World Models

DS 542 / DL4DS — Spring 2026

Boston University

> Under construction

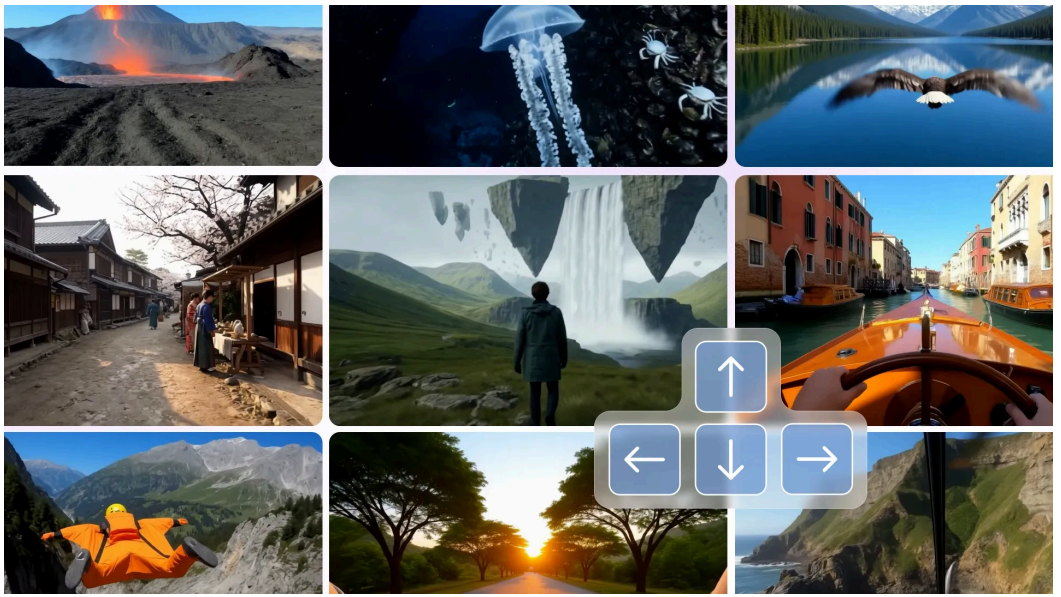
A lecture tracing the full historical arc — from Ha & Schmidhuber (2018) to the JEPA frontier (2025–2026).

* Created with Claude

Part 1 — Motivation & Definition

Hook: What does it take to *generate* a world?

DeepMind's **Genie 3** generates interactive, navigable 3D environments at **24 fps** from a single text prompt — with consistency for several minutes.



What does the network need to "know" to do this? *Source: DeepMind, Genie 3 (Aug 2025).*

Definition

A **world model** is a learned function predicting how the world (or its representation) evolves:

$$\hat{s}_{t+1} = f_\theta(s_t, a_t)$$

Often probabilistic:

$$p_\theta(s_{t+1} \mid s_t, a_t)$$

The choice of *what counts as s*, *what counts as a*, and *what we use the prediction for* is the central source of disagreement in the field.

Why learn a world model?

- **Sample efficiency** — train policies in imagination, not in the real world

Why learn a world model?

- **Sample efficiency** — train policies in imagination, not in the real world
- **Planning** — roll out counterfactual futures before acting

Why learn a world model?

- **Sample efficiency** — train policies in imagination, not in the real world
- **Planning** — roll out counterfactual futures before acting
- **Self-supervised representation learning** — prediction as the pretraining objective

Why learn a world model?

