Proceedings of Machine Learning Research 101:32–47, 2019 ACML 2019

Towards Governing Agent's Efficacy: Action-Conditional β-VAE for Deep Transparent Reinforcement Learning

John Yang YJOHN@SNU.AC.KR
Gyuejeong Lee REGULATION.LEE@SNU.AC.KR
Simyung Chang TIMELIGHTER@SNU.AC.KR
Nojun Kwak NOJUNK@SNU.AC.KR
Seoul National University, Seoul, Korea

Abstract

We tackle the blackbox issue of deep neural networks in the settings of reinforcement learning (RL) where neural agents learn towards maximizing reward gains in an uncontrollable way. Such learning approach is risky when the interacting environment includes an expanse of state space because it is then almost impossible to foresee all unwanted outcomes and penalize them with negative rewards beforehand. We propose Action-conditional β-VAE (AC-β-VAE) that allows succinct mappings of action-dependent factors in desirable dimensions of latent representations while disentangling environmental factors. Our proposed method tackles the blackbox issue by encouraging an RL policy network to learn interpretable latent features by distinguits influenshing ices from uncontrollable environmental factors, which closely resembles the way humans understand their scenes. Our experimental results show that the learned latent factors not only are interpretable, but also enable modeling the distribution of entire visited state-action space. We have experimented that this characteristic of the proposed structure can lead to ex post facto governance for desired behaviors of RL agents.

Keywords: Transparent Policy Network, AI Governance

1. Introduction

Despite many recent successful achievements that deep neural networks (DNN) have allowed in machine learning fields (Krizhevsky et al. (2012); LeCun et al. (2015); Mnih et al. (2015)), the legibility of their high-level representations are noticeably less studied compared to the relevant studies which rather prioritize performance enhancements or task completions (Burrell (2016)). While the opaqueness of DNNs comes handy when strict labels are available for every data sample, its blackbox issue is a great element of risk especially in reinforcement learning (RL) settings where machines, or agents, are allowed to have highly intertwined interactions with their environments. Since an RL agent's policy on action selection is optimized towards maximizing the rewards, it may produce undesirable outcomes if these outcomes are not primarily penalized with negative reward signals. Yet, too much regulation would, contrarily, result in misusing the full potential of the technology (Rahwan (2018)); RL is proven of its strength over humans by, for an example of learning the game of Go, figuring to learn unprecedented winning moves (Silver et al. (2017)). Our problem setting is thus whether human is able to control and even govern machine's *efficacy* resulted by precedently optimized for an environment. Motivated so, we desire to build a deep RL framework that is interpretable while learning its way towards maximizing the cumulative rewards so that we can control its behavior according to our preference.

To induce a transparent policy network, our fundamental aim is to disentangle the controllable factors and uncontrollable factors in the reinforcement learning settings. Sharing the

same hypothesis with the works of Thomas et al. (2017), Sawada (2018) and Oh et al. (2015), we, in this paper, propose a method that allows training a deep RL policy network of which latent features are disentangled into independently controllable and uncontrollable factors so that their inner mechanism becomes interpretative. We intend to accomplish this by training an RL agent with an action-conditional β-VAE (AC-β-VAE). Adding supplementary implication of action-conditions between sequential state transitions, β-VAE, an unsupervised method that disentangles meaningful factors of variation from a distribution, efficiently disentangles controllable and uncontrollable factors. We strategically design the AC-β-VAE module to share a backbone structure with a policy network to overcome the blackbox issue, supporting the transparency of deep RL. We, later in the paper, empirically compare our proposed method against the baseline models, and furthermore extends the experiment on assessing regulating the learned behavioral components.

2. Related Works

Deep learning methods are praised of their unruled pattern extraction that yields better performance in many tasks than machines trained under human prior knowledge (Günel; Moore and Lu (2011); Vanderbilt (2012)), but as stated earlier, the blackbox characteristic of DNNs can be precarious especially in the RL setting. One of the safety factors of AI development suggested in (Amodei et al. (2016)) is avoidance of negative side effects when training an agent to complete a goal task with a strict reward function.

Attempts to open the blackbox of DNN and to understand the inner system of neural networks have been made in many recent works (Lipson and Kurman (2016); Zeiler and Fergus (2014); Bojarski et al. (2017); Greydanus et al. (2017)). Its inherent learning phenomena are reversely analyzed by observing the resultingly learned understructure. While the training progress is also analytically interpreted via information theory (Shwartz-Ziv and Tishby (2017); Saxe et al. (2018)), it is still challenging to anticipate how and why high-level features in neural models are learned in a certain way before training them. Since learning a disentangled representation encourages its interpretability (Bengio et al. (2013); Higgins et al. (2016)), it is previously reported that features of convolutional neural networks (CNN) can also be learned in a visually explainable way (Zhang and Zhu (2018)) through disentangled representation learning.

Prospection of future states conditioned by current actions is meaningful to RL agents in many ways, and action-conditional (variational) autoencoders are learned to predict sequent states in the works of Ha and Schmidhuber (2018); Oh et al. (2015) and Thomas et al. (2017). DARLA (Higgins et al. (2017)) utilizes disentangled latent representations for cross-domain zero-shot adaptations. It aims to prove its representation power in multiple similar but different environments.

While having good data representations is important for learning success of a model (Bengio et al. (2013)), disentangled representation learning has been reported to be a catalyst for many AI tasks, allowing faster convergences (Jaderberg et al. (2016); Lake et al. (2017)). A disentangled latent factor is largely dependable for an independent data generative factor while being insensitive to other factors. InfoGAN (Chen et al. (2016)), an extension of the generative adversarial network (GAN) (Goodfellow et al. (2014)), maximizes the mutual information between the data and a subset latent noise. Compact information, as a result, is encoded in each dimension of the latent representation. Dual swap disentangling method (Feng et al. (2018)) switches certain dimension(s) of two encoded latent vectors of paired input data and trains an auto-encoder to reconstruct the observation, forcing the factor in the switched dimension(s) to

learn the same attribute among the pair inputs. Additional to β-VAE (Higgins et al. (2016)), β-TCVAE (Chen et al. (2018)) and FactorVAE (Kim and Mnih (2018)) are introduced with better disentanglement performances and finer synthesized generations, but the unsupervised disentanglement methods require careful calibration of the β to ensure disentangling controllable and uncontrollable factors.

