Dream to Control: Learning Behaviors by Latent Imagination

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, Mohammad Norouzi

Topic: Model Based RL

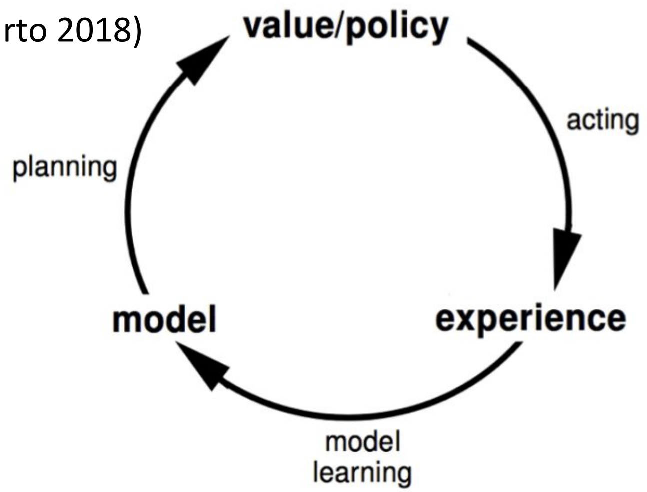
Presenter: Haotian Cui

Motivation and Main Problem

1-4 slides

Should capture

- High level description of problem being solved (can use videos, images, etc)
- Why is that problem important?
- Why is that problem hard?
- High level idea of why prior work didn’t already solve this (Short description, later will go into details)

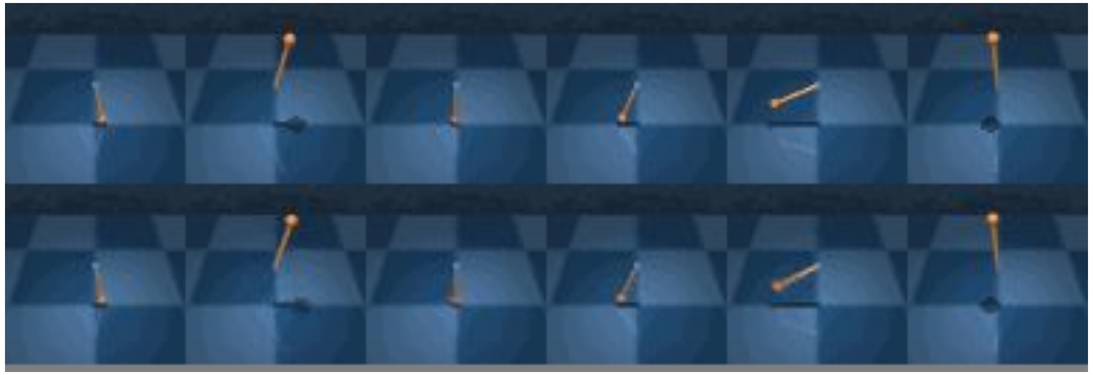


(Credit:Sutton & Barto 2018)

What is model based RL?

- "Model" often refers to world models, which capture the state transitions. A model M_i includes $[S, A, P_i, R_i]$
- Benefits of world models:
  - it can be more data efficient by leveraging a richer training signal.
  - has the potential to transfer to other tasks given the same env.
- Challenges of model-based RL:
  - model bias -> error compounding (model error + policy error)
- Dyna(Sutton 1990) - model rollout trajectories + real trajectories
- When to trust your model (arxiv.org/abs/1906.08253) - short model-generated rollouts branched from real data
- PlaNet(Hafner et al., 2018) - latent space planning enables fast planning

Motivation



Video here: https://dreamrl.github.io/

- Model is essential for:
  - Intelligent agents can achieve goals in complex environments even though they never encounter the exact same situation twice.
  - A parametric model can make predictions about future.
- latent world model is particularly:
  - Fast and small memory footprint
  - Able to imagine thousands of trajectories in parallel
- Operational problem - difficulty in building latent dynamic models:
  - Hard to find analytic gradients – existing works used derivative-free optimizations
  - Need accurate trajectory prediction

Contributions – so how does this work build a world model instead?

- **Analytic gradients**: propagating analytic value gradients back through the latent dynamics using reparameterization.
- **Learning long-horizon behaviors by latent imagination** is achieved by (1) predicting both actions and state values, (2) training purely by imagination in a latent - efficiently learn the policy. (Squeeze the algorithm to learn well in latent space)
- **Empirical performance for visual control**: Dreamer exceeds previous agents in terms of data-efficiency, computation time, and final performance.