- **Sample efficiency** — train policies in imagination, not in the real world
- **Planning** — roll out counterfactual futures before acting
- **Self-supervised representation learning** — prediction as the pretraining objective
- **Simulation** — synthetic data, games, robotics, autonomous driving

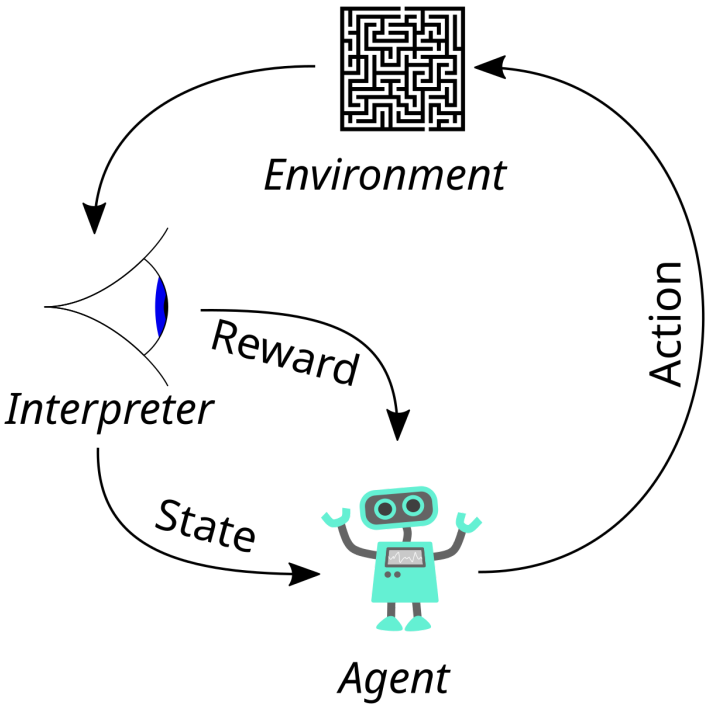
Callbacks to other things we've covered

Component	**Lectures it draws on**
Encoder of observations	VAE, ViT, CNN
Sequence dynamics	RNN, Transformer
High-fidelity decoding	Diffusion, GAN
Joint-embedding objectives	GNN, contrastive SSL
Action conditioning	(new today: RL)

World models are a *recombination* of techniques you already know.

Part 2 — Reinforcement Learning Primer

The agent–environment loop



At each step t: agent observes s_t, takes action a_t, receives reward r_{t+1}, and lands in s_{t+1}.

Goal: learn a **policy** $\pi(a \mid s)$ that maximizes long-term reward. *Source: Wikipedia / Sutton & Barto.*

The MDP formalism

A Markov Decision Process: (S, A, P, R, γ)

- **Transition function** $P(s_{t+1} \mid s_t, a_t)$ — *this is exactly the world model*
- **Reward** $R(s, a)$
- **Discount** $\gamma \in [0, 1)$

Return: $G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$

Value function: $V^\pi(s) = \mathbb{E}_\pi [G_t \mid s_t = s]$

In standard ("model-free") RL, we **never learn P** — we just learn from samples.

Model-free vs. model-based RL

	Model-free	Model-based
Learns	Policy / value directly	Policy *and* a model $\hat P$ of the world
Examples	DQN, PPO, SAC	World Models, Dreamer, MuZero
Sample efficiency	Poor (millions of steps)	Strong
Real-world cost	Often prohibitive	Can train *in imagination*
Risk	Stable	Errors in $\hat P$ compound

> *If you have a good world model, RL becomes much cheaper.* The rest of this lecture is **how do we learn one**? — especially important for robotics and autonomous driving, where real rollouts are expensive, slow, and dangerous.

Part 3 — The Ha & Schmidhuber Blueprint (2018)

