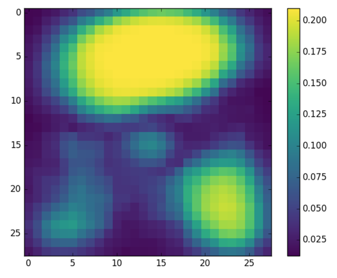


EXPLORATION FOR MULTI-TASK REINFORCEMENT LEARNING WITH DEEP GENERATIVE MODELS

SAI PRAVEEN B, JS SUHAS, BALARAMAN RAVINDRAN

PROBLEM

Learning to solve multiple tasks simultaneously is the Multi-task reinforcement learning(MTRL) problem. MTRL can be solved by either planning after deducing the current MDP or ignoring MDP deduction and learning a policy over all the MDPs combined.
This is a difficult problem due to the following reasons.

1. Learning and discovering common structure in a distribution of MDPs is hard
2. Partial observability makes modeling the MDPs distribution harder

In our work, we try to solve the problem of planning in the MTRL setting by interleaving MDP deduction with planning.

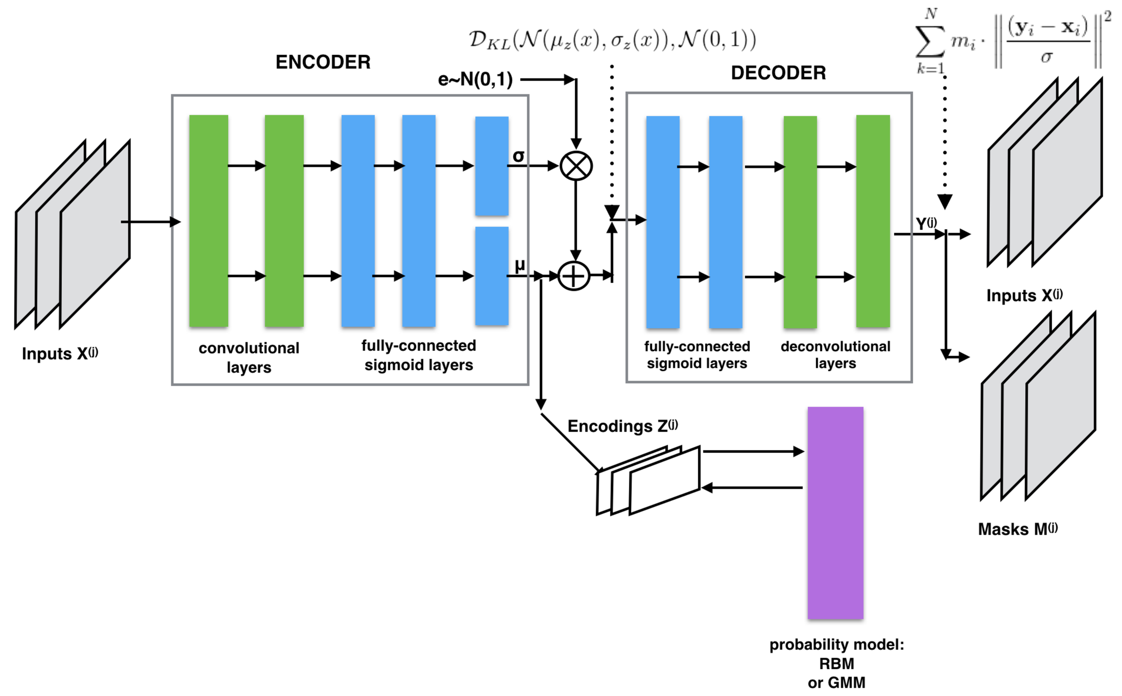
CONTRIBUTIONS

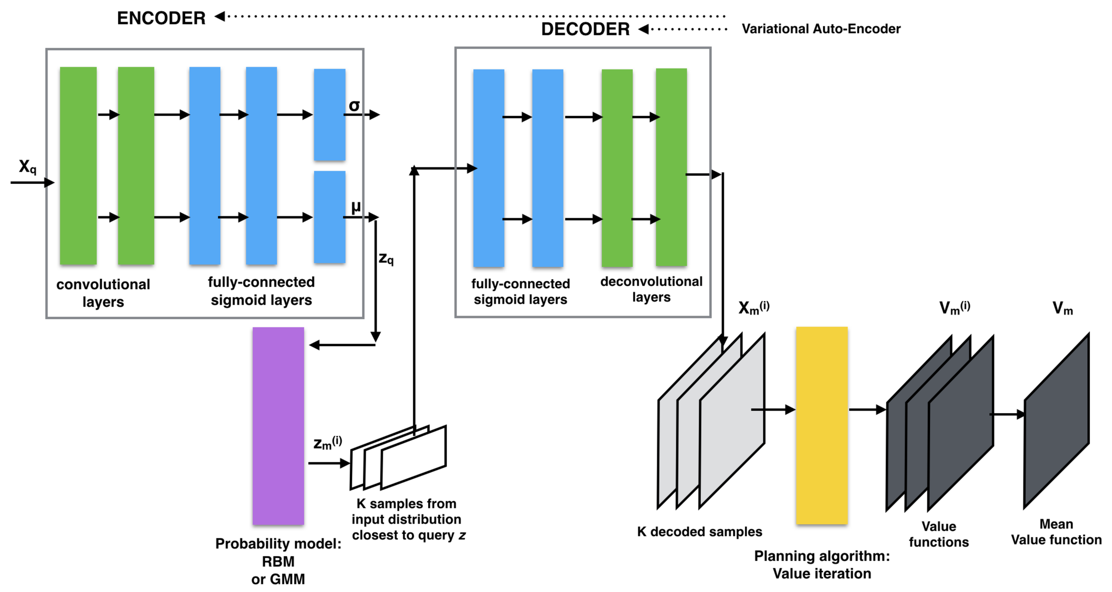
We use deep generative models to learn the MDP distribution posterior conditioned on the observational evidence. The contributions of this paper are two-fold.

1. We propose a new architecture, *Deep Generative Model*, using Convolutional Variational Autoencoders and Gaussian-Bernoulli RBMs, taking a Bayesian approach to the MTRL problem
2. We propose a new exploration bonus using Jacobian of the encoder with a simple interpretation

To the best of our knowledge, our work is the first to propose such a model and the exploration bonus.

DEEP GENERATIVE MODEL





