete tasks and CNNs can be adopted in future research. One potential negative social impact is that the safety issue remains when deploying the shared control system in real world application. As shown in the autonomous driving experiment, our method derived from empirical study has no theoretical guarantee on the safety of human subject. The unsuccessful collaboration or conflict between human and AI may lead to potential dangers.

References

[1] O. Amir, E. Kamar, A. Kolobov, and B. Grosz. Interactive teaching strategies for agent training. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, 2016.

[2] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. *arXiv preprint arXiv:1707.01495*, 2017.

[3] A. Atrey, K. Clary, and D. Jensen. Exploratory not explanatory: Counterfactual analysis of saliency maps for deep reinforcement learning. *arXiv preprint arXiv:1912.05743*, 2019.

[4] A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, et al. Vector-based navigation using grid-like representations in artificial agents. *Nature*, 557 (7705):429–433, 2018.

[5] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba. Gan dissection: Visualizing and understanding generative adversarial networks. *arXiv preprint arXiv:1811.10597*, 2018.

[6] D. Bau, J.-Y. Zhu, H. Strobelt, A. Lapedriza, B. Zhou, and A. Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020.

[7] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[8] J. Bragg and E. Brunskill. Fake it till you make it: Learning-compatible performance support. In *Uncertainty in Artificial Intelligence*, pages 915–924. PMLR, 2020.

[9] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba. Openai gym, 2016.

[10] Z. Cao, E. Biyik, W. Wang, A. Raventos, A. Gaidon, G. Rosman, and D. Sadigh. Reinforcement learning based control of imitative policies for near-accident driving. In *Robotics: Science and Systems*, 2020.

[11] S. Carter and M. Nielsen. Using artificial intelligence to augment human intelligence. *Distill*, 2(12):e9, 2017.

[12] X. Chen, Z. Wang, Y. Fan, B. Jin, P. Mardziel, C. Joe-Wong, and A. Datta. Reconstructing actions to explain deep reinforcement learning. *arXiv preprint arXiv:2009.08507*, 2020.

[13] J. Cho, D.-H. Lee, and Y.-G. Yoon. Inducing functions through reinforcement learning without task specification. *arXiv preprint arXiv:2111.11647*, 2021.

[14] K. Cobbe, O. Klimov, C. Hesse, T. Kim, and J. Schulman. Quantifying generalization in reinforcement learning. In *International Conference on Machine Learning*, pages 1282–1289. PMLR, 2019.

[15] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700. IEEE, 2018.

[16] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. *PyBullet*, 2016.

[17] C. J. Cueva and X.-X. Wei. Emergence of grid-like representations by training recurrent neural networks to perform spatial localization. *arXiv preprint arXiv:1803.07770*, 2018.

[18] A. K. Dhawale, S. B. Wolff, R. Ko, and B. P. Ölveczky. The basal ganglia can control learned motor sequences independently of motor cortex. *BioRxiv*, page 827261, 2019.

[19] L. Fridman. Human-centered autonomous vehicle systems: Principles of effective shared autonomy. *arXiv preprint arXiv:1810.01835*, 2018.

[20] M. Graziano. *The intelligent movement machine: An ethological perspective on the primate motor system*. Oxford University Press, 2008.

[21] M. S. Graziano, C. S. Taylor, and T. Moore. Complex movements evoked by microstimulation of precentral cortex. *Neuron*, 34(5):841–851, 2002.

[22] S. Guo, R. Zhang, B. Liu, Y. Zhu, D. Ballard, M. Hayhoe, and P. Stone. Machine versus human attention in deep reinforcement learning tasks. *Advances in Neural Information Processing Systems*, 34, 2021.

[23] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

[24] R. H. Hahnloser, A. A. Kozhevnikov, and M. S. Fee. An ultra-sparse code underliesthe generation of neural sequences in a songbird. *Nature*, 419(6902):65–70, 2002.