Related works

- Control with latent dynamics:
  - E2C (Watter et al., 2015) and RCE (Banijamali et al., 2017), PlaNet (Hafner et al., 2019)
- Imagined Multi -step returns:
  - VPN (Oh et al., 2017), MVE (Feinberg et al., 2018), and STEVE (Buckman et al., 2018)
- Analytic value gradients:
  - DPG (Silver et al., 2014), DDPG (Lillicrap et al., 2015), and SAC (Haarnoja et al., 2018)

Approach / Algorithm / Methods (if relevant)

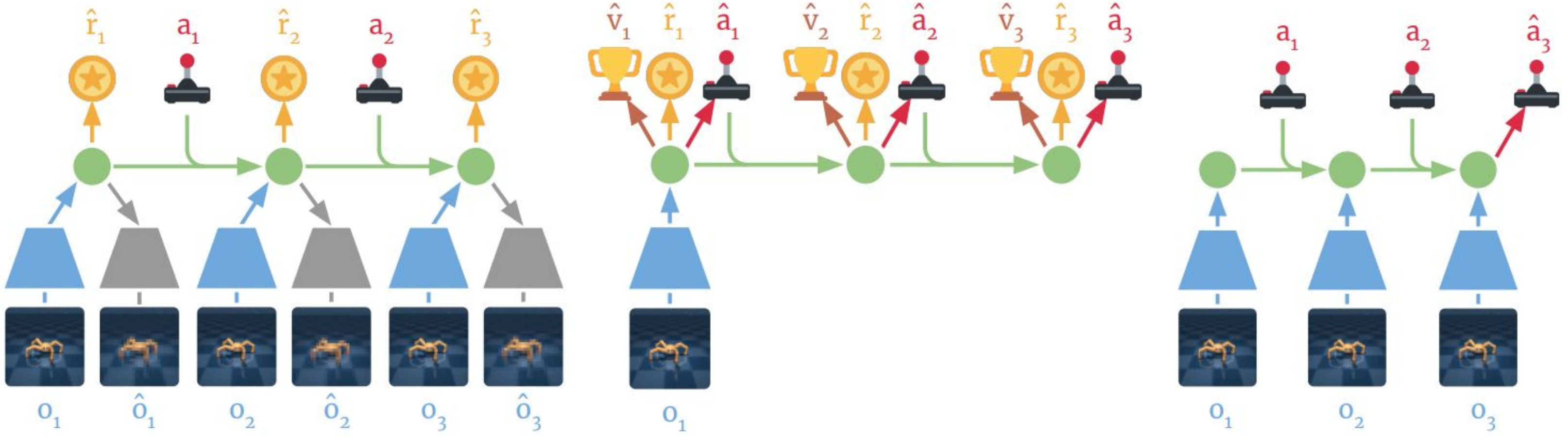
Likely >1 slide

Describe algorithm or framework (pseudocode and flowcharts can help)

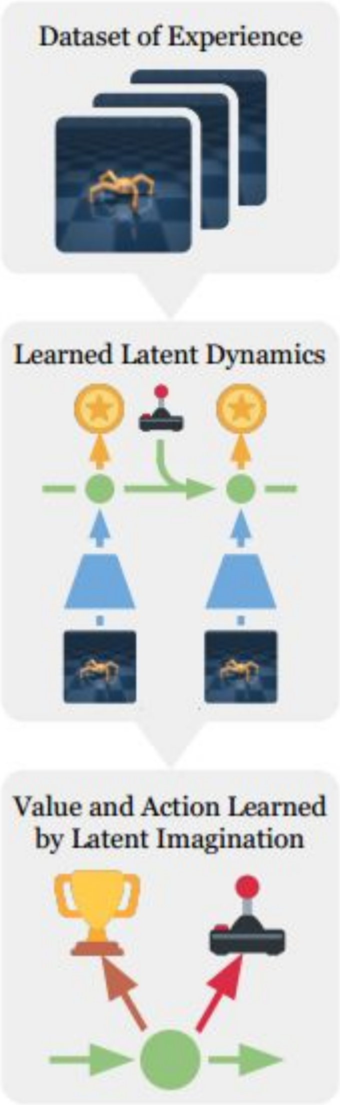
What is it trying to optimize?

Implementation details should be left out here, but may be discussed later if its relevant for limitations / experiments

Method –overview



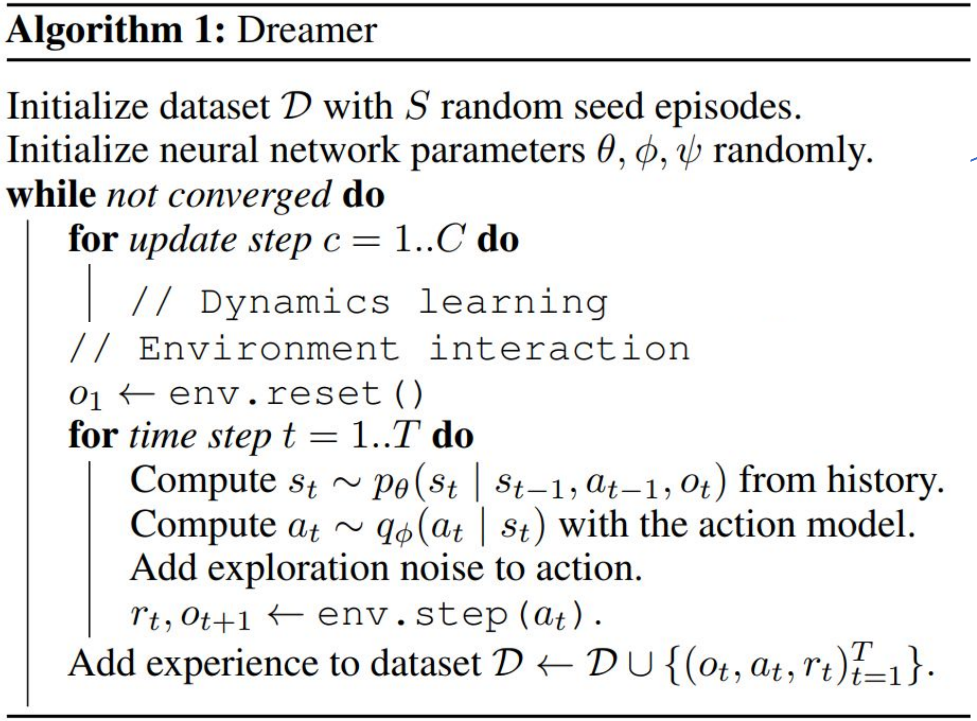
(a) Learn dynamics from experience (b) Learn behavior in imagination (c) Act in the environment



