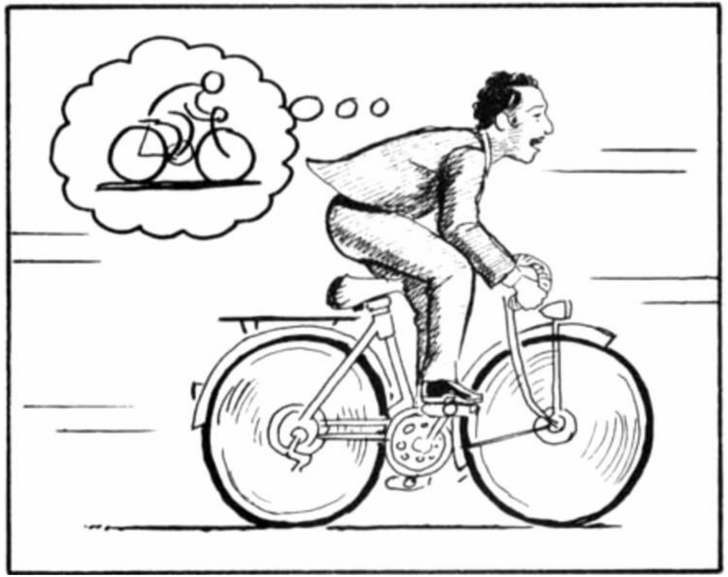
DreamerV2: Learning Behaviours by Latent Imagination Agents on MinAtar

Reinforcement Learning
Sapienza University of Rome

Introduction - World Models

- Humans develop a **mental model** of the world based on their **perception** with their **limited senses**
- **Decisions** are based on this internal model
- **Artificial agents** can benefit from having a good **representation of past and present states**, and a good predictive model of the future



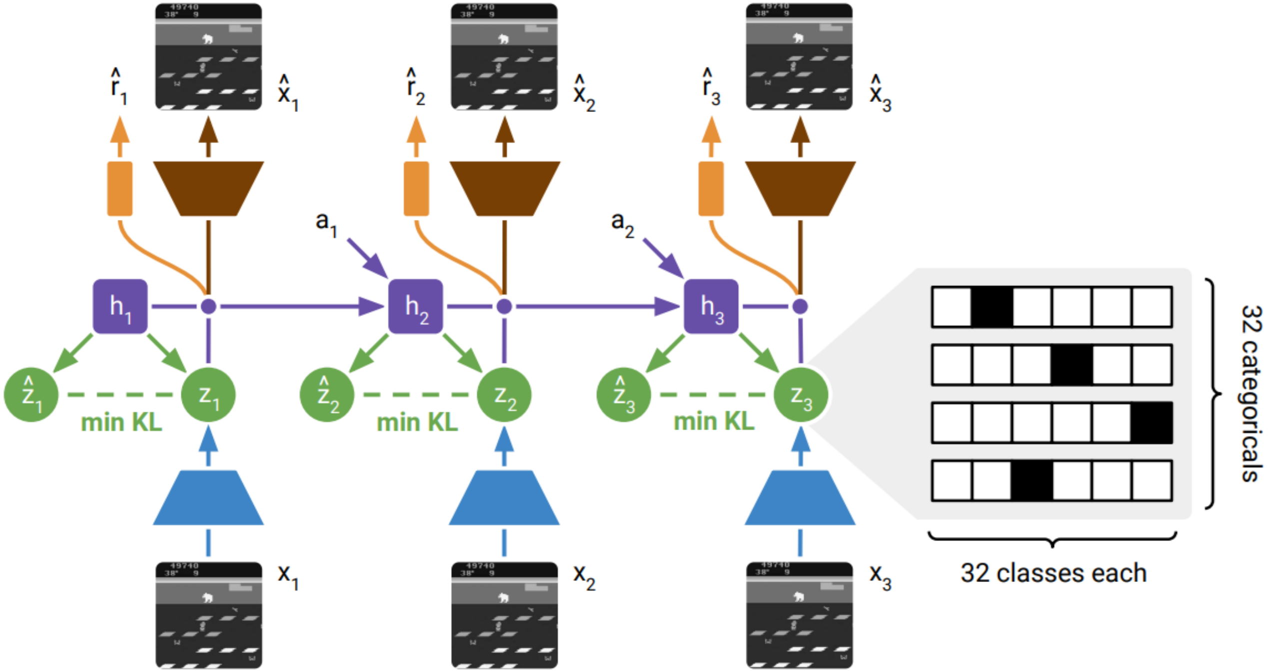
Introduction - World Models

- World models **facilitate generalization** from past experience and allow **learning behaviors from imagined outcomes** to increase sample-efficiency
- Learning successful behaviors purely within the world model demonstrates that the **world model learns to accurately represent the environment**
- Reinforcement Learning with World Models

