

CIS 6280 · FALL 2026

WORLD MODELS

LECTURE 2

History · Foundations · Probabilistic Formulation

Course Capacity Will Be 70

We will increase the capacity from 49 to 70.

01

Still waiting to enroll?

Please wait for a seat to open as enrollment settles and some students drop.

02

Planning to audit?

Email the TAs. They will collect your information and add you to Canvas later.

ENROLL · Wait for an official seat. **AUDIT** · Contact the TAs for Canvas access.

Three Questions Organize This Course

Across very different worlds, the same three questions kept recurring.

01

Representation

What information must persist?

02

Prediction

What could happen next?

03

Interaction

What can the system change—and what feedback returns?

TODAY · Where did the prediction question come from, and how do we state it precisely?

ONE IDEA · MANY LINEAGES

Modern World Models Are a Convergence

Internal surrogates, predictive state, action–consequence models, learned dynamics, and imagination came from different fields.

World Models (2018) packaged the idea. It did not invent it.



History Roadmap — What Each Field Added

Read this as a convergence, not one family tree.

Cognition	internal surrogates and flexible representation
Control	predictive state, feedback, and uncertainty
AI + system ID	executable or learned transitions
Reinforcement learning	planning and learning from imagination
Deep learning	pixels, latent variables, and generative futures

54 HYPOTHESIS ON THE NATURE OF THOUGHT

of operations possible for it than for any other single process or model.

Of course we have still to face the question *why* these analogies between different mechanisms—these similarities of relation-structure—should exist. To see common principles and simple rules running through such complexity is at first perplexing though intriguing. When, however, we find that the apparently complex objects around us are combinations of a few almost indestructible units, such as electrons, it becomes less perplexing. For it is inevitable that processes corresponding to arithmetical addition of these elementary units—electrons, protons, etc.—should manifest themselves in many instances. That is to say, if all pieces of pure iron consist of similar groupings of similar units, it is very likely that two pieces of iron placed end to end will add in length according to some simple law, and that pieces of other substances will do the same. The emergence of common principles and similarities is, then, not so surprising if it is shown that all substances are composed of similar ultimate units, for the appearance of uniformity and similarity is then the reappearance of a uniformity and similarity which were in fact ever present. We are still faced with the more ultimate question, why diverse materials should consist of combinations of a very few types of ultimate particles. The short life of some particles, such as positrons, suggests that in the vicinity of other particles, such as electrons, they are not stable; if it could be shown that in any such encounter the electron is more stable and the positron less, some kind of explanation would have been given as to why electrons are more frequent. It would still be conceivable that innumerable entirely different types of ultimate particle could have existed; if there is only one type in existence at a time there is nothing for it to be inconsistent with (apart from such factors as the mutual repulsion

HYPOTHESIS ON THE NATURE OF THOUGHT 55

of similar particles and their consequent inability to form combinations). If, however, we conceive the world as made of a number of different types of ultimate unit, it is possible that they would prove to be mutually unstable and that all particles must acquire the same properties in order to exist, much as water in different tubes on a common arm always finds a constant level.

This, however, is very speculative; the point of interest for our present enquiry is that physical reality is built up, apparently, from a few fundamental types of units whose properties determine many of the properties of the most complicated phenomena, and this seems to afford a sufficient explanation of the emergence of analogies between mechanisms and similarities of relation-structure among these combinations without the necessity of any theory of objective universals.

We have now to enquire how the neural mechanism, in producing numerical measurement and calculation, has managed to function in a way so much more universal and flexible than any other. Our question, to emphasize it once again, is not to ask what kind of thing a number is, but to think what kind of mechanism could represent so many physically possible or impossible, and yet self-consistent, processes as number does.

The key may possibly lie in the following fact: in causal chains and physical or chemical combinations, the possibility of a given combination tends to be limited by other factors than the mere self-consistency of the combination. If you try to determine whether the series of integers can be extended to infinity by piling bricks on top of one another, you find that after a time the bricks fall down, or you cannot reach to pile any more up, or you run short of bricks or die; all these are extraneous difficulties. More subtle are the difficulties of adding nine oranges to nine apples, or of trying

Craik, 1943 · [Open six-page excerpt ↗](#)



Kenneth Craik, first director of the MRC Applied Psychology Unit

Internal Model — Try Before Acting

Internal rehearsal can replace some costly trial and error.

Craik's surrogate included external reality—and the organism's own possible actions.

1943–48

1948–60s

1960s–70s

1970s–90s

2000–14

2015–20

2020s

Small-Scale Model — Preserve What Matters

A model is a selective surrogate—not a miniature photograph.

01

Situation

Retain the conditions that matter.

02

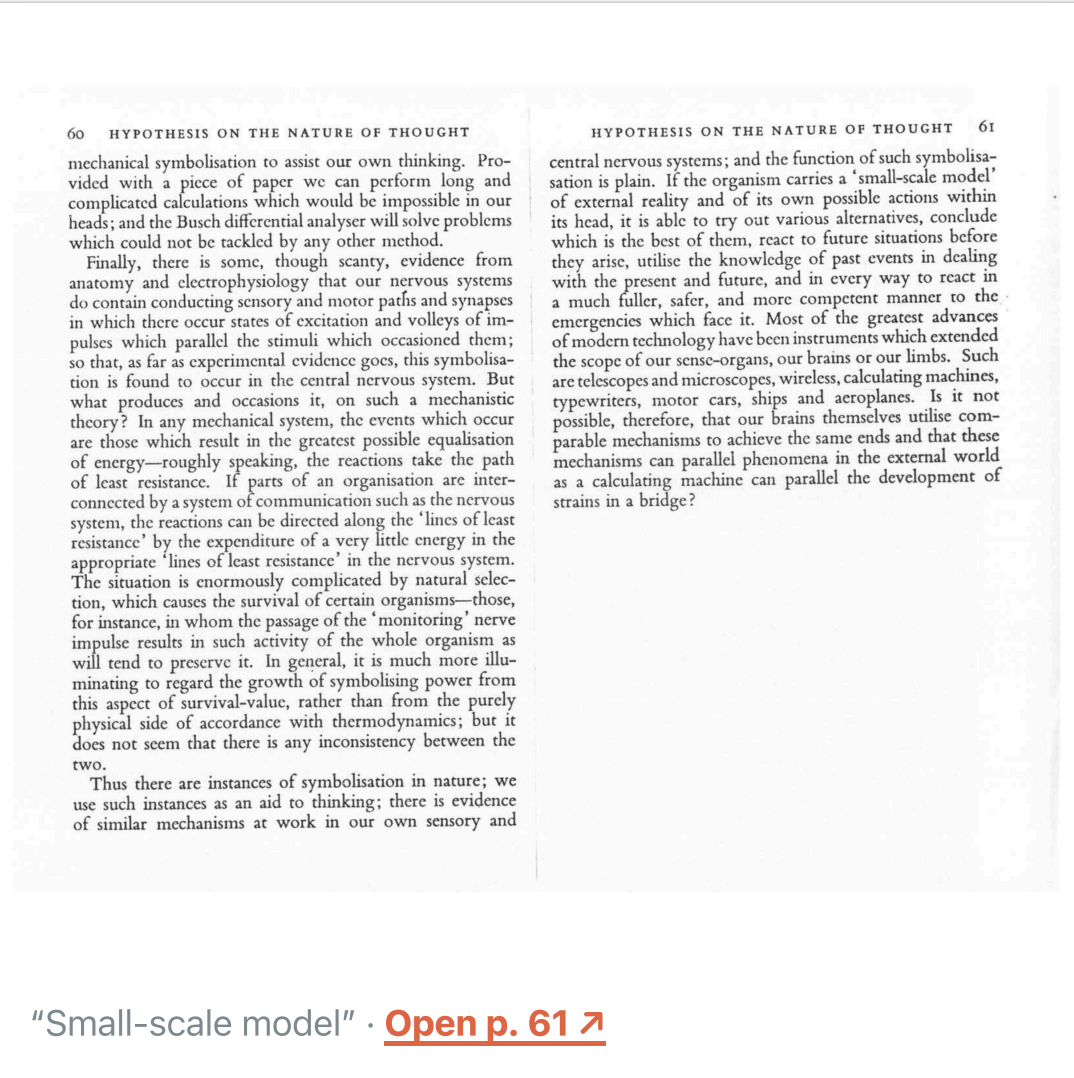
Possible action

Represent alternatives that could be tried.

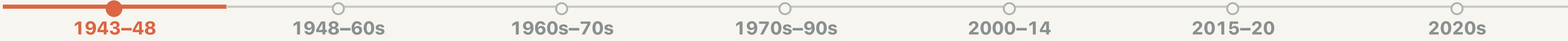
03

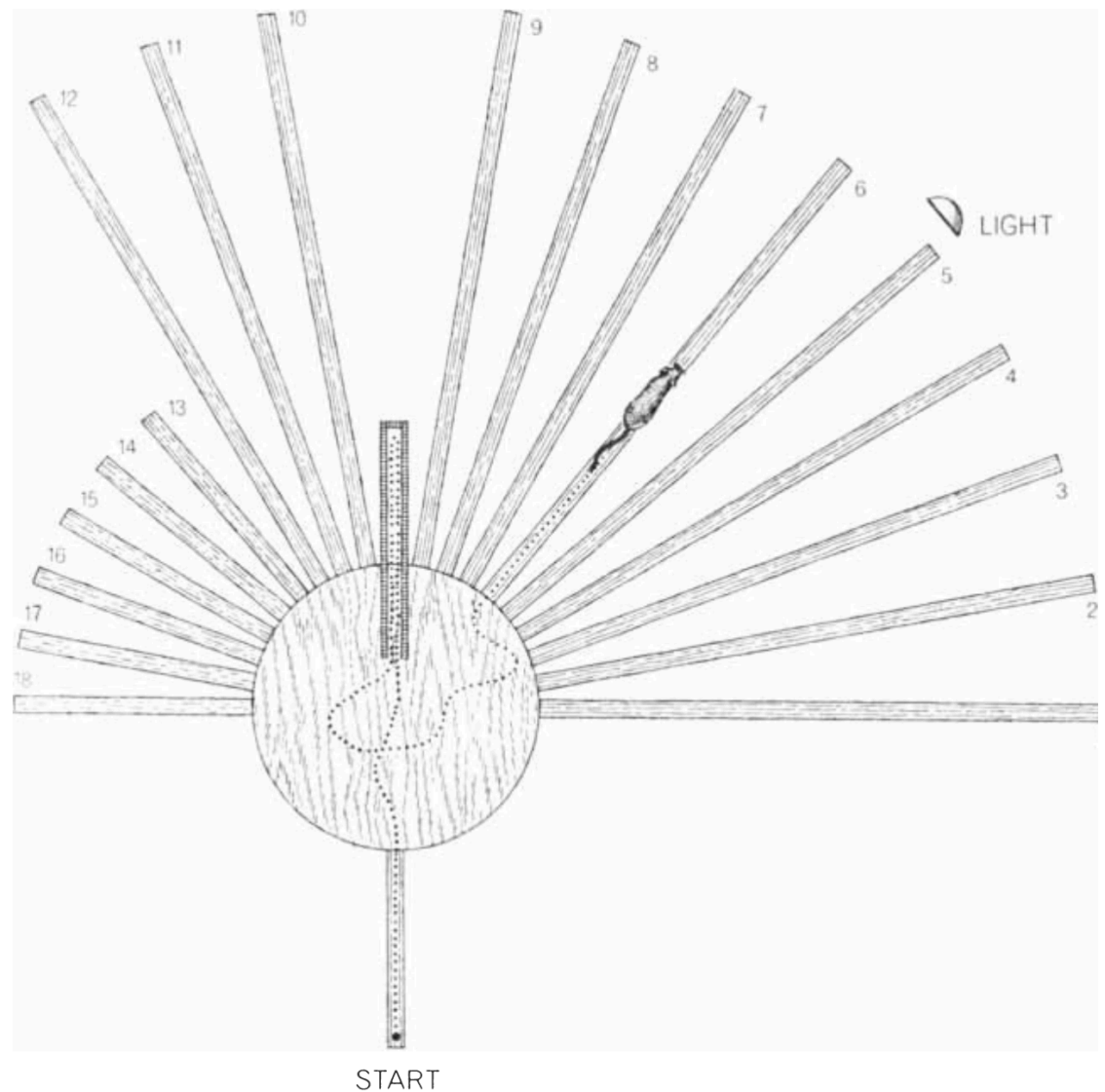
Consequence

Preserve differences that guide a choice.



A bridge model keeps forces; a game model keeps legal moves. Neither needs every surface detail.





"SUNBURST" MAZE is a modification of the training maze shown at the top of this page. A

Sunburst maze: the familiar route is blocked

Cognitive Map — Find a New Route

Flexible behavior suggests more than a memorized response chain.

Relational knowledge can be recombined when the familiar path is blocked.

Latent Learning — Knowledge Before Reward

CLASSIC DELAYED-REWARD RESULT

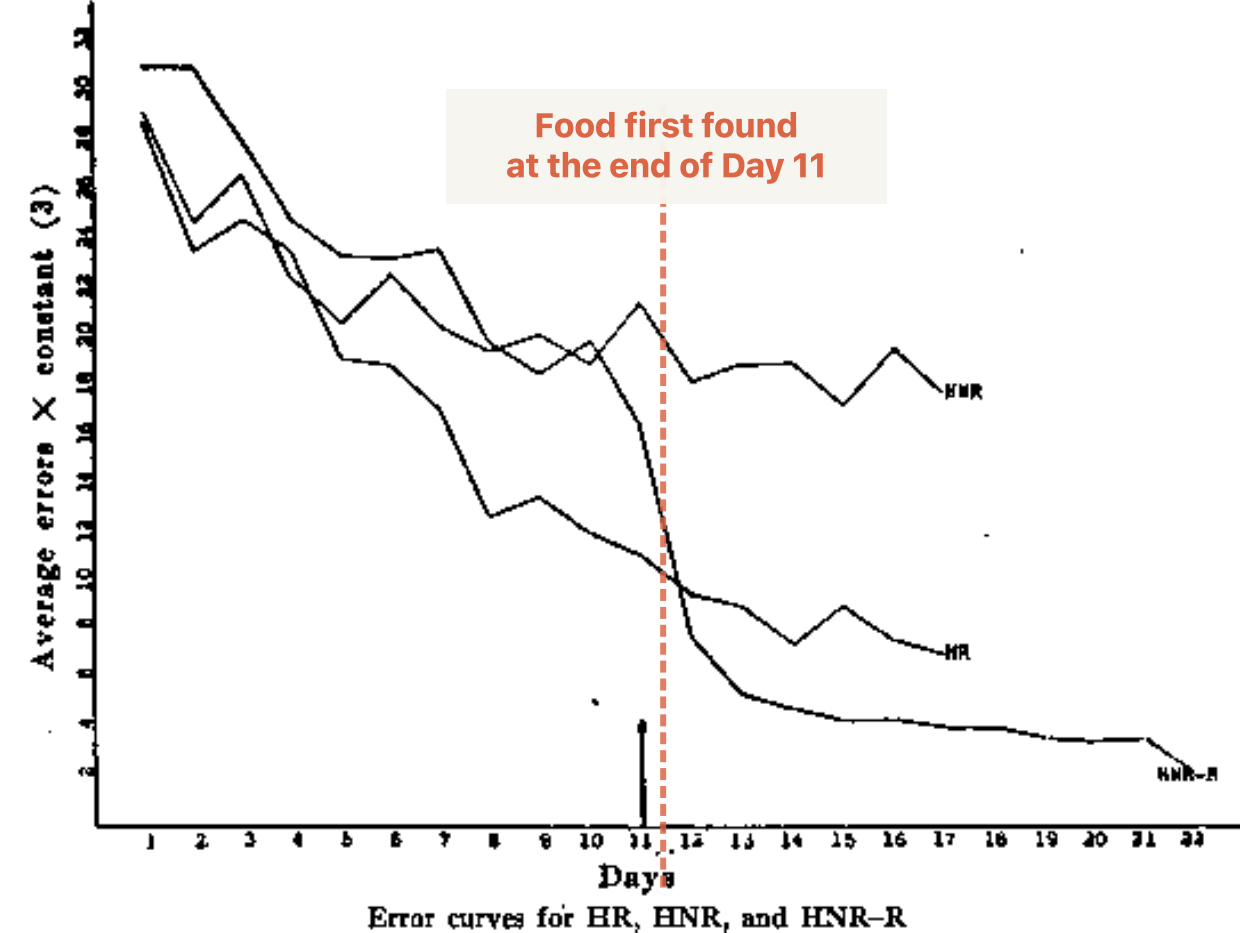


FIG. 6

(From E. C. Tolman and C. H. Honzik, Introduction and removal of reward, and maze performance in rats. *Univ. Calif. Publ. Psychol.*, 1930, 4, No. 19, p. 267.)

HRreward from start · HNRno reward · HNR-Rreward from Day 11

1 · EXPLORE

No food reward for ten days

The delayed-reward group runs the same maze, but its error curve remains high.

2 · REWARD ARRIVES

Food appears on Day 11

The rats discover food only after completing that day's run.

3 · LEARNING BECOMES VISIBLE

Errors collapse on Day 12

Performance rapidly approaches the group rewarded from the start.

Learning $\neq$ current performance

Reward changed what behavior revealed; useful maze knowledge was likely acquired earlier.