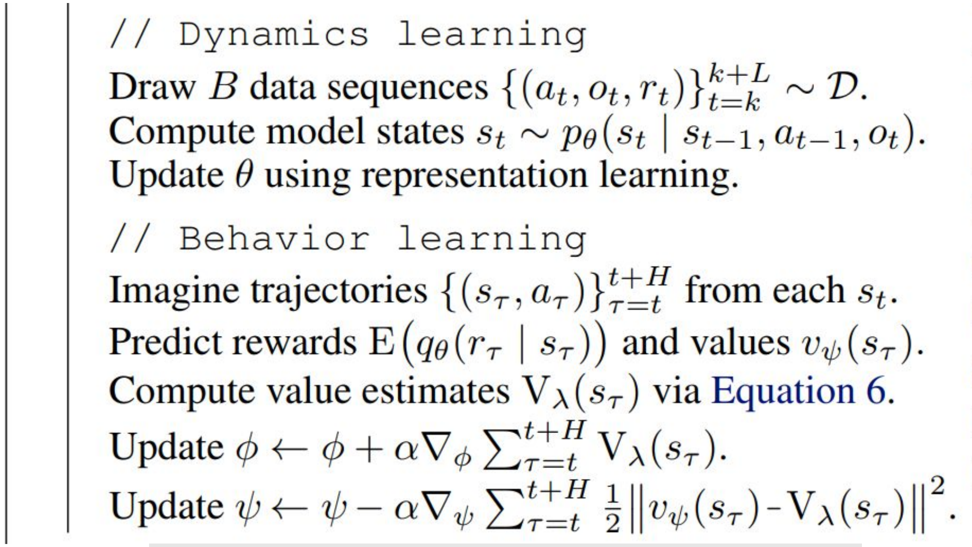
(a)From the dataset of past experience, the agent learns to encode observations and actions into compact latent states. (b) In the compact latent space, Dreamer predicts state values (c) The agent encodes the history of the episode to compute the current model state and predict the next action to execute in the environment

$$
\begin{aligned}
&\text{Representation model:} && p(s_t \mid s_{t-1}, a_{t-1}, o_t) \\
&\text{Transition model:} && q(s_t \mid s_{t-1}, a_{t-1}) \\
&\text{Reward model:} && q(r_t \mid s_t).
\end{aligned}
$$

Method – algorithm 1

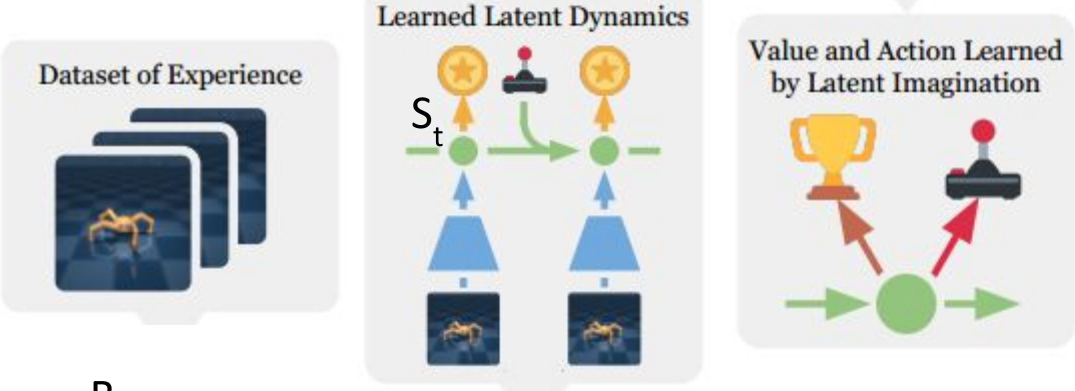


```
Algorithm 1: Dreamer

Initialize dataset $\mathcal{D}$ with $S$ random seed episodes.
Initialize neural network parameters $\theta, \phi, \psi$ randomly.
while not converged do
    for update step $c = 1..C$ do
        // Dynamics learning
    // Environment interaction
    $o_1 \leftarrow$ env.reset()
    for time step $t = 1..T$ do
        Compute $s_t \sim p_\theta(s_t \mid s_{t-1}, a_{t-1}, o_t)$ from history.
        Compute $a_t \sim q_\phi(a_t \mid s_t)$ with the action model.
        Add exploration noise to action.
        $r_t, o_{t+1} \leftarrow$ env.step($a_t$).
    Add experience to dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_t, a_t, r_t)_{t=1}^T\}$.
```