Introduction - DreamerV2

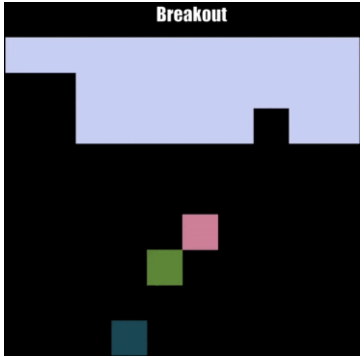
- **Model-based agent**
- Learns behaviors purely from the **latent-space predictions** of a separately trained world model
- Achieves **human-level performance** on the Atari 200M benchmark
- Shows that model-based RL can outperform top model-free algorithms on the most competitive RL benchmarks

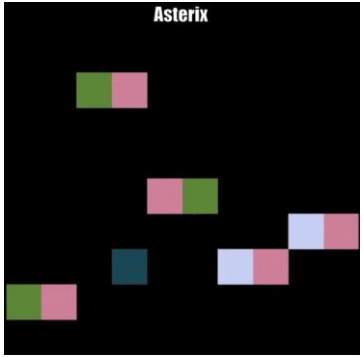
Learning The World Model



Learning The World Model - MinAtar

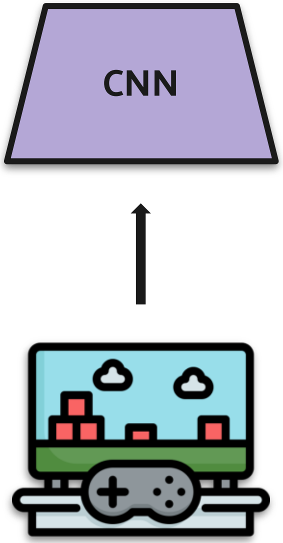
- MinAtar implements **miniaturized and simplified** versions of several Atari 2600 games
- Simplifies the games
- Experimentation with the environments more accessible and **efficient**
- **10x10xn state representation**





Learning The World Model - Image Encoder

- The 84x84 grayscale image is **downscaled** to 64x64 pixels so that convolution can be applied (ATARI)
- The image encoder is implemented as a Convolutional Neural Network (**CNN**)
- It extracts features from the image and encodes them into a useful embedding



Learning The World Model - RSSM

$$\text{RSSM} \begin{cases} \text{Recurrent model:} & h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) \\ \text{Representation model:} & z_t \sim q_\phi(z_t \mid h_t, x_t) \\ \text{Transition predictor:} & \hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t) \end{cases}$$

- It uses a **GRU** to compute the deterministic recurrent states
- Deterministic recurrent state $\mathbf{h_t}$
- Posterior state $\mathbf{z_t}$ incorporates information about the current image $\mathbf{x_t}$

Learning The World Model - RSSM

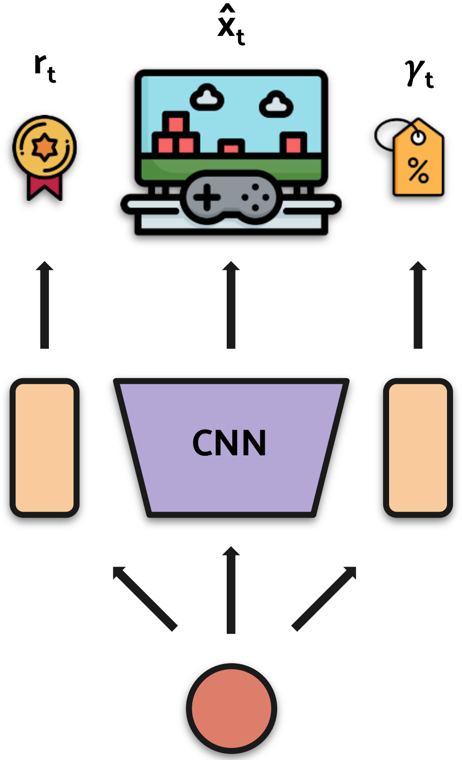
$$\text{RSSM} \begin{cases} \text{Recurrent model:} & h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}) \\ \text{Representation model:} & z_t \sim q_\phi(z_t \mid h_t, x_t) \\ \text{Transition predictor:} & \hat{z}_t \sim p_\phi(\hat{z}_t \mid h_t) \end{cases}$$