The First Era Contributed **Purpose + Representation**

PURPOSE

Rehearse before acting

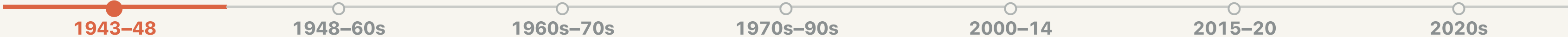
Use an internal surrogate to compare alternatives without trying every one in the world.

REPRESENTATION

Keep useful structure

Preserve relationships that support flexible prediction and behavior—not every visible detail.

A world model is judged by what it enables, not by how literally it copies the world.



Cybernetics — Control in a Feedback Loop

Act, measure what happened, correct the next action, and repeat.

Wiener put communication and control in animals and machines into one shared language of feedback.



IV Feedback and Oscillation

A patient comes into a neurological clinic. He is not paralyzed, and he can move his legs when he receives the order. Nevertheless, he suffers under a severe disability. He walks with a peculiar uncertain gait, with eyes downcast on the ground and on his legs. He starts each step with a kick, throwing each leg in succession in front of him. If blindfolded, he cannot stand up, and totters to the ground. What is the matter with him?

Another patient comes in. While he sits at rest in his chair, there seems to be nothing wrong with him. However, offer him a cigarette, and he will swing his hand past it in trying to pick it up. This will be followed by an equally futile swing in the other direction, and this by still a third swing back, until his motion becomes nothing but a futile and violent oscillation. Give him a glass of water, and he will empty it in these swings before he is able to bring it to his mouth. What is the matter with him?

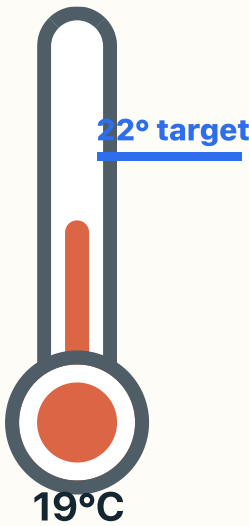
Both of these patients are suffering from one form or another of what is known as *ataxia*. Their muscles are strong and healthy enough, but they are unable to organize their actions. The first patient suffers from *tabes dorsalis*. The part of the spinal cord which ordinarily receives sensations has been damaged or

Wiener · *Cybernetics* · [Open Chapter IV ↗](#)

Feedback Is **Not Yet** a World Model

FEEDBACK CONTROLLER

Correct after an error



- 1 Set target: 22°C
- 2 Measure now: 19°C
- 3 Current error → heater ON

It can react to the present error without simulating several future temperature trajectories.

WORLD MODEL

Compare before acting

Current
state

Action A → Predicted future A

Action B → Predicted future B

It adds an internal prediction of consequences under candidate actions.

A feedback controller may be purely reactive. A world model adds a way to anticipate consequences. · [Wiener's thermostat example, p. 131 ↗](#)

1943–48

1948–60s

1960s–70s

1970s–90s

2000–14

2015–20

2020s

Bellman Connected Dynamics to Long-Term Decisions

One action changes both the payoff now and the distribution of future states.

$$f_N(i) = \max_q \left[\underbrace{b_i(q)}_{\text{now}} + \sum_{j=1}^M \underbrace{a_{ij}(q)}_{\text{dynamics}} \underbrace{f_{N-1}(j)}_{\text{future}} \right]$$

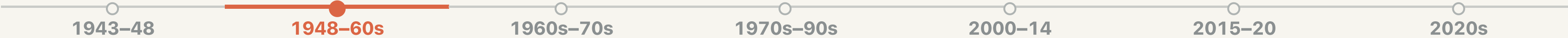
WORLD-MODEL PIECE

$$a_{ij}(q) = P(s_{t+1} = j \mid s_t = i, a_t = q)$$

CONCRETE CONTEXT: EQUIPMENT REPLACEMENT

state = current machine condition · **action** = keep or replace · the choice changes today's return and tomorrow's condition.

Historical bridge · A known transition model can be used to compare consequences across multiple stages. Bellman did not explain how to learn that model from pixels.



A Markovian Decision Process

RICHARD BELLMAN

1. Introduction. The purpose of this paper is to discuss the asymptotic behavior of the sequence $\{f_N(i)\}$, $i = 1, 2, \dots, M$, $N = 1, 2, \dots$, generated by the non-linear recurrence relations

(1) $f_N(i) = \text{Max}_q \left[b_i(q) + \sum_{j=1}^M a_{ij}(q) f_{N-1}(j) \right]$, $N = 1, 2, \dots$,
 $f_0(i) = c_i$, $i = 1, 2, \dots, M$.

Although these equations are non-linear, they possess certain quasi-linear properties which permit a more thorough discussion than might be imagined upon first glance.

As we shall discuss below, this question arises from the consideration of a dynamic programming process. A related process gave rise to an equation of the above form which was discussed in [1], under a particular set of assumptions concerning the functions $b_i(\alpha)$ and the matrices $A(\alpha) = (a_{ij}(\alpha))$. Here we shall

Original recurrence · [Open source page ↗](#)

Observation Is Not State — Predict, Then Correct

DYNAMICS · PREDICT

$$x_{t+1} = Ax_t + Bu_t + w_t$$

SENSOR · OBSERVE

$$o_t = Cx_t + v_t$$

x_t

latent dynamic state

u_t

control input

o_t

noisy, incomplete observation

w_t, v_t

process and measurement noise

A single image may show position while hiding velocity, friction, contact, or an occluded cause.

OPTIONAL · ABOUT 60 SECONDS

Visual state-and-measurement intuition ↗

Play roughly 0:12–1:12

Predict with dynamics → Correct with the new observation

1943–48

1948–60s

1960s–70s

1970s–90s

2000–14

2015–20

2020s

Feedback, State, Uncertainty — Three New Ingredients

Prediction now sits inside interaction and must cope with incomplete evidence.

01 Closed loop

Actions change the observations that arrive next.

02 Predictive state

The current measurement may not be sufficient.

03 Uncertainty

A useful estimate carries confidence, not only a point guess.

distributions and linear dynamics are natural, mutually plausible assumptions particularly when the statistical data are scant.

How is a dynamic system (linear or nonlinear) described? The fundamental concept is the notion of the state. By this is meant, intuitively, some quantitative information (a set of numbers, a function, etc.) which is the least amount of data one has to know about the past behavior of the system in order to predict its future behavior. The dynamics is then described in terms of state transitions, i.e., one must specify how one state is transformed into another as time passes.

A linear dynamic system may be described in general by the vector differential equation

$$\left. \begin{aligned} \frac{dx}{dt} &= F(t)x + D(t)u(t) \\ \text{and} \\ y(t) &= M(t)x(t) \end{aligned} \right\} \quad (12)$$

where x is an n -vector, the state of the system (the components x_i of x are called state variables); $u(t)$ is an m -vector ($m \leq n$) representing the inputs to the system; $F(t)$ and $D(t)$ are $n \times n$, respectively, $n \times m$ matrices. If all coefficients of $F(t)$, $D(t)$, $M(t)$ are constants, we say that the dynamic system (12) is time-invariant or stationary. Finally, $y(t)$ is a p -vector denoting the outputs of the system; $M(t)$ is an $n \times p$ matrix; $p \leq n$.

The physical interpretation of (12) has been discussed in detail elsewhere [18, 20, 23]. A look at the block diagram in Fig. 1 may be helpful. This is not an ordinary but a matrix block diagram (as revealed by the fat lines indicating signal flow). The integrator in

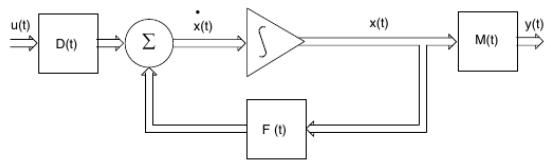
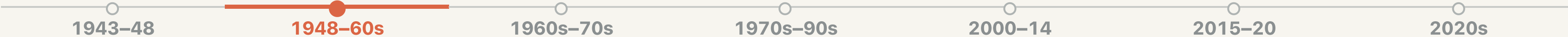


Fig 1. Matrix block diagram of the general linear continuous-dynamic

Kalman, 1960: state carries forward what prediction needs

These ideas will later meet explicit action rules, learned dynamics, and imagined experience.



Good Regulator Theorem — What “Model” Means Here

The simplest optimal regulator must preserve the system distinctions that change successful regulation.

Example	if hot and cold require different actions, the regulator must distinguish hot from cold
“Model” here	a task-relevant mapping from system situations to regulating responses
Not implied	a full simulator, neural module, complete world copy, or arbitrary rollout engine

The formal statement · [Open original paper ↗](#)

We consider the regulatory situation described earlier, in which the set of regulatory events R and the set of events S in the rest of the system (i.e. in the ‘reguland’, S , which we view as R ’s opponent) jointly determine, through a mapping ψ , the outcome events Z . By all optimal regulator we will mean a regulator which produces regulatory events in such a way that $H(Z)$ is minimal. Then under very broad conditions stated in the proof below, the following theorem holds:

Theorem: The simplest optimal regulator R of a reguland S produces events R which are related to the events S by a mapping $h : S \rightarrow R$.

Restated somewhat less rigorously, the theorem says that the best regulator of a system is one which is a model of that system in the sense that the regulator’s actions are merely the system’s actions as seen through a mapping h . The type of isomorphism here is that expressed (in the form used above) by

$$\exists h : \forall i : \rho(i) = h[\sigma(i)] \tag{8}$$

where ρ and σ are the mappings that R and S impose on their common input I . This form is essentially that of (5) above,

Proof: The sets R , S , and Z and the mapping $\Psi : R \times S \rightarrow Z$ are presumed given. We will assume that over the set S there exists a probability distribution $p(S)$ which gives the relative frequencies of the events in S . We will further assume that the behaviour of any particular regulator R is specified by a conditional distribution $p(R|S)$ giving, for each event in S , a distribution on the regulatory events in R . Now $p(S)$ and $p(R|S)$ jointly determine $p(R, S)$ and hence $p(Z)$ and $H(Z)$, the entropy in the set of outcomes. ($H(Z) = - \sum_{z_i \in Z} p(z_i) \log p(z_i)$.) With $p(S)$ fixed, the class of optimal regulators therefore

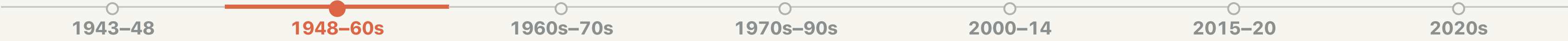
corresponds to the class of optimal distributions $p(R|S)$ for which $H(Z)$ is minimal. We will call this class of optimal distributions π .

It is possible for there to be very different distributions $p(Z)$ all having the same minimal entropy $H(Z)$. To consider that possibility would merely complicate this proof without affecting it in any essential way, so we will suppose that every $p(R|S)$ in π determines, with $p(S)$ and ψ , the same (unique) $p(Z)$. We now select for examination an arbitrary $p(R|S)$ from π .

The heart of the proof is the following lemma:

Lemma: $\forall S_j \in S$, the set $\{ \psi(r_i, s_j) : p(r_i, s_j) > 0 \}$ has only one element. That is, for every s_j in S , $p(R|s_j)$ is such that all r_i with positive probability map, with s_j under ψ to the same z_k in Z .

Proof of lemma: Suppose, to the contrary, that $p(r_1|s_j) > 0$, $p(r_2|s_j) > 0$, $\psi(r_1, s_j) = z_1$, and $\psi(r_2, s_j) = z_2 \neq z_1$. Now $p(r_1, s_j)$ and $p(r_2, s_j)$ contribute to $p(z_1)$ and $p(z_2)$ respectively, and by varying these probabilities (by subtracting Δ from $p(r_1, s_j)$ and adding Δ to $p(r_2, s_j)$) we could vary $p(z_1)$ and $p(z_2)$ and thereby vary $H(Z)$. We could make Δ either positive or negative, whichever would make $p(z_1)$ and $p(z_2)$ more unequal. One of the



Shakey — Planning with Symbols

A goal could be given without prescribing every motor step.

Planning connected an integrated acting system to an explicit symbolic model of action consequences.

SYSTEM

Shakey

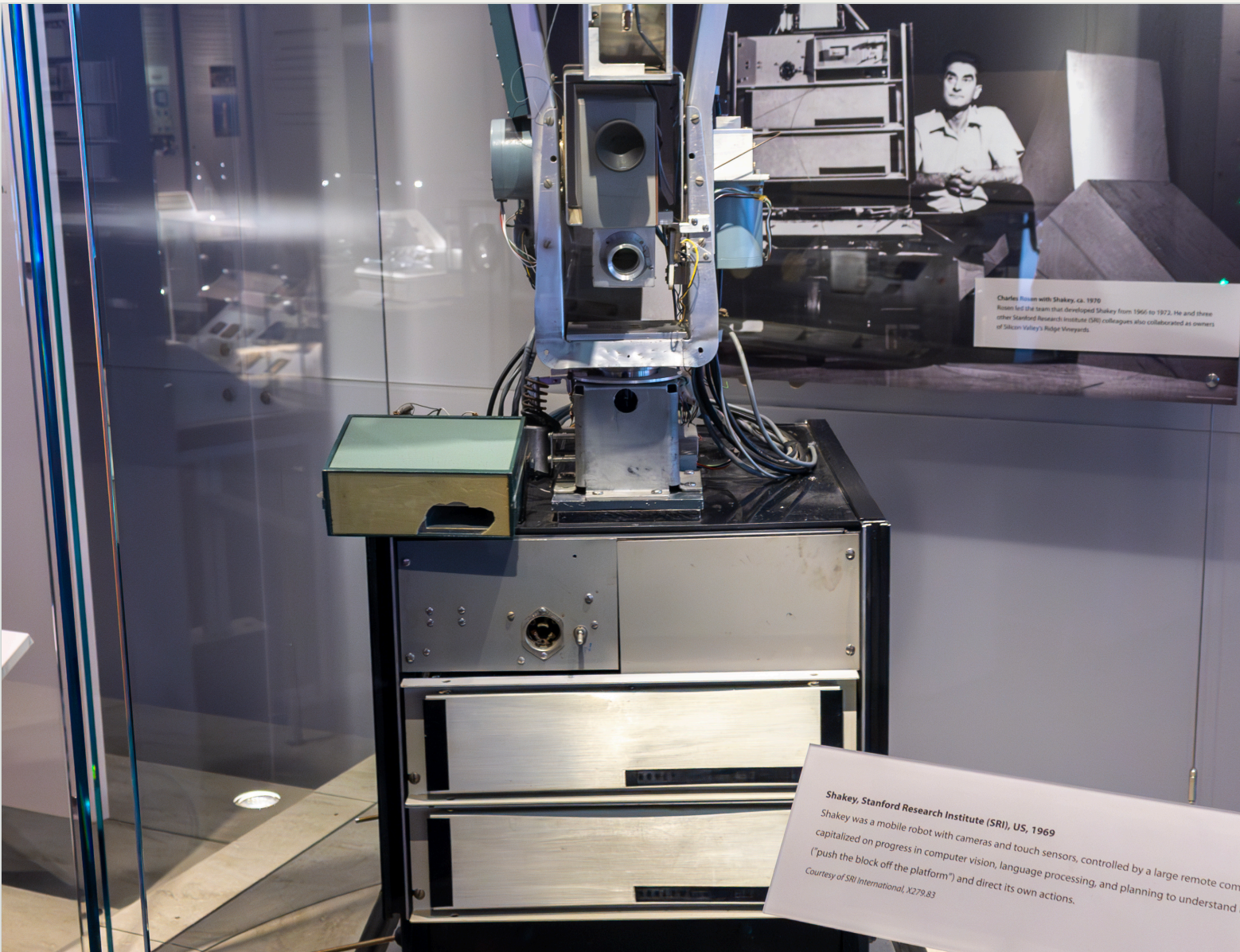
Perception · world description · planning · execution

PLANNING FORMALISM

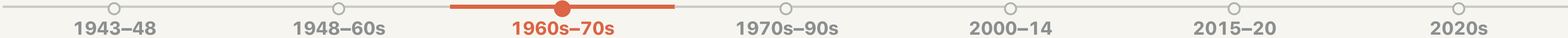
STRIPS

Symbolic facts · action preconditions · action effects

Relationship: STRIPS was developed within the Shakey project and gave its planner an explicit model of how actions change the represented world.



Shakey made symbolic reasoning about action visible



A World State Can Be Symbolic

STRIPS = STanford Research Institute Problem Solver

STATE

Symbolic facts

ROBOT_AT(A)
BOX_AT(A)

ACTION MODEL

Operator

Preconditions say when it applies;
effects say which facts change.

NEXT STATE

Updated facts

ROBOT_AT(B)
BOX_AT(B)

YES—WITH ONE ADDITION. The state is symbolic; the world model also includes rules predicting how actions transform those symbols.

STRIPS: A New Approach to the Application of Theorem Proving to Problem Solving¹

Richard E. Fikes
Nils J. Nilsson
Stanford Research Institute, Menlo Park, California

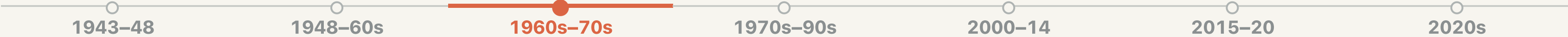
Recommended by B. Raphael

Presented at the 2nd IJCAI, Imperial College, London, England, September 1–3, 1971.