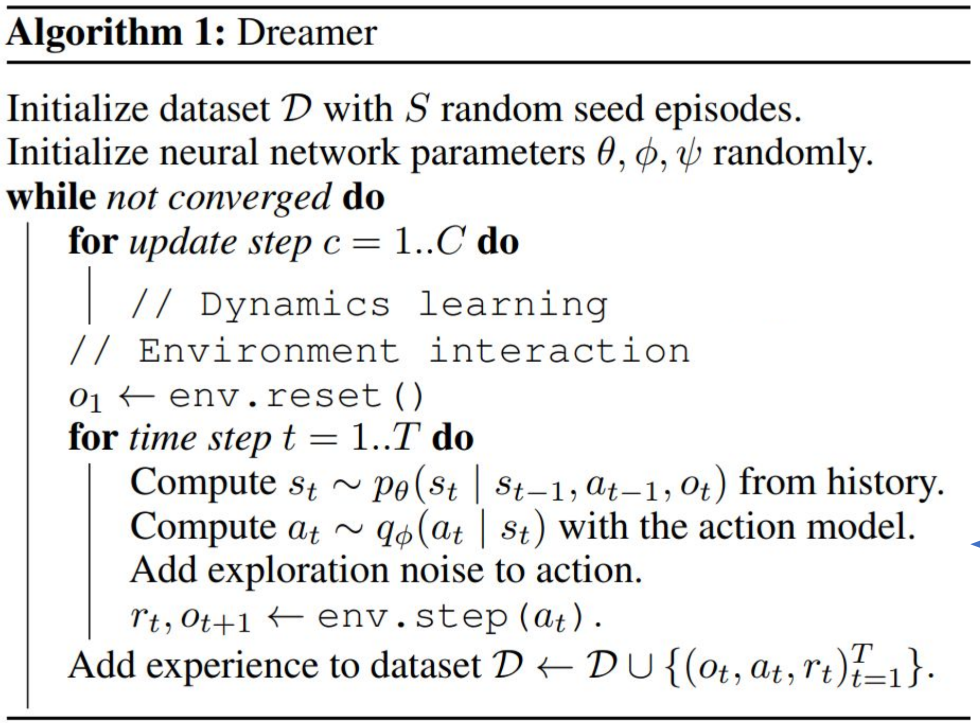
```
// Dynamics learning
Draw $B$ data sequences $\{(a_t, o_t, r_t)\}_{t=k}^{k+L} \sim \mathcal{D}$.
Compute model states $s_t \sim p_\theta(s_t \mid s_{t-1}, a_{t-1}, o_t)$.
Update $\theta$ using representation learning.

// Behavior learning
Imagine trajectories $\{(s_\tau, a_\tau)\}_{\tau=t}^{t+H}$ from each $s_t$.
Predict rewards $\mathrm{E}\big(q_\theta(r_\tau \mid s_\tau)\big)$ and values $v_\psi(s_\tau)$.
Compute value estimates $\mathrm{V}_\lambda(s_\tau)$ via Equation 6.
Update $\phi \leftarrow \phi + \alpha \nabla_\phi \sum_{\tau=t}^{t+H} \mathrm{V}_\lambda(s_\tau)$.
Update $\psi \leftarrow \psi - \alpha \nabla_\psi \sum_{\tau=t}^{t+H} \frac{1}{2} \big\| v_\psi(s_\tau) - \mathrm{V}_\lambda(s_\tau) \big\|^2$.
```

Actor-Critic in the imagined world



B

Method – algorithm 1



Algorithm 1: Dreamer

```
Initialize dataset $\mathcal{D}$ with $S$ random seed episodes.
Initialize neural network parameters $\theta, \phi, \psi$ randomly.
while not converged do
    for update step $c = 1..C$ do
        // Dynamics learning
    // Environment interaction
    $o_1 \leftarrow$ env.reset()
    for time step $t = 1..T$ do
        Compute $s_t \sim p_\theta(s_t \mid s_{t-1}, a_{t-1}, o_t)$ from history.
        Compute $a_t \sim q_\phi(a_t \mid s_t)$ with the action model.
        Add exploration noise to action.
        $r_t, o_{t+1} \leftarrow$ env.step($a_t$).
    Add experience to dataset $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_t, a_t, r_t)_{t=1}^{T}\}$.
```

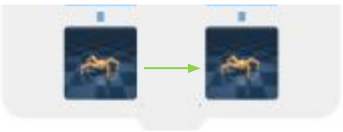
In the real world:

- Blind execution with no learning

A comparison to other (model-based) RL

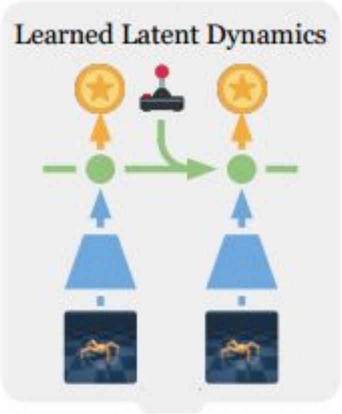
- Dreamer moves the transition arrow – the world model transition, upward to the latent space.