Although it is known that deep RL methods naturally connote attentiveness on important features that influence much on maximizing cumulative rewards (Wang et al. (2015); Greydanus et al. (2017)), performance can be enhanced if additional attentive features with agent's self-identity are engineered along (Jaderberg et al. (2016); Pathak et al. (2017); Choi et al. (2018)). Many recent related works engage dynamic model to distinguish or identify controllable features from (inverse) modeling dynamics of the environment. (Oh et al. (2015); Agrawal et al. (2016)), while some works focus on learning the meaningful feature representations (Jaderberg et al. (2016); Achiam et al. (2018)).

Disentanglement of controllable and uncontrollable factors have been recently studied and introduced of its importance. The most noticeable related works on disentangling visual controllable and uncontrollable factors within the latent space (Sawada (2018); Thomas et al. (2018, 2017)) require more than one network to solve the problem. In the work of Thomas et al. (2018), a selectivity loss is applied additional to the reconstruction term on an auto-encoding structure to grasp independently controllable factors through interactions with a given world. A separate network that optimizes the disentanglement of independently controllable factors is required to evaluate the selectivity score. However, the method is not able to disentangle the uncontrollable factors, which is why Sawada (2018) has proposed another method structured with two DNNs each of which separately disentangles controllable and uncontrollable factors with the pretrained models from the work of Thomas et al. (2018). Our proposed method is unique in its way of learning disentangled latent representations of independently controllable and uncontrollable factors in the latent space with a single network of variational auto-encoder, which greatly lowers computation burden. And by sharing the backbone structure with a policy network, AC-β-VAE enables an RL agent to learn latent features that are disentangled and interpretable on-policy.

3. Preliminary: β-VAE

Variational autoencoder (VAE) (Kingma and Welling (2013)) works as a generative model based on the distribution of training samples (Co-Reyes et al. (2018); Babaeizadeh et al. (2017)). VAE's goal is to learn the marginal likelihood of a sample x from a distribution parametrized by generative factors z. In doing so, a tractable proxy distribution $q_\phi(z|x)$ is used to estimate an intractable posterior $p_\theta(z|x)$ with two different parameter vectors ϕ and θ. The marginal likelihood of a data point x can be defined as:

$$\log p_\theta(x) = D_{KL}(q_\phi(z|x)||p_\theta(z|x)) + L(\theta, \phi, x, z). \tag{1}$$

Since the KL divergence term $D_{KL}(\cdot||\cdot)$ is non-negative, $L_{vae} \triangleq L(\theta, \phi, \mathbf{x}, \mathbf{z})$ sets a variational lower bound for the likelihood $\log p_\theta(x)$ and the best approximation $q_\phi(z|x)$ for $p_\theta(z|x)$ can be obtained by maximizing this lower bound:

$$L_{vae} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\phi(z|x)||p(z)). \tag{2}$$

In practice, q_ϕ and p_θ are respectively encoder and decoder that are parameterized by deep neural networks, and the prior $p(z)$ is usually set to follow Gaussian distribution $\mathcal{N}(0, I)$. The gradients of the lower bound can be approximated using the *reparametrization trick*.

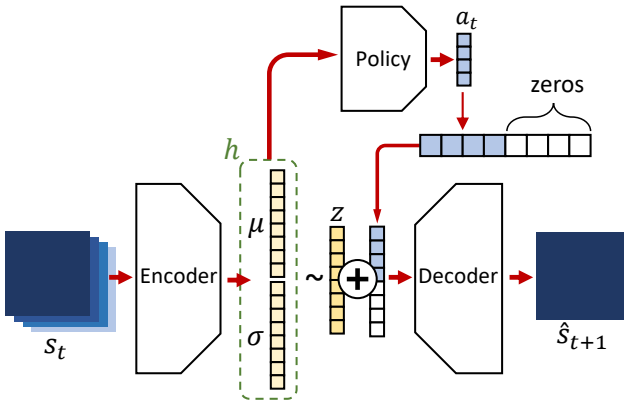


Figure 1: *Structural diagram of AC-β-VAE with a sharing policy network. Without a sharing policy network, AC-β-VAE can be trained with MDP tuple datasets. A data sample must be composed of s_t, a_t and s_{t+1} which flow with the model structure as illustrated. s_t can be represented as a stack along with few previous states, depending on environment complexity. With a policy network attached, AC-β-VAE can also be trained on-the-fly through interactions with an environment, assisting policy network.*

β-VAE (Higgins et al. (2016)) extends the work and drives VAE to learn disentangled latent features, weighting the KL-divergence term from the VAE loss function (negative of the lower bound) with a hyper-parameter $\beta > 1$:

$$L_{\beta vae} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta D_{KL}(q_\phi(z|x)||p(z)). \tag{3}$$

When β is ideally selected and does not severely interfere the reconstruction optimization, each latent factor of z is learned to be not only independent of each other, but also often interpretable, producing features with physio-visual characteristics of a given world. For better training stability, equation 3 is re-engineered into the following loss function in Burgess et al. (2018):

$$L_{\beta vae} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \gamma|D_{KL}(q_\phi(z|x)||p(z)) - C|. \tag{4}$$

where C represents the annealing capacity control hyperparemeter targetted by the KL-divergence term.