The 2018 paper that started the modern field

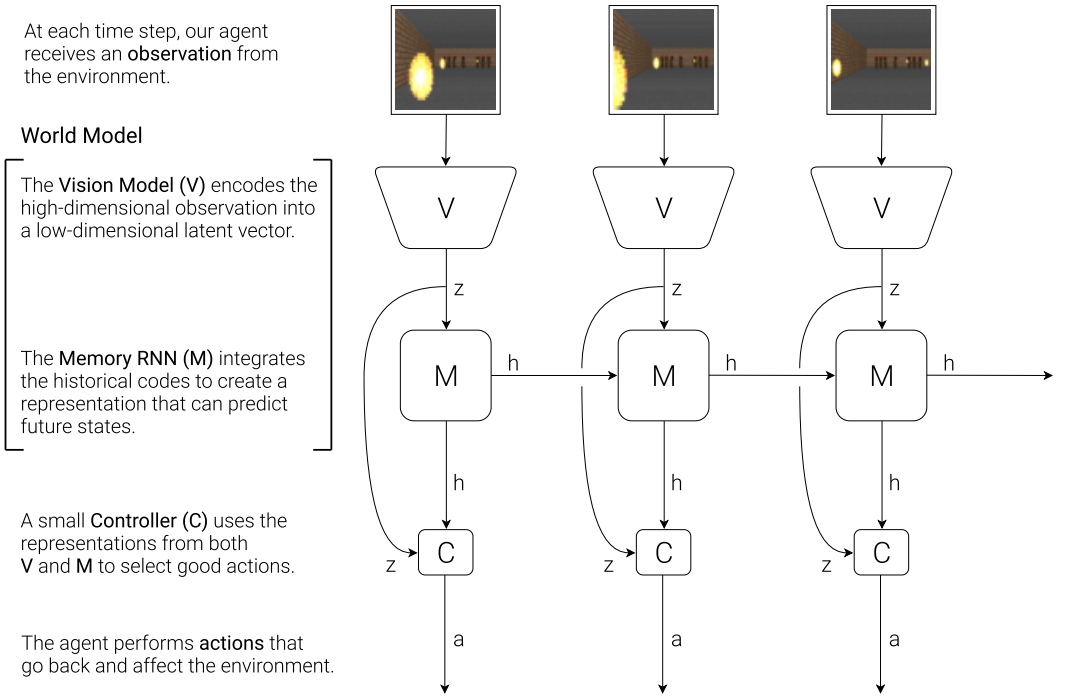
David Ha and Jürgen Schmidhuber, **"World Models"** (NeurIPS 2018).

worldmodels.github.io — an interactive paper with live demos. Still the cleanest pedagogical resource in the field.

Demos: **CarRacing-v0** and **ViZDoom: Take Cover**.

> *"We can even train our agent entirely inside of its own hallucinated dream generated by its world model, and transfer this policy back into the actual environment."*

Three components: V, M, C



- **V** (Vision): VAE — compresses each frame into latent z_t
- **M** (Memory): MDN-RNN — predicts $p(z_{t+1} \mid z_t, a_t, h_t)$
- **C** (Controller): tiny linear policy — outputs a_t from $[z_t, h_t]$

Source: Ha & Schmidhuber 2018.

V model — VAE on game frames

A **convolutional VAE** compresses each $64 \times 64 \times 3$ frame into a 32-dimensional latent z_t.

Callback to VAE lecture — same architecture, same ELBO loss:

$$\mathcal{L}_{\text{VAE}} = \mathbb{E}_{q(z|x)}[\log p(x|z)] - D_{\text{KL}}(q(z|x) \| p(z))$$

Why VAE rather than autoencoder? Smooth, sampleable latent space — essential when we want to *dream* (sample futures).

M model — MDN-RNN

The memory predicts the *next* latent given the current latent, action, and hidden state:

$$p(z_{t+1} \mid z_t, a_t, h_t) = \sum_{k=1}^{K} \pi_k(h_t)\, \mathcal{N}\!\left(z_{t+1};\, \mu_k(h_t),\, \sigma_k^2(h_t)\right)$$

A **mixture of Gaussians** — because the future is multimodal.

A fireball might or might not appear next frame. A single Gaussian would average those futures and produce mush.

The temperature trick

Sampling temperature τ controls the dream's uncertainty:

$$p_\tau(z_{t+1}) \propto p(z_{t+1})^{1/\tau}$$

- $\tau \to 0$: deterministic, "easy" dream
- $\tau \to \infty$: very noisy, "hard" dream

Counterintuitive result: training the controller in a *harder* dream than reality produces a more *robust* policy. The agent learns to handle imagined adversity.