ABSTRACT
We describe a new problem solver called STRIPS that attempts to find a sequence of operators in a space of world models to transform a given initial world model into a model in which a given goal formula can be proven to be true. STRIPS represents a world model as an arbitrary collection of first-order predicate calculus formulas and is designed to work with models consisting of large numbers of formulas. It employs a resolution theorem prover to answer questions of particular models and uses means-ends analysis to guide it to the desired goal-satisfying model.

DESCRIPTIVE TERMS
Problem solving, theorem proving, robot planning, heuristic search.

Fikes & Nilsson, 1971 · [Open original paper ↗](#)



An Operator Predicts Action Effects

PUSH(box, A → B) is an abstract action rule—not a motor command.

PRECONDITIONS

Must already hold

ROBOT_AT(A)
BOX_AT(A)

DELETE

Facts made false

ROBOT_AT(A)
BOX_AT(A)

ADD

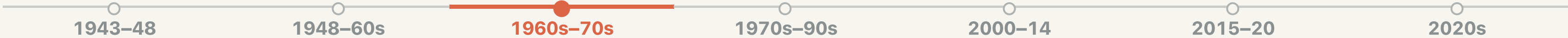
Facts made true

ROBOT_AT(B)
BOX_AT(B)

STATE UPDATE · successor = (current state – delete list) ∪ add list. The model predicts consequences; the planner searches for a sequence that reaches the goal.

```
Delete list: AT(ROBOT($),NEXTTO(ROBOT,$))
Add list: NEXTTO(ROBOT,m)
push(m,n): robot pushes object m next to item n
Precondition:
  PUSHABLE(m) ∧ ONFLOOR ∧ NEXTTO(ROBOT,m) ∧ {(∃x)[INROOM(m,x)
    ∧ INROOM(n,x)] ∨ (∃x,∃y)[INROOM(m,x) ∧ CONNECTS(n,x,y)]}
Delete list: AT(ROBOT($)) NEXTTO(ROBOT $) NEXTTO($,m)
             AT(m$)          NEXTTO(m$)
Add list: NEXTTO(m,n)
          NEXTTO(n,m)
          NEXTTO(ROBOT,m)
Artificial Intelligence 2 (1971), 189–208
```

Original push operator · [Open p. 197 ↗](#)



Symbolic Planning vs. System Identification: **Write or Estimate Dynamics**

SYMBOLIC AI

Write the transition rules

Facts and action operators define how one represented situation becomes another.

SYSTEM IDENTIFICATION

Estimate dynamics from data

Applied inputs and measured responses reveal a numerical predictive model.

Fig. 1. A typical CTDIC search tree.

Programmed operators transform explicit world models

Fig. 1. Identification of a process in a closed loop.

An attempt to identify H_p from measurements of y will give

Measured inputs and outputs identify numerical dynamics

System Identification — Learn Dynamics from Input–Output Data

Unknown dynamics need not be hand-written: intervene, record the response, fit, and test.

01

Choose a model class

State what kind of dynamics can be represented.

02

Design informative inputs

Excite the system and record input–output trajectories.

03

Fit the dynamics

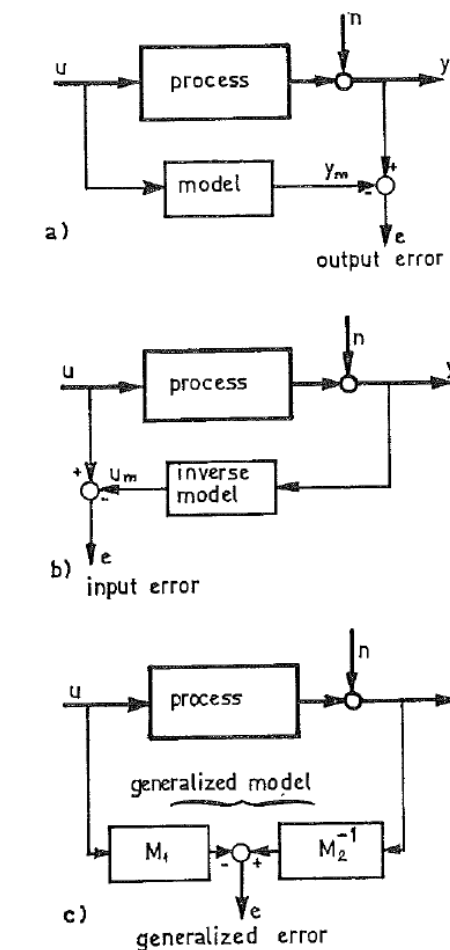
Estimate unknown parameters from measured responses.

04

Validate on new data

Test prediction beyond the trajectories used for fitting.

HISTORICAL POINT · Neural world models scale this recipe to pixels, tokens, and latents—they did not invent learned dynamics.



Fit against the process · [Open survey ↗](#)

MPC — Choose with a World Model

The world model predicts. MPC uses those predictions to choose the next action.

1 · CANDIDATES

Try several short action paths.

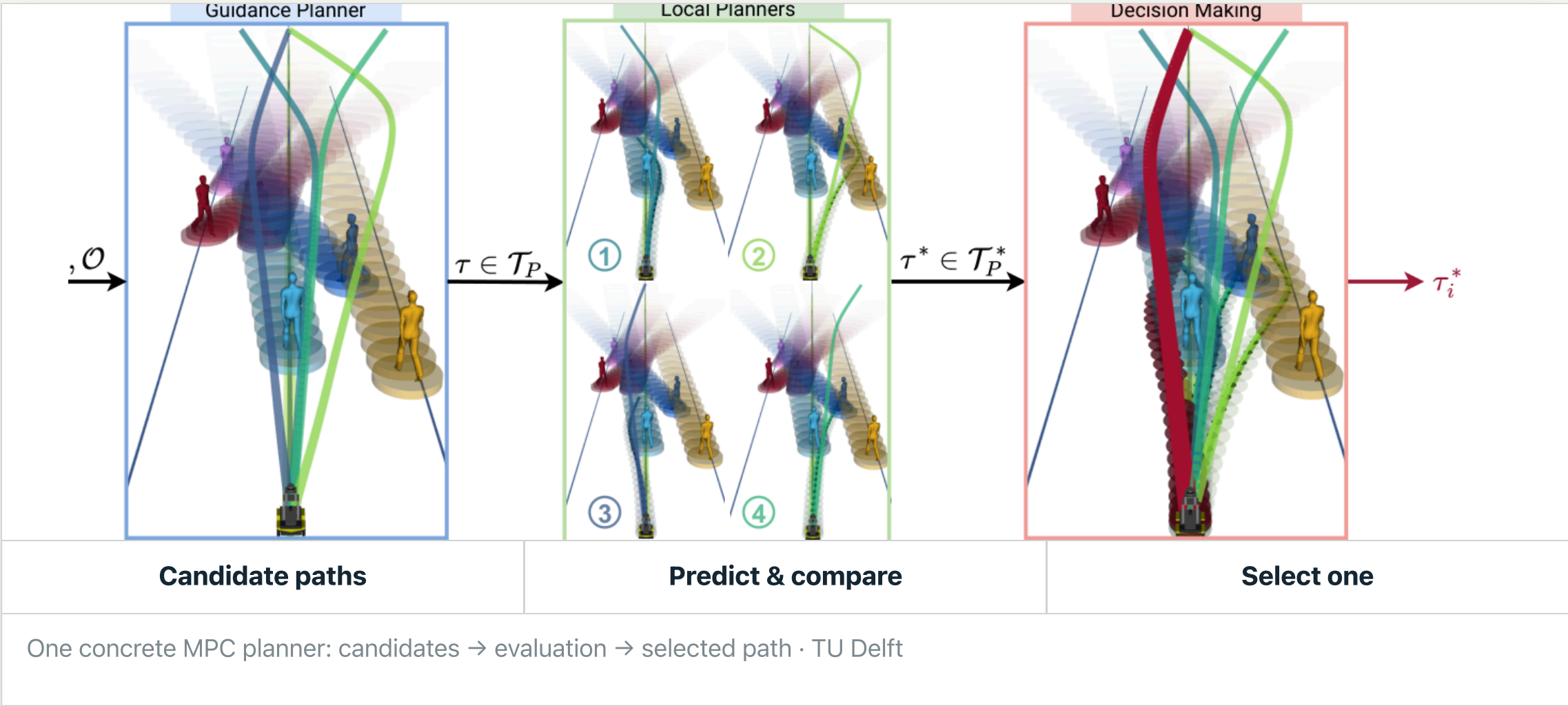
2 · PREDICT & SCORE

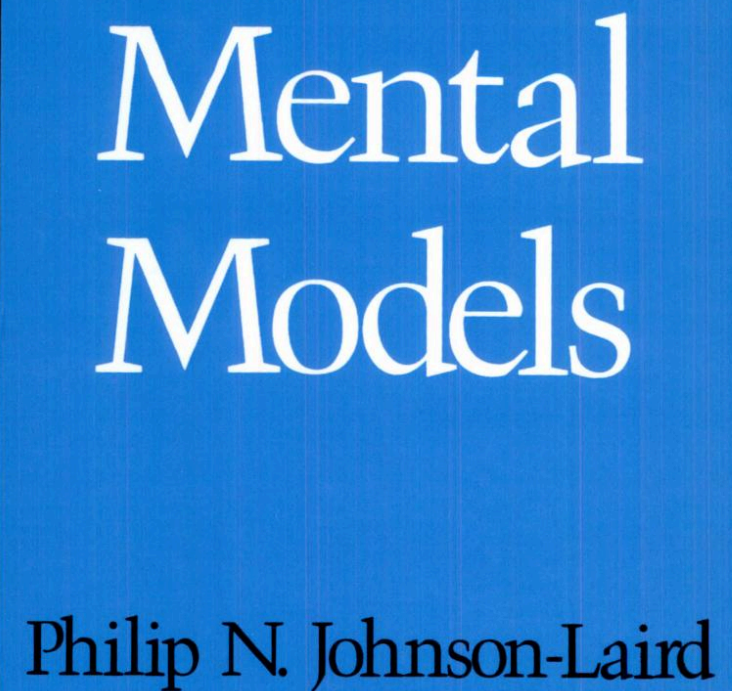
Use the model to compare where each path may lead.

3 · ACT & REPLAN

Choose one path, execute only its first step, then look again.

MPC = model + goal + look-ahead choice + feedback.





Philip Johnson-Laird, *Mental Models*, 1983

Reasoning — Compare Possible Situations

A conclusion is necessary only if it survives every situation allowed by the premises.

One counterexample can refute it. These possibilities are not yet probabilities, temporal dynamics, or action-conditioned futures.

1943–48

1948–60s

1960s–70s

1970s–90s

2000–14

2015–20

2020s

Neural System Identification: Learn a Nonlinear Plant

Unknown nonlinear dynamics became something to identify from input–output experience.

INPUTS

Intervene

Apply commands to an unknown dynamical system.

RESPONSES

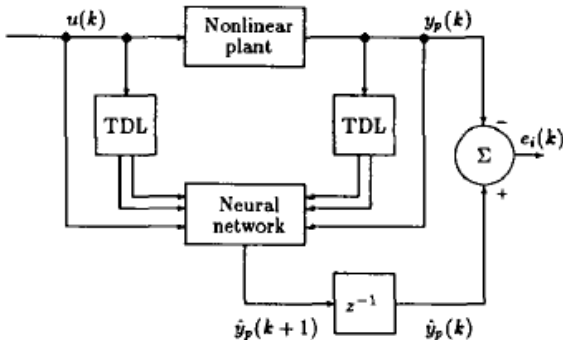
Observe

Measure how the physical plant actually changes.

SURROGATE

Learn

Fit a neural model that predicts the response.

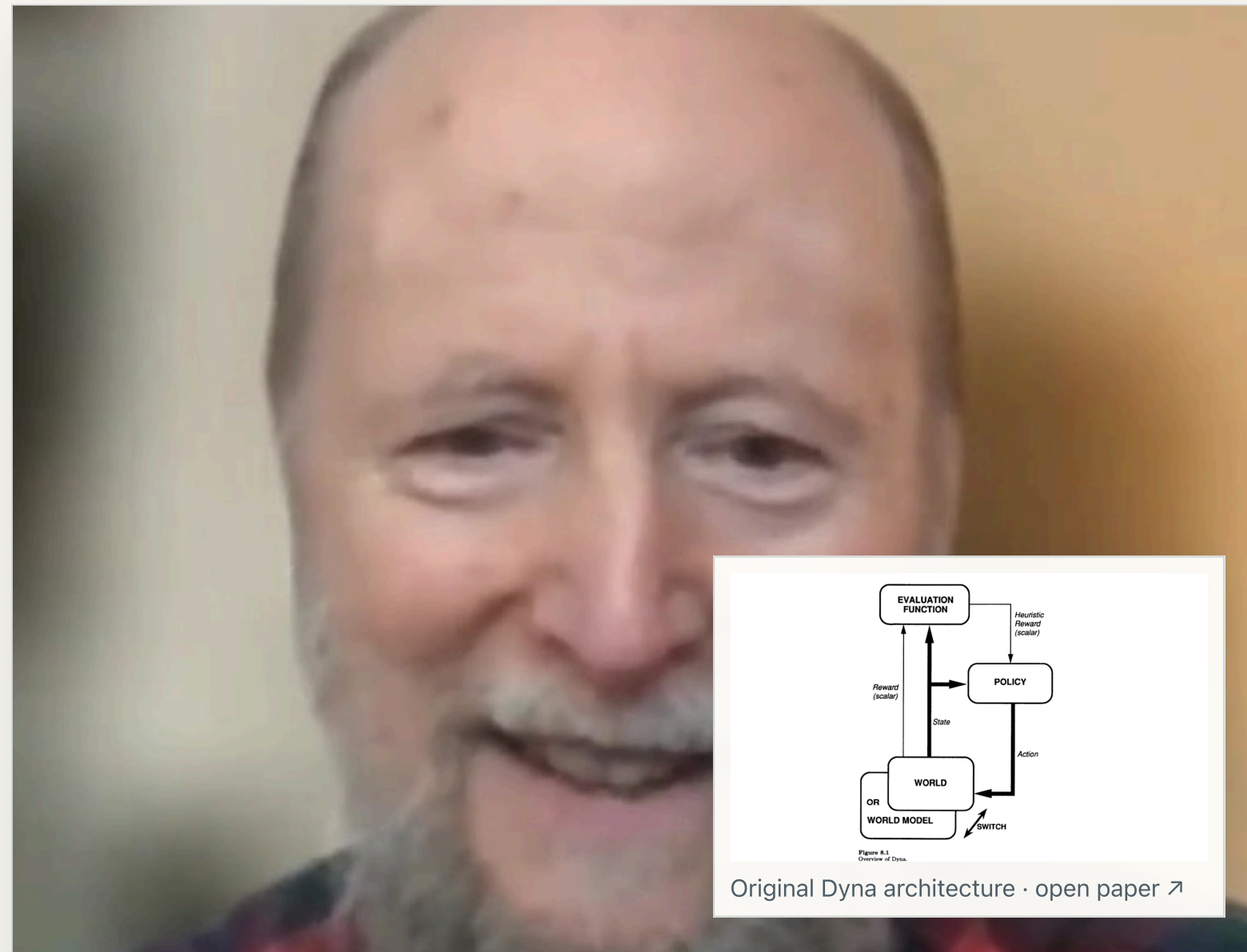


g. 11. Identification of nonlinear plants using neural networks.

A neural surrogate learns to match the nonlinear plant's response

This was neural and adaptive control—not yet the modern deep-learning era.