From



p$(s_t \mid s_{t-1}, a_{t-1})$

To



$q(s_t \mid s_{t-1}, a_{t-1})$

- Terminology analogy

Terminology	Usually	Dreamer

Action and value models

The action and value models are trained cooperatively as typical in policy iteration:

- the action model aims to maximize an estimate of the value,
- the value model aims to match an estimate of the value that changes as the action model change
- use reparameterization for continuous actions and latent states and straight-through gradients (Bengio et al., 2013)

$$\text{Action model:} \quad a_\tau \sim \; q_\phi(a_\tau \mid s_\tau) \qquad a_\tau = \tanh\big(\mu_\phi(s_\tau) + \sigma_\phi(s_\tau)\,\epsilon\big), \quad \epsilon \sim \mathrm{Normal}(0, \mathbb{I})$$

$$\text{Value model:} \quad v_\psi(s_\tau) \approx \mathrm{E}_{q(\cdot|s_\tau)}\big(\textstyle\sum_{\tau=t}^{t+H} \gamma^{\tau-t} r_\tau\big).$$

Objectives

$$\mathrm{E}_{q_\theta, q_\phi}\Big(\sum_{\tau=t}^{t+H} \mathrm{V}_\lambda(s_\tau)\Big)$$

$$\min_\psi \mathrm{E}_{q_\theta, q_\phi}\Big(\sum_{\tau=t}^{t+H} \frac{1}{2}\Big\| v_\psi(s_\tau) - \mathrm{V}_\lambda(s_\tau))\Big\|^2\Big)$$

- Choice of value model:

$$\mathrm{V}_\mathrm{R}(s_\tau) \doteq \mathrm{E}_{q_\theta, q_\phi}\Big(\sum_{n=\tau}^{t+H} r_n\Big),$$

V_R simply sums the rewards from τ until the horizon

$$\mathrm{V}_\mathrm{N}^k(s_\tau) \doteq \mathrm{E}_{q_\theta, q_\phi}\Big(\sum_{n=\tau}^{h-1} \gamma^{n-\tau} r_n + \gamma^{h-\tau} v_\psi(s_h)\Big) \quad \text{with} \quad h = \min(\tau + k, t + H),$$

V_N uses k-step look ahead

$$\mathrm{V}_\lambda(s_\tau) \doteq (1 - \lambda) \sum_{n=1}^{H-1} \lambda^{n-1} \mathrm{V}_\mathrm{N}^n(s_\tau) + \lambda^{H-1} \mathrm{V}_\mathrm{N}^H(s_\tau),$$

V_λ exponentially-weighted average of the estimates for different k to balance bias and variance.

Action and value models

Dreamer uses V_λ

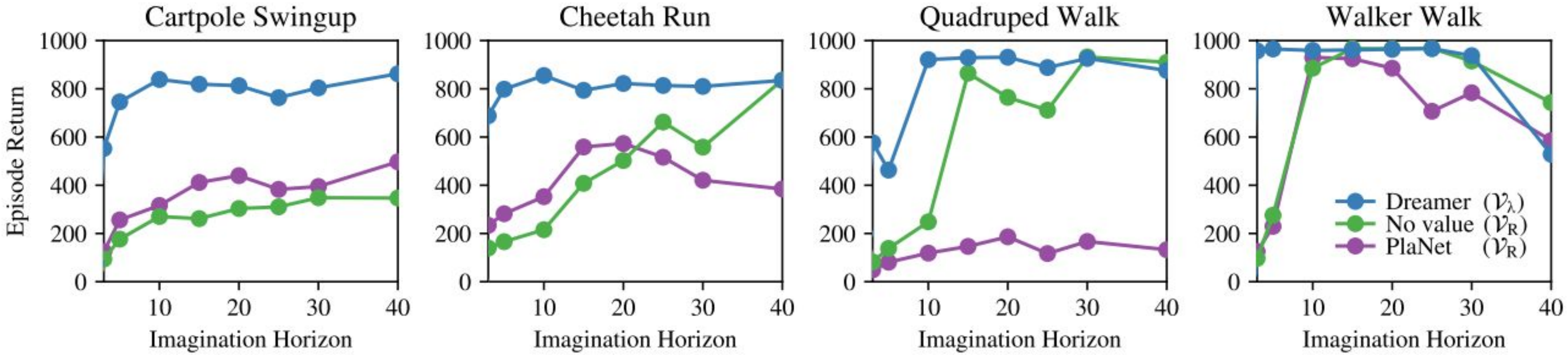


Figure 4: Imagination horizons. We compare the final performance of Dreamer, learning an action model without value prediction, and online planning using PlaNet. Learning a state value model to estimate rewards beyond the imagination horizon makes Dreamer more robust to the horizon length. The agents use pixel reconstruction for representation learning and an action repeat of $R = 2$.

LEARNING LATENT DYNAMICS

- Reward prediction
  - match the reward prediction to the real outcomes.