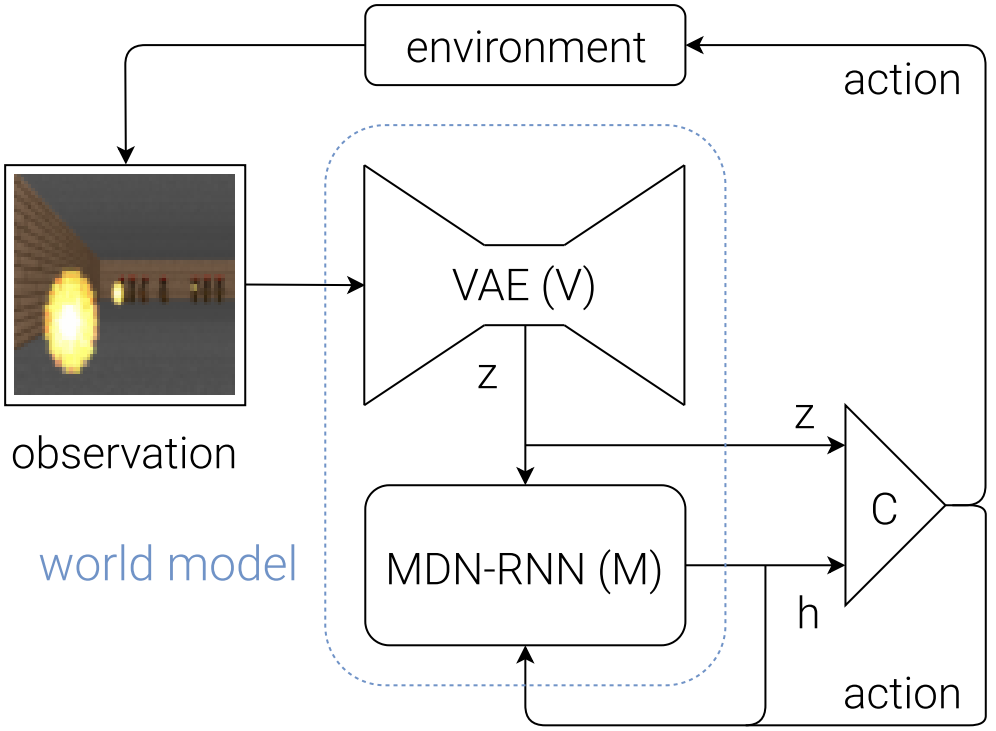
C model — a tiny linear controller

$$a_t = W_c \, [z_t, h_t] + b_c$$

About **1000 parameters**. Trained with **CMA-ES** (Covariance Matrix Adaptation Evolution Strategy) — no gradient descent.

Why so small? With V and M doing the heavy lifting on perception and prediction, the controller only needs to map a compact state to an action.

The kicker: train entirely in the dream



1. Collect random rollouts in the real environment.
2. Train V on frames; train M on the latent sequence.
3. **Train C entirely inside M's dreams** (no real rollouts).
4. Deploy C in the real environment — it works.

Source: Ha & Schmidhuber 2018.

Why this paper still matters

- Established the **V + M + C decomposition** that everything afterward refines.
- Demonstrated **prediction-as-pretraining** — prefigures modern self-supervised learning.
- Proved you can **transfer from imagination to reality**.

Limits: tiny scale; brittle outside narrow domains; only 2D pixel games; MDN-RNN poor at long horizons.

Part 4 — Scaling Up: PlaNet → Dreamer → MuZero

What needed to change

Ha & Schmidhuber's blueprint had three weaknesses at scale:

- **Pixel reconstruction is wasteful** — VAE spends capacity on textures
- **Long-horizon credit assignment fails** in MDN-RNN dreams
- **Continuous control** (robotics) needs gradients, not just evolution

The Dreamer line (Hafner et al., 2019–2025) addresses all three.