- Prior state $\hat{\mathbf{z}}_t$ aims to predict the posterior without access to the current image
- The **concatenation of deterministic and stochastic states** forms the compact model state
- The representation model and the transition predictor are **MLPs**
- The world model can be interpreted as a **sequential VAE**

Learning The World Model - Image Predictor

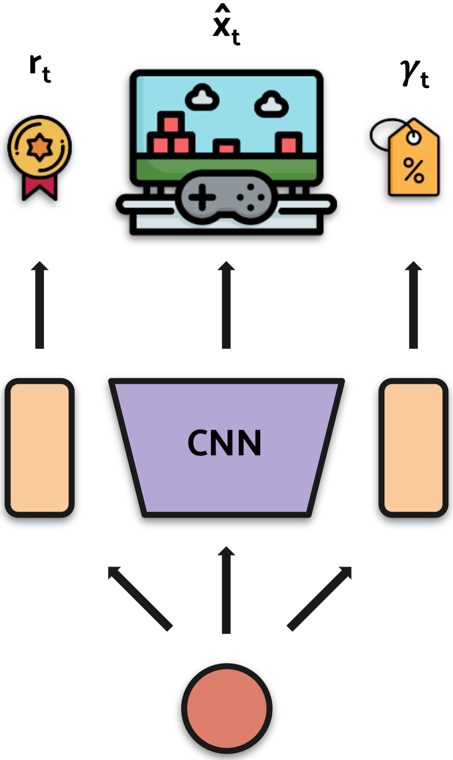
Image predictor: $\hat{x}_t \sim p_\phi(\hat{x}_t \mid h_t, z_t)$

- From the compact RSSM state, the image predictor **predicts the current image**
- The image predictor is a Deconvolutional Neural Network (**Transposed CNN**)



Learning The World Model - Reward Predictor

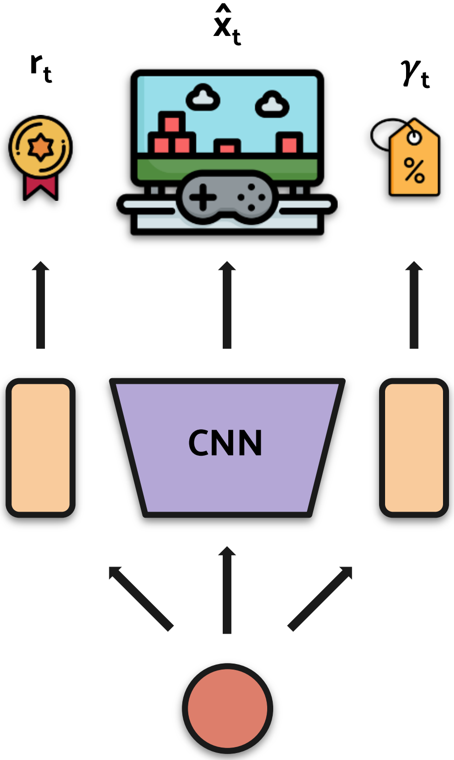
Reward predictor: $\hat{r}_t \sim p_\phi(\hat{r}_t \mid h_t, z_t)$



- The reward predictor allows to predict the **reward** given only the posterior stochastic state z_t and the deterministic state h_t
- The reward predictor is a **MLP**

Learning The World Model - Discount Predictor

Discount predictor: $\hat{\gamma}_t \sim p_\phi(\hat{\gamma}_t \mid h_t, z_t)$



- The **discount** predictor allows to estimate the discount value of the reward
- The discount predictor is a **MLP**