Richard S. Sutton

Dyna: A Learned Model Could Support **Imagination**

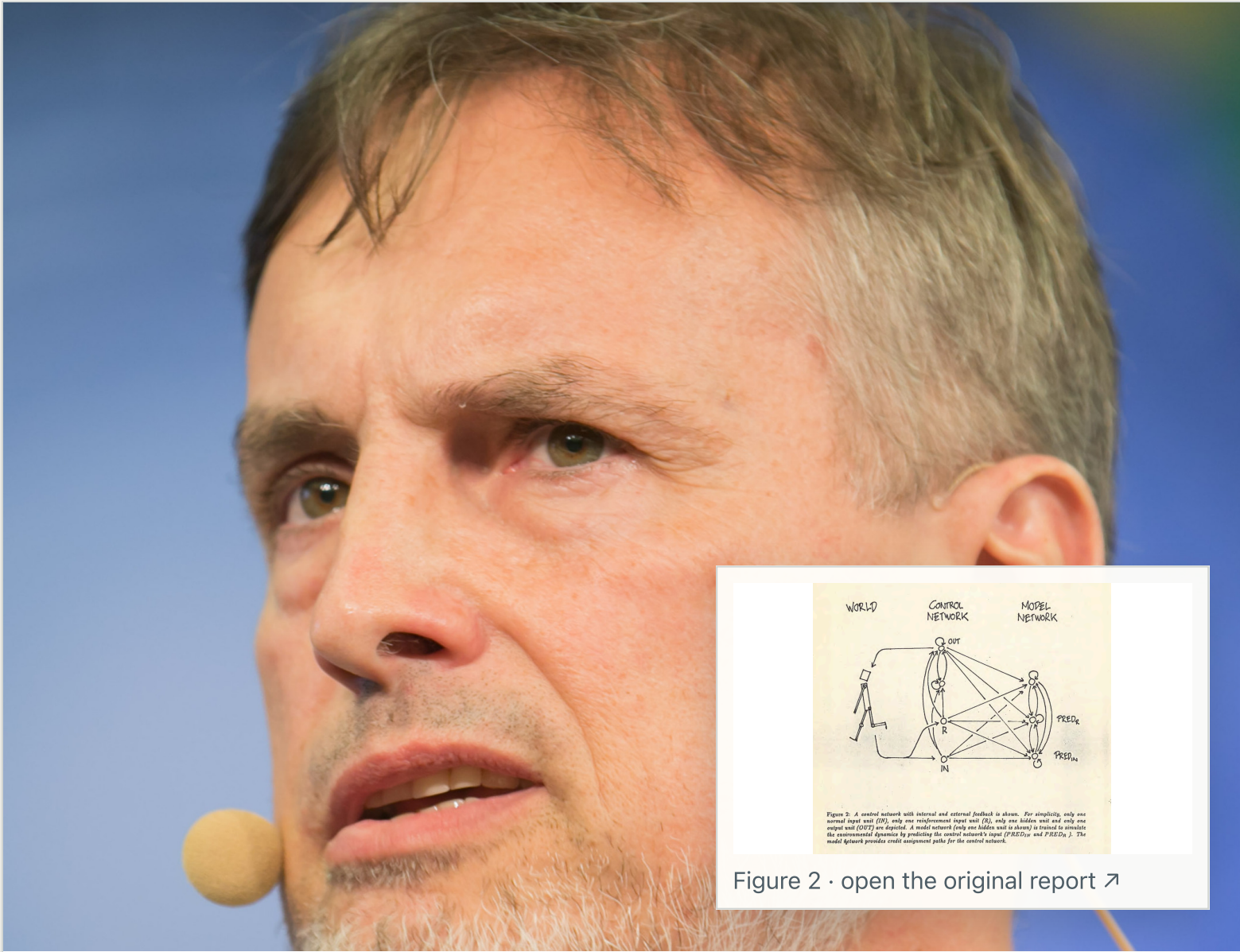
The model became something an agent could use internally—not only something to fit.

Dyna connected experience from the world with hypothetical experience produced by an internal model.

HISTORICAL TAKEAWAY

Models entered the learning loop.

We return to planning mechanics and Dyna algorithms later in the course.



Jürgen Schmidhuber

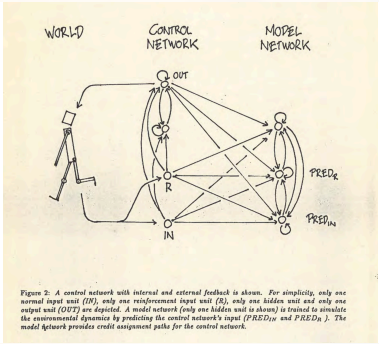


Figure 2 · open the original report ↗

Making the World Differentiable

A learned recurrent environment model became a surrogate for training a recurrent controller.

- | | |
|------------------------|---|
| 01 · Controller | produces an action |
| 02 · Environment model | predicts sensory input and reinforcement |
| 03 · Backpropagate | task error passes through the frozen model to update the controller |

The real world need not be differentiable; the learned surrogate provides an approximate learning path.

Forward and Inverse Models Ask Opposite Questions

Use the same arm example; change which variable is given and which must be predicted.

FORWARD MODEL · CONSEQUENCE PREDICTOR

"What will this command do?"

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

GIVEN
current arm state x_t

GIVEN
motor command a_t

predict next hand pose $\hat{y}_{t+1}$

INVERSE MODEL · ACTION SELECTOR

"Which command reaches this target?"

$$a_t = g_{\phi}(x_t, y^{\star})$$

GIVEN
current arm state x_t

GIVEN
desired hand pose $y^{\star}$

produce motor command a_t

World-model role

A forward model is the core predictive direction of an action-conditioned world model—but it may be much narrower.

Controller role

The inverse model selects an action and may be ambiguous when several commands reach the same target.



A Forward Model Can Teach a Controller from a Distant Outcome

Distal teacher: the target says where the hand should end up—not which motor command was correct.

GOAL

desired outcome $y^\star$

CONTROLLER

$a = g_\phi(x, y^\star)$

FROZEN FORWARD MODEL

$\hat{y} = f_\theta(x, a)$

OUTCOME ERROR

$\mathcal{L} = \|\hat{y} - y^\star\|^2$

ERROR RETURNS THROUGH THE LEARNED MODEL

$$\frac{\partial \mathcal{L}}{\partial a} = \left(\frac{\partial \hat{y}}{\partial a} \right)^\top \frac{\partial \mathcal{L}}{\partial \hat{y}}$$

The forward model's Jacobian converts outcome error into an action-learning signal.

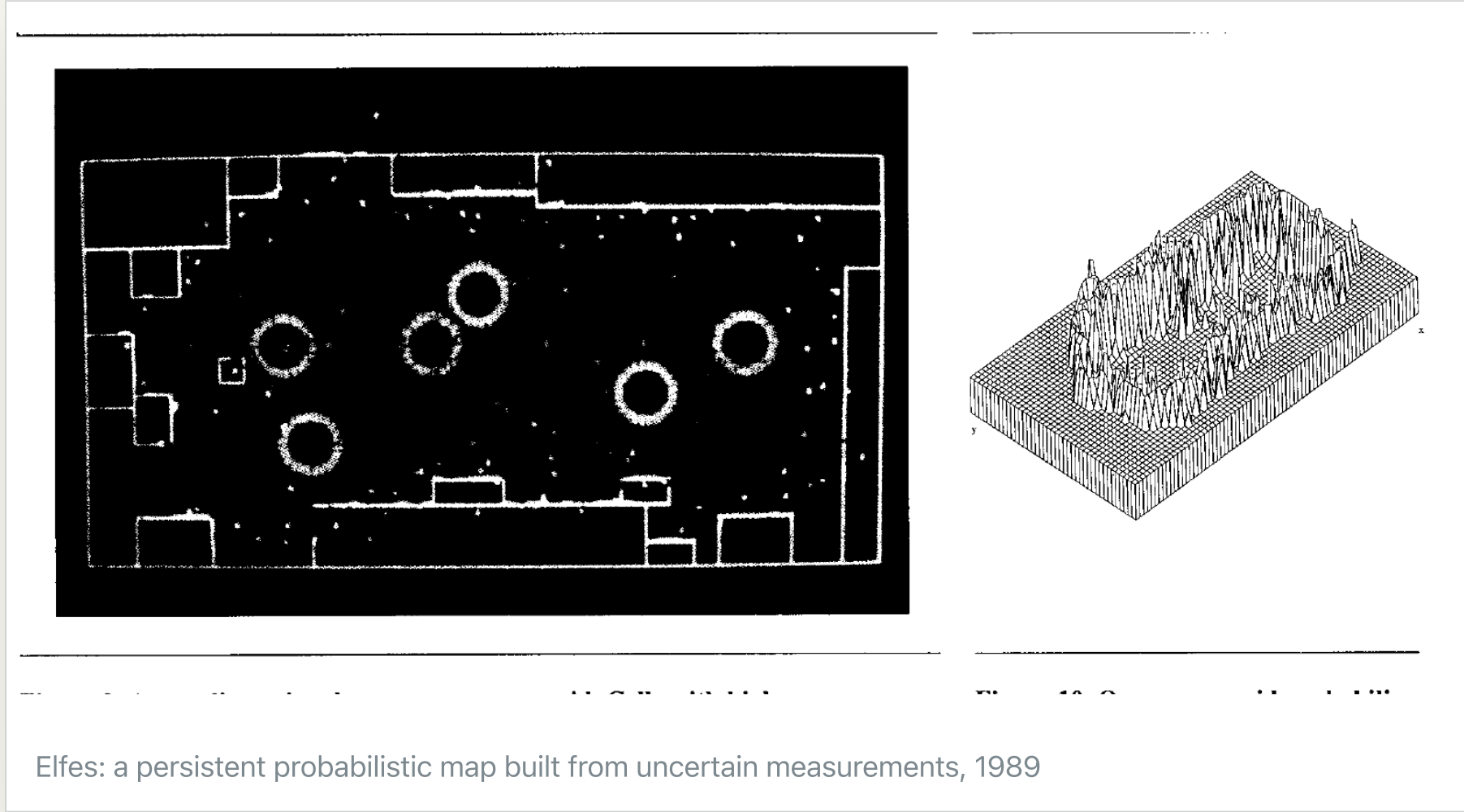
312 JORDAN AND RUMELHART

Figure 3. Learning the forward model using the prediction error $y[n-1] - \hat{y}[n]$.

First learn the forward map · [Open paper ↗](#)

What this is—and is not

A learned, differentiable credit-assignment path for a relatively low-dimensional motor task; not yet a general high-dimensional generative world model.



Occupancy Grids & SLAM — Build a Persistent Spatial State

Fuse partial, noisy views into a map that survives after the camera moves.

- Occupancy grids represent free, occupied, and uncertain space
- SLAM estimate the agent’s pose while updating the map

Early maps were often 2D. The enduring idea is a world-centered spatial state that can support localization, prediction, and planning.

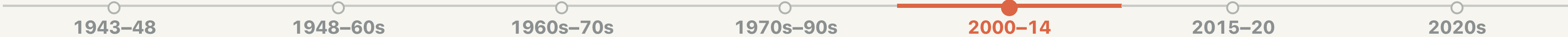
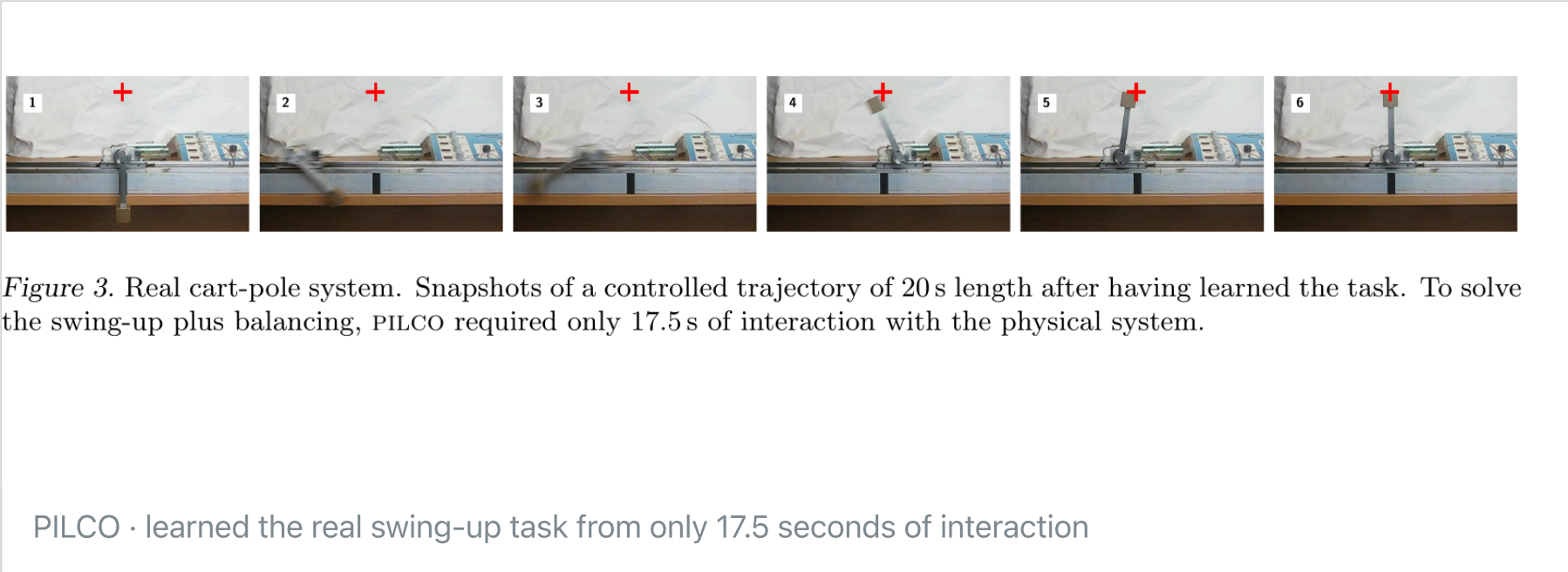


PILCO — Learn Dynamics, Then Improve the Policy

Use a probabilistic learned model to get more value from every real trial.

Learn dynamics	fit a Gaussian process that predicts state change + uncertainty
Imagine rollouts	propagate a distribution—not one guessed trajectory
Improve policy	minimize expected long-horizon cost, then test in reality

What changed from Dyna: uncertainty was carried through continuous imagined futures and used to optimize the policy. The experiments were still low-dimensional—not pixels.



Deep Predictive Models — Pixels or Latents?

ACTION-CONDITIONED VIDEO

Predict the visible future

Oh et al. generated high-dimensional Atari frames conditioned on the chosen action.

LATENT DYNAMICS

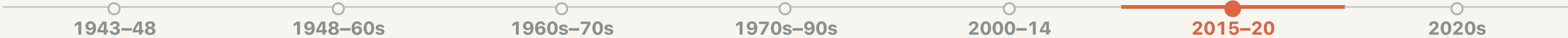
Predict a compact state

E2C encoded raw images and learned action-conditioned transitions in a control-friendly latent space.

Step	MLP	naFf	Feedforward	Recurrent	Ground Truth	Action
255						↑
256						↑

Oh et al.: action-conditioned pixel futures, 2015

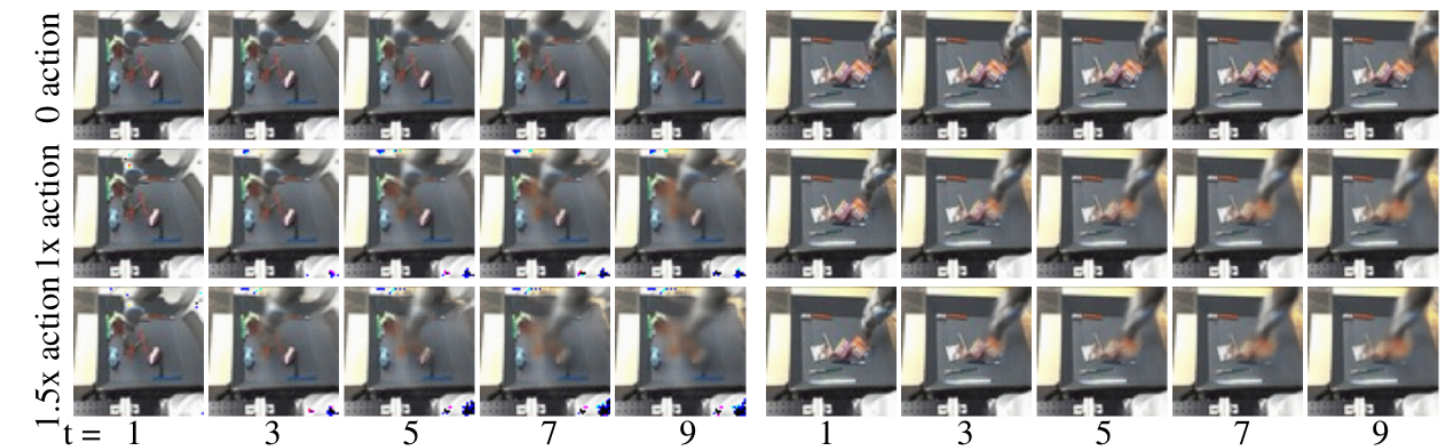
Latent prediction was explicit in 2015—well before JEPA. PlaNet later extended latent dynamics to harder image-based tasks.



Action-Conditioned Video Prediction — Futures Branch with Action

Hold the visual history fixed. Change the action.
The predicted future should change.

Oh showed this in Atari; Finn and colleagues showed it in physical interaction. Robotics is evidence—not the definition.



Finn, Goodfellow & Levine: action-conditioned video, 2016

World Models — Vision, Memory, Controller

Compress what is seen, predict how it evolves, then act through the compact model.

V
Vision

Turn high-dimensional observations into compact features.