The figures above show the *Train* and *Query* architectures. We use a modified loss function for training the variational autoencoder to work with partial inputs. The encoding produced for the input $\mathbf{X}$ is denoted by $\mathbf{Z}$. Output of the decoder is denoted by $\mathbf{Y}$. We model the MDP distribution in the encoding space, $\mathbf{Z}$, using a Gaussian-Bernoulli Restricted Boltzmann Machine.
To query the model, we use the partial image as input to obtain the reconstructed encoding, $\mathbf{Z}$.

PLANNING

At each step, we use the partial image as input to the *Query* architecture and obtain the reconstructed encoding, $\mathbf{Z}$. Using $\mathbf{Z}$ as the visible layer inputs,$\vec{v}$, we sample the hidden layer, $\vec{h}$. We then sample K visible layers from the posterior $\mathbf{p}(\vec{v}|\vec{h})$. These are then decoded to obtain K MDPs.
Planning is done using an *aggregate* value function. For each state, s, define its aggregate value function, $\bar{V}(s)$ as

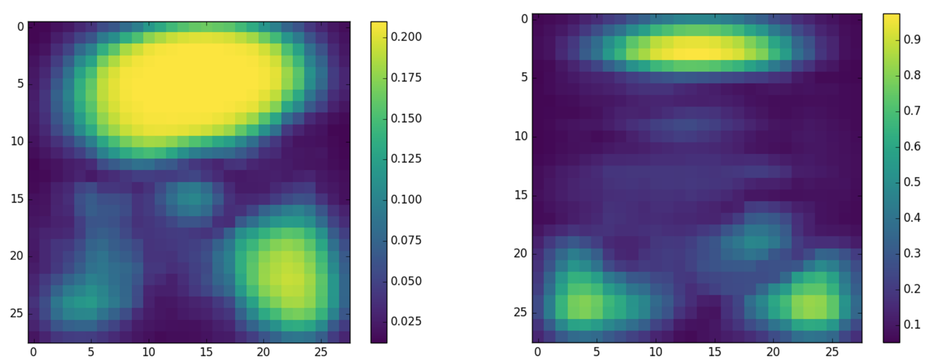
$$\bar{V}(s) = \mathbb{E}_{\mathbf{m}\sim p(\mathbf{y}|\mathbf{x})}\left[V_{\mathbf{m}}(s)\right] \approx \frac{\sum_{k=0}^{K} V_{\mathbf{m}_k}(s)}{K}$$

With actions persisting only for τ steps, we use Value Iteration (40 iters) to obtain approximate $V_{\mathbf{m}_k}$. A quicker estimate can be obtained using Monte-Carlo methods when the state-space is large.