Learning The World Model - Loss Function

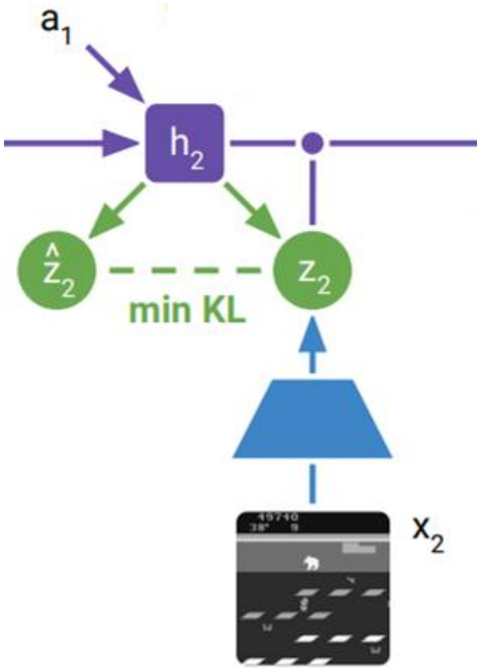
- World model's components are **optimized jointly**
- The distributions produced by the predictors are trained to maximize the **log-likelihood** of their corresponding targets:

$$- \ln p_\phi(\cdot \mid h_t, z_t) \qquad x_t, r_t, \gamma_t$$

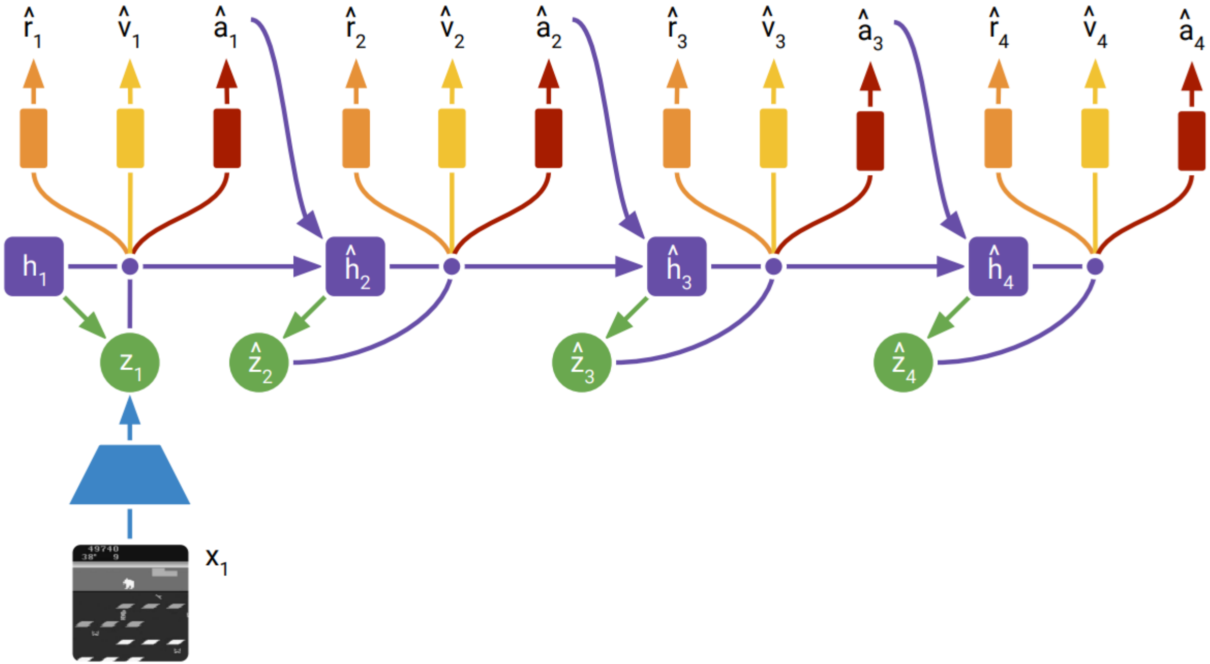
- **KL loss** is a key component of the model's loss

Learning The World Model - KL Loss

- It **trains the prior** toward the representations
- It **regularizes** how much information the **posterior** incorporates from the image
- **Increases robustness** to novel inputs
- Encourages reusing existing **information from past steps**, thus learning long-term dependencies
- KL loss is minimized faster with respect to the prior than the representations by using different learning rates (**KL Balancing**)