Higgins et al. (2016) models an environment with the β-Variational Autoencoder (β-VAE) to generate disentangled latent features, inducing the learned features to be interpretable. However, unsupervised methods of learning disentangled representations without labeled samples (Tang et al. (2013); Chen et al. (2018, 2016); Cohen and Welling (2014); Kim and Mnih (2018)) including β-VAE are considered unstable when certain factors are desired to be disentangled. We, in the following section, propose a method that exploits the strong disentanglement power of β-VAE with additional action-conditioning in desirable dimensions to stably disentangle controllable and uncontrollable factors of the latent representations.

4. The Proposed Model

Our proposed model is composed of two structures: a policy gradient RL method and the action-conditional β-VAE (AC-β-VAE). As shown in Figure 1, both components are designed

Algorithm 1 AC-β-VAE with an actor-critic policy network

Initialize encoder $q_\phi(h|s)$ and decoder $p_\theta(s|z)_{z\sim\mathcal{N}(h)}$.
Initialize critic $V_w(s)$, actor $\pi_\psi(a|h)$ and state s.
while *NOT stop-criterion* **do**
  $t_{start} = t$
  while $t - t_{start} \geq$ *number of step* ***or*** *NOT* $s_{terminal}$ **do**
    Take an action a_t with policy π_ψ
    Receive new state s_{t+1} and reward r_t
  end
  $R = \begin{cases} 0 & \text{for terminal } s_t \\ V_w(s_t) & \text{for non-terminal } s_t \end{cases}$
  while $i \in \{t-1, ..., t_{start}\}$ **do**
    $R \Leftarrow r_i + \gamma R$
    Compute $A(s_i, a_i)$ (for A2C or PPO)
    Sample $z_i \sim \mathcal{N}(h_i)$ and create a_i^{map}
    Predict $p_\theta(\hat{s}_{i+1}|z_i + a_i^{map})$
    Compute L_{policy} and $L_{ac-\beta vae}$ Update encoder, actor and decoder based on:
    $L_{total} = L_{policy} + \alpha L_{ac-\beta vae}$
    Update critic by minimizing the loss:
    $L_{critic}(w) = (R - V_w(s_i))^2$
  end
end

to strategically share first layers of the encoding network so that the latent features of AC-β-VAE can also become the input of the policy network. This simple shared architecture enables human-level interpretations on behaviors of deep RL methods.

Consider a reinforcement learning setting where an actor plays a role of learning policy $\pi_\psi(a_t|s_t)$ and selects an action $a \in \mathcal{A}$ given a state $s \in \mathcal{S}$ at time t, and there exists a critic that estimates value of the states $V_w(s)$ to lead the actor to learn the optimal policy. Here, ψ and w respectively denote the network parameters of the actor and the critic. Training progresses towards the direction of maximizing the objective function based on cumulative rewards, $J(\theta) = \mathbb{E}_{\pi_\psi}[\sum_t \gamma^t r_t]$ where r_t is the instantaneous reward at time t and γ is a discount factor. The policy update objective function to maximize is defined as follows:

$$L_{policy} = \mathbb{E}_\pi[\log \pi_\psi(s_t, a_t) A^\pi(s_t, a_t)]. \tag{5}$$

Here, $A^\pi(s, a)$ is an advantage function, which is defined as it is in asynchronous advantage actor-critic method (A3C) (Mnih et al. (2016)):

$$A^\pi(s_t, a_t) = \sum_{i=0}^{k-1} \gamma^i r(s_{t+i}, a_{t+i}) + \gamma^k V_w^\pi(s_{t+k}) - V_w^\pi(s_t),$$

where k denotes the number of steps. We have used the update method of *Advantage Actor Critic* (A2C) (Wu et al. (2017)), a synchronous and batched version of A3C, for Atari domain environments (Bellemare et al. (2013)). *Proximal Policy Optimization* (PPO) (Schulman and Klimov (2017)) is also used for our experiments in continuous control environments, which reformulates the update criterion with the use of clipping objective constraint $\mathcal{C}$ in the form of:

$$L_{policy} = \mathbb{E}_{\pi}\left[\frac{\pi_{\psi}(a|s)}{\pi_{\psi}^{old}(a|s)}A(s,a)\right] - \mathcal{C}D_{KL}(\pi_{\psi}^{old}(\cdot|s)||\pi_{\psi}(\cdot|s)). \tag{6}$$

Here, the subscript t for a, s and A is omitted for brevity.

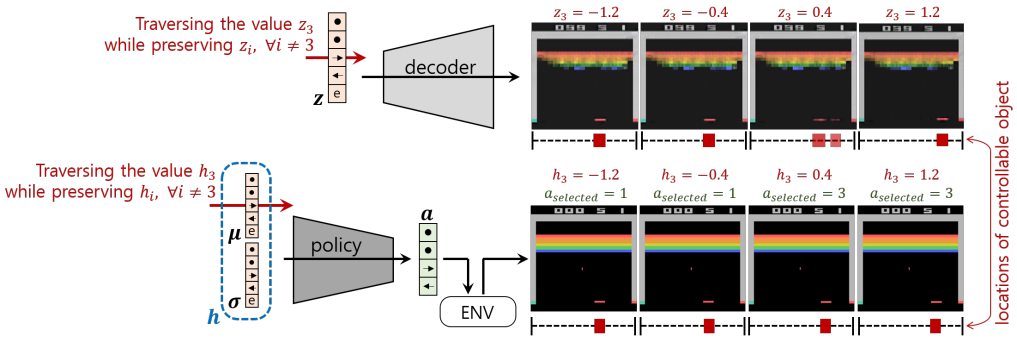


Figure 2: *The results of traversing the latent factor of our trained model on Atari game environment* BREAKOUT *with* $z \in \mathbb{R}^5$*, where* $z_{1:4}$ *are mapped with variant features of* $a \in \mathbb{R}^4$ *and* z_5 *is condensed with other environmental factors. Since the factors in the latent vector* z *of AC-*β*-VAE are defined by the vectors of mean and standard deviation* μ, σ*, traversing* i*-th value of the latent vector* z_i *is almost equivalent to traversing* μ_i*. The input DNN feature* h *of the policy network is the concatenation of* μ *and* σ*, and thus the next state due to its output actions* $a_{selected}$ *caused by traversed* μ_i *factor would be probabilistically predictable by the visual consequence estimated by the decoder with traversed* z_i*.*