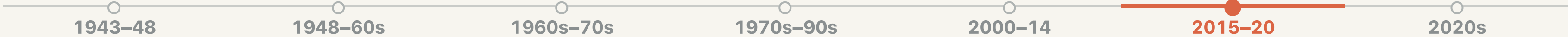
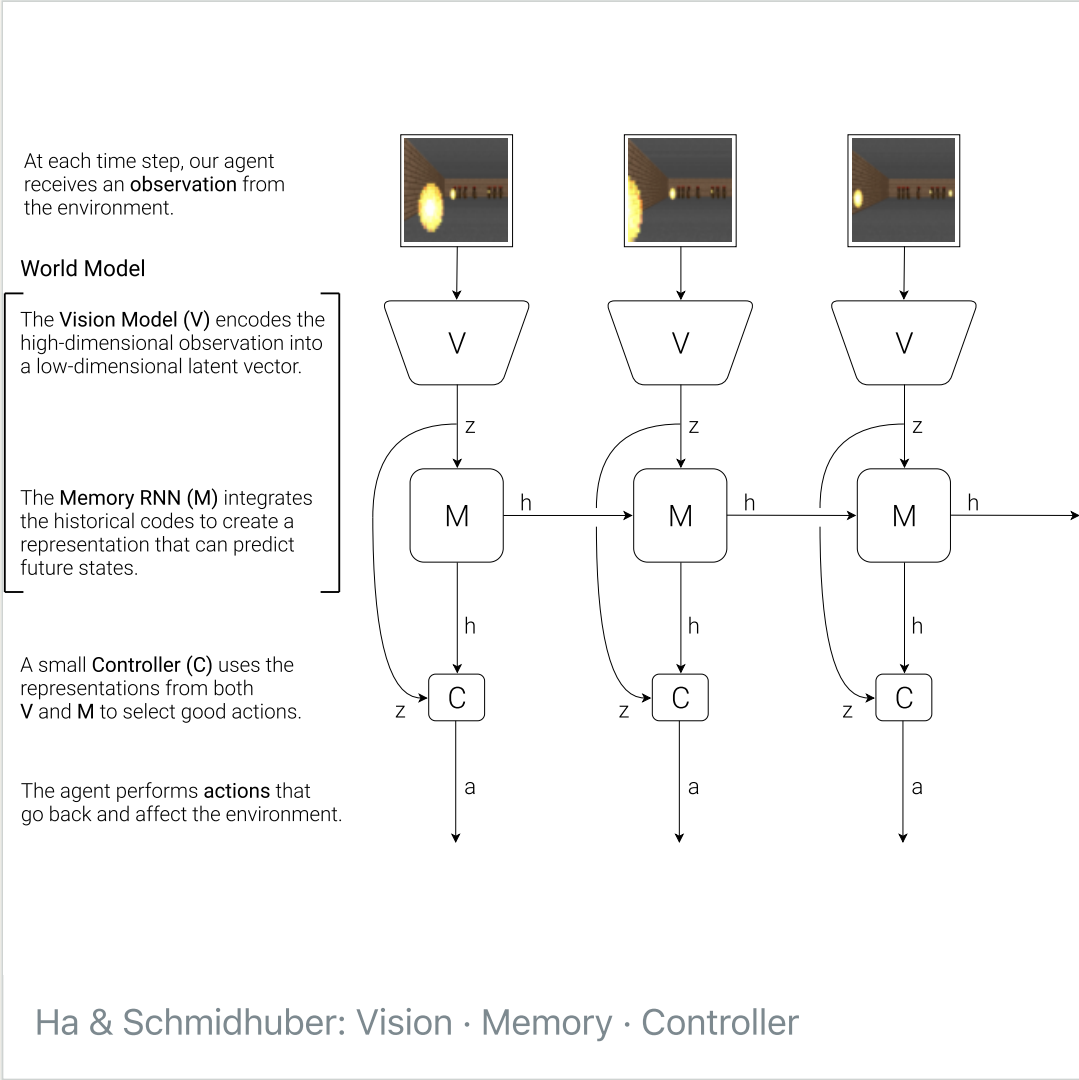
M
Memory

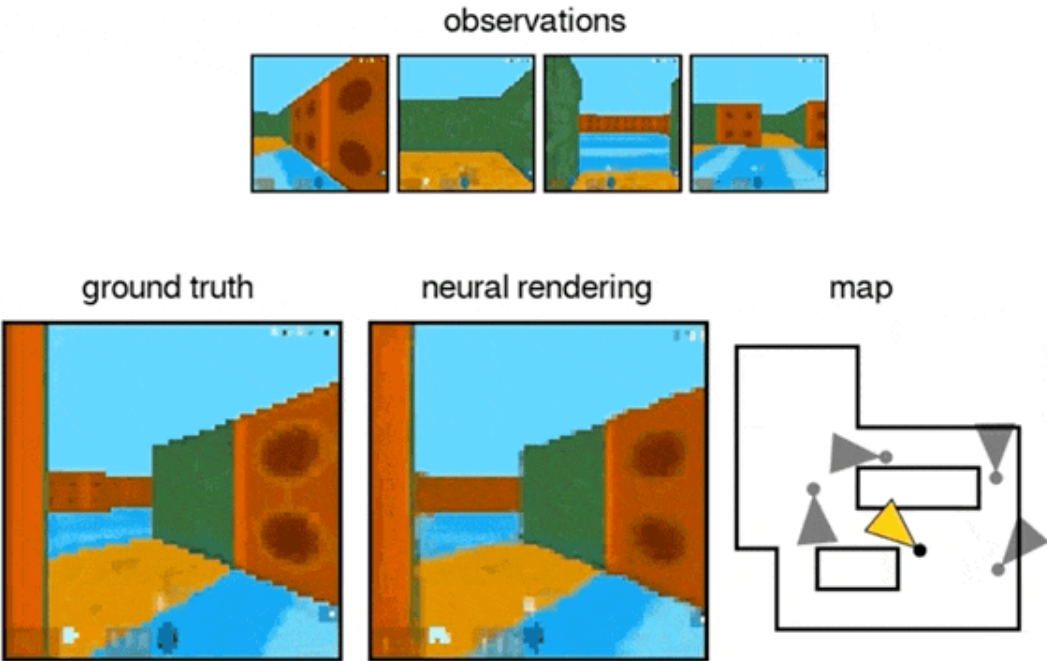
Predict how the internal situation changes over time.

C
Controller

Choose actions using the learned internal situation.

A memorable neural package—not the origin of internal models. Its learned “dream” echoes Dyna’s reuse of imagined experience.





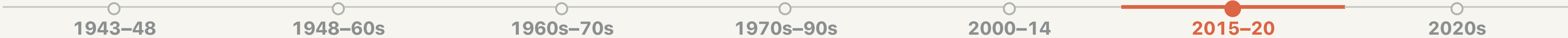
GQN · observations + query viewpoint → imagined unseen view

GQN — Infer a Scene, Then Query a New View

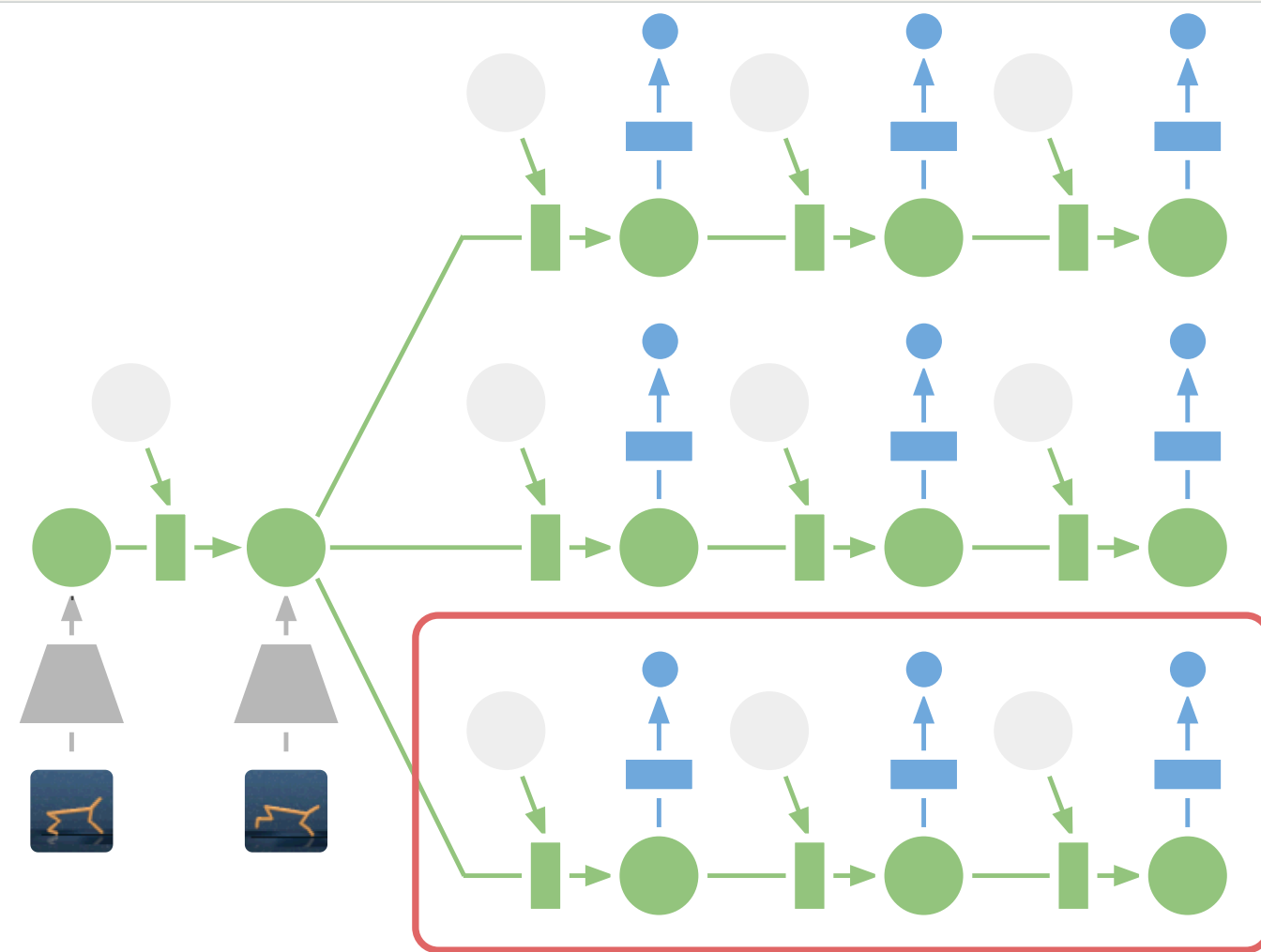
Compress several posed observations into one scene representation.

Observe	a few images and their camera poses
Represent	the scene in a compact learned state
Query	“what should I see from this new viewpoint?”

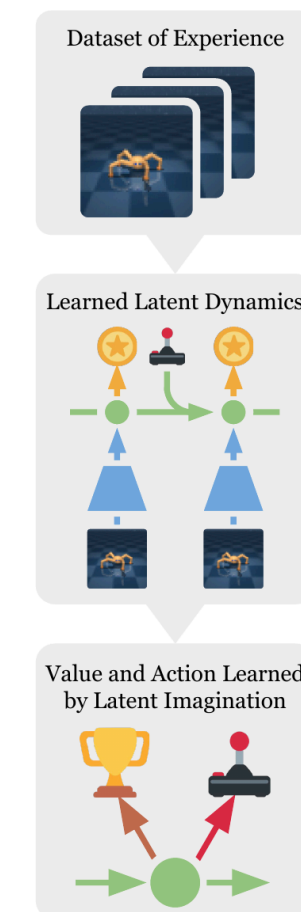
GQN learned geometry-aware scene structure without an explicit hand-designed 3D map—but it did not yet model action-conditioned dynamics.



PlaNet → Dreamer — Plan and Learn in Latent Space



PlaNet (2019) · stochastic latent dynamics support action choice



Dreamer (2020) · behavior learned from imagined latent trajectories

A learned world model became not only a place to look ahead, but a place to train behavior.

1943–48

1948–60s

1960s–70s

1970s–90s

2000–14

2015–20

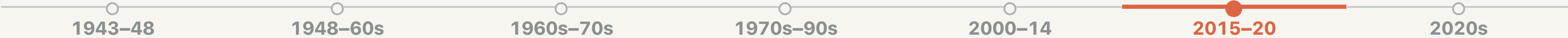
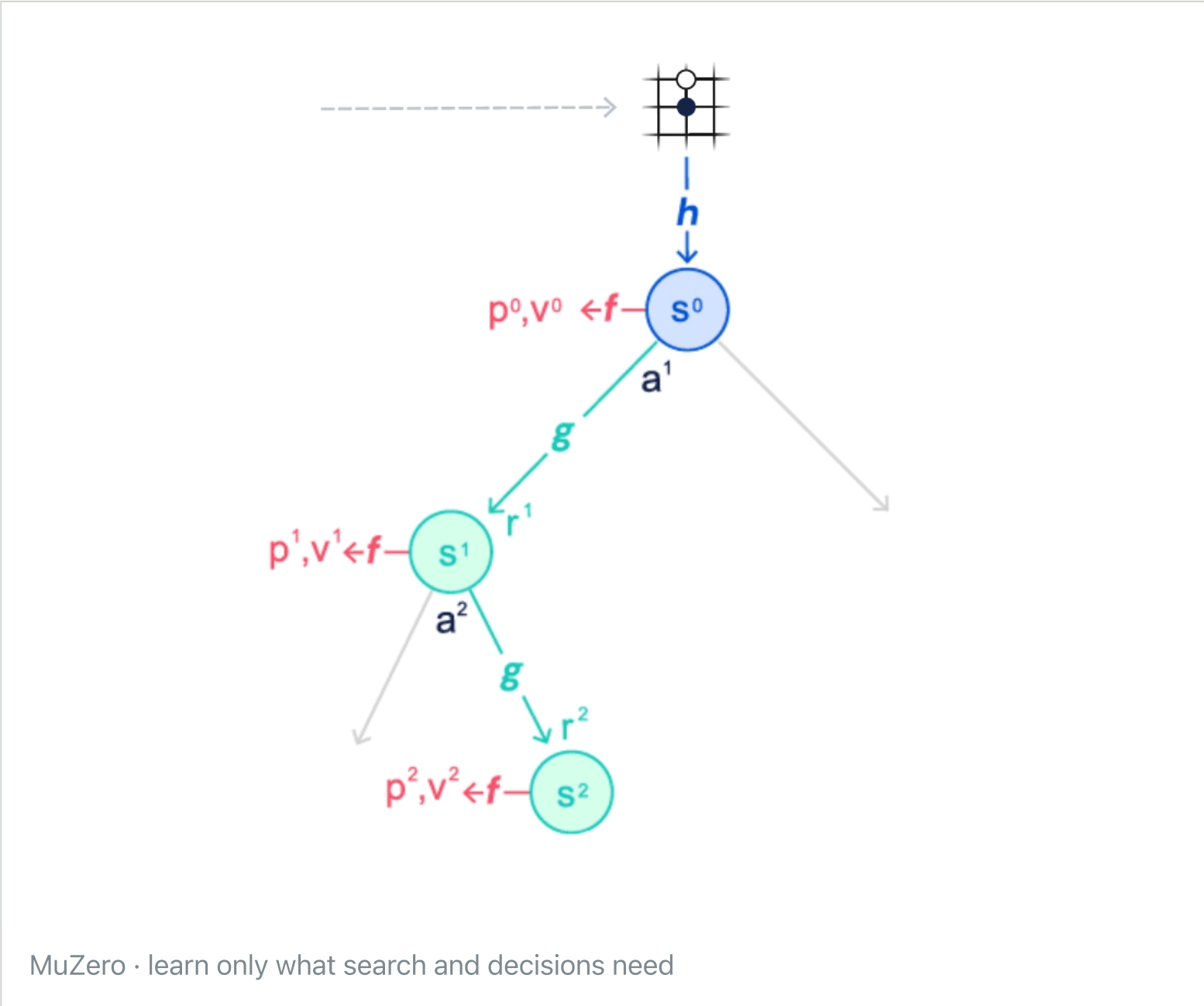
2020s

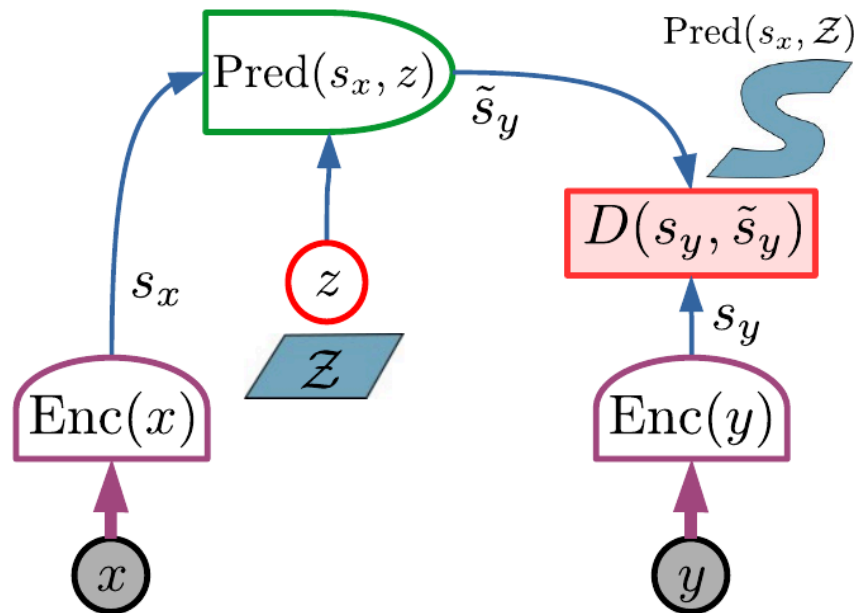
MuZero — Predict What Decisions Need

A useful model need not reconstruct every pixel.

Predict	reward, value, and policy-relevant quantities
Use	the learned dynamics inside look-ahead search
Discard	visible detail that does not affect the decision

MuZero made “task-relevant model” a famous modern design choice—not the only definition of a world model.





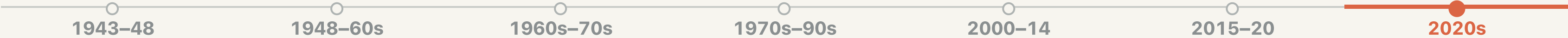
JEPA · predict a representation of the missing target

JEPA — Predict Representations, Not Pixels

Match an abstract target in representation space instead of generating every visible detail.