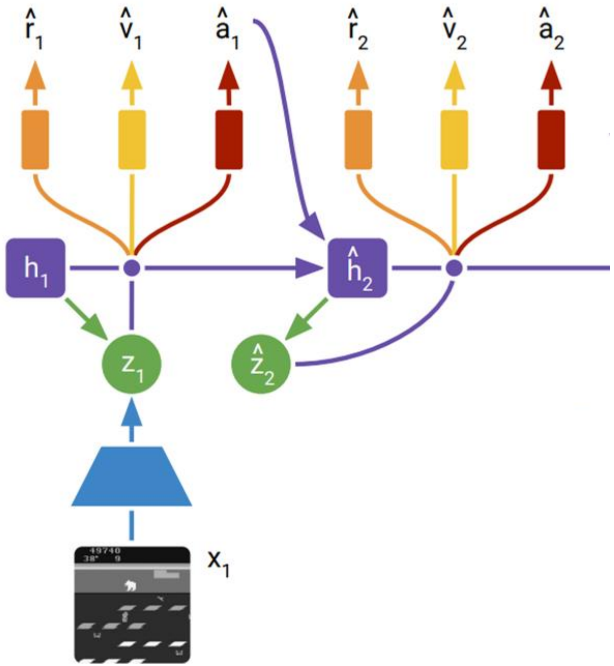


Behaviour Learning



Behaviour Learning - Imagination

- The transition predictor outputs sequences of compact model states $\hat{z}_t$ up to the **imagination horizon**
- The reward predictor predicts the rewards for each state
- The discount predictor outputs the **discount sequence**, used to down-weight the rewards



Behaviour Learning - Actor Critic

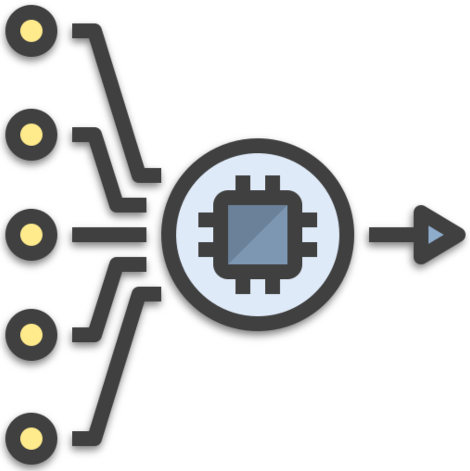
- DreamerV2 learns **long-horizon** behaviors purely within its world model using an **actor** and a **critic**
- Both the **actor and critic operate on top of the learned model states**
- The **world model is fixed** during behavior learning, so the actor and critic gradients do not affect its representations

Behaviour Learning - Actor Critic

- The actor and critic are **trained cooperatively**
- The actor predicts an **action based on the prior stochastic state**, concatenated to the deterministic state
- The critic estimates the **sum of future rewards** achieved by the actor from each imagined state
- The latent state sequence is **Markovian**

Behaviour Learning - Actor Critic

- The actor outputs a **categorical distribution** over actions and the critic has a deterministic output
- The actor and critic are both **MLP**s with ELU activations
- The two components are trained from the same **imagined trajectories** but optimize **separate loss functions**

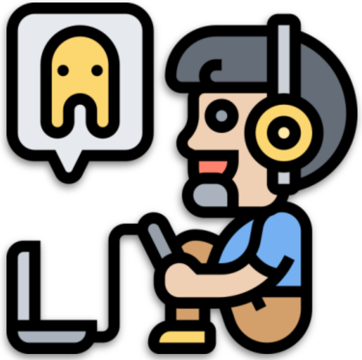


Behaviour Learning - Critic loss

- The critic aims to **predict the discounted sum of future rewards** (state value) that the actor achieves in a given model state
- **Temporal-difference learning** is used, the critic is trained towards a value target that is constructed from intermediate rewards and critic outputs for later states
- The value target is the λ-target, a weighted average of **n-step returns** for different horizons where longer ones are weighted exponentially less
- The value learning is stabilized using a **target network**, the targets are computed using a copy of the critic that is updated every 100 gradient steps

Behaviour Learning - Actor loss

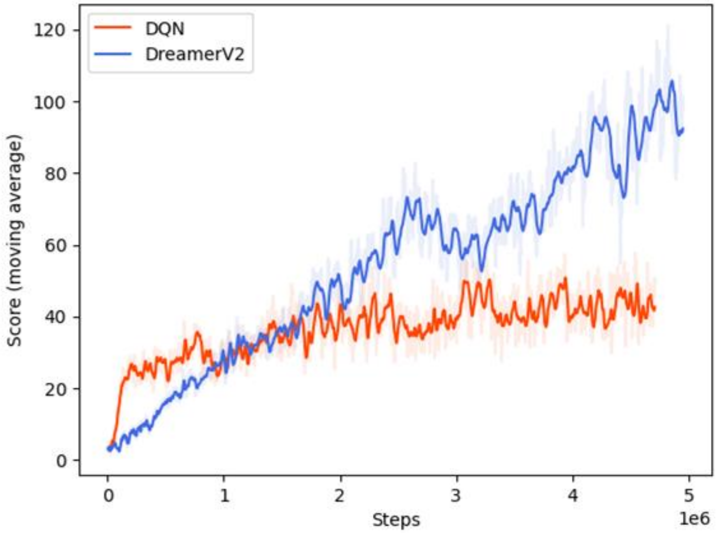
- The actor aims to output actions that **maximize the prediction of long-term future rewards made by the critic**
- DreamerV2 combines unbiased but high-variance **Reinforce gradients** with biased but low-variance **straight-through gradients**
- **Entropy** of the actor is **regularized** to encourage exploration where feasible, while allowing the actor to choose precise actions when necessary