4.1. Action-Conditional β-VAE (AC-β-VAE)

With a given environment, the policy network combined with the encoder produces rollouts of typical Markov tuples that consist of (s_t, a_t, r_t, s_{t+1}). A raw state s_t feeds into the encoder model and gets encoded into a representation $h \in \mathbb{R}^{2n}$, where n is the dimension of the the latent space. Since the policy network and AC-β-VAE share the parameters until this encoding process, the representation $h = [\mu^T, \sigma^T]^T$ represents a DNN feature which is inputted to the policy network while also representing a concatenated form of the mean and the standard deviation vectors $\mu, \sigma \in \mathbb{R}^n$. The vectors are reparametrized into a posterior variable $z \in \mathbb{R}^n$ through the AC-β-VAE pipeline. The output of the encoder feed-flows into the policy network $\pi(a|h)$ to output an optimal action $a \in \mathbb{R}^m$ where $m < n$ so that an RL environment responds accordingly. The action vector a is then concatenated with a vector of zeros in length of $\mathbb{R}^{n-m}$ to create, we call, an *action-mapping vector* $a^{map} = [a^T, 0^T]^T \in \mathbb{R}^n$. An element-wise sum of the latent variable z and the action-mapping vector a^{map} is performed in order to map action-controllable factors into the latent vector. This causes the latent variable sampled to be constrained by the probability of actions. The resultant vector $z_t + a_t^{map}$ is fed into the decoder network to predict the next state $\hat{s}_{t+1}$. The prediction is then compared with the real state s_{t+1} given by the environment after the action taken. For an MDP tuple collected at time t, the loss of AC-β-VAE is computed with the following loss function:

$$L_{ac-\beta vae} = \mathbb{E}_{q_\phi(h_t|s_t)}[\log p_\theta(s_{t+1}|z_t + a_t^{map})]_{z_t \sim \mathcal{N}(h_t)} - \beta D_{KL}(q_\phi(z_t|s_t)||\mathcal{N}(0, I)). \tag{7}$$

As one can see, the AC-β-VAE model can be trained either simultaneously with the policy network or separately, and all our experiments are performed with the former because it is more

practical. At each iteration of update, the total objective function value is calculated with the weighted sum of objective function values from both models:

$$L_{total} = L_{policy} + \alpha L_{ac-\beta vae} \tag{8}$$

where α is the weight balance parameter. Since exploration based on the error between generated outputs and the ground-truths have already been proven on the training enhancement in many RL related works (Oh et al. (2015); Ha and Schmidhuber (2018); Tang et al. (2017)) our model rather focuses on feasible training of a transparent neural policy network and modeling self-efficacy of agents, not on RL performance improvement. We thus choose relatively small-valued α not to confuse the policy network too much. A basic pseudo-code for the training scenario of our proposed structure is provided in Algorithm 1.

4.2. Mapping Action-Controllable Factors

Learning visual influence was previously introduced of its importance and implicitly solved in the works of Oh et al. (2015) and Greydanus et al. (2017). Distinguishing directly-controllable objects and environment-dependent objects reflects much of how a human perceives the world. Restricting in the world of Atari game domains as an example, it is intuitive for a human agent to first figure out 'where I am in the screen' or 'what I am capable of with my actions' and then work their ways towards achieving the highest score.

We show in the experiment section that AC-β-VAE allows RL agents not only to explicitly learn visual influences of their actions, but also learn them in a human-friendly way. By traversing each element of the latent vector, we are able to interpret which dimensions are mapped with actions and which are mapped with other environmental factors.

4.3. Transparent Policy Network

As mentioned earlier, the encoder and the policy network can be grouped as one bigger policy network model with an interpretable layer constrained by the AC-β-VAE loss. Unlike high-level features from conventional DNN models, the inner features of our policy network are consequentially interpretable.

Figure 2 illustrates how our policy network becomes transparent. If the action-dependent factors are disentangled in the latent vector $z \in \mathbb{R}^n$ and mapped into $z_{1:m}$, then so they are in $\mu_{1:m}$ and $\sigma_{1:m}$ because they define the sampling distribution of z_i where i denotes the dimensional location. The variational samplings from the latent space of VAE is defined as: $z_i = \mu_i + \sigma_i \epsilon_i$ where ϵ is an auxiliary noise variable $\epsilon \sim \mathcal{N}(0, 1)$. Since the σ value controls mainly the scale of sampled ϵ, traversing z_i is almost equivalent to traversing μ_i Thus, traversing μ_i encourages the policy network to cause actions as predictions of each traversing value of z_i for $i \leq m$.

5. Experiments

In this section, we present experimental results that demonstrate the following key aspects of our proposed method:

- By mapping actions into the latent vector of β-VAE, action-controllable factors are disentangled from other environmental factors.
- Governance over the optimized behavior of an agent can be made based on human-level interpretation of learned latent behavioral factors.