[25] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah, et al. Urban driving with conditional imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 251–257. IEEE, 2020.

[26] A. A. Ismail, M. Gunady, H. C. Bravo, and S. Feizi. Benchmarking deep learning interpretability in time series predictions. *arXiv preprint arXiv:2010.13924*, 2020.

[27] H. J. Jeon, D. P. Losey, and D. Sadigh. Shared autonomy with learned latent actions. *arXiv preprint arXiv:2005.03210*, 2020.

[28] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[29] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah. Learning to drive in a day. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8248–8254. IEEE, 2019.

[30] J. Kim and J. Canny. Interpretable learning for self-driving cars by visualizing causal attention. In *Proceedings of the IEEE international conference on computer vision*, pages 2942–2950, 2017.

[31] P. Knott, M. Carroll, S. Devlin, K. Ciosek, K. Hofmann, A. D. Dragan, and R. Shah. Evaluating the robustness of collaborative agents. *arXiv preprint arXiv:2101.05507*, 2021.

[32] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[33] N. Kriegeskorte and J. Diedrichsen. Peeling the onion of brain representations. *Annual review of neuroscience*, 42:407–432, 2019.

[34] N. Kriegeskorte, M. Mur, and P. Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2, 2008. ISSN 1662-5137. doi: 10.3389/neuro.06.004.2008.

[35] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[36] A. Levy, R. Platt, and K. Saenko. Hierarchical reinforcement learning with hindsight. *arXiv preprint arXiv:1805.08180*, 2018.

[37] Q. Li, Z. Peng, Z. Xue, Q. Zhang, and B. Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *arXiv preprint arXiv:2109.12674*, 2021.

[38] X.-H. Li, C. C. Cao, Y. Shi, W. Bai, H. Gao, L. Qiu, C. Wang, Y. Gao, S. Zhang, X. Xue, et al. A survey of data-driven and knowledge-aware explainable ai. *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[39] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020.

[40] K. Mason and S. Grijalva. A review of reinforcement learning for autonomous building energy management. *Computers & Electrical Engineering*, 78:300–312, 2019.

[41] J. Merel, L. Hasenclever, A. Galashov, A. Ahuja, V. Pham, G. Wayne, Y. W. Teh, and N. Heess. Neural probabilistic motor primitives for humanoid control. *arXiv preprint arXiv:1811.11711*, 2018.

[42] J. Merel, D. Aldarondo, J. Marshall, Y. Tassa, G. Wayne, and B. Ölveczky. Deep neuroethology of a virtual rodent. *arXiv preprint arXiv:1911.09451*, 2019.

[43] O. Nachum, S. Gu, H. Lee, and S. Levine. Data-efficient hierarchical reinforcement learning. *arXiv preprint arXiv:1805.08296*, 2018.

[44] H. Nili, C. Wingfield, A. Walther, L. Su, W. Marslen-Wilson, and N. Kriegeskorte. A toolbox for representational similarity analysis. *PLoS computational biology*, 10(4):e1003553, 2014.

[45] D. Omeiza, H. Webb, M. Jirotka, and L. Kunze. Explanations in autonomous driving: A survey. *arXiv preprint arXiv:2103.05154*, 2021.

[46] Z. Peng, Q. Li, C. Liu, and B. Zhou. Safe driving via expert guided policy optimization. In *5th Annual Conference on Robot Learning*, 2021.

[47] N. Puri, S. Verma, P. Gupta, D. Kayastha, S. Deshmukh, B. Krishnamurthy, and S. Singh. Explain your move: Understanding agent actions using specific and relevant feature attribution. *arXiv preprint arXiv:1912.12191*, 2019.

[48] S. Reddy, A. D. Dragan, and S. Levine. Shared autonomy via deep reinforcement learning. *arXiv preprint arXiv:1802.01744*, 2018.

[49] S. Reddy, A. Dragan, S. Levine, S. Legg, and J. Leike. Learning human objectives by evaluating hypothetical behavior. In *International Conference on Machine Learning*, pages 8020–8029. PMLR, 2020.