RSSM — Recurrent State-Space Model

The core innovation. Latent state has **two parts**:

- h_t — **deterministic** GRU hidden state (reliable memory)
- z_t — **stochastic** sample (uncertainty)

This combo gives both *consistent long-term memory* and *expressive multimodal prediction*.

RSSM equations

$$ \begin{aligned} h_t &= f_\phi(h_{t-1},\, z_{t-1},\, a_{t-1}) && \text{recurrent backbone} \\[4pt] z_t &\sim q_\phi(z_t \mid h_t,\, o_t) && \text{posterior — uses observation} \\[4pt] \hat z_t &\sim p_\phi(z_t \mid h_t) && \text{prior — used for dreaming} \end{aligned} $$

Plus prediction heads for: reconstructed observation $\hat o_t$, reward $\hat r_t$, episode continuation $\hat c_t$.

The prior p_ϕ is trained to match the posterior q_ϕ (KL term) — so we can *imagine* without observations.

DreamerV1/V2 — actor–critic in imagined latent space

(a) Train the world model from real experience.
(b) Imagine latent rollouts; train **actor** and **critic** entirely inside these imagined trajectories.

Source: Hafner et al., DreamerV3 / Nature 2025 arXiv.

DreamerV3 — one algorithm, 150+ tasks

Same architecture, hardened with engineering tricks:

- **Symlog** predictions for reward, value, observation (handles wildly different scales)
- **KL balancing** + **free bits** to prevent posterior collapse
- **Categorical latents** — 32 distributions × 32 classes, with straight-through gradients
- **Adaptive gradient clipping**, RMSNorm, SiLU

Result: **a single fixed hyperparameter set** solves continuous control, Atari, BSuite, Crafter, **and Minecraft from scratch**.

The Minecraft Diamond milestone

DreamerV3 was the **first algorithm to collect a diamond in Minecraft** — no human data, no curriculum, ~30M environment steps from raw pixels.

Dreamer 4 (Sept 2025) does it from *offline data alone*: a sequence of >20,000 keyboard/mouse actions planned entirely in imagination.

This validates the central claim: a sufficiently good world model + planning-in-imagination = far-sighted strategic behavior.

MuZero — implicit world models

A different lineage (Schrittwieser et al., DeepMind 2020).

- **Never reconstructs pixels**.
- Predicts only what planning needs: **reward**, **value**, **policy**.
- Beats AlphaZero on Go, Chess, Shogi *without being given the rules*.

	Generative (Dreamer)	**Implicit (MuZero)**
Reconstructs observations?	Yes	No
Latent trained for	Reconstruction + reward	Reward + value + policy
Pros	Interpretable; debuggable	Compact; no wasted capacity
Cons	Wastes capacity on textures	Latent is opaque

> Maybe pixel reconstruction is the *wrong* objective entirely. — This idea returns as **JEPA** in Part 6.

Part 5 — World Models Meet Generative Video

The 2024 regime shift

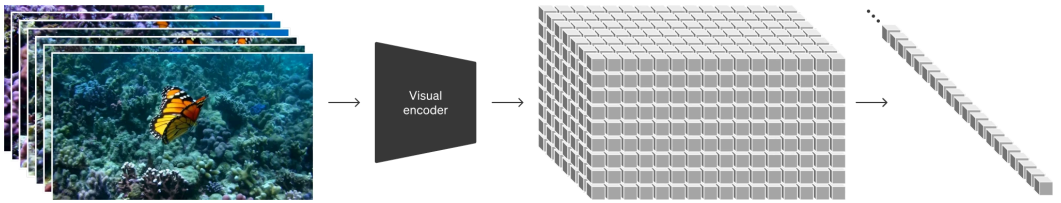
Pre-2023 world models were **small, task-specific, trained on game data**.

Post-2023, the same recipe that worked for LLMs — **internet-scale pretraining** + **transformers** — gets applied to video.