Context	encode what is available
Predictor	infer the representation of what is missing or future
Target	compare in feature space, not raw pixels

JEPA changes the prediction target and learning objective. It did not originate latent dynamics: action-conditioned latent transitions were explicit by 2015.

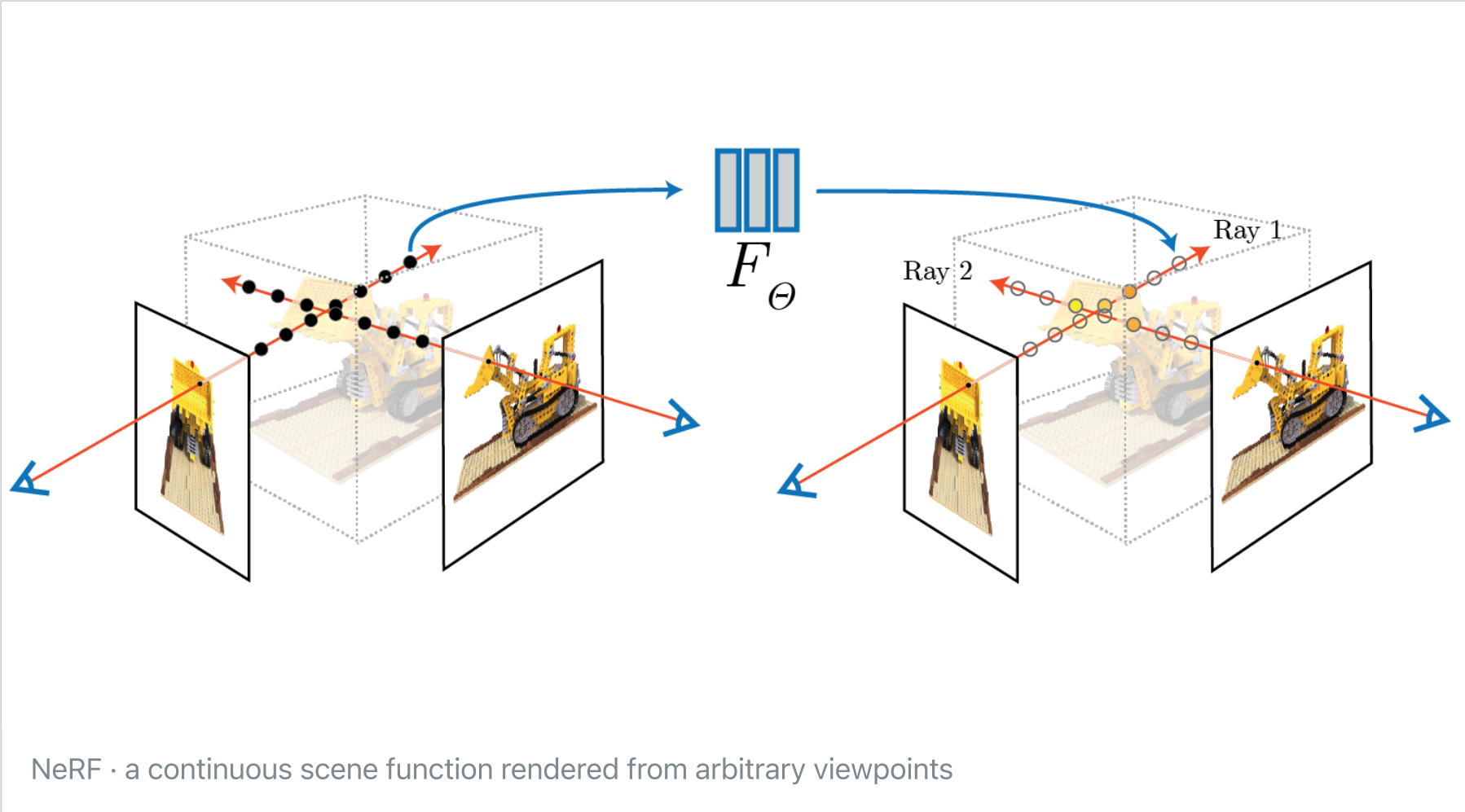


NeRF — Put a Scene Inside a Neural Field

Represent color and density continuously across 3D position and viewing direction.

Fit	a static scene from images with known camera poses
Query	points along a new camera ray
Render	a photorealistic novel view

Important boundary: NeRF is a powerful scene model. By itself, it does not predict how the world changes after an action.



Genie — Interactive Video Worlds

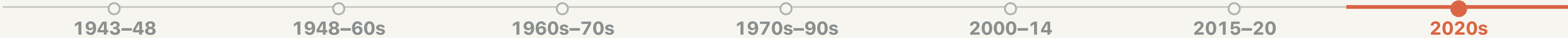
Generate the next sensory experience while the user acts inside it.

Condition on	history, prompt, and the current action
Generate	a rich visual continuation in real time
Maintain	enough consistency to keep interacting

This is neither MuZero’s task-signal target nor JEPA’s feature target. It models a controllable stream of sensory futures.



Genie 3 · generated sensory futures respond to user actions

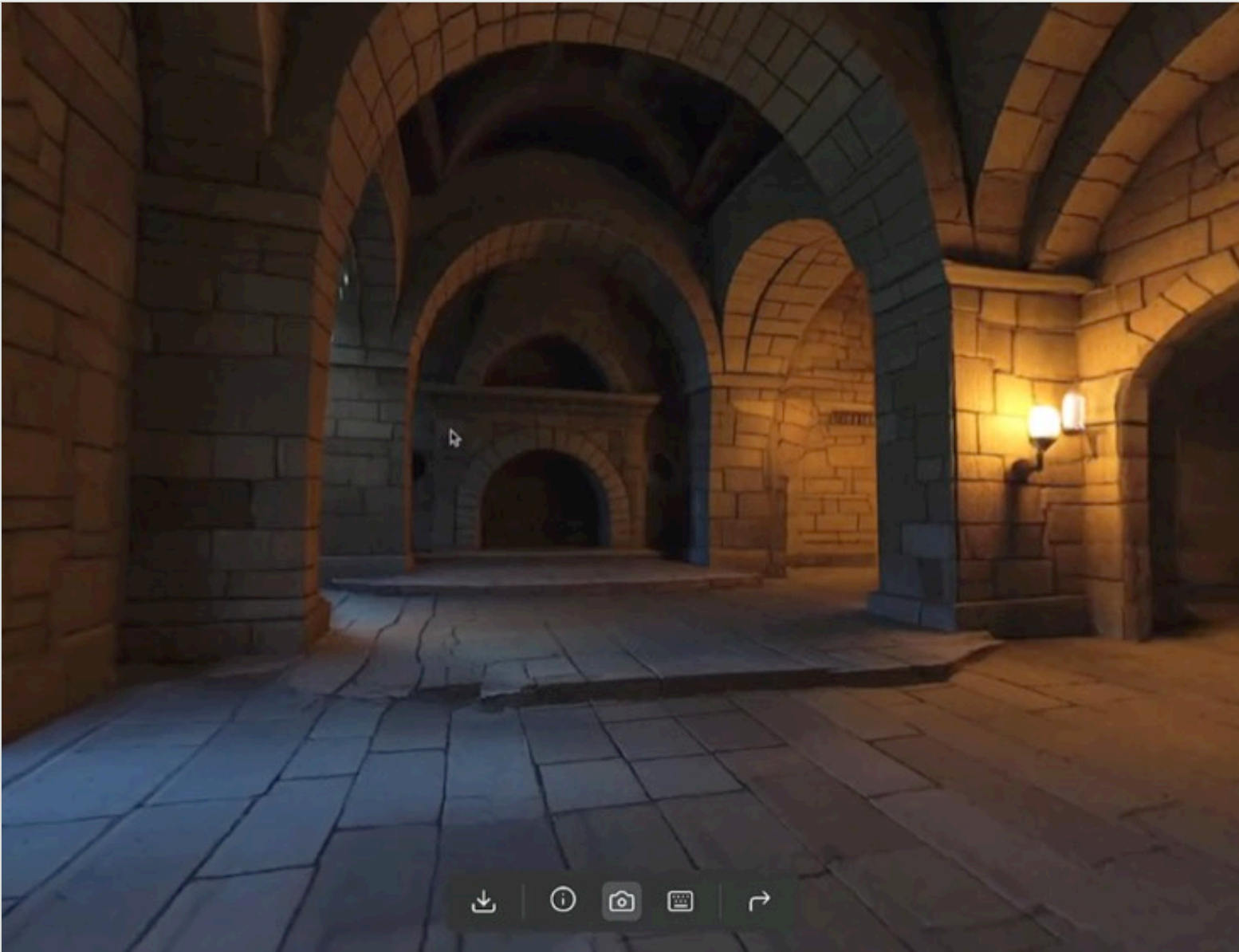


Marble — Persistent 3D Worlds You Can Revisit

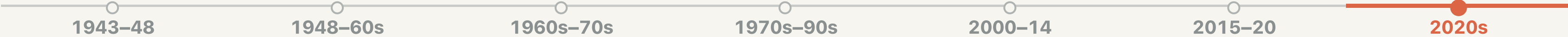
Generate a spatial world, not only the next camera frame.

Create	a 3D world from text, image, video, or coarse geometry
Navigate	through persistent, explorable scene structure
Edit & export	the world as a reusable spatial asset

This extends static scene modeling toward generative spatial worlds. The course’s stronger test still asks: how do state and observations change with action and time?



World Labs · a generated 3D world that remains explorable



World Models Ask a Concrete Question

World models ≠ robotics. Games, software, science, and embodied systems can all ask the same conditional-future question.

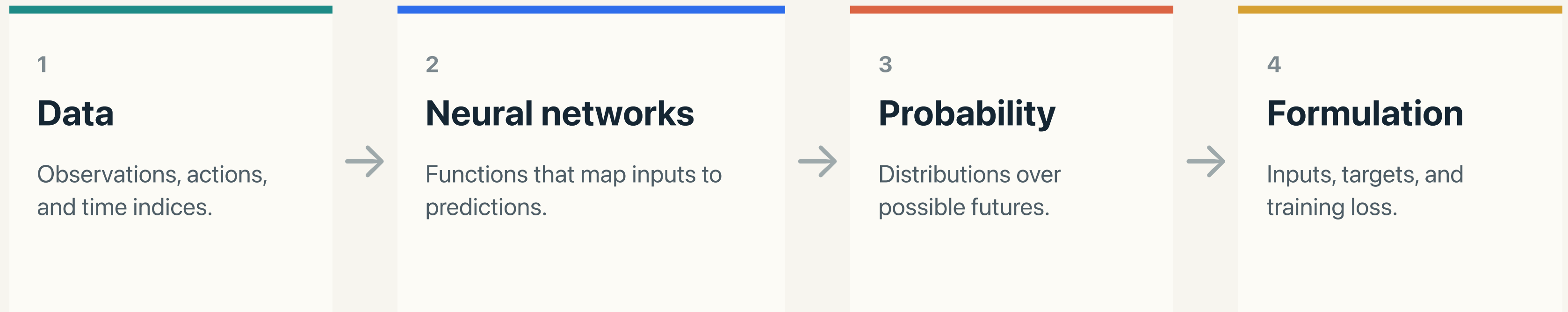


From a historical family to one concrete prediction question

NEXT: DATA → NEURAL COMPUTATION → PROBABILITY → FORMULATION



Foundations for World Models



Common Tensor Shapes

1

Scalar

one number

d

Vector

d numbers

$H_{\text{img}} \times W_{\text{img}} \times C$

Image

height × width × channels

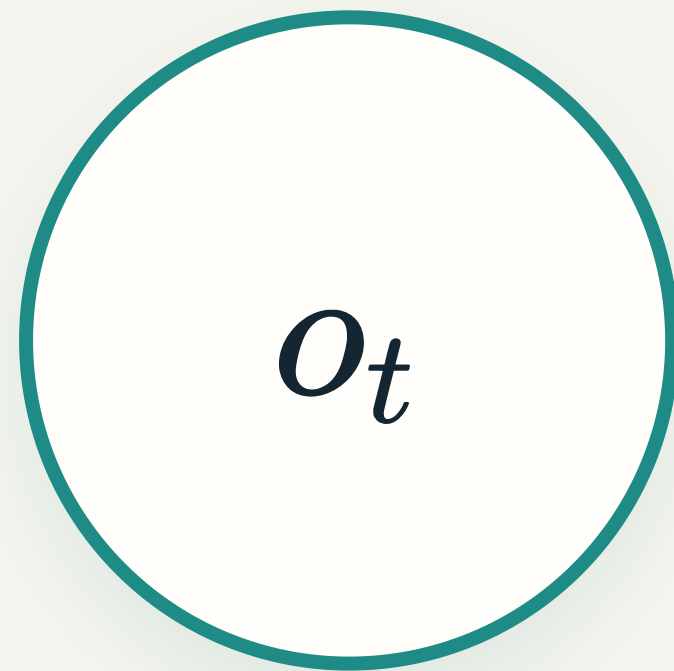
$T \times H_{\text{img}} \times W_{\text{img}} \times C$

Video

time × height × width × channels

Tensor shape records feature, space, channel, and time axes.

Observation and Action at Time t



Observation

What the system can measure at time t : pixels, sensors, text, or a screen.

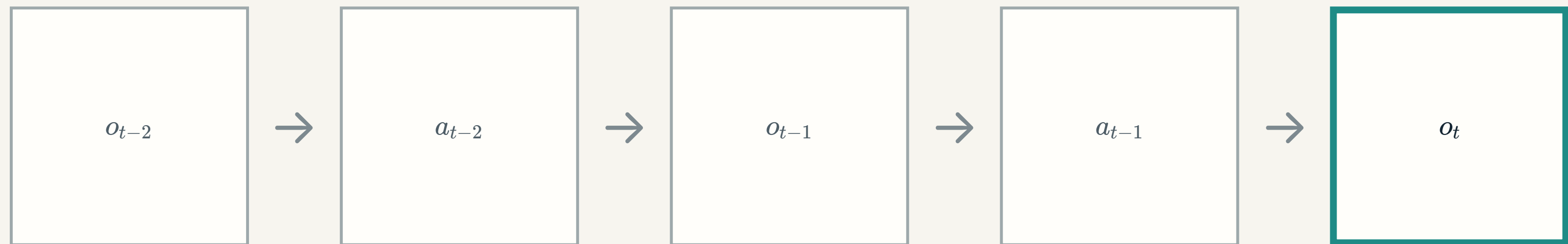


Action

An input applied to the world: a push, motor command, game move, or click.

o_t is the observation; a_t is the action at time t .

History Contains Past **Observations and Actions**



$$h_t = (o_{\leq t}, a_{< t})$$

a_t stays separate because we want to predict its consequence.

Specify the Prediction **Horizon** and **Target**

HORIZON

How far ahead?

$$H$$

One step or a segment extending through time $t + H$.

TARGET

What information?

$$Y_{t+1:t+H}$$

Pixels, pose, contact, task signals, or a learned representation.

One Neural-Network Layer



$\mathbf{v} = \sigma(W\mathbf{u} + \mathbf{b})$ · the weights and biases are learned from data.

Two Affine Layers Are Still **One Affine Map**

NO ACTIVATION

Equivalent to one layer

$$W_2(W_1\mathbf{u} + \mathbf{b}_1) + \mathbf{b}_2$$

Combine the two matrices and the two biases.

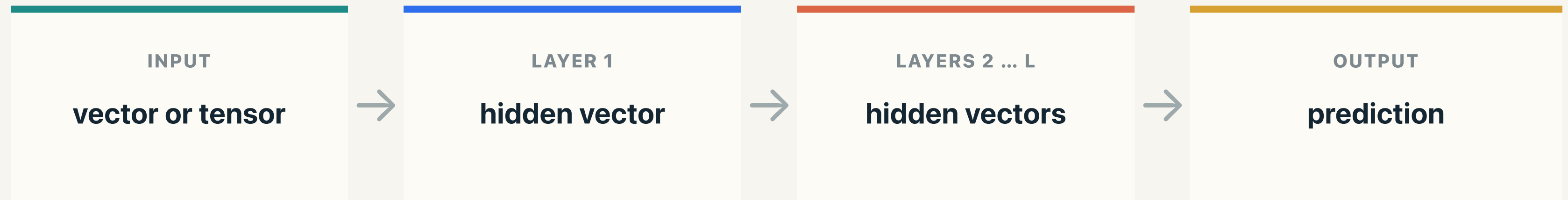
WITH AN ACTIVATION

**Cannot collapse
to one affine layer**

$$W_2 \sigma(W_1\mathbf{u} + \mathbf{b}_1) + \mathbf{b}_2$$

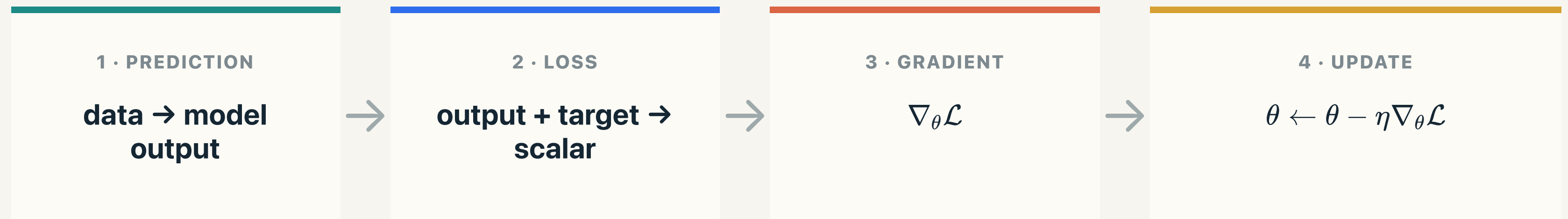
The nonlinearity makes the composition more expressive.