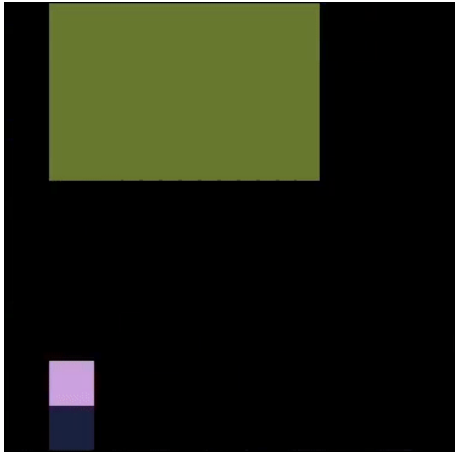


Results & Simulations

- We compared our model to **DQN**
- We tested out our model **without** some of its core features:
  - **KL balancing**
  - **Categorical** latent variables
- We tested our model using **action repeat**

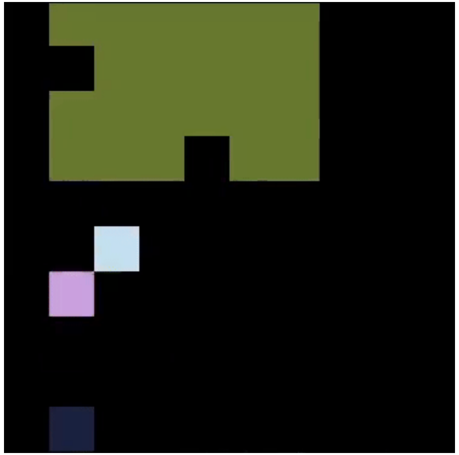
Results & Simulations - Space Invaders



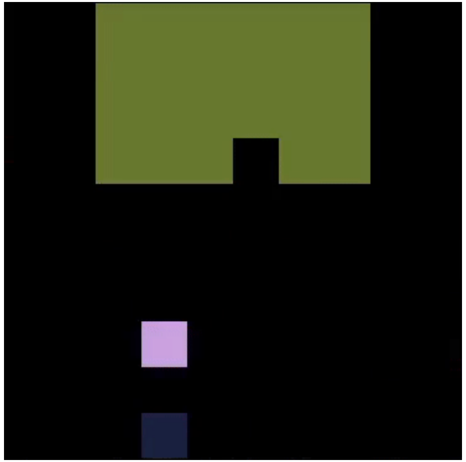


Our implementation

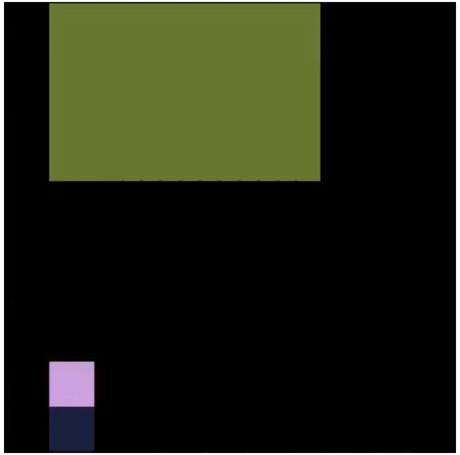
Results & Simulations - Space Invaders



1 MLN
- Poor predictive ability
- Poor consistency

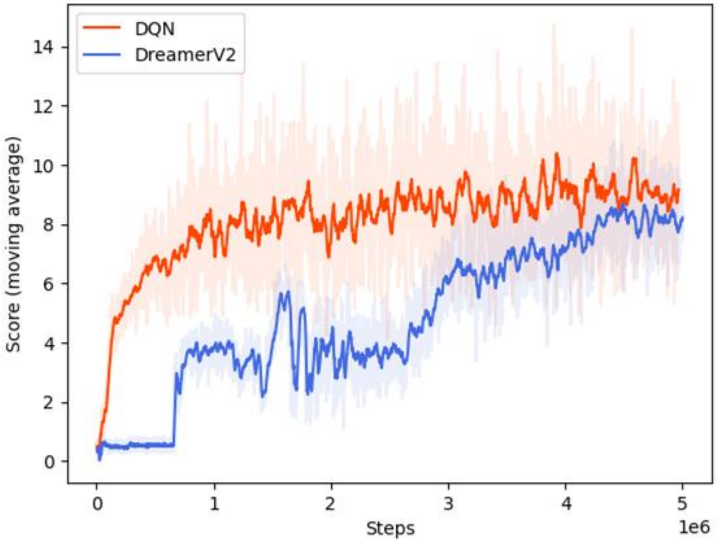


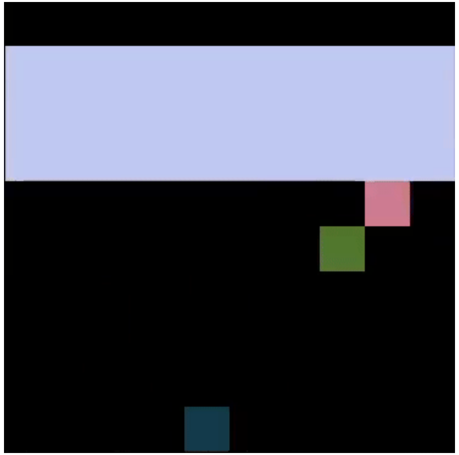
3 MLN
- Good predictive ability
- Poor consistency