- Reconstruction   Increase the variational lower bound (ELBO; Jordan et al., 1999)

$$\mathcal{J}_{\mathrm{REC}} \doteq \mathrm{E}_p\Big(\sum_t \big(\mathcal{J}_{\mathrm{O}}^t + \mathcal{J}_{\mathrm{R}}^t + \mathcal{J}_{\mathrm{D}}^t\big)\Big) + \mathrm{const} \qquad \mathcal{J}_{\mathrm{O}}^t \doteq \ln q(o_t \mid s_t)$$

$$\mathcal{J}_{\mathrm{R}}^t \doteq \ln q(r_t \mid s_t) \qquad \mathcal{J}_{\mathrm{D}}^t \doteq -\beta\,\mathrm{KL}\big(p(s_t \mid s_{t-1}, a_{t-1}, o_t) \,\big\|\, q(s_t \mid s_{t-1}, a_{t-1})\big)$$

- Contrastive estimation

$$\mathcal{J}_{\mathrm{NCE}} \doteq \mathrm{E}\Big(\sum_t \big(\mathcal{J}_{\mathrm{S}}^t + \mathcal{J}_{\mathrm{R}}^t + \mathcal{J}_{\mathrm{D}}^t\big)\Big) \qquad \mathcal{J}_{\mathrm{S}}^t \doteq \ln q(s_t \mid o_t) - \ln\Big(\sum_{o'} q(s_t \mid o')\Big)$$

Reconstruction Objective

Derive from the information bottleneck (Tishby et al., 2000)

$$\max \mathrm{I}(s_{1:T};(o_{1:T},r_{1:T}) \mid a_{1:T}) - \beta\, \mathrm{I}(s_{1:T},i_{1:T} \mid a_{1:T})$$

$$\begin{aligned}
&\mathrm{I}(s_{1:T};(o_{1:T},r_{1:T}) \mid a_{1:T}) \\
=\ &\mathrm{E}_{p(o_{1:T},r_{1:T},s_{1:T},a_{1:T})}\Big(\sum_t \ln p(o_{1:T},r_{1:T} \mid s_{1:T},a_{1:T}) - \underbrace{\ln p(o_{1:T},r_{1:T} \mid a_{1:T})}_{\text{const}}\Big) \\
\overset{\pm}{=}\ &\mathrm{E}\Big(\sum_t \ln p(o_{1:T},r_{1:T} \mid s_{1:T},a_{1:T})\Big) \\
\geq\ &\mathrm{E}\Big(\sum_t \ln p(o_{1:T},r_{1:T} \mid s_{1:T},a_{1:T})\Big) - \mathrm{KL}\Big(p(o_{1:T},r_{1:T} \mid s_{1:T},a_{1:T}) \,\Big\|\, \prod_t q(o_t \mid s_t)q(r_t \mid s_t)\Big) \\
=\ &\mathrm{E}\Big(\sum_t \ln q(o_t \mid s_t) + \ln q(r_t \mid s_t)\Big).
\end{aligned}$$

$$\begin{aligned}
&\mathrm{I}(s_{1:T};i_{1:T} \mid a_{1:T}) \\
=\ &\mathrm{E}_{p(o_{1:T},r_{1:T},s_{1:T},a_{1:T},i_{1:T})}\Big(\sum_t \ln p(s_t \mid s_{t-1},a_{t-1},i_t) - \ln p(s_t \mid s_{t-1},a_{t-1})\Big) \\
=\ &\mathrm{E}\Big(\sum_t \ln p(s_t \mid s_{t-1},a_{t-1},o_t) - \ln p(s_t \mid s_{t-1},a_{t-1})\Big)
\end{aligned}$$

Non negativity of KL divergence

$$\begin{aligned}
\leq\ &\mathrm{E}\Big(\sum_t \ln p(s_t \mid s_{t-1},a_{t-1},o_t) - \ln q(s_t \mid s_{t-1},a_{t-1})\Big) \\
=\ &\mathrm{E}\Big(\sum_t \mathrm{KL}\big(p(s_t \mid s_{t-1},a_{t-1},o_t) \,\big\|\, q(s_t \mid s_{t-1},a_{t-1})\big)\Big).
\end{aligned}$$

Contrastive Objective

$$\mathcal{J}_{\mathrm{NCE}} \doteq \mathrm{E}\Big(\sum_t \big(\mathcal{J}_{\mathrm{S}}^t + \mathcal{J}_{\mathrm{R}}^t + \mathcal{J}_{\mathrm{D}}^t\big)\Big) \qquad \mathcal{J}_{\mathrm{S}}^t \doteq \ln q(s_t \mid o_t) - \ln\Big(\sum_{o'} q(s_t \mid o')\Big)$$

$$\begin{aligned}
& \mathrm{E}\big(\ln q(o_t \mid s_t) + \ln q(r_t \mid s_t)\big) \\
\overset{+}{=}\ & \mathrm{E}\big(\ln q(o_t \mid s_t) - \ln q(o_t) + \ln q(r_t \mid s_t)\big) \\
=\ & \mathrm{E}\big(\ln q(s_t \mid o_t) - \ln q(s_t) + \ln q(r_t \mid s_t)\big) \\
\geq\ & \mathrm{E}\Big(\ln q(s_t \mid o_t) - \ln \sum_{o'} q(s_t \mid o') + \ln q(r_t \mid s_t)\Big)
\end{aligned}$$

InfoNCE mini-batch bound (Poole et al., 2019)