A new question emerges: *if you train a generative video model at sufficient scale, does it become a world model?*

Sora — "video generation models as world simulators"

OpenAI, Feb 2024. Architecture: a **diffusion transformer** operating on **spacetime patches** of video latents.



A pretrained video VAE compresses each clip; the result is cut into *spacetime patches*; a DiT denoises them.

Callback to ViT (patches) and diffusion (denoising) lectures.

Source: [OpenAI Sora technical report](#).

Sora's emergent properties — and failures

Emergent:

- Object permanence across cuts
- Primitive 3D consistency
- Multi-character interactions

Failure modes documented in the technical report and follow-up analyses:

- Glasses fall *through* tables
- Hands with seven fingers
- Wolves spontaneously appearing
- No conservation of mass or causality

Photorealism is not understanding.

Genie — the *interactive* turn

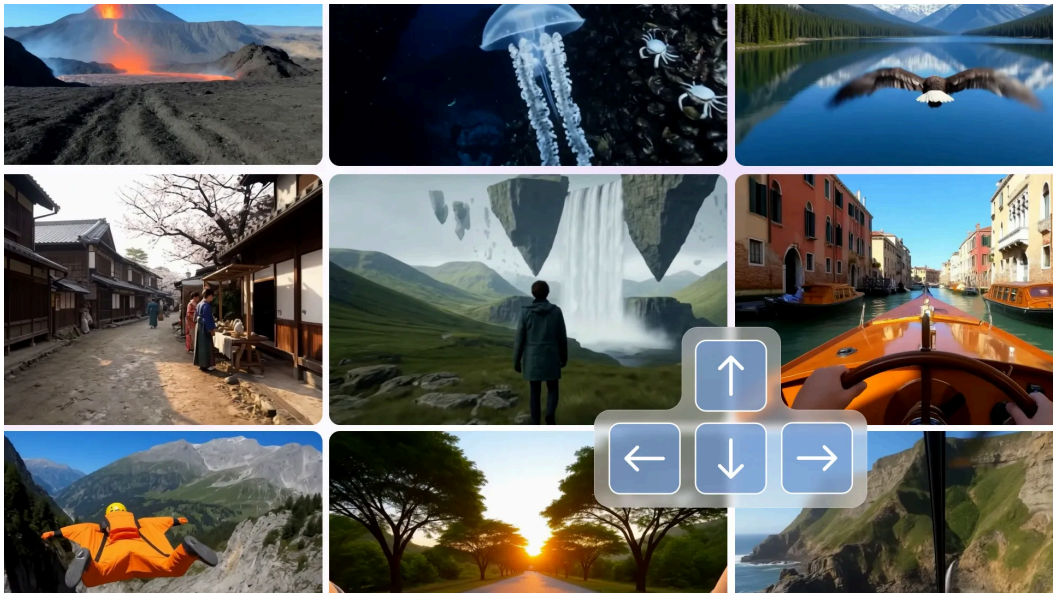
Bruce et al., DeepMind, Feb 2024 (arXiv:2402.15391). **11B parameters**, trained on **200,000 hours** of unlabeled internet gameplay video.

Three components:

- **Video tokenizer** — ST-Transformer (spatiotemporal) VQ-VAE
- **Latent action model** — infers a *learned* discrete action vocabulary (8 actions, embedding size 32) from observed transitions, **with no action labels**
- **Dynamics model** — autoregressive ST-Transformer with MaskGIT, predicts next frame token given past tokens + latent action

Subtlety (Bandaru, 2025): the latent actions are *not interpretable* — "action 1" might mean diagonal-up, not "jump." So is Genie a true *world model* or just a *controllable video generator*? Debated.

Genie 3 — real-time playable worlds



DeepMind, Aug 2025. **24 fps, 720p, several minutes** of consistent dynamics from a text prompt. No hard-coded physics.

Key new mechanism: **promptable world events** — mid-session text commands ("start a thunderstorm") modify the running simulation.