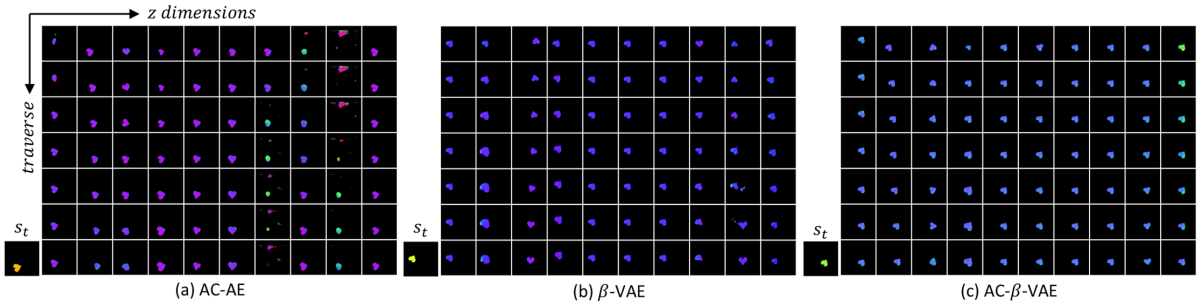


Figure 3: *The figures are better seen when zoomed in. Visual qualitative results of AC-β-VAE trained on-policy with a random policy through interactions with the dSprites environment without extrinsic rewards. AC-AE and AC-β-VAE are conducted with action-conditioning. Actions of vertical/horizontal moving, rotating, and scaling are respectively mapped into the first four dimensions ordered 1st to 10th from left to right. Deterministic AC-AE is not able to disentangle the environmental factor (the color variations), and is also composed of unintererepretable high-level representations. β-VAE's disentanglement results do not guarantee disentanglement of controllable and uncontrollable factors. Training hyper-parameters and the objective function are selected as done in Burgess et al. (2018).*

	VAE (β=1)	β-VAE (β=6)	AC-VAE (β=1)	AC-β-VAE (β=6)
Avg. Disent.	0.191	0.813	0.798	**0.865**
Avg. Compl.	0.217	0.741	0.754	**0.766**

Table 1: *The quantitative scores of disentanglement and completeness averaged over dimensions of the latent vector learned with (s_t, a_t, s_{t+1}) tuples from dSprites environment.*

We have experimented our method in three different environment types: dSprites, Atari and MuJoCo.

dSprites Environment is an environment we have designed with the *dSprites* dataset (Matthey et al. (2017)). The environment provides 64×64 sized visual states of a heart-shaped object based on action inputs without extrinsic rewards. When an episode starts with the object randomly positioned, randomly rotated, randomly scaled, and randomly colored, and each episode ends when the number of interactions reaches 30. At each step, the heart-shaped object responds to a 4-dimensional action vector composed of the following discrete independent action input: move vertically (upward, downward or no-action) and horizontally (left, right or no-action), scale (enlarge, shrink or no-action) and rotate (left, right or no-action) by a unit. Each action factor is independently chosen by an uniform distribution, and the color of the heart object is changing from red to green, to blue, and then back to red (R→G→B→R) independently from the agent's actions[1]

Atari Learning Environment is a software framework for assessing RL algorithms (Bellemare et al. (2013)). Each frame is considered as a state and immediate rewards are given for every state transitions. Our method is experimented in the Atari game environments of Breakout, Seaquest and Space-Invaders.

1. If environmental factor is not patterned during sequent transitions from s_t to s_{t+1}, VAE framework is not able to learn transition distribution.

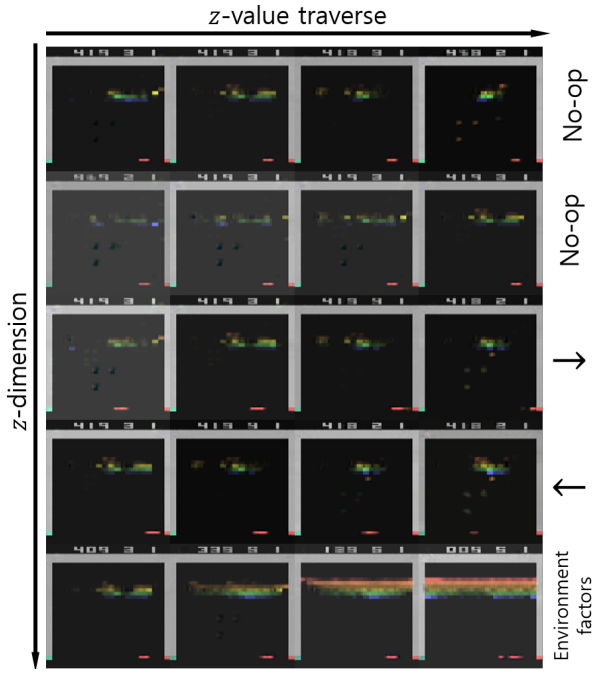


Figure 4: *The images are the estimated next states obtained by traversing the latent vector* $z \in \mathbb{R}^5$ *learned by AC-β-VAE with* β*=10 and* α*=0.001 on the Atari game environment* Breakout. *The factors at* $z_{1:4}$ *are mapped with the control factors such as movements of the paddle, and* z_5 *is mapped with the environmental factors such as bricks and the scoreboard.*

MuJoCo Environment provides a physics engine system for rigid body simulations (E. Todorov and Tassa. (2012); G. Brockman and Zaremba. (2016)). Four robotics tasks are engaged in our experiments: Walker2d, Hopper, Half-Cheetah and Swimmer. A state vector represents the current status of a provided robotic figure, each factor of which is unknown of its physical meaning.

As an encoder and a decoder, we have used a convolutional neural network (CNN) for Atari environments and fully-connected MLP networks for dSprites and MuJoCo environments. For the stochastic policy network, we have used a fully-connected MLP. PPO and A2C are applied to optimize agent's policy for continuous control and discrete actions, respectively. Most of hyper-parameters for the policy optimization are referred from the works of (Schulman and Klimov (2017); Wu et al. (2017)).

5.1. Disentanglement & Interpretability

To demonstrate the disentanglement performance and interpretability of the proposed algorithm, we have experimented our method with (s_t, a_t, s_{t+1}) tuples from environments mentioned above.

Figure 3 and Table 1 illustrate the results for the dSprites environment. The metric framework suggested by Eastwood and Williams (2018) with a random forest regressor is applied to present the quantitative results of disentanglement and completeness. Since the metric system is based on the disentanglement for the conventional VAE and β-VAE where predictions are targetted by the inputs, our metric results are not strictly comparable to the ones reported in the original work.

