MuDreamer: Learning Control-Focused World Models without Pixel Reconstruction

Withdraw From ICLR 2024

November 10, 2025

Problem Setup in DreamerV3

DreamerV3 trains latents by reconstructing pixels.

- Pixel reconstruction shapes the representation toward *all* visual details (incl. backgrounds).
- Under strong visual distractors, this can dilute task-relevant features.
- Reconstruction is compute-heavy and not strictly necessary for control.

Goal: Focus the learned latent state ($\boldsymbol{s}_t = \{h_t, z_t\}$) on *control-relevant* information (rewards, continuation, value, action effects), not textures.

MuDreamer's Core Idea

Drop pixel reconstruction as a shaping signal. Learn a latent world model by predicting *task-relevant* quantities only:

- Reward ($\hat{r}_t$) and Continue ($\hat{c}_t$)
- **Value** (distributional, discretized (λ)-return)
- **Past Action** (inverse dynamics)

Optional: keep a decoder for visualization **with stop-grad**, so it does not influence latents.

Why: Dense supervision each step, but aligned with *control*.

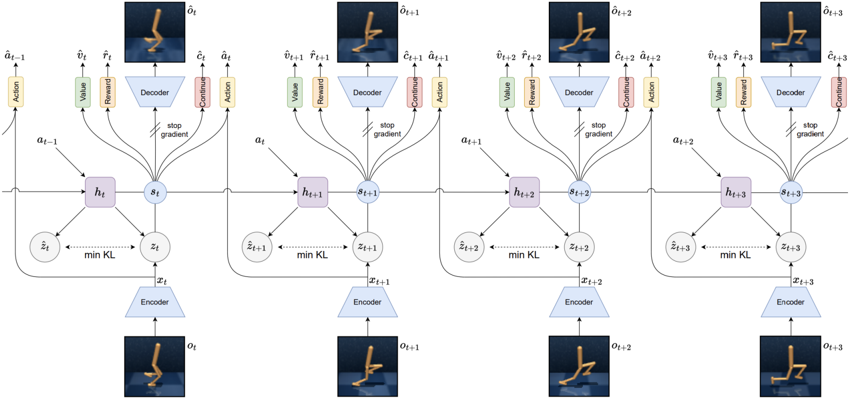
Architecture at a Glance

Per-timestep (strictly sequential)

1. Update memory: $(h_t = f_\phi(h_{t-1}, z_{t-1}, a_{t-1}))$
2. Prior over state: $(\hat{z}_t \sim p_\phi(z \mid h_t))$
3. Encode image: $(x_t = \mathrm{enc}_\phi(o_t))$
4. Posterior: $(z_t \sim q_\phi(z \mid h_t, x_t))$
5. Bundle: $(s_t = \{h_t, z_t\})$
6. Heads: $(\hat{r}_t, \hat{c}_t, \hat{v}_t)$ from (s_t); $\boldsymbol{(}\widehat{a}_{t-1}\boldsymbol{)}$ from $((x_t, s_{t-1}))$; optional decoder $(\hat{o}_t)$ from $(\mathrm{sg}(s_t))$.

Normalization: Use BatchNorm in the representation net to avoid collapse.

Architecture Diagram



Loss Stack (World Model)

Total model loss (sequence length (T)):

$$\mathcal{L}_{\text{model}} = \sum_{t=1}^{T} \Big(\beta_{\text{pred}} \mathcal{L}_{\text{pred},t} + \beta_{\text{dyn}} \mathcal{L}_{\text{dyn},t} + \beta_{\text{rep}} \mathcal{L}_{\text{rep},t}\Big).$$

Prediction loss ($\mathcal{L}_{\text{pred}}$) trains:

- Reward: $(-\log p_\phi(r_t \mid s_t))$
- Continue: $(-\log p_\phi(c_t \mid s_t))$
- **Value**: $(-\log p_\phi(R_t^\lambda \mid s_t))$ (two-hot symlog discretization)
- **Past Action**: $(-\log p_\phi(a_{t-1} \mid x_t, s_{t-1}))$
- Optional reconstruction (aux): $(-\log p_\phi(o_t \mid \mathrm{sg}(s_t)))$

Loss Stack (KL Terms)

Dynamics KL (train prior to match posterior)

$$\mathcal{L}_{\mathsf{dyn},t} = \max\big(1, \mathrm{KL}[\mathrm{sg} q_\phi(z_t \mid h_t, x_t) \mid p_\phi(z \mid h_t)]\big).$$

Representation KL (train posterior to be predictable)

$$\mathcal{L}_{\mathsf{rep},t} = \max\big(1, \mathrm{KL}[q_\phi(z_t \mid h_t, x_t) \mid \mathrm{sg} p_\phi(z \mid h_t)]\big).$$

Typical weights: $(\beta_{\mathsf{pred}} = 1.0, \beta_{\mathsf{dyn}} = 0.95, \beta_{\mathsf{rep}} = 0.05)$.
Note: KLs are capped below at 1 for stability.

Value Head Target (on Real Sequences)

Use a slow **EMA teacher** ($v_{\phi'}$) to define (λ)-return targets:

$$R_t^\lambda = r_{t+1} + \gamma c_{t+1}\big((1-\lambda)v_{\phi'}(s_{t+1}) + \lambda R_{t+1}^\lambda\big), \quad R_T^\lambda = v_{\phi'}(s_T).$$

- Transform with symlog, discretize with two-hot bins; train the student value head (v_ϕ) with cross-entropy.
- Purpose: shape **representations** (not used for behavior learning on imagined rollouts).

Behavior Learning (Dreamer-style)

Imagination in latent space (unchanged from V3):

- Start from a posterior state, roll out ($H = 15$) steps with the **prior** and the **actor**.
- Use model heads ($\hat{r}, \hat{c}$) on imagined states to construct discretized (λ)-return targets.
- **Critic** (v_ψ) learns on imagined trajectories, with EMA regularizer toward ($v_{\psi'}$).
- **Actor** maximizes normalized advantages plus entropy; backprop-through-model (continuous) or REINFORCE (discrete).

Key separation: behavior uses (v_ψ), not the world-model (v_ϕ).

Why Past-Action & Value Heads?

- **Value head (on real data):** forces latents to encode *long-horizon, return-relevant* features; improves stability.
- **Past-action head:** inverse dynamics from $((x_t, s_{t-1})) \to (a_{t-1})$ gives dense, control-aligned supervision each step, especially when rewards are sparse.
- Together they replace pixel reconstruction as a representation-learning signal, while keeping behavior learning unchanged.

Results: DeepMind Visual Control Suite (1M steps)

- Competitive or better than DreamerV3 on several tasks (e.g., Cheetah Run, Quadruped Walk, Reacher Hard).
- Overall mean/median scores comparable or improved with faster convergence in many cases.
- Long-horizon latent predictions remain accurate without reconstruction gradients.

DeepMind Visual Control Suite Results

Task	Random	TPC‡	DreamerPro‡	DreamerV3†	MuDreamer
Acrobot Swingup	0.3	5.1	13.1	9.1	**41.9**
Cartpole Balance	329.3	792.9	870.1	198.7	**974.8**
Cartpole Balance Sparse	53.9	26.9	198.4	18.4	**898.7**
Cartpole Swingup	67.4	574.8	689.2	145.7	**794.4**
Cartpole Swingup Sparse	0.0	0.2	**17.8**	0.3	0.0
Cheetah Run	6.7	440.8	**380.7**	94.3	318.1
Cup Catch	31.5	451.5	437.5	27.9	**904.5**
Finger Spin	0.9	696.8	**724.2**	96.5	644.2
Finger Turn Easy	48.8	**479.5**	232.4	197.8	229.4
Finger Turn Hard	35.0	198.3	**228.3**	39.8	226.7
Hopper Hop	0.0	0.2	**1.4**	0.6	0.2
Hopper Stand	1.9	14.5	**296.5**	3.0	5.4
Pendulum Swingup	2.0	**778.7**	777.6	8.0	606.8
Quadruped Run	8.8	162.9	470.8	108.9	**735.0**
Quadruped Walk	110.0	681.4	784.5	61.2	**872.8**
Reacher Easy	52.6	642.4	692.7	154.2	**914.4**
Reacher Hard	7.4	7.0	9.4	10.6	**13.5**
Walker Run	25.9	137.9	402.9	78.7	**432.8**
Walker Stand	139.4	935.4	940.6	254.4	**966.7**
Walker Walk	36.8	428.3	736.1	164.7	**759.0**
Mean	47.9	372.8	445.2	83.6	**517.0**
Median	28.7	409.8	416.2	57.1	**620.0**

Results: Natural Background (Robustness to Visual Distractors)

- Natural-video backgrounds make pixel reconstruction overly track textures.
- MuDreamer's aux decoder reconstructions visually **filter irrelevant background**, evidence the latent ignores distractors.
- Reported mean/median (example): Mean $\approx$ 517, Median $\approx$ 620 (vs. Dreamer baselines under same setting).

Results: Atari

			Lookahead search		No lookahead search			
Game	Random	Human	MuZero	EffZero	SimPLe	IRIS	DreamerV3	MuDreamer
Alien	228	7128	530	1140	617	420	**959**	951
Amidar	6	1720	39	102	74	143	139	**153**
Assault	222	742	500	1407	527	**1524**	706	891
Asterix	210	8503	1734	16844	1128	854	932	**1411**
Bank Heist	14	753	193	362	34	53	**649**	156
Battle Zone	2360	37188	7688	17938	4031	**13074**	12250	12080
Boxing	0	12	15	44	8	70	78	**96**
Breakout	2	30	48	406	16	**84**	31	34
Chopper Com.	811	7388	1350	1794	979	**1565**	420	808
Crazy Climber	10780	35829	56937	80125	62584	59324	**97190**	96128
Demon Attack	152	1971	3527	13298	208	**2034**	303	553
Freeway	0	30	22	22	17	**31**	0	5
Frostbite	65	4335	255	314	237	259	909	**1652**
Gopher	258	2412	1256	3518	597	2236	**3730**	1500
Hero	1027	30826	3095	8530	2657	7037	**11161**	8272
James Bond	29	303	88	459	100	**463**	445	409
Kangaroo	52	3035	63	962	51	838	4098	**4380**
Krull	1598	2666	4891	6047	2205	6616	7782	**9644**
Kung Fu Mas.	258	22736	18813	31112	14862	21760	21420	**26832**
Ms Pacman	307	6952	1266	1387	1480	999	1327	**2311**
Pong	–21	15	-7	21	13	15	**18**	**18**
Private Eye	25	69571	56	100	35	100	882	**1042**
Qbert	164	13455	3952	15458	1289	746	3405	**4061**
Road Runner	12	7845	2500	18512	5641	9615	**15565**	8460
Seaquest	68	42055	208	1020	**683**	661	618	428
Up N Down	533	11693	2897	16096	3350	3546	7600	**26494**
#Superhuman	0	N/A	5	13	1	10	9	**11**
Human Mean	0%	100%	56%	190%	33%	105%	112%	**126%**
Human Median	0%	100%	23%	116%	13%	29%	**49%**	43%

Ablations

Action & Value Heads

- Remove **Action** $\rightarrow$ performance drops (notably on sparse-reward tasks like Hopper Hop, Cartpole Swingup Sparse).
- Remove **Value** $\rightarrow$ less stable training.
- Remove **Both** $\rightarrow$ larger drop.

Conclusion: Both heads help the model learn dynamics-aware, control-relevant latents.

BatchNorm & KL Balancing

- **BatchNorm in representation** prevents collapse without pixel reconstruction; improves stability and speed.
- **KL balance** ((β_{rep}) vs. (β_{dyn})) matters: too little rep-KL destabilizes; too much slows learning.
- Reported curves (mean score across 20 DMC-Visual tasks) highlight sensitivity.

Implementation Notes

- **Horizon:** ($H = 15$). **Discount/Trace:** ($\gamma = 0.997$), ($\lambda = 0.95$).
- **EMA teacher for (v_ϕ):** momentum ($\tau = 0.01$).
- **Weights:** ($\beta_{\text{pred}} = 1.0$), ($\beta_{\text{dyn}} = 0.95$), ($\beta_{\text{rep}} = 0.05$).
- **Normalization:** BatchNorm in representation net.
- **Heads:** reward/continue/value from (s_t); past-action from ((x_t, s_{t-1})).
- **Optional decoder:** stop-grad only (visualization).

My Skepticism (Failure Modes)

- **Action–state ambiguity:** inverse dynamics may be ill-posed; weak learning signal.
- **Partial observability/occlusion:** (x_t) may not reflect (a_{t-1})'s effect.
- **Shortcut risks:** leaks if using (s_t) as input; overfit to trivial cues without strong data augs.
- **Non-controllable but relevant factors:** inverse dynamics won't emphasize them; value head helps but not always enough.

What I'd Test or Swap In

- **Stress tests:** sparse reward + distractors; measure linear probes for velocities/contacts; MI ($I(a_{t-1}; x_t \mid s_{t-1})$); rollout error.
- **Alternatives:** action-conditioned contrastive (InfoNCE), bisimulation-style regularization, forward-consistency KL/JSD, controllability Jacobian regularizers.

Takeaway: Treat past-action & value heads as *toggleable biases toward controllability and long-horizon relevance*; keep them if ablations pay off.