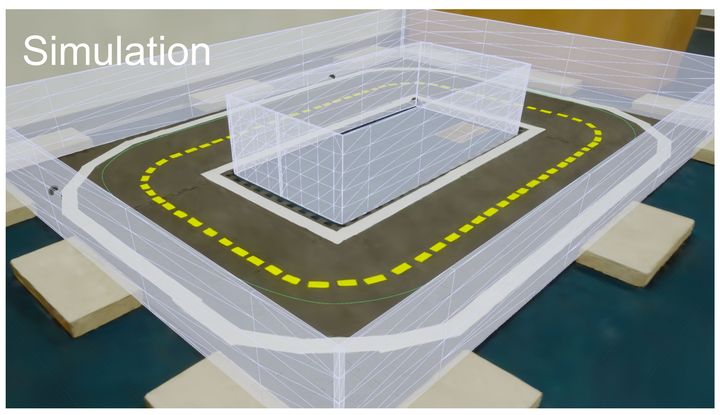
Adapting World Models with Latent-State Dynamics Residuals

JB Lanier, Kyungmin Kim, Armin Karamzade, Yifei Liu,
Ankita Sinha, Kat He, Davide Corsi, Roy Fox

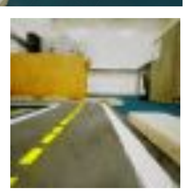
UCI University of California, Irvine

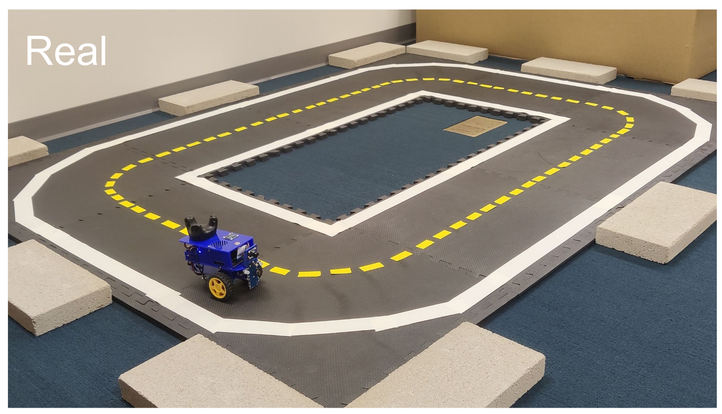
0. Motivation

- Real dynamics differ from simulator dynamics.
- Can calibrate simulator by training residual error model on simulator next-state predictions.
- What if we can't use/measure low dimensional states to apply a residual correction?
  - ➢ e.g. we only record actions+images in real.
  - ➢ Our solution: **Train a latent-state world model. Learn a residual to that instead**



Sim Image Observation:





Real Image Observation:

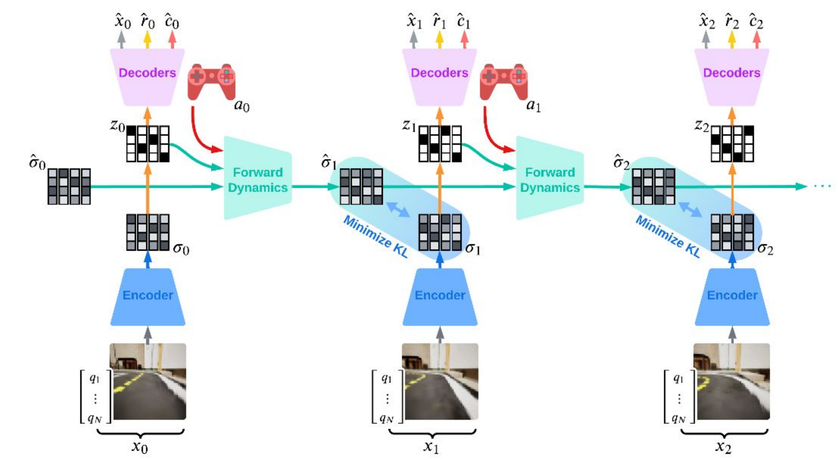
1. Latent-State World Model (DRAW)

- Learn to simulate trajectories in latent state space.
- RL agent trains by interacting with the world model instead of the environment.