5 MLN
- Good predictive ability
- Good consistency

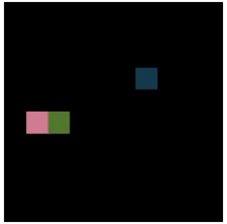
Results & Simulations - Breakout

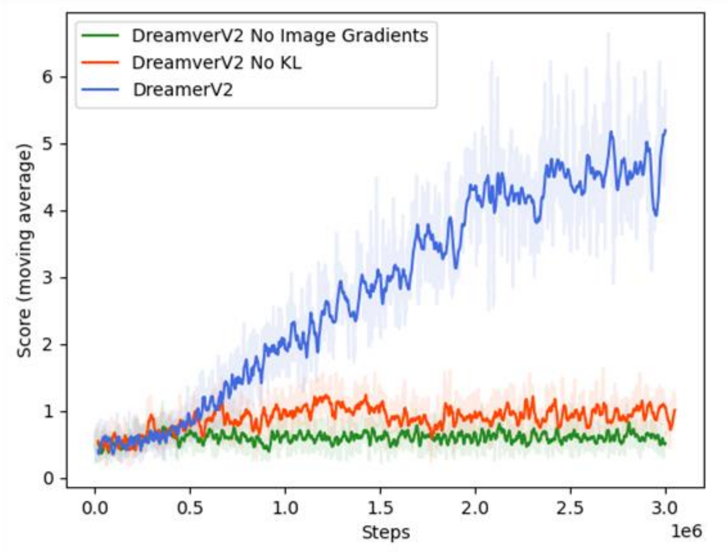


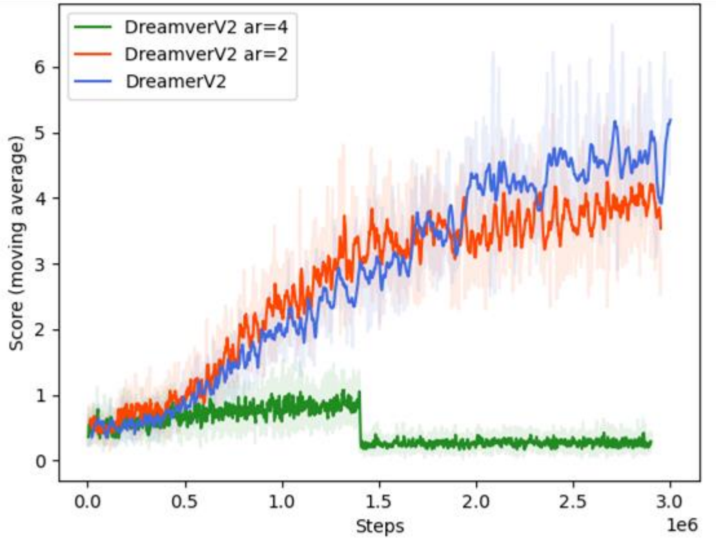


Our implementation

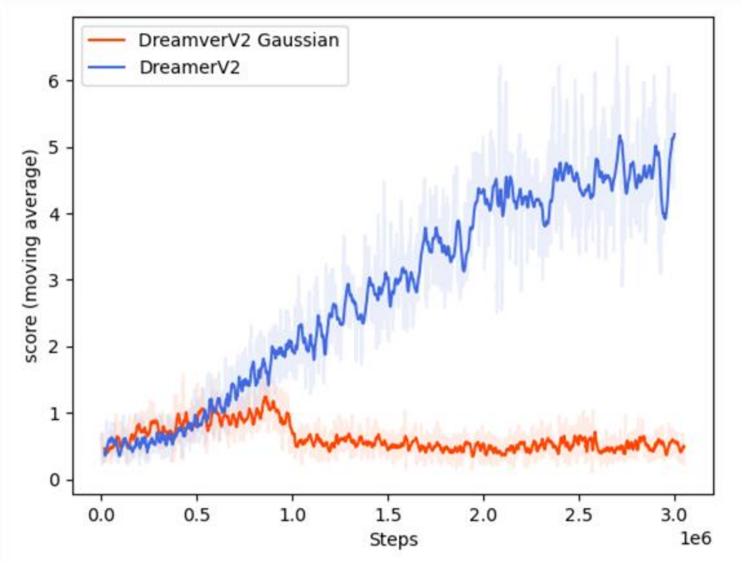
Results & Simulations - Ablations

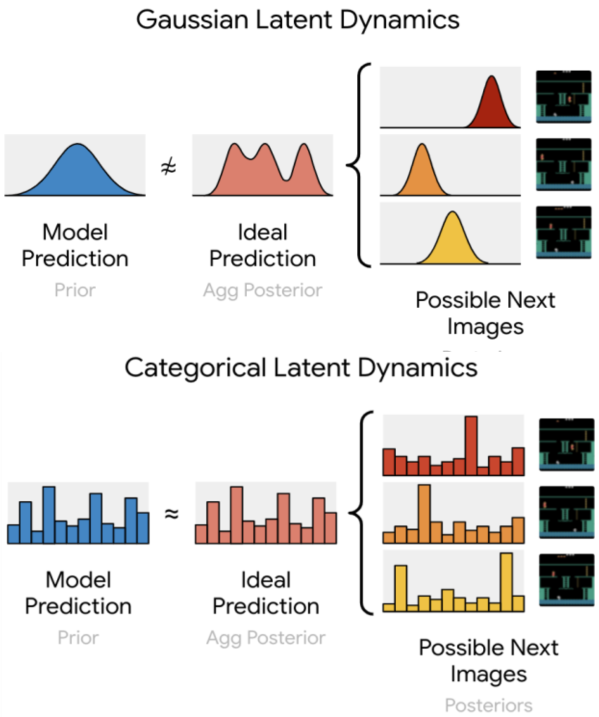






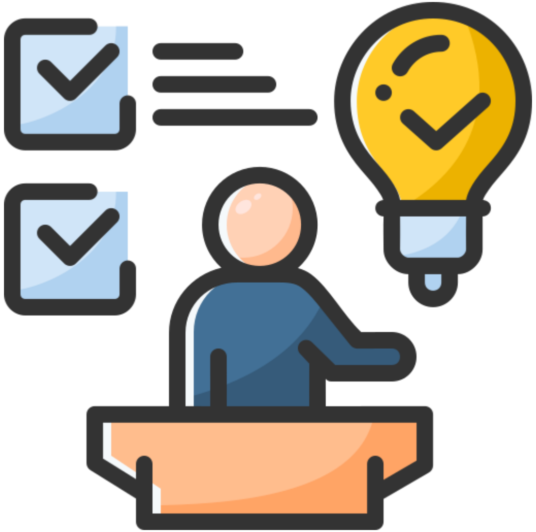
Results & Simulations - Categorical vs Gaussians





Conclusions

- Experiments shows that learning a **categorical latent space** and using **KL balancing** improves the performance of the agent.
- **Image information** is crucial for learning generally useful representations
- Huge number of parameters, so **long training time**
- Can outperform model free methods in many games



Thank you for the attention!