These three papers are to highlight MBRL algorithms that have utilized latent-variable world models to encode low-dimensional latent states from high-dimensional inputs.

Dream to Control: <https://arxiv.org/abs/1912.01603>

Goal: Learn a model able to, with a given input observation, predict the next observation and future hidden states in order to train an effective policy.

To do so, we implement dynamics learning, for which we need to learn three models: representation, transition, and reward.

In collected experiences, you are given observations, actions taken, and reward received. We want to learn to represent each observation as a hidden state (encoding), the transition (the paper uses recurrent state space model – RSSM), and learn to predict the reward given a state.

Pseudocode for dynamics learning:

While not converged:

for update step:

draw data sequences

compute model states (encode for hidden states)

update parameter theta using “representation learning”

In Dreamer, the model learns to reconstruct the same state as the input observation. If you can learn a model that can accurately reconstruct the observation, then your representation is an informative one.

You now have a model able to represent an observation in a hidden state (or latent state), predict the reward, and learn the transition from one state to another, which will be helpful in imagination. This model is also trained with the ELBO loss function that helps keep in latent representation only the information that is needed.

Next is behavior learning, where we use the RSSM model to imagine future trajectories of hidden states. We are given an observation, and using our policy, determine an action, then, using our dynamics model, predict the probable next hidden state, value, and reward. We repeat with backpropagation to learn an effective policy.

Background: What is latent space? Latent space is an encoding of raw data into a meaningful internal representation. This allows us to compress data from a large number of inputs into a low-dimensional latent space, which can then be used for tasks. In this particular paper, a latent dynamics model is implemented to predict “latent trajectories”, which are generated samples through which an agent can learn behavior from in order to learn action and value models. In the paper, a latent state refers to the representation of the environment (in a way summarizes the agent’s observation).

The action model represents the agent’s policy–it predicts actions to optimize the cumulative reward of trajectories generated by the world model. The value model is used to estimate the cumulative reward of a latent state in the trajectories. The value model works with the action model – providing the action model with how valuable a state is.

Intelligent agents can utilize a world model, which represents an agent’s knowledge about its environment (its world) in a model that can make predictions about the future. Behavior is learned by predicting hypothetical trajectories in the compact latent space of the world model. The paper proposes an agent called Dreamer that 1) learns dynamics from experience by collecting data sequences and “learns to encode observations and actions into compact latent states and predicts environment rewards”, 2) maximizes future value predictions by propagating gradients back through imagined trajectories, and 3) encodes history of episode to compute current model state and predict next action to execute.

Dreamer also uses an actor critic approach to learn behaviors past the imagination horizon.

Learning Latent Dynamics: Latent imagination required a representation model (to encode observations and actions into model states), transition model (predicts future model states), and reward model (predicts rewards given model state). Goal: maximize expected imagined rewards with respect to the policy.

Latent imagination can be done by simply learning to predict future rewards given actions and past observations, but with a finite dataset, “learning about observations that correlate with rewards is likely to improve the world model”.

Dreamer 1) learns latent dynamics model from experiences, 2) learns action and value models from imagined latent trajectories and 3) executes actions in the environment to collect new experiences.

Mastering Atari with Discrete World Models: <https://arxiv.org/abs/2010.02193>

DreamerV2 – Sequel to Dreamer, with modifications such as “discrete latents” and “balancing terms within the KL loss”.

Same process as Dreamer algorithm – learn the world model from a dataset of past experiences, learn an actor and critic model with imagined trajectories of model states, and execute actions in environment to grow the experience dataset.

The world model includes an image encode, and a RSSM (from a previous paper of the same authors), and predictors for the image, reward, and discount factor.

Similar ot Dreamer V1, there is an actor and critic model for learning behaviors. The actor chooses actions for predicting imagined sequences of model states. The critic “accumulates future predicted rewards to take into account rewards beyond planning horizon”.

Deep RL with Latent Variable Model: <https://arxiv.org/abs/1907.00953>

Overview: Proposed algorithm learns a compact latent representation of environment to then perform RL.

An obstacle in RL is that a considerable portion of learning is spent acquiring good representations of the observation space. This approach explicitly learns a latent variable model of the MDP.

Blog Post: