CSCI: 1470
The Dreamer Series

Randall Balestriero

DreamerV1 (2020) | DreamerV2 (2021) | DreamerV3 (2023) | Dreamer 4 (2025)


Dreamer 4


Dreamer 4

Why model-based RL?

Motivation: from model-free to world models

✗ Model-free limitations

- Requires millions of environment interactions
- No explicit understanding of dynamics
- Poor sample efficiency: every lesson must be experienced
- Cannot generalize to new tasks without retraining

✓ Model-based advantages

- Learn a world model: predict what happens next
- Plan by imagining trajectories — no env needed
- 10–100× more sample efficient in practice
- Transfer knowledge across tasks via shared dynamics

⭐ If we can learn an accurate model, we can train policies by “dreaming” — imagining outcomes without costly real interactions.

The Dreamer series at a glance

Four generations of world-model RL (2020–2025)

V1 — **Dream to Control** — ICLR 2020

Continuous latent imagination with RSSM + actor-critic. First to beat model-free on DMC from pixels.

V2 — **Mastering Atari with Discrete World Models** — ICLR 2022

Discrete categorical latents + KL balancing. First model-based to reach human-level Atari.

V3 — **Mastering Diverse Domains through World Models** — Nature 2025

Fixed hyperparameters across 150+ tasks. First to get diamonds in Minecraft without human data.

V4 — **Training Agents Inside of Scalable World Models** — arXiv 2025

Transformer + shortcut forcing. 2B params. First to get Minecraft diamonds purely from offline video.

DreamerV1 — core idea

Hafner et al. (2020) “Dream to Control: Learning Behaviors by Latent Imagination”

1. Learn world model
Encode obs → latent, predict forward

→

2. Learn behavior
Actor + critic on imagined rollouts

→

3. Collect experience
Execute policy in real env

Key innovation: analytic gradients

- Backpropagate value gradients directly through the world model’s predicted trajectory
- Much more efficient than REINFORCE (model-free) or CEM shooting (PlaNet)
- Actor and critic learn entirely from dreamed experience — no environment needed

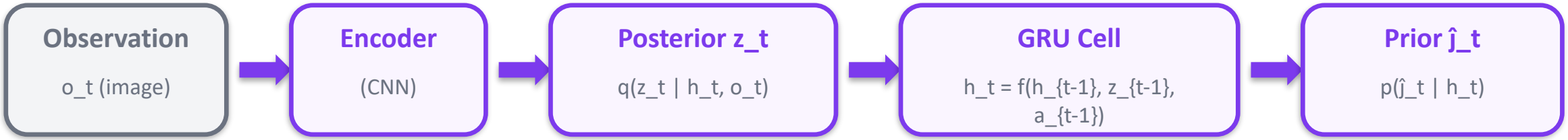
Actor $\pi(a|s)$
Trained on dreams
Reparameterized gradients

Critic $v(s)$
Estimates V^{λ}
Bootstraps at horizon H

⭐ DreamerV1 proved that backprop through a learned world model enables highly sample-efficient RL from pixels.

DreamerV1 — the RSSM world model

Recurrent State-Space Model: deterministic memory + stochastic predictions



State = (h_t, z_t)

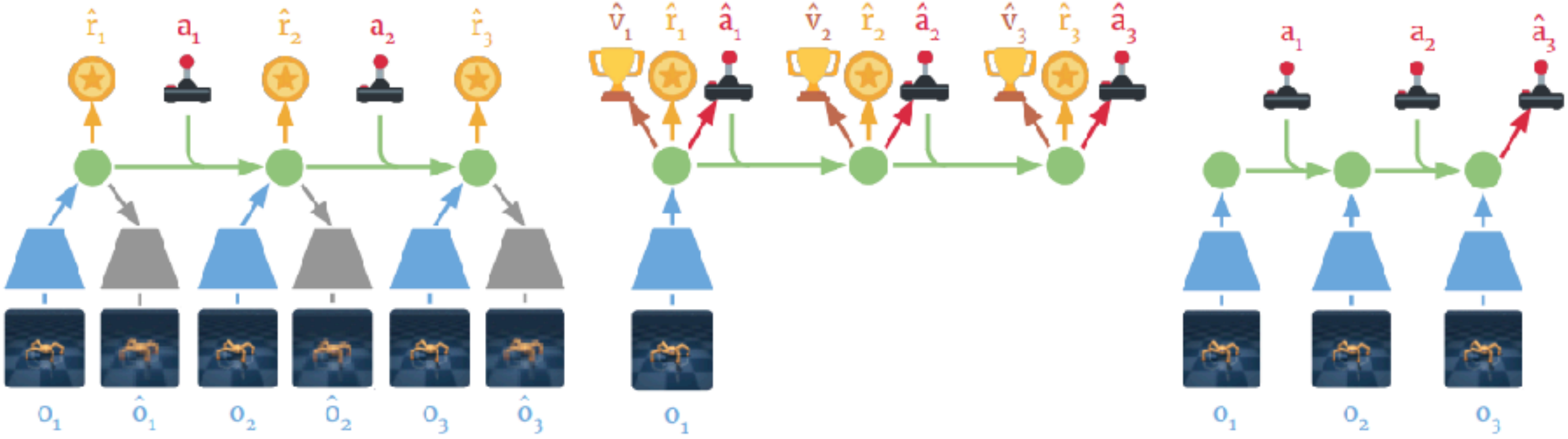
- h_t: deterministic (GRU) — aggregates history
- z_t: stochastic — captures uncertainty (Gaussian in V1)
- Prior predicts z without current obs → used in dreams
- KL[q ‖ p] trains the prior & regularizes posterior

Prediction heads

- Image decoder: reconstructs obs from (h_t, z_t)
- Reward predictor: predicts r_t from state
- Trained jointly via ELBO (reconstruction + KL)

⭐ The RSSM is the backbone of V1–V3. During imagination only the prior + GRU run — no encoder, no real observations.

DreamerV1 — the RSSM world model



DreamerV1 — results & limitations

DeepMind Control Suite: 20 continuous control tasks from pixels

✓ Strengths

- Beat PlaNet (prior SOTA) by a large margin
- Surpassed D4PG using 20× fewer interactions
- Single GPU, thousands of parallel dream trajectories
- First proof that backprop through world models scales

✗ Limitations

- Gaussian latents fail on discrete dynamics (Atari)
- Continuous actions only — no discrete support
- Hyperparameters require per-domain tuning
- Reconstruction loss wastes capacity on irrelevant pixels

⭐ V1 proved latent imagination works for continuous control. The Gaussian bottleneck motivates V2's move to categoricals.

DreamerV2 — from continuous to discrete latents

Hafner et al. (2021) “Mastering Atari with Discrete World Models”

Major changes from V1

- Categorical latents: 32 variables × 32 classes (straight-through gradients)
- Discrete + continuous actions via straight-through estimators
- KL balancing (α = 0.8): train prior faster to prevent collapse
- Learned discount predictor γ_t for variable-length episodes
- Mixed actor gradients: REINFORCE + dynamics backprop

V1 latent

$z \sim N(\mu, \sigma^2)$

Continuous, unimodal

V2 latent

$z = [c_1, c_2, \ldots, c_{32}]$

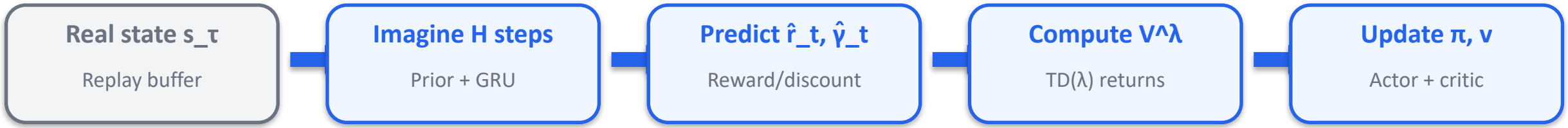
32 categoricals × 32 classes

Multimodal & expressive

⭐ Categorical latents can represent multi-modal distributions — critical for Atari where the next state can be one of several discrete possibilities.

DreamerV2 — actor-critic in imagination

Policy learning entirely through imagined rollouts



Actor gradient (mixed)

- Straight-through dynamics backprop through transitions
- REINFORCE for discrete action gradients
- Entropy regularization prevents premature convergence

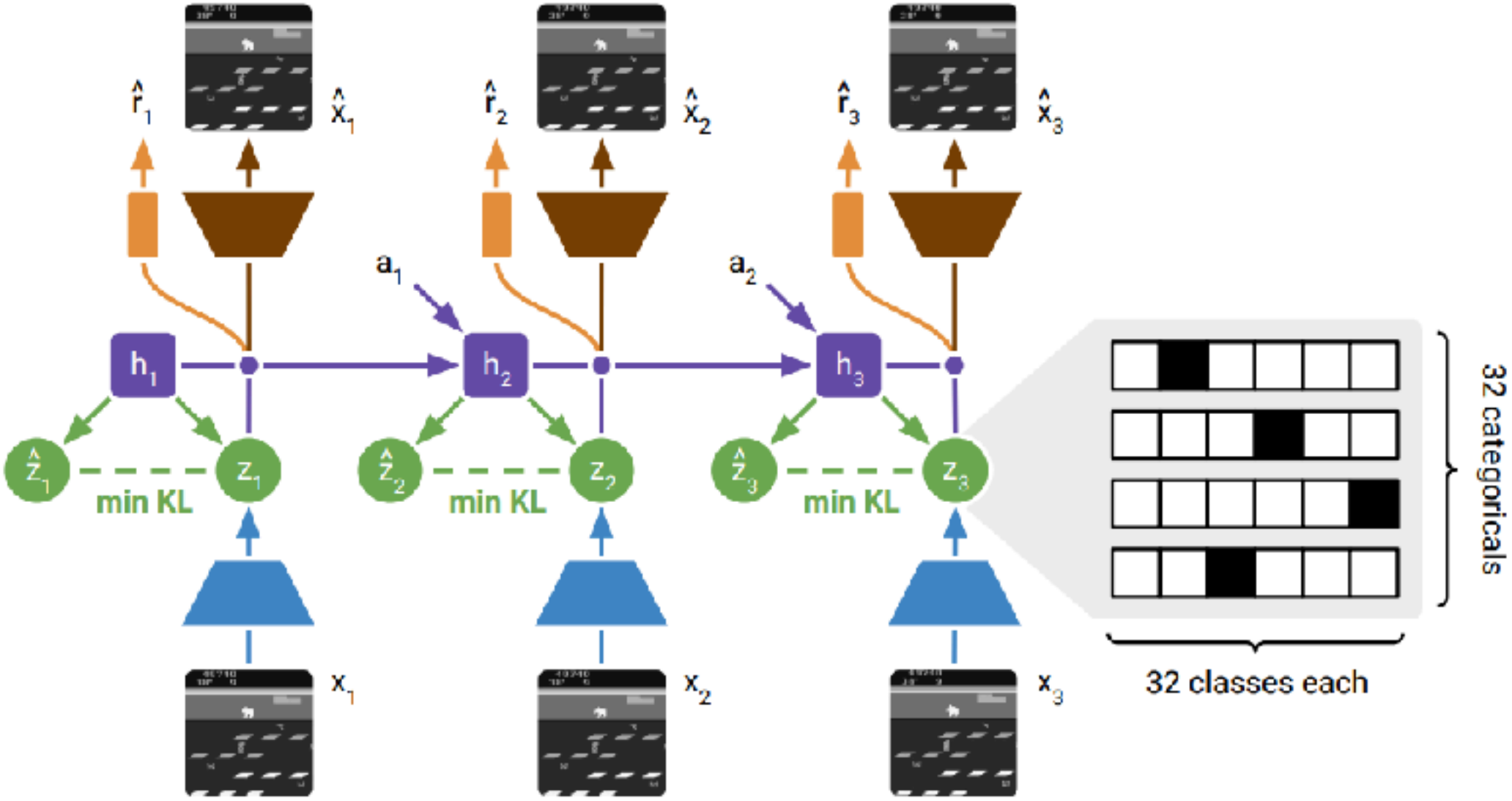
Critic

- Predicts V^λ(s_t) via λ-returns bootstrapped by target net
- EMA target network for stability
- All training happens inside the dream

⭐ No environment contact during policy learning. The world model generates all training data for the actor and critic.

DreamerV2 — actor-critic in imagination

Policy learning entirely through imagined rollouts



DreamerV2 — results

First model-based agent to achieve human-level Atari (200M frames)

✓ Achievements

- Human-level median across 55 Atari games
- Outperformed Rainbow, IQN, and other model-free baselines
- Competitive at 200M frames with far-higher-compute methods
- Maintained V1's DMC strength (continuous control)
- Atari has abrupt transitions — multimodal latents capture this
- Each categorical independently encodes a factor of variation

✗ Remaining limitations

- Still needs per-domain hyperparameter tuning
- Different reward scales destabilize training
- Larger models don't reliably help (scaling bottleneck)
- Open-world sparse reward (Minecraft) still unsolved

DreamerV3 — one algorithm, fixed hyperparameters

Hafner et al. (2023) “Mastering Diverse Domains through World Models” — Nature 2025

Core philosophy

- Same architecture + hyperparams across ALL domains
- The algorithm adapts to the problem, not the researcher
- Architecture = DreamerV2 RSSM; innovations are in normalization & losses

Robustness techniques

- Symlog: sign(x)·ln(|x|+1) compresses magnitudes
- Symexp twohot: distributional prediction over exp-spaced bins
- Percentile return normalization: [5th, 95th] running range

More techniques

- KL balancing + free bits (clip KL < 1 nat)
- 1% unimix on all categoricals (prevents mode collapse)
- Block GRU + SiLU activation + RMSNorm

Why this matters

- Zero tuning → applicable out of the box
- Scaling works: bigger model = better performance + efficiency
- Makes RL practical for real new application domains

DreamerV3 — symlog & return normalization

The tricks that enable domain-agnostic learning

Symlog transform

symlog(x) = sign(x) · ln(|x| + 1)

Compresses magnitudes symmetrically

Brings 0.01 and 1000 to comparable loss scales

Applied to encoder, decoder, reward, critic

Percentile return normalization

Track running 5th & 95th percentiles of V^λ returns

R_norm = (R - P5) / max(1, P95 - P5)

Actor always sees returns in ~[0, 1] range

No sensitivity to reward magnitude

Twohot encoding

Scalar → soft distribution over discrete bins

Target spread across two adjacent bins

Cross-entropy loss (not MSE) — robust to outliers

Bins are exponentially spaced via symexp

Unimix categoricals (1%)

Every categorical blends 1% uniform probability

Prevents any class from reaching p = 0

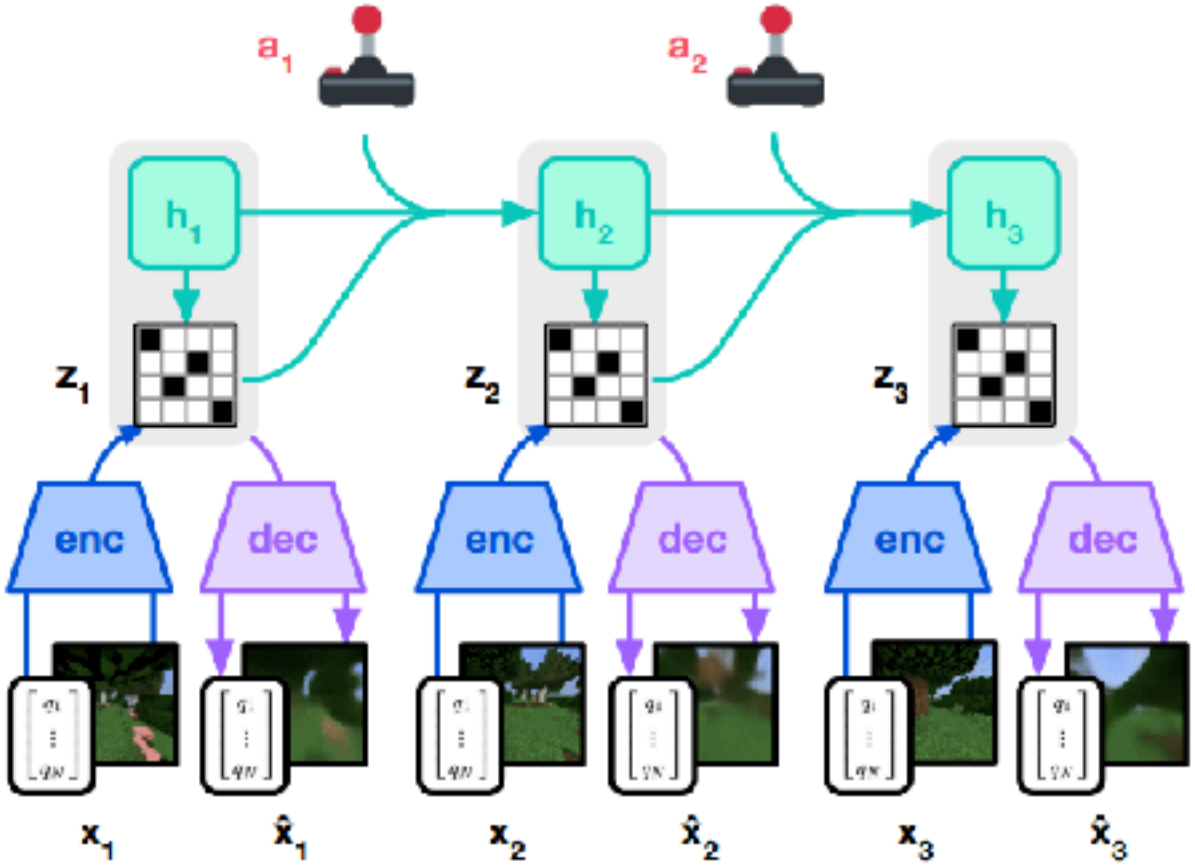
Avoids infinite KL / training instability

Simple yet critical for cross-domain robustness

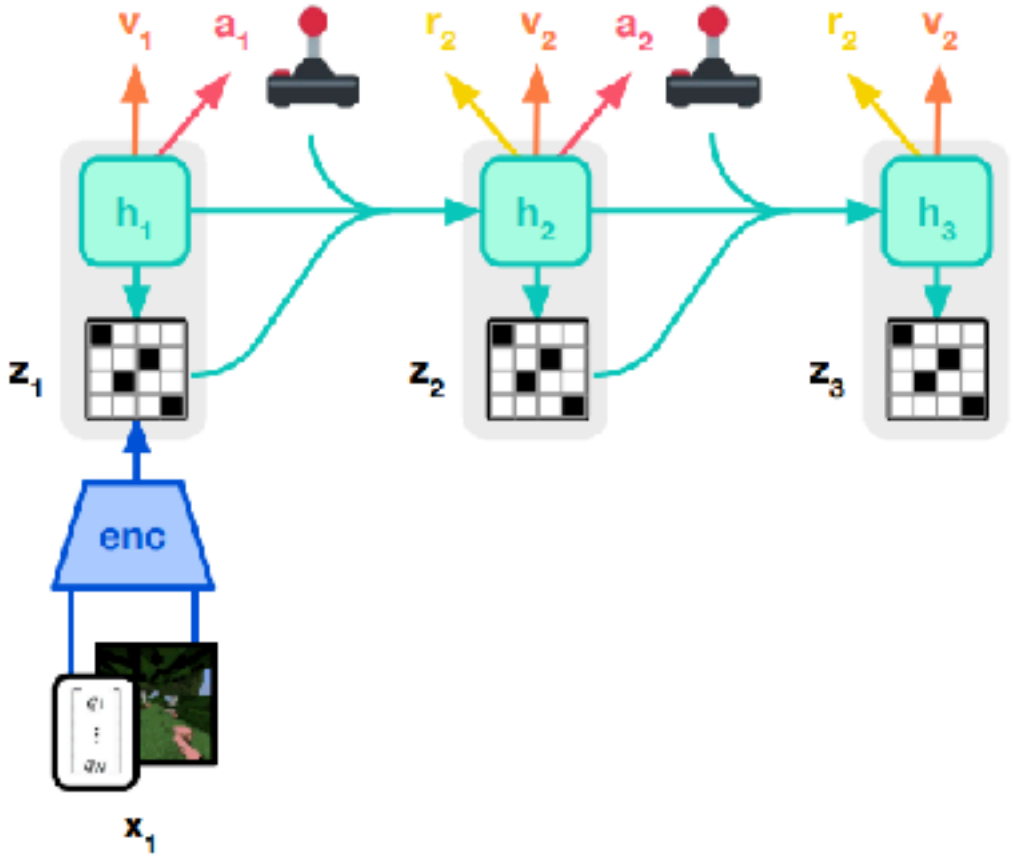
⭐ Together these ensure the same loss function produces comparable gradients whether rewards are 0.001 or 1,000,000.

DreamerV3 — symlog & return normalization

The tricks that enable domain-agnostic learning



(a) World Model Learning



(b) Actor Critic Learning

DreamerV3 — results across 150+ tasks

Fixed hyperparameters, single algorithm, diverse domains

7	150+	0	✓
Benchmarks	Total tasks	Configs tuned	Minecraft diamonds

Benchmarks conquered

- Proprioceptive & Visual Control Suite (DMC)
- Atari 100k and Atari 200M
- BSuite (468 configs), Crafter, DMLab, ProcGen
- Minecraft diamonds from scratch (no human data)

Key findings

- Beats specialized agents: DrQ-v2, TD-MPC, ImpactNet
- Substantially outperforms PPO (Acme implementation)
- Scaling: bigger model → better perf. AND data efficiency
- More gradient steps → higher data efficiency

⭐ DreamerV3 is the first single RL algorithm to work out-of-the-box across such diverse domains. Published in Nature (2025).

Dreamer 4 — a paradigm shift

Hafner, Yan, Lillicrap (2025) “Training Agents Inside of Scalable World Models”

RSSM (GRU)	→	**Block-causal Transformer**
Categorical latents	→	**Continuous tanh bottleneck**
ELBO (variational)	→	**Shortcut forcing (diffusion)**
Online RL	→	**Offline (video data only)**
~20M params	→	**~2B params (100×)**

Why the shift?

- Transformers parallelize training (RNNs can’t)
- Diffusion objectives model visual complexity better than VAEs
- Vast unlabeled video data available — no env interaction needed
- Shortcut forcing: 4 steps (not 64) → real-time inference on 1 GPU

⭐ Dreamer 4: from “learn a model while acting” to “learn a model from data, then act purely in imagination.”

Dreamer 4 — architecture

Causal tokenizer + block-causal transformer dynamics model

Causal tokenizer

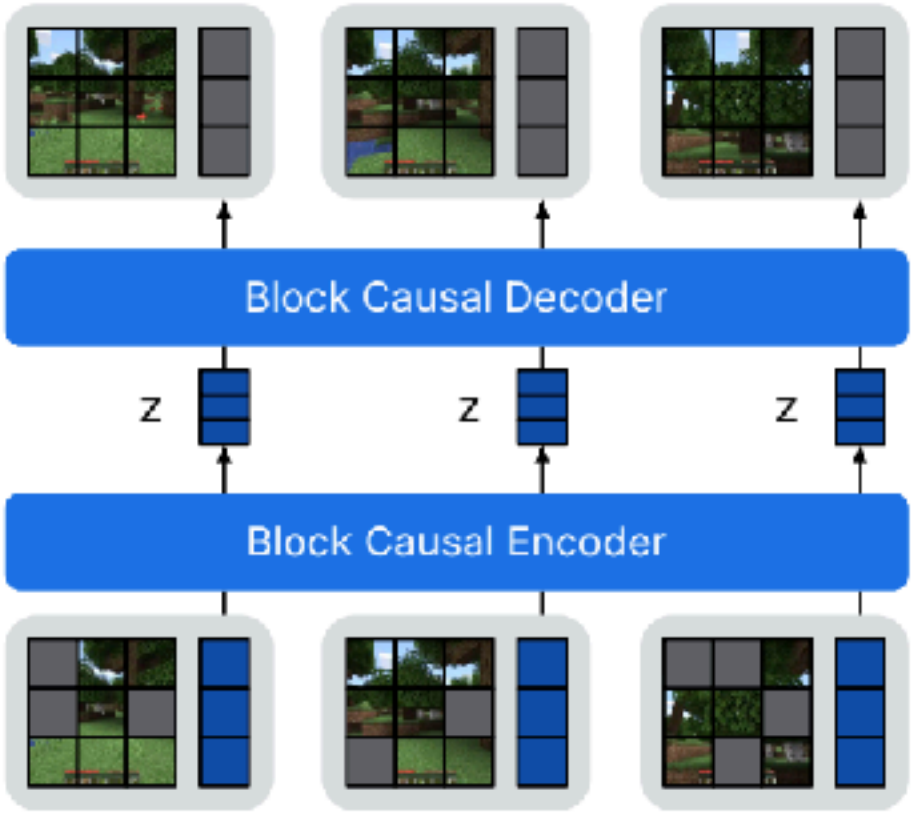
- Masked autoencoding: patches + learned tokens → decode from learned tokens only
- Tanh bottleneck (no VQ, no VAE): simple, continuous, bounded
- Causal temporal attention: frame-by-frame decoding
- Trained with MSE + LPIPS perceptual loss

Block-causal transformer

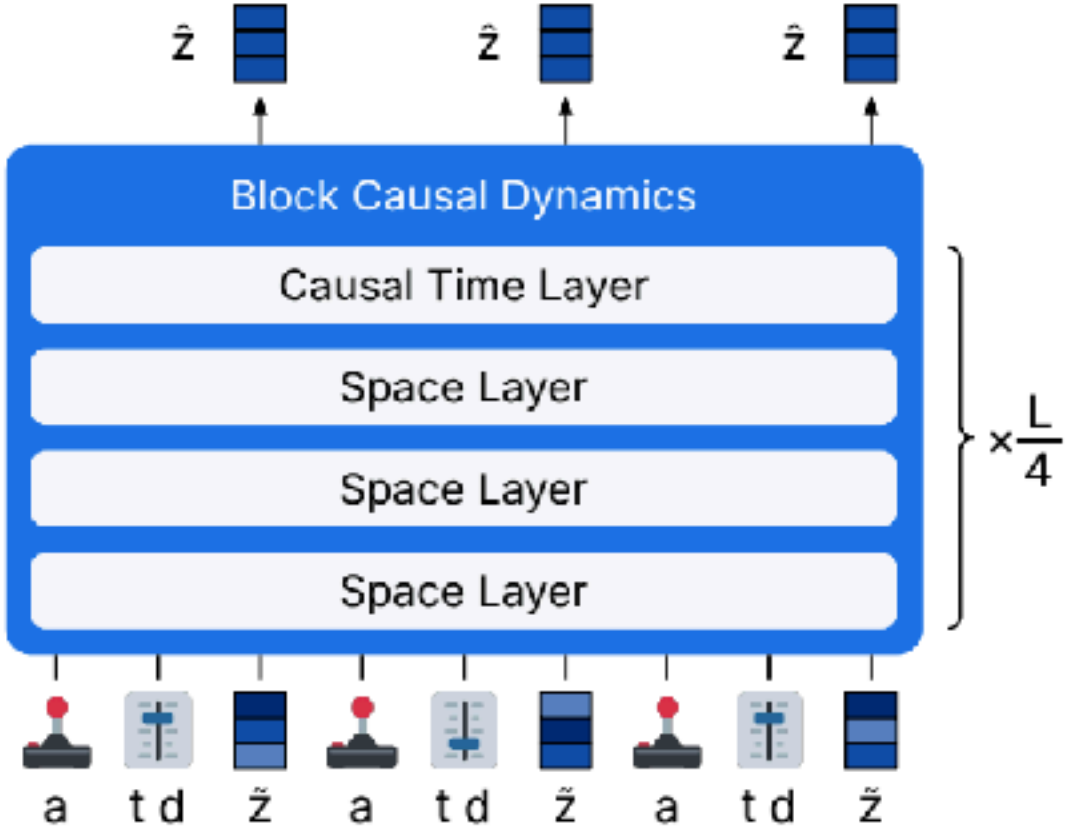
- Full attention within time step (spatial), causal across (temporal)
- Interleaved: [action, noise level, step size, latent tokens]
- Most layers spatial-only (efficient) — few temporal layers
- Scales to ~2B params with manageable compute

Dreamer 4 — architecture

Causal tokenizer + block-causal transformer dynamics model



(a) Causal Tokenizer



(b) Interactive Dynamics

Dreamer 4 — shortcut forcing & results

Fast diffusion + offline RL = diamonds from video

Shortcut forcing objective

- Generalizes diffusion forcing — denoise at any noise level, any step size
- Small steps: standard denoising (predict clean from noisy)
- Large steps: bootstrap — combine two smaller steps (self-distillation)
- x-prediction (predict clean directly) → less error accumulation

Agent training pipeline

- 4 steps match quality of 64 → 16× speedup
- Behavioral cloning from action-labeled subset (multi-token prediction)
- RL fine-tuning: policy + value heads, world model frozen, train in imagination

Headline results

⛏ **Minecraft diamonds**
0.7% success — first from pure video

💾 **Training data**
2,541 hrs VPT (100× less than VPT web)

⚡ **Inference speed**
~20 FPS on single H100 GPU

🎯 **Action labels**
Only ~10% of data needs labels

🧩 **Object interactions**
Crafting, mining, inventory predicted

Dreamer evolution — side-by-side comparison

How each version improved on its predecessor

	V1 (2020)	V2 (2021)	V3 (2023)	V4 (2025)
World model	RSSM (GRU)	RSSM (GRU)	RSSM (Block GRU)	Block-causal TF
Latent type	Gaussian	32×32 Categ.	32×32 Categ.	Continuous (tanh)
Objective	ELBO	ELBO	ELBO + symlog	Shortcut forcing
Actions	Continuous	Disc. + Cont.	Disc. + Cont.	Disc. + Cont.
Actor gradient	Reparam.	Mixed (ST+RF)	Mixed (ST+RF)	BC + RL fine-tune
Data regime	Online	Online	Online	Offline (video)
Hyperparams	Per-domain	Per-domain	Fixed (all)	Fixed
Scale	~5M	~20M	~20M	~2B
Flagship	DMC SOTA	Atari human	MC ⛏ (online)	MC ⛏ (offline)

Beyond reward-driven world models

Not all world models are for RL — the broader landscape

World models without rewards

- Video generation (Sora, Genie, GameNGen): dynamics for generation, no policy
- Self-supervised features (JEPA, V-JEPA): world model → representations
- Curiosity-driven exploration: model errors as intrinsic reward
- Learned simulators: robotics, driving — predict physics, not value
- LLM world models: predict action consequences in text

The unifying idea

- A world model answers: “what happens next?”
- Useful whether or not a reward signal exists
- Same model can serve RL, planning, generation, understanding
- Trend: world models becoming foundation models (pretrain → fine-tune)

🔮 Looking ahead: The boundary between “world model for RL” and “video foundation model” is rapidly blurring. Dreamer 4’s architecture is essentially a video diffusion model with a policy head.

See you on Wednesday!