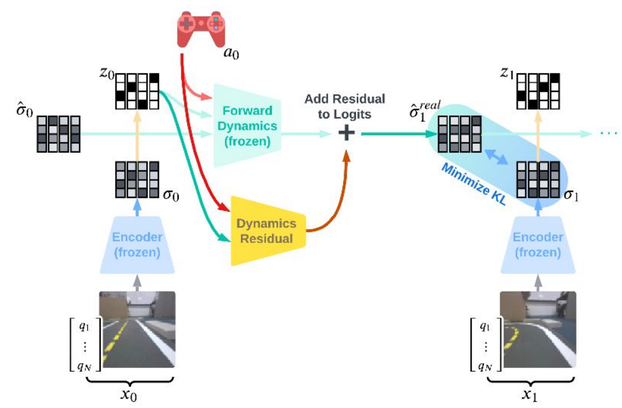
2. Latent-State Dynamics Residual (ReDRAW)

- Use offline real data to learn residual on latent-state dynamics.
- Train agent to perform task on calibrated world model that matches real environment.

Step 1. Pretain World Model in Simulation



Step 2. Train Dynamics Residual with Real Data



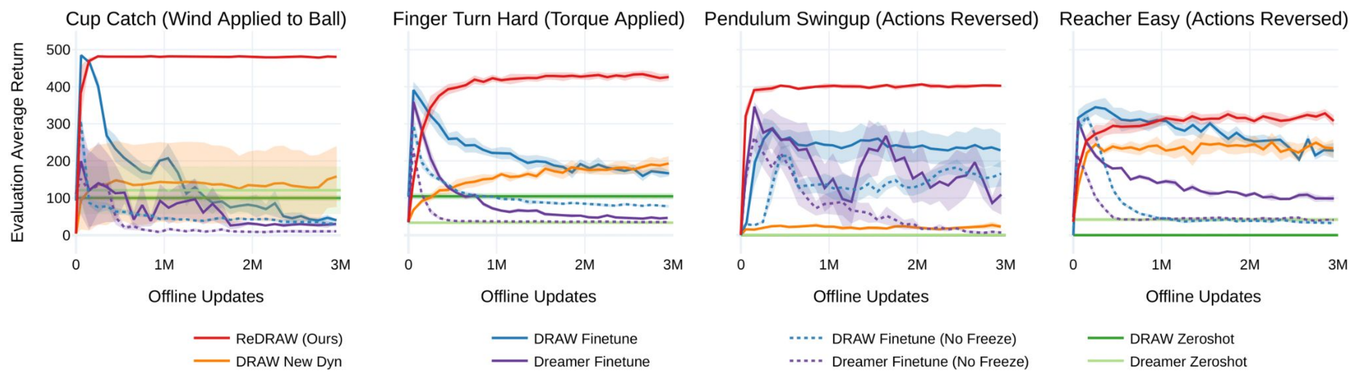
$$\text{State Encoder} \quad z_t \sim q_\theta(z_t|x_t) \tag{1}$$
$$\text{Forward Dynamics} \quad \hat{u}_t = f_\theta(z_{t-1}, \hat{o}_{t-1}, a_{t-1}) \tag{2}$$
$$\text{Forward Belief} \quad \hat{o}_t = p_\theta(\hat{z}_t|z_{t-1}, \hat{o}_{t-1}, a_{t-1}) \tag{3}$$
$$= \text{softmax}(\hat{u}_t) \tag{4}$$
$$\text{Forward Sample} \quad \hat{z}_t \sim \text{MultiCategorical}(\hat{o}_t) \tag{5}$$
$$\text{Reward} \quad \hat{r}_t \sim p_\theta(\hat{r}_t|z_t) \tag{6}$$
$$\text{Continuation} \quad \hat{c}_t \sim p_\theta(\hat{c}_t|z_t) \tag{7}$$
$$\text{State Decoder} \quad \hat{x}_t \sim p_\theta(\hat{x}_t|z_t) \tag{8}$$
$$\text{Policy} \quad a_t \sim \pi_\phi(a_t|z_t) \tag{9}$$
$$\text{Value Function} \quad v_t = V_\phi(z_t) \tag{10}$$

$$\hat{u}_t = f_\theta(z_{t-1}, \hat{o}^{real}_{t-1}, a_{t-1}) \tag{15}$$
$$\hat{e}_t = \delta_\psi(z_{t-1}, a_{t-1}) \tag{16}$$

$$\hat{o}^{real}_t = p_{\theta,\psi}(\hat{z}^{real}_t|z_{t-1}, \hat{o}^{real}_{t-1}, a_{t-1}) \tag{17}$$
$$= \text{softmax}(\hat{u}_t + \hat{e}_t) \tag{18}$$
$$z^{real}_t \sim \text{MultiCategorical}(\hat{o}^{real}_t). \tag{19}$$