JACOBIAN EXPLORATION BONUS

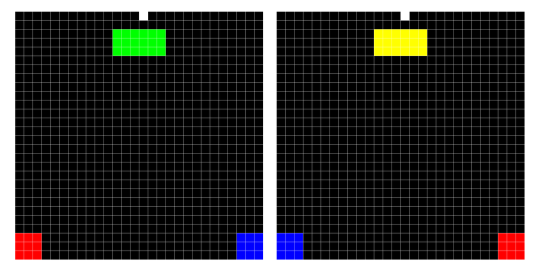
To incentivize the agent to visit decisive locations, we introduce the Jacobian Exploration Bonus based on the change in the embedding $\mathbf{Z}$. Intuitively, the embedding $\mathbf{Z}$ has the highest change when input contains information decisive in aiding reconstruction. Formally, we define this as

$$B_{\alpha}(s) = \alpha \cdot tanh\left(\epsilon + \left\|\frac{\partial \mathbf{z}}{\partial \mathbf{x_s}}\right\|\right)$$

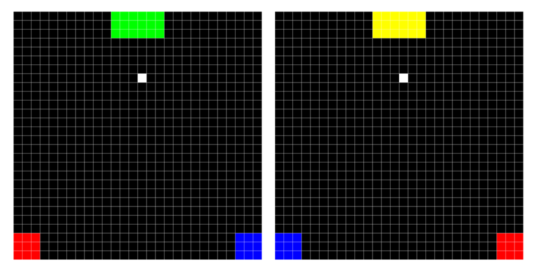


These figures show the final *Jacobian* exploration bonuses in BW-E and BW-H worlds.

EXPERIMENTS



BW-E World



BW-H World

We propose the Back-World Easy(BW-E) and Back-World Hard(BW-H) worlds with two 28x28 grid-world MDPs each. In each world, color of *marker* pixels indicates goal location. BW-E is easier to solve with markers lying on most paths from start to end, while BW-H requires a detour from otherwise optimal path.
We benchmark **STRL**(Value Iteration), **MTRL-0**(Deep Generative Model without any bonuses) and **MTRL-α**(Deep Generative Model with Jacobian exploration bonus) on BW-E and BW-H worlds.

Table 1: Average Reward

World	STRL	MTRL-0	MTRL-α
BW-E	0.21	**0.99**	**0.99**
BW-H	0.23	0.92	**0.99**

Table 2: Average Episode Length

World	STRL	MTRL-0	MTRL-α
BW-E	184.19	**46.20**	46.29
BW-H	183.64	54.0	**45.8**

Since the deep generative model aids in input reconstruction, performance on BW-E and BW-H is better for **MTRL-0** and **MTRL-α**. STRL fails to identify decisive *marker* locations and because they lie on most paths from start to end, **MTRL-0** is able to reach **MTRL-α** performance.
In BW-H, optimal strategy requires taking a detour and visiting markers before ascertaining goal locations. **MTRL-0** has no incentive to take the detour while **MTRL-α** uses Jacobian Exploration Bonus to guide planning.
As expected, **MTRL-α** has the smallest episode lengths and highest rewards in all experiments, demonstrating the need for an exploration bonus to reduce uncertainty about the current MDP.

REFERENCES

[1] Sai Praveen Bangaru, JS Suhas and Balaraman Ravindran. *Exploration for Multi-task Reinforcement Learning with Deep Generative Models* arXiv:1611.09894 [cs.AI]
[2] Junhyuk Oh, Valliappa Chockalingam, Satinder P. Singh and Honglak Lee *Control of Memory, Active Perception, and Action in Minecraft* arXiv:1605.09128 [cs.AI]

A FUTURE DIRECTION

Can we use the MDP reward structure to further refine the Jacobian Bonus to get some sort of an *utility* interpretation? We would like to explore this by formulating new exploration bonuses.

Can our model scale to Minecraft-like 3D environments used in [2] with minimal architectural changes? How well does it perform on these tasks?

SOURCE CODE

The source code for our work is available at `https://github.com/SaipraveenB/super-duper-octo-lamp`