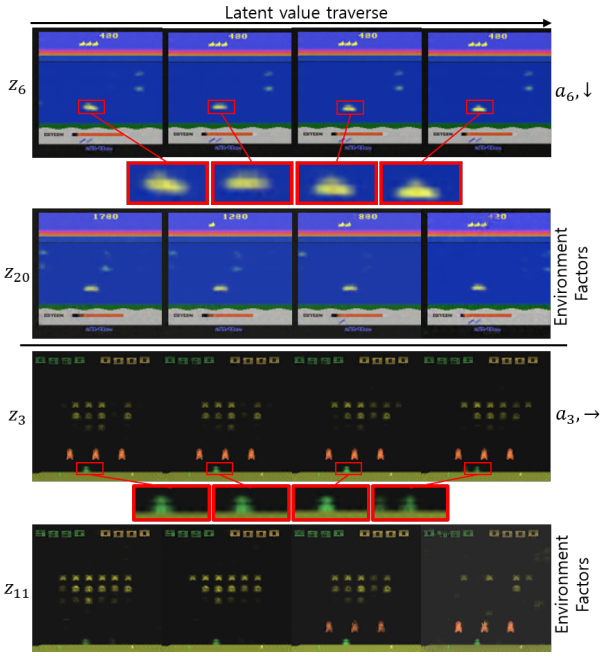
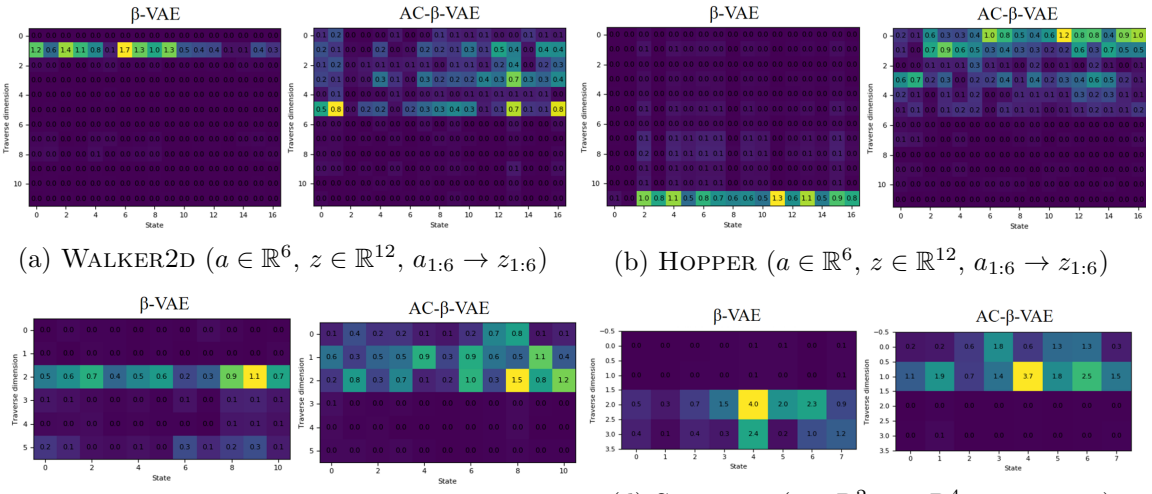


Figure 5: *The images are the estimated sequent states obtained by traversing the latent vector* $z \in \mathbb{R}^{20}$ *learned by A2C policy and AC-*β*-VAE with* β*=10 and* α*=0.001 on Atari game environments* SEAQUEST *(top) and* SPACE-INVADERS *(bottom) with action spaces of* $\mathbb{R}^{18}$ *and* $\mathbb{R}^6$*, respectively. Because of a small movement per action, we have enlarged the ego at a fixed location (red box).*

For Figure 3, AC-β-VAE and other entries are trained on-policy with 1.1M time-steps of interactions by a random policy. In other words, no policy network is needed to train in this experiment, but the datasets for the candidate entries depend on a policy that generates random actions. The baseline models include an action-conditional version of a deterministic autoencoder, labeled as AC-AE, and β-VAE trained with inputs and targets of either s_t or s_{t+1}, each with 50% of chance for the fair comparison. In this paper, β-VAE is considered sufficient to represent other unsupervised methods of disentangling latent representations that have followed its work since our aim is to disentangle indenpendently controllable factors and uncontrollable factors, rather than disentangling all variational factors. All candidate methods have learned to generate latent vectors with action-dependent factors mapped in desired dimensions (first-four dimensions from the left in the figure), except for β-VAE. AC-AE is shown to be difficult to strictly disentangle uncontrollable factors (the color variations), and some of the captured variations are high-level and uninterpretable. β-VAE strongly disentangles variational factors, but as mentioned earlier, its learning is unstable to guarantee the disentanglement of controllable and uncontrollable factors. Although the same weight on the KL-diveregence term and the same action trajectories are applied for both AC-β-VAE and β-VAE during traning, AC-β-VAE performs better disentangling independently controllable and uncontrollable factors.

The results for the Atari environments in Figure 4 and Figure 5 show that the latent vector trained with our method models the given environment successfully. All the visited state space and learned behaviors can be projected by traversing each dimension of the latent vector. In that sense, our method can be considered as an action-conditional generative model. Because AC-β-VAE can model the world in an egocentric perspective, all the sequences of (state-action-next state) can be re-simulated. Such trait may advance many RL methods since similar models are

used for an exploration guidance (Tang et al. (2017)) or as the imagery rehearsals for training (Ha and Schmidhuber (2018)).