A Deep Network Composes **Multiple Layers**



$$f_{\theta} = f_{\theta_L}^{(L)} \circ \dots \circ f_{\theta_2}^{(2)} \circ f_{\theta_1}^{(1)}$$

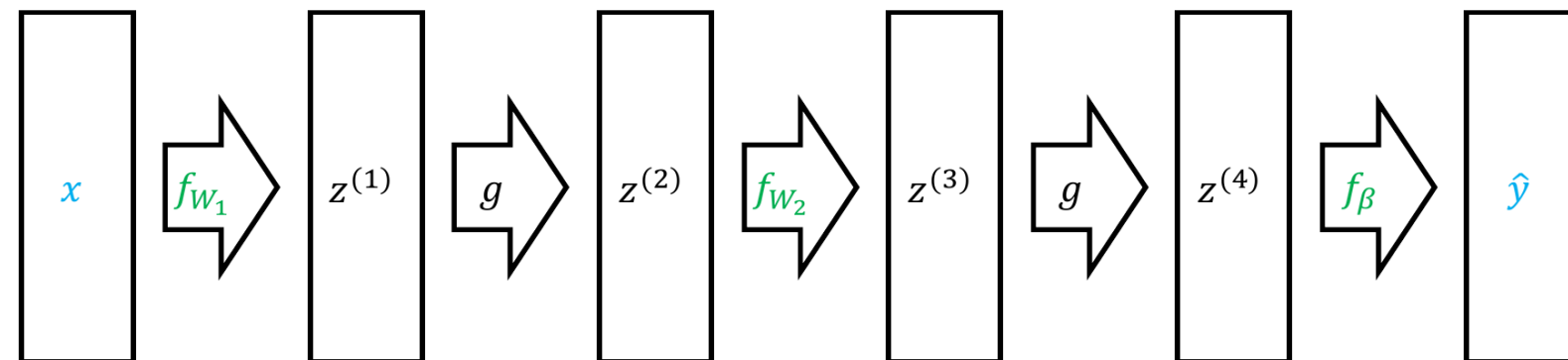
Training Minimizes a Loss



Backpropagation computes the gradient.

MLP: Fully Connected Layers

Feedforward Neural Networks



CIS 4190/5190 · Spring 2026 · L17 p.48

DEFINITION

All-to-all connections

Each output unit uses all input coordinates.

TYPICAL USE

Fixed-length vectors

State vectors, action vectors, and small output heads.

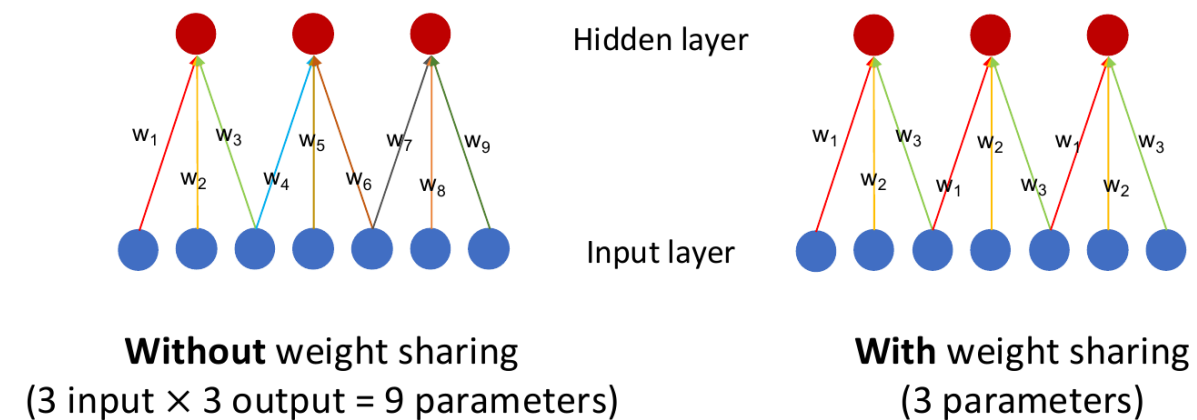
LIMITATION

No built-in space or time

Flattening an image removes its grid structure.

ConvNet: Local Filters with Shared Weights

Convolution Layers



Slide credit: Jia-Bin Huang

CIS 4190/5190 · Spring 2026 · L15 p.48

STEP 1

Restrict connectivity

Each output sees a local receptive field instead of the whole image.

STEP 2

Share the weights

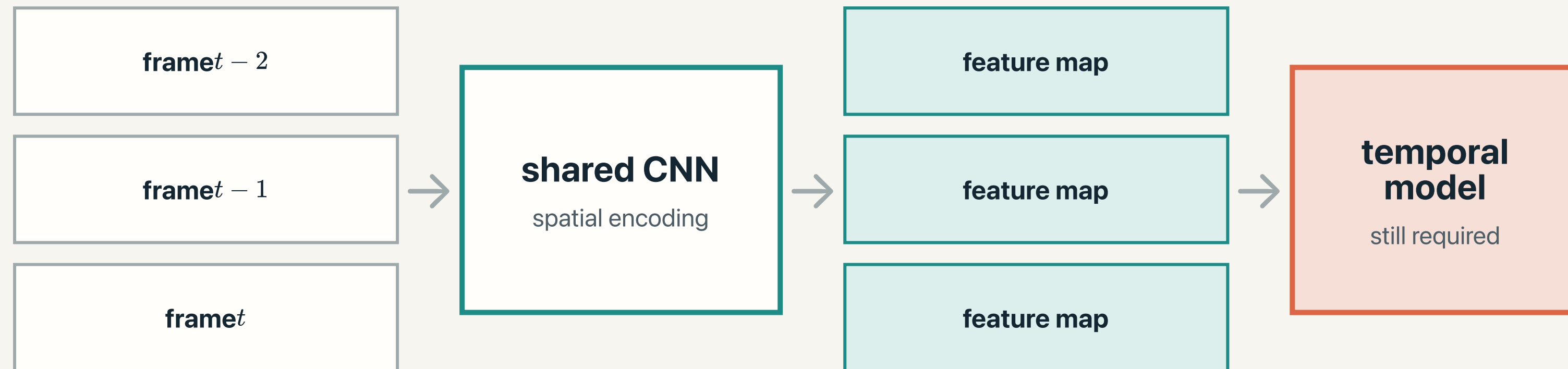
The same learned detector is reused at many spatial locations.

WORLD-MODEL USE

Encode spatial evidence

A CNN can encode each frame; a separate mechanism must still connect time and actions.

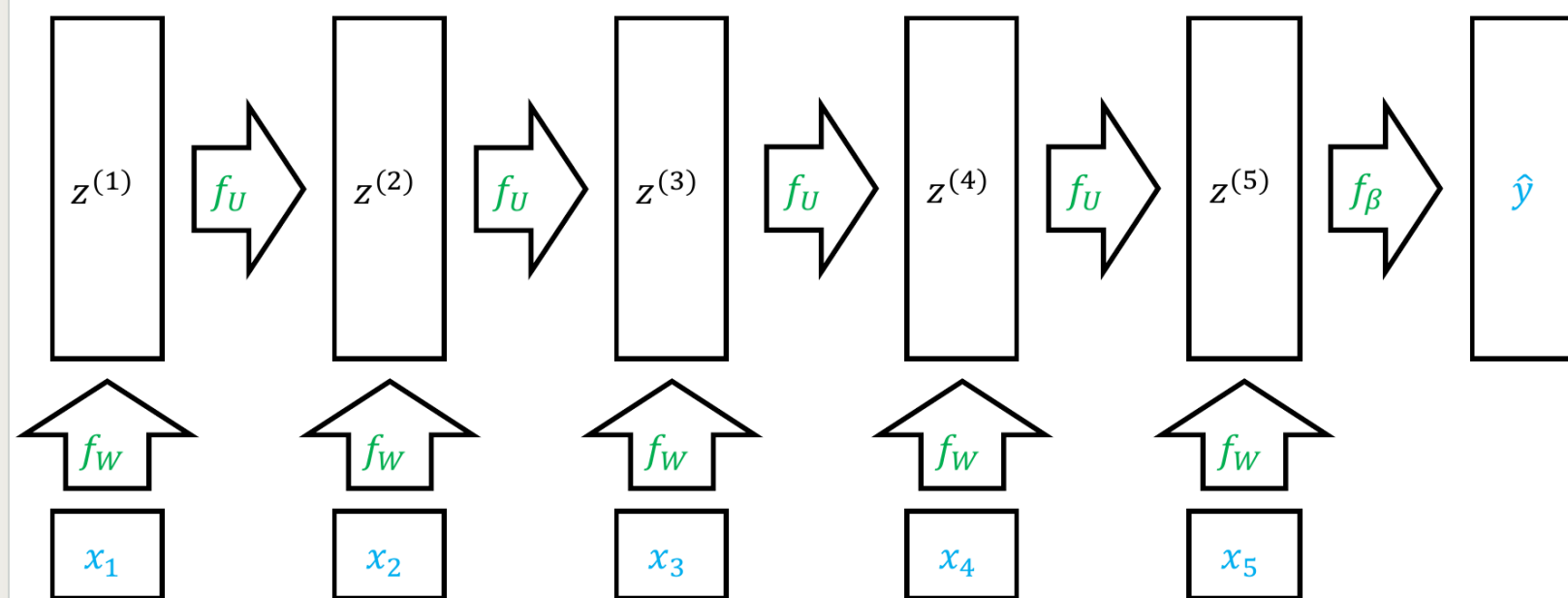
A Frame Encoder Does Not **Model Time**



Motion requires combining multiple frames and actions.

RNN: Update Memory at Each Time Step

Recurrent Neural Networks



CIS 4190/5190 · Spring 2026 · L17 p.50

UPDATE

Reuse the same function

The same input and recurrent maps are used at every time step.

MEMORY

Carry a hidden vector forward

Earlier inputs can affect later outputs.

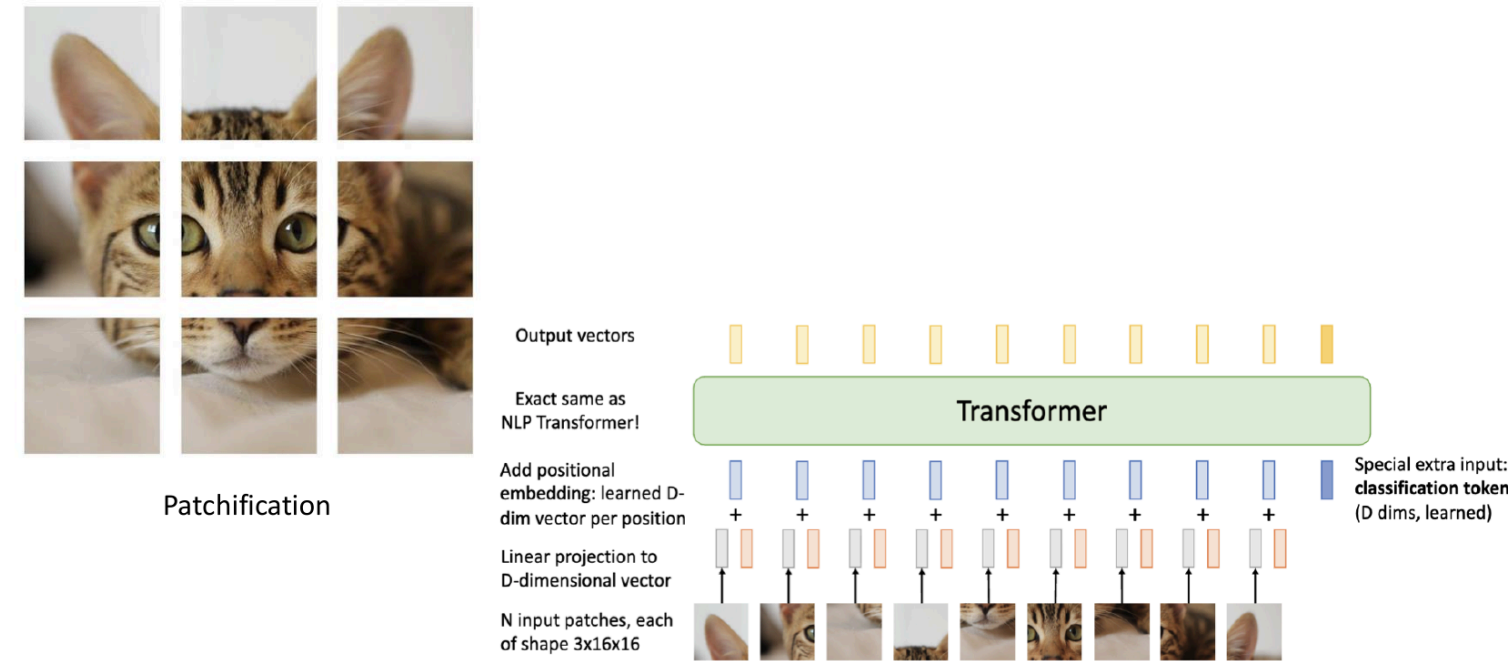
PREDICTION

Read from the hidden vector

The current memory can be mapped to a prediction.

Transformer: Process a Sequence of Tokens

Vision Transformers



CIS 4190/5190 · Spring 2026 · L18 p.83

TOKEN

One input item represented as a vector

An image patch, sensor reading, or action can be one token.

INPUT

A sequence of tokens

All tokens enter the Transformer together.

OUTPUT

One updated vector per token

Position and time encodings mark where and when each token occurs.

Self-Attention: Compare Tokens and Mix Values

Self-Attention Layer

- **Self-attention layer:**

$$y[t] = \sum_{s=1}^T \text{attention}(x[s], x[t]) \cdot f(x[s])$$

- Input first processed by local layer f
- All inputs can affect $y[t]$
- But weighted by $\text{attention}(x[s], x[t])$
- Resembles convolution but connection is learned instead of hardcoded

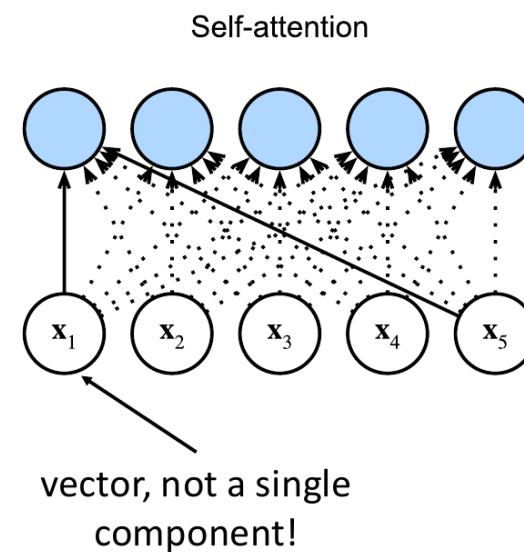


Figure credit to d2l.ai

CONVOLUTION

Use a fixed local neighborhood

Each position receives information from nearby positions.

SELF-ATTENTION

Compare each token with all tokens

Query-key scores become attention weights.

OUTPUT

Take a weighted sum of values

Tokens with larger weights contribute more.