Source: [DeepMind Genie 3 blog](#).

Cosmos — physical AI foundation model

NVIDIA, Jan 2025 (arXiv:2501.03575). Explicitly a **"World Foundation Model"** for physical AI.

- Trained on **20M hours of real-world video** (vs. Genie's gameplay)
- Both **diffusion** and **autoregressive** variants
- Built-in evaluation for **physical alignment** — conservation, plausibility
- Aimed at robotics, autonomous driving, simulation

Strategy: the **Hugging Face of world models** — open weights, downstream fine-tuning recipes.

Architectural comparison

	Sora	Genie 3	Cosmos
Input	Text prompt	Text + interactive	Text / image / video
Output	Fixed-length video	Real-time playable world	Video, conditioned
Backbone	DiT (diffusion + transformer)	Autoregressive latent diffusion	Both DiT and AR
Conditioning	Text only	Text + per-frame actions	Multimodal
Training data	Mixed video	Gameplay video	Real-world video
Primary goal	Cinematic generation	Interactive simulation	Physical realism

The central tension

Genie produces playable worlds. Sora produces beautiful clips. Cosmos targets physics fidelity.

But:

- **Compounding error** in long autoregressive rollouts
- **No causal grounding** — pixel-level loss does not enforce physics
- **Unclear utility for planning** — can an agent actually use these for decision-making?

This sets up Part 6.

Part 6 — The JEPA Alternative & Open Problems

LeCun's critique (2022 AMI position paper)

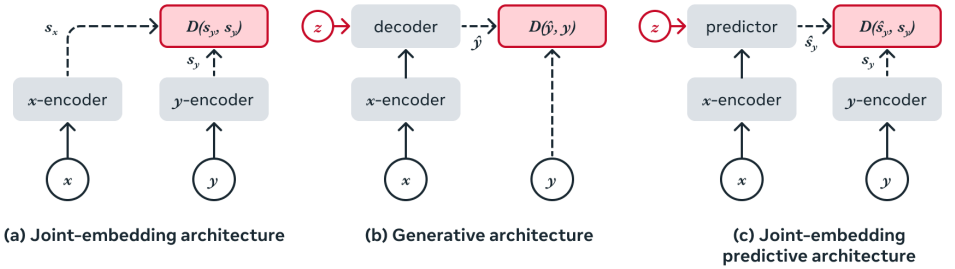
> *Generative pixel-prediction is fundamentally wasteful.*

Most pixels carry no decision-relevant information. Predicting them spends capacity on irrelevant texture detail.

LeCun proposes an alternative: predict **representations** of the future, not pixels.

This becomes the **Joint-Embedding Predictive Architecture (JEPA)**.

JEPA architecture



- **Generative** (left, Masked AE): decode pixel y from x
- **JEPA** (right): predict the *embedding* s_y from s_x, conditioned on action / latent z

Loss is computed in **representation space**, not pixel space.

Source: [Assran et al., I-JEPA (CVPR 2023)](#).

JEPA loss (conceptually)

$$\mathcal{L}_{\text{JEPA}} = \big\| \,
\text{Predictor}\big(\text{Enc}_\theta(x),\, a\big) \;-\;
\text{sg}\!\left[\text{Enc}_{\bar\theta}(y)\right] \,\big\|^2$$

Plus a **non-collapse mechanism** (EMA target encoder, VICReg regularization, or similar).

	MAE	**Contrastive**	**JEPA**
Predicts	Pixels	Distance to negatives	Representations
Needs negatives?	No	Yes	No
Collapse risk	Low	Low	**High** (needs reg)

V-JEPA 2 (Meta, June 2025)

Two-stage training:

1. **Pretrain** on >1 million hours of internet video — *action-free* masking objective
2. **Post-train** an action-conditioned variant on just **62 hours** of unlabeled robot video

Demonstrates **zero-shot robotic planning** from image goals — V-JEPA 2-AC.