Experiments setup

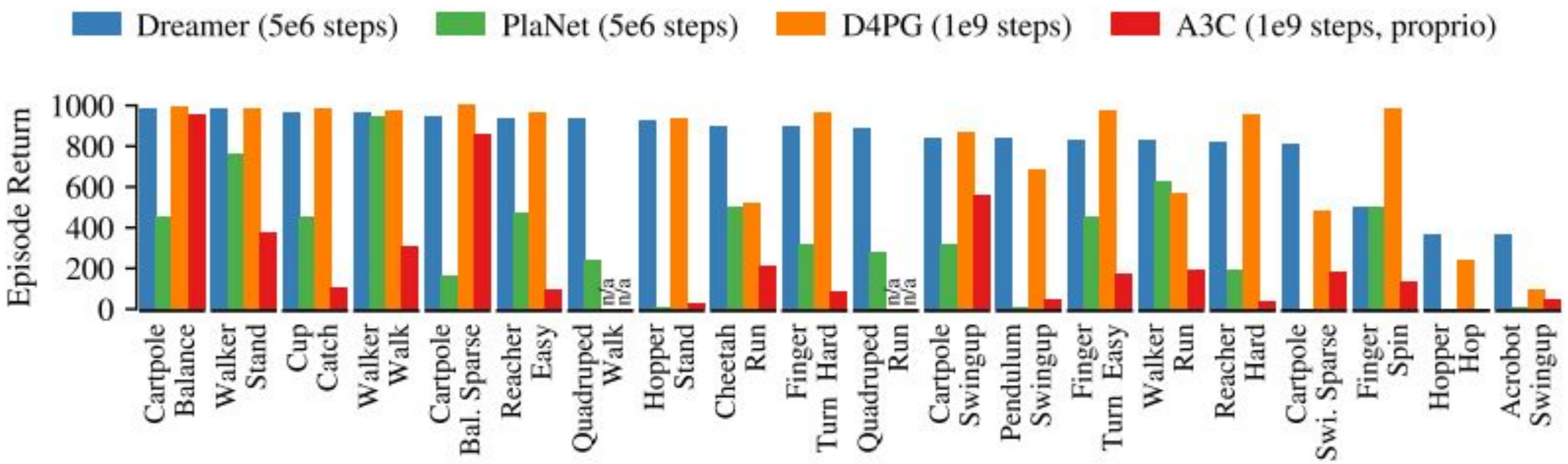
Evaluate Dreamer on 20 visual control tasks of the DeepMind Control Suite (Tassa et al., 2018)

- Agent observations are images of shape 64 × 64 × 3,
- actions range from 1 to 12 dimensions, rewards range from 0 to 1,
- episodes last for 1000 steps and have randomized initial states. Horizon 10 - 15.

Baseline:

- D4PG(Barth-Maron et al., 2018) - highest reported performance
- A3C (Mnih et al., 2016) , PlaNet (Hafner et al., 2018)

Results – performance comparison



Dreamer(average performance of 823) exceeds the performance of the strong model-free D4PG agent that achieves an average of 786 within 10^9 environment steps. At the same time, Dreamer inherits the data-efficiency of PlaNet, confirming that the learned world model can help to generalize from small amounts of experience.