(a) Walker2d ($a \in \mathbb{R}^6$, $z \in \mathbb{R}^{12}$, $a_{1:6} \to z_{1:6}$)

(b) Hopper ($a \in \mathbb{R}^6$, $z \in \mathbb{R}^{12}$, $a_{1:6} \to z_{1:6}$)

(c) Half-Cheetah ($a \in \mathbb{R}^3$, $z \in \mathbb{R}^6$, $a_{1:3} \to z_{1:3}$)

(d) Swimmer ($a \in \mathbb{R}^2$, $z \in \mathbb{R}^4$, $a_{1:2} \to z_{1:2}$)

Figure 6: *Traverse results in the MuJoCo environments. The vertical axis represents the dimensions of the latent vector, and the horizontal axis represents the dimensions of the states. The numbers in the boxes represent the standard deviations of each dimensional factor of the following state, s_{t+1}, when traversing the corresponding dimensional factor of the latent vector. Compared to the traverse for unmapped dimensions, the standard deviations of state values in the action-mapped dimensions are larger. Right arrows indicate action-mapping dimensional locations.*

Figure 6 shows the quantitative results of the traverse experiment on the MuJoCo environment. Numbers on the heat-map represent the standard deviations for each dimension's state values when traversing dimensional factor. The higher standard deviation value in the traverse of a specific dimension means the more effects the traversing dimension have on immediate state changes. Unlike other environments, the MuJoCo environment has no environmental factors, and the current state is represented by the preceding movement of the given robotic body. As shown in Figure 6, since the standard deviation of the state values during the traverse of the dimensions that are mapped with actions is larger than the unmapped ones, we can see the proposed algorithm is able to learn the disentangled action-dependent latent features. However, it is limited from clear visual interpretation compared to the experimental cases in other environments because the actions in the MuJoCo environment is defined as a continuous control of torques for all joints and it is conjectured that the movement of one joint affects the whole status of the body.

5.2. Controlling and Governing efficacy

To verify the controllability of an agent's optimized efficacy, we traverse the latent factors over the environment-specific range during an episode on the learned network. In order to examine s_{t+1}, the environment output, the traversal is conducted before reparameterization (μ vector). Furthermore, to get a clear view on the effect of action-mapped dimensions of the latent vector, we set all of the value of action mapped dimensions to zero except for the traversing one and

those unmapped dimension of the latent vector. These experiments are conducted on the Mujoco environments, and traverse range is set as [-5, 5] for every tasks.

The learned behavior in each latent dimension is also depicted in Figure 7. The resultant traverses of action-mapped dimensions on latent factors yield in behavioral movements that are combinations of multiple joint torque values. Unlike in Atari environments with discrete action spaces, AC-β-VAE is constrained with various combinations of continuous action values during training simulations. When the policy network is optimized to accomplish a goal behavior such as walking, the action-mapped latent factors are learned to represent required behavioral components of spreading or gathering the legs. Therefore, μ vector represents variations in combinations of multiple joint movements, which allows for ease of visual comprehension on agent's optimized efficacy. This clearly shows the possibility of governance over an RL agent's efficacy with human-level interpretations through controlling the values of the μ vector in the latent space.

We have taken the advantage of our transparent policy network and derived another behavior by controlling learned behavioral components. An RL agent is able to learn with a reward function defined by human preference to perform, for example, a back-flip motion in Hopper environment (Christiano et al. (2017)). Showing a promising result of human enforcements on an RL model, our method enables governance over the agent's optimized behavior in Half-Cheetah environment. After identification of behavioral components by traversing each element of the μ vector, we are able to express another behavior of the agent, a back-flip in this case, as shown in Figure 8.

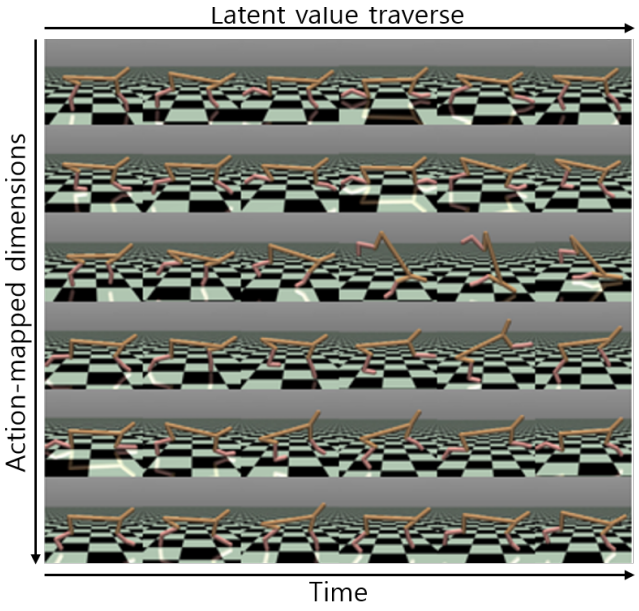


Figure 7: *For* Half-Cheetah *environment with continuous control, latent behavioral factors can be interpreted by traversing latent values in time. As a result, each action-mapped latent feature is responsible for a behavioral factor.*

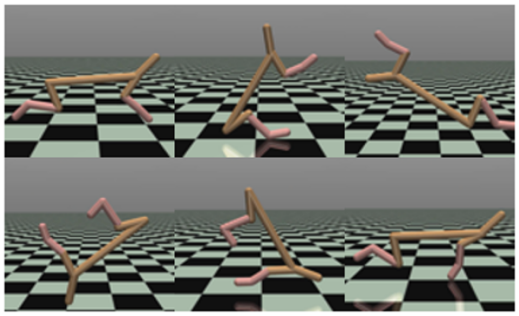


Figure 8: *Example of governing the agent movement in MuJoCo environment of* Half-Cheetah. *The robotic body is conducting a back-flip movement which is induced by controlling latent values at first and second dimensions of the learned μ vector shown in Figure 7 .*

6. Conclusion

In this paper, we propose the action-conditional β-VAE (AC-β-VAE) which, for a given input state s_t at time t, predicts next state s_{t+1} conditioned on an action a_t, sharing a backbone structure with a policy network during a deep reinforcement learning process. Our proposed model not only learns disentangled representations but distinguishes action-mapped factors and uncontrollable factors by partially mapping control-dependent variant features into the latent vector. Since the policy network combined with the preceding encoder can be considered as one bigger policy network that takes raw states as inputs, with AC-β-VAE, we are able to build a transparent RL agent of which latent features are interpretable to human, overcoming conventional blackbox issue of Deep RL. Such transparency allows human governance over the agent's optimized behavior with adjustments of learned latent factors. We plan on the relevant studies for applications of the action-mapped latent vector.