77.3% top-1 on Something-Something v2; 39.7 R@5 on Epic-Kitchens-100.

Connects pretraining at scale (LLM lineage) with planning (Dreamer lineage).

Where JEPA fits in our lecture map

JEPA is in the same family as:

- **Contrastive image SSL** (SimCLR, MoCo, DINO)
- **GNN representation learning** (callback to your GNN lecture)
- **MuZero** (implicit prediction, no reconstruction)

The "implicit world model" lineage running parallel to the "generative" one.

The big debate: do LLMs already have world models?

Pro (autoregressive prediction is sufficient):

- **Othello-GPT** (Li et al. 2023) — an internal board representation emerges from move sequences alone, no spatial supervision
- Sora exhibits primitive object permanence
- Mechanistic interpretability finds world-model-like structures in LLMs

Con (LeCun's view):

- LLMs hallucinate physics
- Token-level loss doesn't enforce world consistency
- True world models need representation-space prediction + planning

This is an *open* question. Two reasonable views.

Evaluation crisis & persistent failure modes

Next-frame prediction loss does **not** measure physical understanding. New benchmarks shift toward causally grounded metrics:

- **IntPhys 2** (Bordes et al., 2025) — intuitive physics: object permanence, gravity, solidity
- **CausalProbe** — causal ordering of events
- **Long-horizon consistency** — does the world stay coherent over 60s?

Even with photorealistic outputs, persistent failure modes remain in 2025: **compounding error** in long autoregressive rollouts, **conservation violations** (mass, energy, identity), **action-conditioning brittleness**, lack of **causal vs. correlational** structure, and the **compute and licensing** burden of video pretraining at scale.

The 2025–2026 inflection

- **Dreamer 4** solves Minecraft Diamond from offline data (Sep 2025)
- **Genie 3** real-time interactive worlds (Aug 2025)
- **V-JEPA 2** demonstrates zero-shot robotic planning (Jun 2025)
- **Cosmos 2.5** — open World Foundation Models, 2M+ downloads (Oct 2025)
- **World Labs** launches **Marble** — commercial 3D world generation (Nov 2025)
- **LeCun leaves Meta** to start **AMI Labs**, raising €500M to build JEPA-style world models (late 2025)

The field is forking into a generative-video lineage and a JEPA lineage.

Wrap-Up

Synthesis: the V/M/C blueprint, evolved

Module	2018	2025
V (perception)	ConvVAE	VQ-VAE → ViT → JEPA encoder
M (dynamics)	MDN-RNN	RSSM → ST-Transformer → DiT
C (control)	Linear + CMA-ES	Actor-critic in imagination → text prompts → goal-conditioned planning

Every prior lecture in the course (FCN, CNN, VAE, GAN, transformer, ViT, diffusion, GNN) plays a role somewhere in this table.

Three open questions for you

1. Is the path to general AI through **scaling LLMs**, **scaling world models**, or **fusing them**?
2. Can a model trained only on *pixels* learn *causal* physics, or do we need **action-conditioned data**?
3. What is the right **evaluation metric** for a "good" world model?

Reading list

Tutorials & overviews:

- Ha & Schmidhuber, *World Models* — interactive paper at worldmodels.github.io
- Rohit Bandaru, "World Models" (2025) — rohitbandaru.github.io/blog/World-Models
- Survey: arXiv:2411.14499 (ACM CSUR 2025)

Primary sources:

- DreamerV3 — Hafner et al., *Nature* 2025; arXiv:2301.04104
- Dreamer 4 — Hafner et al., arXiv:2509.24527
- Sora — OpenAI technical report (Feb 2024)
- Genie — Bruce et al., arXiv:2402.15391; DeepMind blogs for Genie 2 / 3
- Cosmos — NVIDIA, arXiv:2501.03575
- V-JEPA 2 — Assran et al., arXiv:2506.09985
- LeCun, "A Path Towards Autonomous Machine Intelligence" (2022) OpenReview

Questions?