Results – imagined trajectories

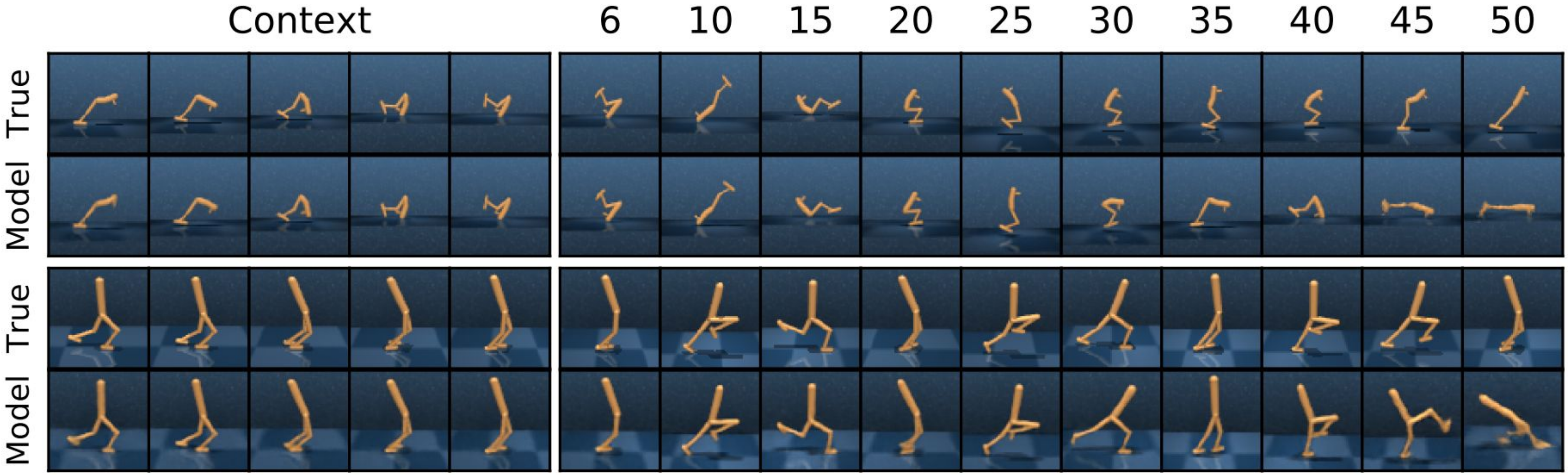
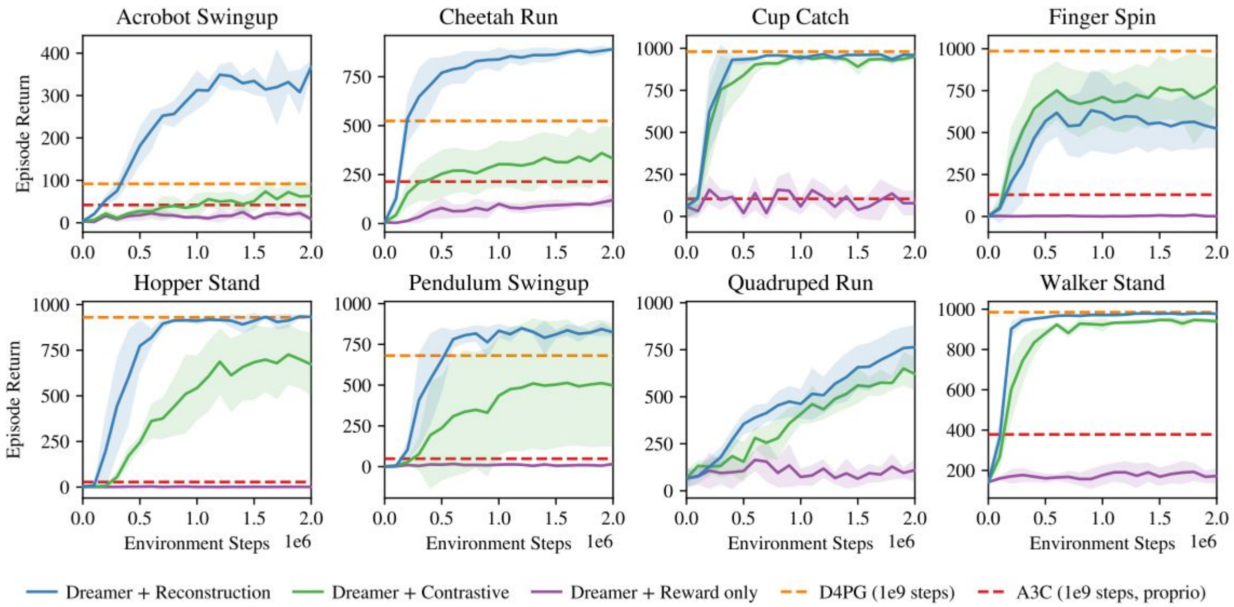


Figure 5: Reconstructions of long-term predictions. We apply the representation model to the first 5 images of two hold-out trajectories and predict forward for 45 steps using the latent dynamics, given only the actions. The recurrent state space model (RSSM; Hafner et al., 2018) performs accurate long-term predictions, enabling Dreamer to learn successful behaviors in a compact latent space.

Results – Representation learning



- Compare three natural choices described: pixel reconstruction, contrastive estimation, and pure reward prediction
- Figure shows clear differences for different representation learning approaches, with pixel reconstruction outperforming contrastive estimation on most tasks.
- This suggests that future improvements in representation learning are likely to translate to higher task performance with Dreamer.

Discussion of results

>=1 slide

What conclusions are drawn from the results?

Are the stated conclusions fully supported by the results and references? If so, why? (Recap the relevant supporting evidences from the given results + refs)

Conclusions

- The proposed approach learns long-horizon behaviors purely by latent imagination.
- Developed analytic gradients of multi-step values back through learned latent dynamics.
- outperforms previous methods in data-efficiency, computation time, and final performance on a variety of challenging continuous control tasks with image inputs.

Critique / Limitations / Open Issues

1 or more slides: What are the key limitations of the proposed approach / ideas? (e.g. does it require strong assumptions that are unlikely to be practical? Computationally expensive? Require a lot of data? Find only local optima?)

- If follow up work has addressed some of these limitations, include pointers to that. But don’t limit your discussion only to the problems / limitations that have already been addressed.

Contributions (Recap)

Approximately one bullet for each of the following (the paper on 1 slide)

- Problem the reading is discussing
- Why is it important and hard
- What is the key limitation of prior work
- What is the key insight(s) (try to do in 1-3) of the proposed work
- What did they demonstrate by this insight? (tighter theoretical bounds, state of the art performance on X, etc)

Questions & Limitations

- Scale latent imagination to environments of higher visual complexity
    - Complex environments？
    - Does the emphasis on long horizon imagination still help in other tasks?

Questions for recap

Where does this work use the variational loss?

How to backprop the stochastic actions, latent states et.al?

Question from me

- Is this on-policy or off-policy? Neither
- Is it actually an actor-critic jointly optimized upon a VAE.
- How to match the imaginary rewards with real rewards?