[50] F. Richter, R. K. Orosco, and M. C. Yip. Open-sourced reinforcement learning environments for surgical robotics. *arXiv preprint arXiv:1903.02090*, 2019.

[51] W. Saunders, G. Sastry, A. Stuhlmueller, and O. Evans. Trial without error: Towards safe reinforcement learning via human intervention. *arXiv preprint arXiv:1707.05173*, 2017.

[52] C. Schaff and M. R. Walter. Residual policy learning for shared autonomy. *arXiv preprint arXiv:2004.05097*, 2020.

[53] T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal value function approximators. In *International conference on machine learning*, pages 1312–1320. PMLR, 2015.

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

[55] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *CoRR*, abs/1610.02391, 2016. URL http://arxiv.org/abs/1610.02391.

[56] Y. Shen and B. Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, 2021.

[57] J. Spencer, S. Choudhury, M. Barnes, M. Schmittle, M. Chiang, P. Ramadge, and S. Srinivasa. Learning from interventions. In *Robotics: Science and Systems (RSS)*, 2020.

[58] F. P. Such, V. Madhavan, R. Liu, R. Wang, P. S. Castro, Y. Li, J. Zhi, L. Schubert, M. G. Bellemare, J. Clune, et al. An atari model zoo for analyzing, visualizing, and comparing deep reinforcement learning agents. *arXiv preprint arXiv:1812.07069*, 2018.

[59] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[60] P. S. Thomas and A. G. Barto. Motor primitive discovery. In *2012 IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL)*, pages 1–8. IEEE, 2012.

[61] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[62] E. Ugur, E. Şahin, and E. Oztop. Self-discovery of motor primitives and learning grasp affordances. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3260–3267. IEEE, 2012.

[63] G. Warnell, N. Waytowich, V. Lawhern, and P. Stone. Deep tamer: Interactive agent shaping in high-dimensional state spaces. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[64] J. Wu, Z. Huang, C. Huang, Z. Hu, P. Hang, Y. Xing, and C. Lv. Human-in-the-loop deep reinforcement learning with application to autonomous driving. *arXiv preprint arXiv:2104.07246*, 2021.

[65] Y. Xu, Y. Shen, J. Zhu, C. Yang, and B. Zhou. Generative hierarchical features from synthesizing images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4432–4442, 2021.

[66] C. Yang, Y. Shen, and B. Zhou. Semantic hierarchy emerges in deep generative representations for scene synthesis. *International Journal of Computer Vision*, 129(5):1451–1466, 2021.

[67] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. *arXiv preprint arXiv:2107.03996*, 2021.

[68] N. Yoshida, T. Daikoku, Y. Nagai, and Y. Kuniyoshi. Embodiment perspective of reward definition for behavioural homeostasis. In *Deep RL Workshop NeurIPS 2021*, 2021.

[69] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord. Explainability of vision-based autonomous driving systems: Review and challenges. *arXiv preprint arXiv:2101.05307*, 2021.

[70] Q.-s. Zhang and S.-c. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[71] R. Zhang, F. Torabi, L. Guan, D. H. Ballard, and P. Stone. Leveraging human guidance for deep reinforcement learning tasks. *arXiv preprint arXiv:1909.09906*, 2019.

[72] R. Zhang, F. Torabi, G. Warnell, and P. Stone. Recent advances in leveraging human guidance for sequential decision-making tasks. *Autonomous Agents and Multi-Agent Systems*, 35(2):1–39, 2021.

[73] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[74] B. Zhou, D. Bau, A. Oliva, and A. Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5, Limitations
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Hyperparameters are in appendix, and experimental settings are in Section 4
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4. Experimental Setting
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]

(b) Did you mention the license of the assets? [N/A]

(c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [Yes]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [Yes] See Section 4. Experimental Setting

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [Yes] See Section 4. Experimental Setting