Comparison of Neural Architectures

ARCHITECTURE	INPUT	OPERATION	COMMON USE
MLP	fixed-length vector	fully connected layers	state/action model or output head
CNN	image or spatial grid	shared local filters	frame encoder or decoder
RNN	sequence	recurrent memory update	temporal model
Transformer	token sequence	self-attention	long or multimodal context

A World Model Specifies **Inputs, Target, and Distribution**

ROLE	SYMBOL	MEANING	EXAMPLE
History	h_t	past observations and actions	frames + past controls
Action	$a_t \dots a_{t+H-1}$	candidate future input	push or motor command
Target	$Y_{t+1} \dots Y_{t+H}$	future quantity to predict	pixels, pose, or contact
Output	$p_{\theta}(Y \mid h, a)$	conditional distribution	probabilities or density

Deterministic Model: One Prediction

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

History and action produce one predicted next target.

 h_t

history

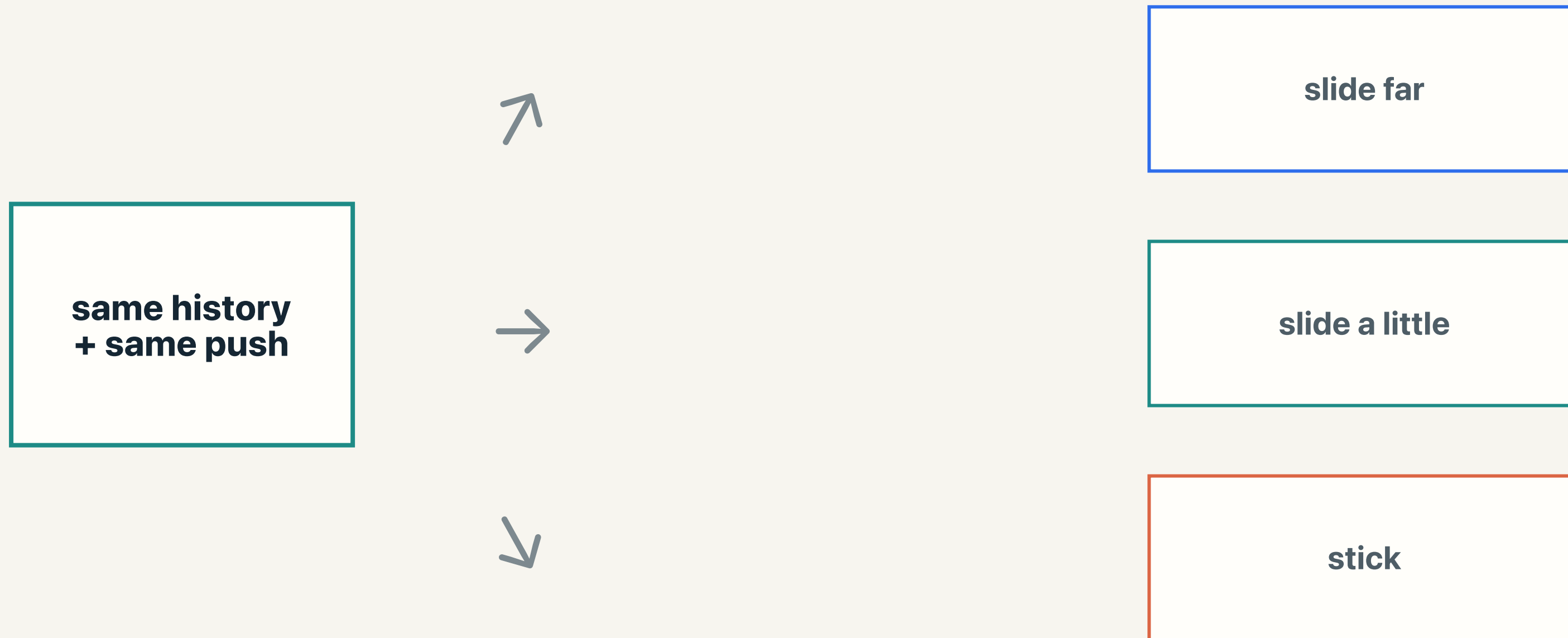
 a_t

action

 $\hat{y}_{t+1}$

predicted target

The Same History and Action Can Lead to **Different Futures**



Unknown friction and contact conditions can change the result.

Random Variable vs. Observed Value

BEFORE OBSERVATION

Possible future values

$$Y_{t+1}$$

Capital Y denotes a random variable.

AFTER OBSERVATION

One recorded value

$$y_{t+1}$$

Lowercase y denotes the value stored in the dataset.

Condition on History and Action

$$p_{\theta}(Y_{t+1} \mid h_t, a_t)$$

PREDICTED VARIABLE

Y_{t+1}

random next target

GIVEN INPUTS

h_t, a_t

history and candidate action

MODEL PARAMETERS

θ

learned during training

The Network Outputs **Distribution Parameters**

DISCRETE TARGET

Categorical probabilities

$$f_{\theta}(h_t, a_t) \rightarrow \ell \rightarrow \text{softmax}(\ell)$$

Logits become probabilities over outcomes such as stick, slide, or topple.

CONTINUOUS TARGET

Gaussian mean and scale

$$f_{\theta}(h_t, a_t) \rightarrow (\mu, \sigma)$$

μ and σ define a Gaussian density over position or velocity.

Higher Probability → Lower NLL

OBSERVED FUTURE

the puck slides far

The same recorded outcome is scored by two models.

MODEL ASSIGNS

$$p_{\theta}(y \mid h, a) = 0.10$$

$$-\log(0.10) = 2.30$$

large loss

MODEL ASSIGNS

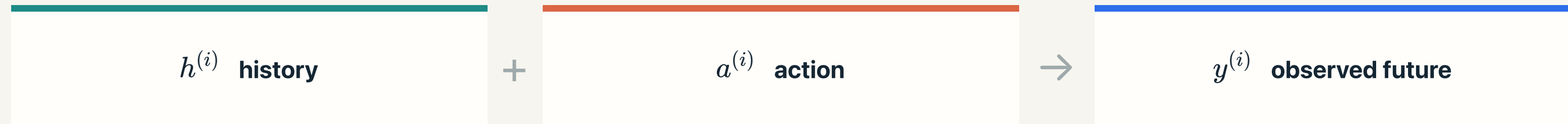
$$p_{\theta}(y \mid h, a) = 0.70$$

$$-\log(0.70) = 0.36$$

small loss

Training Averages NLL Across the Dataset

ONE TRAINING EXAMPLE



$$\mathcal{D} = \left\{ (h^{(i)}, a^{(i)}, y^{(i)}) \right\}_{i=1}^N$$

$$\mathcal{L}(\theta) = -\frac{1}{N} \sum_{i=1}^N \log p_{\theta}(y^{(i)} \mid h^{(i)}, a^{(i)})$$

One-Step Prediction Distribution

$$p_{\theta}(Y_{t+1} \mid h_t, a_t)$$

Predict a distribution over the next target from history and one candidate action.

 h_t

history

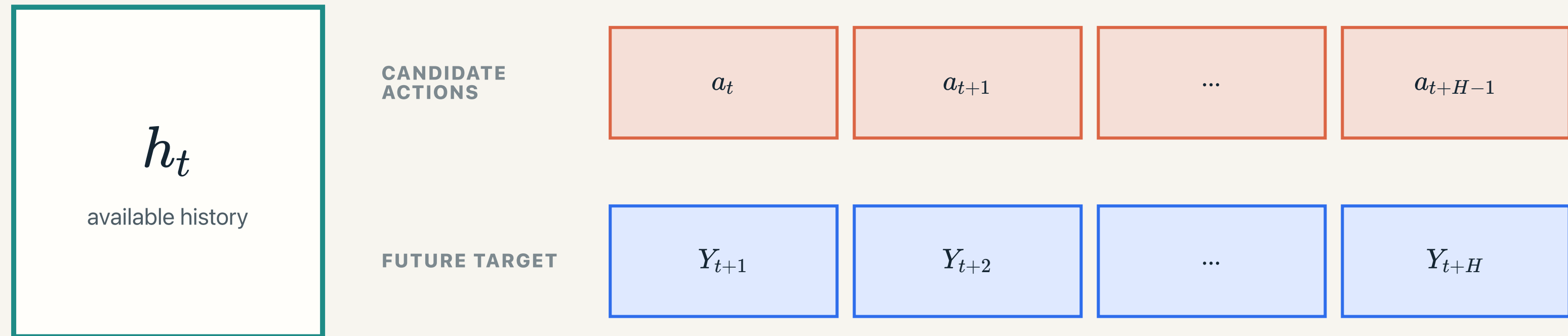
 a_t

candidate action

 Y_{t+1}

random next target

Multi-Step Prediction Distribution



$$p_{\theta}(Y_{t+1:t+H} \mid h_t, a_{t:t+H-1})$$

Observation, History, and Latent State



Encode History into a Latent State

DIRECT INPUT

$$p_{\theta}(Y_{t+1:t+H} \mid h_t, a_{t:t+H-1})$$

Use history and candidate actions.



ENCODER

$$z_t = E_{\theta}(h_t)$$

Map history to a latent vector.

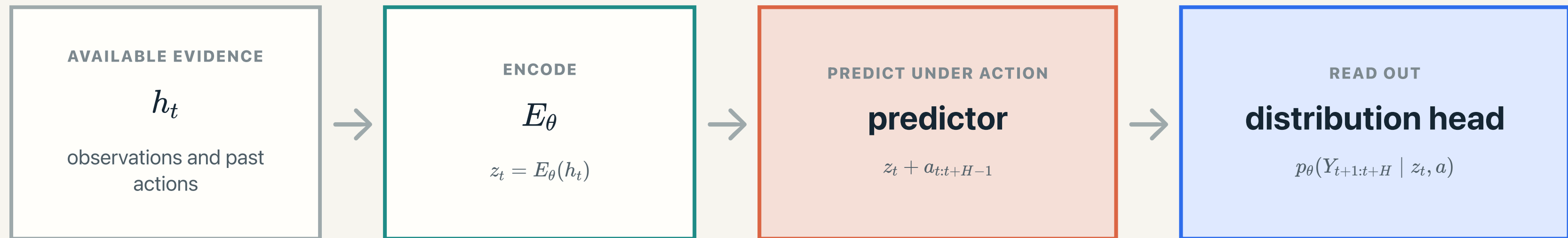


LATENT INPUT

$$p_{\theta}(Y_{t+1:t+H} \mid z_t, a_{t:t+H-1})$$

Use the latent vector and candidate actions.

Encoder → Predictor → Output Distribution



These computations may be separate modules or one shared network.

World Modeling Is a **Capability**, Not One Model

The capability is to answer useful probabilistic questions about how world variables can evolve.

ACTION-FREE INTERFACE

Predict the world

$$p_{\theta}(Y_{t+1:t+H}^{\text{world}} \mid h_t)$$

Video, observations, objects, or learned features.

SUPPLIED ACTIONS

Predict consequences

$$p_{\theta}(Y_{t+1:t+H}^{\text{world}} \mid h_t, a_{t:t+H-1})$$

The canonical controllable query developed today.

JOINT CONTINUATION

Model world + action

$$p_{\theta}(Y_{t+1:t+H}^{\text{world}}, A_{t:t+H-1} \mid h_t)$$

Recent world-action models expose both modalities.

Y^{world} may be pixels, objects, physical variables, or latents. Lowercase a is supplied; uppercase A is modeled as random.

Joint Modeling Connects **Actions and Consequences**

$$p_{\theta}(Y_{\text{future}}^{\text{world}}, A_{\text{future}} \mid h_t) = p_{\theta}(A_{\text{future}} \mid h_t) p_{\theta}(Y_{\text{future}}^{\text{world}} \mid h_t, A_{\text{future}})$$

ACTION QUERY

What tends to be done?

$$p_{\theta}(A_{\text{future}} \mid h_t)$$

A behavior or policy interface.

WORLD QUERY

What follows this action?

$$p_{\theta}(Y_{\text{future}}^{\text{world}} \mid h_t, A_{\text{future}})$$

A consequence-prediction interface.

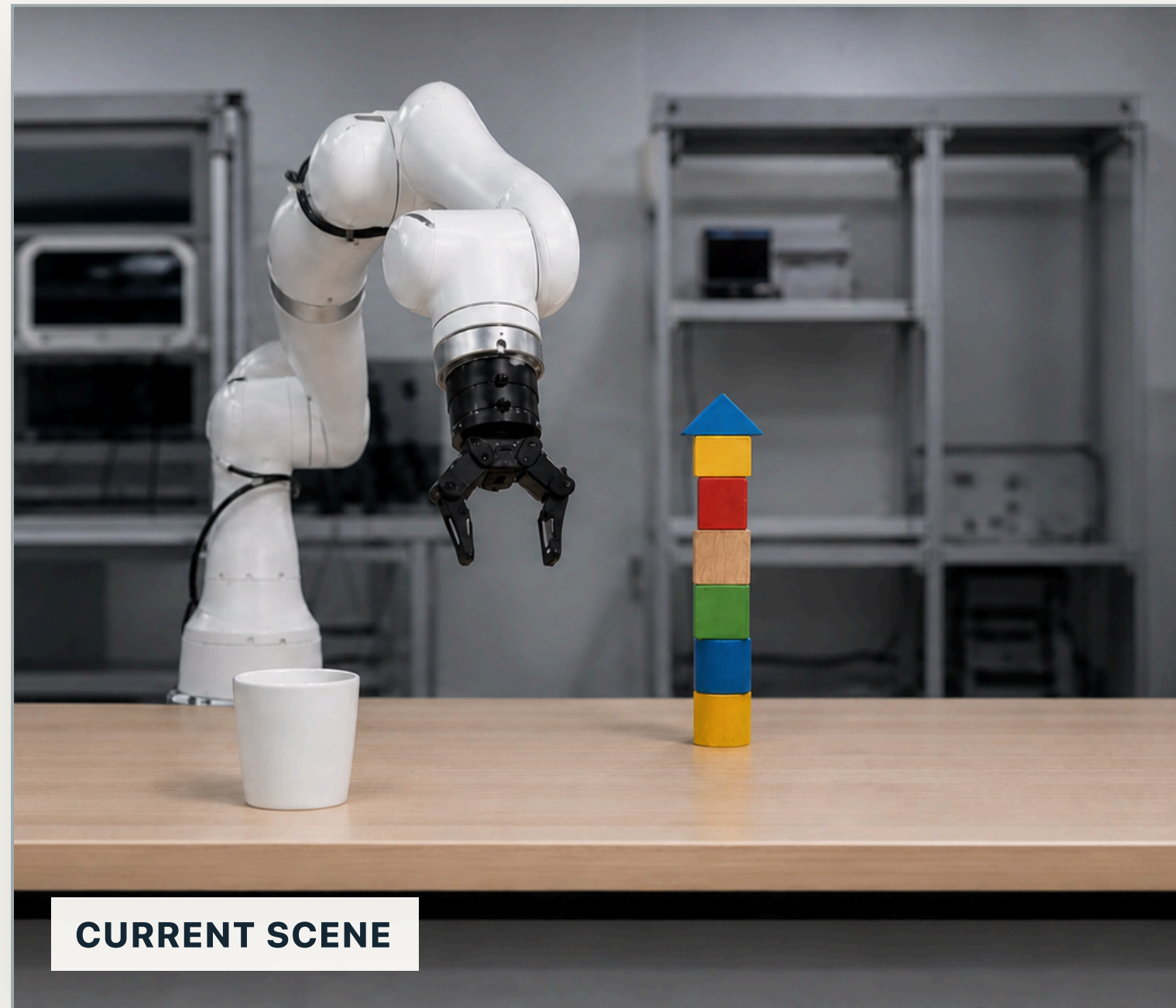
JOINT QUERY

How do both co-evolve?

$$p_{\theta}(Y_{\text{future}}^{\text{world}}, A_{\text{future}} \mid h_t)$$

An aligned world-action interface.

Action-only prediction does not expose world consequences. **Joint behavior modeling** does not by itself choose the best action.



ONE COHERENT EXAMPLE

What Happens After We Push the **Blue Block**?

We will specify one model from beginning to end: history · latent state · action · target · uncertainty · objective.

Recent Frames Reveal Motion Missing from **One Image**



$$h_t = (o_{\leq t}, a_{< t})$$

$z_t = E_{\theta}(h_t)$ compresses the evidence for prediction.

A Push Must Specify Direction, Magnitude, and **Duration**

CANDIDATE A

Left · light · short



A brief, low-magnitude push on the same blue block.

CANDIDATE B

Right · hard · long



A longer, higher-magnitude push in the opposite direction.

The action sequence is supplied to the model; selecting an action belongs to a later decision system.

The Query Must Fix What the Future **Contains**

TARGET

Tower pose + contact

Predict whether blocks move, touch, separate, or begin to fall.

HORIZON

Next H steps

Follow the early response through possible loss of stability.

UNCERTAINTY

Several outcomes remain

Friction, mass, contact, occlusion, and execution are only partially known.

$$Y_{t+1:t+H} = \{\text{future pose and contact variables}\}$$

Each Push Defines a Different **Future Distribution**

SAME LATENT STATE · ACTION A

Light push left

$$p_{\theta}(Y_{t+1:t+H} \mid z_t, a_{t:t+H-1}^A)$$

Most mass may remain near upright, with some probability of leaning.

SAME LATENT STATE · ACTION B

Hard push right

$$p_{\theta}(Y_{t+1:t+H} \mid z_t, a_{t:t+H-1}^B)$$

Probability can shift toward sliding, separation, or collapse.

CNN + RNN vs. Transformer

MODULAR VISUAL SEQUENCE MODEL

frames → shared CNN → RNN memory +
action → distribution head

$$p_{\theta}(Y \mid z_t, a)$$

FUSED TOKEN MODEL

frame patches + action tokens + time →
Transformer → distribution head

$$p_{\theta}(Y \mid h_t, a)$$

Both take history and action as input and output a distribution over Y .

One-Step and Sequence Losses Differ

ONE STEP

Score the next outcome

$$\mathcal{L}_1 = -\log p_\theta(y_{t+1} \mid h_t, a_t)$$

Each training window supervises one immediate target.

DIRECT SEQUENCE

Score the observed segment

$$\mathcal{L}_H = -\log p_\theta(y_{t+1:t+H} \mid h_t, a_{t:t+H-1})$$

One model call scores the chosen future segment.

A sequence target does not by itself imply recursive rollout or autoregressive factorization.

Inputs and Targets Across Domains

DOMAIN	HISTORY	CANDIDATE ACTION	FUTURE TARGET
Physical system	frames + sensors	force or motor command	pose, contact, or pixels
Game	screens + past moves	joystick or discrete move	screen, event, or value-relevant feature
Software agent	page state + interaction log	click, type, or command	next interface state or task event

$$p_{\theta}(Y_{\text{future}} \mid \text{available history, candidate actions})$$

Next: Where Does the **Experience** Come From?

TODAY

Define the prediction problem

Data · neural computation · probability · latent state
· learning objective



LECTURE 3

Environments, simulators, and rollouts

reset · step · trajectories · experience collection