7. Acknowledgement

This work was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) (2017M3C4A7077582).

References

Joshua Achiam, Harrison Edwards, Dario Amodei, and Pieter Abbeel. Variational option discovery algorithms. *arXiv preprint arXiv:1807.10299*, 2018.

Aishwarya Agrawal, Dhruv Batra, and Devi Parikh. Analyzing the behavior of visual question answering models. *arXiv preprint arXiv:1606.07356*, 2016.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.

Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8): 1798–1828, 2013.

Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. *arXiv preprint arXiv:1704.07911*, 2017.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

Jenna Burrell. How the machine 'thinks': Understanding opacity in machine learning algorithms. *Big Data & Society*, 3(1):2053951715622512, 2016.

Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. *arXiv preprint arXiv:1802.04942*, 2018.

Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.

Jongwook Choi, Yijie Guo, Marcin Moczulski, Junhyuk Oh, Neal Wu, Mohammad Norouzi, and Honglak Lee. Contingency-aware exploration in reinforcement learning. *arXiv preprint arXiv:1811.01483*, 2018.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, 2017.

John D Co-Reyes, YuXuan Liu, Abhishek Gupta, Benjamin Eysenbach, Pieter Abbeel, and Sergey Levine. Self-consistent trajectory autoencoder: Hierarchical reinforcement learning with trajectory embeddings. *arXiv preprint arXiv:1806.02813*, 2018.

Taco S Cohen and Max Welling. Transformation properties of learned visual representations. *arXiv preprint arXiv:1412.7659*, 2014.

T. Erez E. Todorov and Y. Tassa. Mujoco: A physics engine for model-based control. *International Conference on Intelligent Robots and Systems (IROS)*, 2012.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. 2018.

Zunlei Feng, Xinchao Wang, Chenglong Ke, An-Xiang Zeng, Dacheng Tao, and Mingli Song. Dual swap disentangling. In *Advances in Neural Information Processing Systems*, pages 5898–5908, 2018.

L. Pettersson J. Schneider J. Schulman J. Tang G. Brockman, V. Cheung and W. Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

Sam Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding atari agents. *arXiv preprint arXiv:1711.00138*, 2017.

Mehmet Günel. Googlenet.

David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

Irina Higgins, Arka Pal, Andrei A Rusu, Loic Matthey, Christopher P Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. *arXiv preprint arXiv:1707.08475*, 2017.

Max Jaderberg, Volodymyr Mnih, Wojciech Marian Czarnecki, Tom Schaul, Joel Z Leibo, David Silver, and Koray Kavukcuoglu. Reinforcement learning with unsupervised auxiliary tasks. *arXiv preprint arXiv:1611.05397*, 2016.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

Hod Lipson and Melba Kurman. *Driverless: intelligent cars and the road ahead*. Mit Press, 2016.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

Matthew Michaels Moore and Beverly Lu. Autonomous vehicles for personal transport: A technology assessment. 2011.

Junhyuk Oh, Xiaoxiao Guo, Honglak Lee, Richard L Lewis, and Satinder Singh. Action-conditional video prediction using deep networks in atari games. In *Advances in neural information processing systems*, pages 2863–2871, 2015.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–17, 2017.

Iyad Rahwan. Society-in-the-loop: programming the algorithmic social contract. *Ethics and Information Technology*, 20(1):5–14, 2018.

Yoshihide Sawada. Disentangling controllable and uncontrollable factors of variation by interacting with the world. *arXiv preprint arXiv:1804.06955*, 2018.

Andrew Michael Saxe, Yamini Bansal, Joel Dapello, Madhu Advani, Artemy Kolchinsky, Brendan Daniel Tracey, and David Daniel Cox. On the information bottleneck theory of deep learning. 2018.

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

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354, 2017.

Haoran Tang, Rein Houthooft, Davis Foote, Adam Stooke, OpenAI Xi Chen, Yan Duan, John Schulman, Filip DeTurck, and Pieter Abbeel. # exploration: A study of count-based exploration for deep reinforcement learning. In *Advances in Neural Information Processing Systems*, pages 2753–2762, 2017.

Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Tensor analyzers. In *International Conference on Machine Learning*, pages 163–171, 2013.

Valentin Thomas, Jules Pondard, Emmanuel Bengio, Marc Sarfati, Philippe Beaudoin, Marie-Jean Meurs, Joelle Pineau, Doina Precup, and Yoshua Bengio. Independently controllable features. *arXiv preprint arXiv:1708.01289*, 2017.

Valentin Thomas, Emmanuel Bengio, William Fedus, Jules Pondard, Philippe Beaudoin, Hugo Larochelle, Joelle Pineau, Doina Precup, and Yoshua Bengio. Disentangling the independently controllable factors of variation by interacting with the world. *arXiv preprint arXiv:1802.09484*, 2018.

Tom Vanderbilt. Let the robot drive: The autonomous car of the future is here. *Wired Magazine, Conde NAST, www. wired. com*, pages 1–34, 2012.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado Van Hasselt, Marc Lanctot, and Nando De Freitas. Dueling network architectures for deep reinforcement learning. *arXiv preprint arXiv:1511.06581*, 2015.

Yuhuai Wu, Elman Mansimov, Roger B Grosse, Shun Liao, and Jimmy Ba. Scalable trust-region method for deep reinforcement learning using kronecker-factored approximation. In *Advances in neural information processing systems*, pages 5279–5288, 2017.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

Quan-shi Zhang and Song-Chun Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.