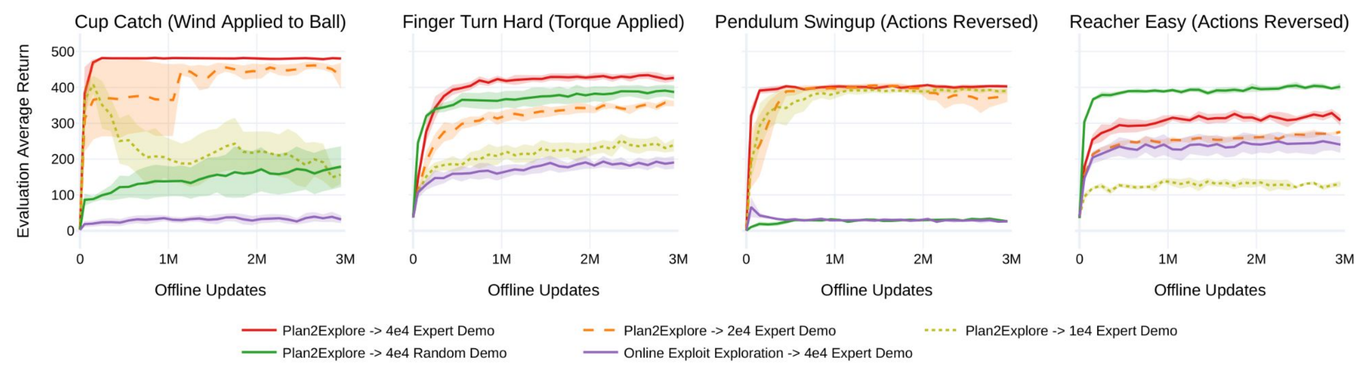
3. ReDRAW Avoids Overfitting to Real

- High capacity world model trains on millions of simulation timesteps.
- Low capacity residual calibrates to small real dataset and doesn't overfit.
- Useful when it's hard to validate checkpoints on a real robot.



4. Diverse Sim Training Data Helps

- Optimal real behaviors differs from sim. The frozen model needs cover real-relevant states.
- Pretrain world model with diverse sim trajectories (Plan2Explore, Sekar et al. 2020)
- World model trained just on reward-seeking sim data fails to adapt.



5. Tested in Visual Sim-to-Real Robot Task

- Real Duckiebots lane-following from images. Unreal Engine digital twin via Gaussian splatting.
- Calibrate with ~10k reward-free real transitions (~17 min)
- Only ReDRAW adapts when we reverse real actions.

Method	Transfer Sim to Unmodified Real: Avg Dense Reward (↑)	Avg Lap Time (sec) (↓)	Avg Center Offset (↓)	Transfer Sim to Actions-Reversed Real: Avg Dense Reward (↑)	Avg Lap Time (sec) (↓)	Avg Center Offset (↓)
Dreamer Zeroshot	-1.18 ± 0.23	–	6.86 ± 0.56	-2.35 ± 0.23	–	13.36 ± 1.13
Dreamer Finetune	-0.87 ± 0.33	–	5.45 ± 1.55	-1.61 ± 0.57	–	7.75 ± 1.53
DRAW Zeroshot	0.07 ± 0.06	**22.41** ± 0.73	5.12 ± 0.41	-2.72 ± 0.42	–	9.39 ± 1.35
ReDRAW (Ours)	**0.38** ± 0.02	**22.75** ± 0.26	**2.47** ± 0.26	**0.39** ± 0.03	**24.21** ± 1.15	**2.10** ± 0.39

Hafner et al. Mastering Diverse Domains through World Models. 2023.

Sekar et al. Planning to Explore via Self-Supervised World Models. 2020.