A PATH TOWARDS AUTONOMOUS MACHINE INTELLIGENCE

Yann LeCun

Paper Review by Hongzheng Wang

Topic

- This is a position paper expressing the vision for a path towards intelligent machines that ***learn more like animals and humans***, that ***can reason and plan***, and whose behavior is ***driven by intrinsic objectives***, rather than by hard-wired programs, external supervision, or external rewards.

- This position paper proposes an architecture and training paradigms with which to construct autonomous intelligent agents.

Main Challenges

- How can machines learn to represent the world, learn to predict, and learn to act largely by observation?
  - *Interactions in the real world are expensive and dangerous.*
- How can machine reason and plan in ways that are compatible with gradient-based learning?
  - *Our best approaches to learning rely on estimating and using the gradient of a loss, which can only be performed with differentiable architectures and is difficult to reconcile with logic-based symbolic reasoning.*
- How can machines learn to represent percepts and action plans in a hierarchical manner, at multiple levels of abstraction, and multiple time scales?
  - *Humans and many animals are able to conceive multilevel abstractions with which long-term predictions and long-term planning can be performed by decomposing complex actions into sequences of lower-level ones.*

Main Challenges

- Animals and humans exhibit learning abilities and understandings of the world that are far beyond the capabilities of current AI and machine learning (ML) systems.

- Humans know how to act in many situation they have never encountered. A human can learn to drive a car in about 20 hours of practice.
- To be reliable, current ML systems need to be trained with very large numbers of trials so that even the rarest combination of situations will be encountered frequently during training.
- Still, our best ML systems are still very far from matching human reliability in real-world tasks such as driving.

Main Contributions

- An overall cognitive architecture in which all modules are differentiable and many of them are trainable.
- JEPA and Hierarchical JEPA: a non-generative architecture for predictive world models that learn a hierarchy of representations.
- A non-contrastive self-supervised learning paradigm that produces representations that are simultaneously informative and predictable.
- A way to use H-JEPA as the basis of predictive world models for hierarchical planning under uncertainty.

World Model

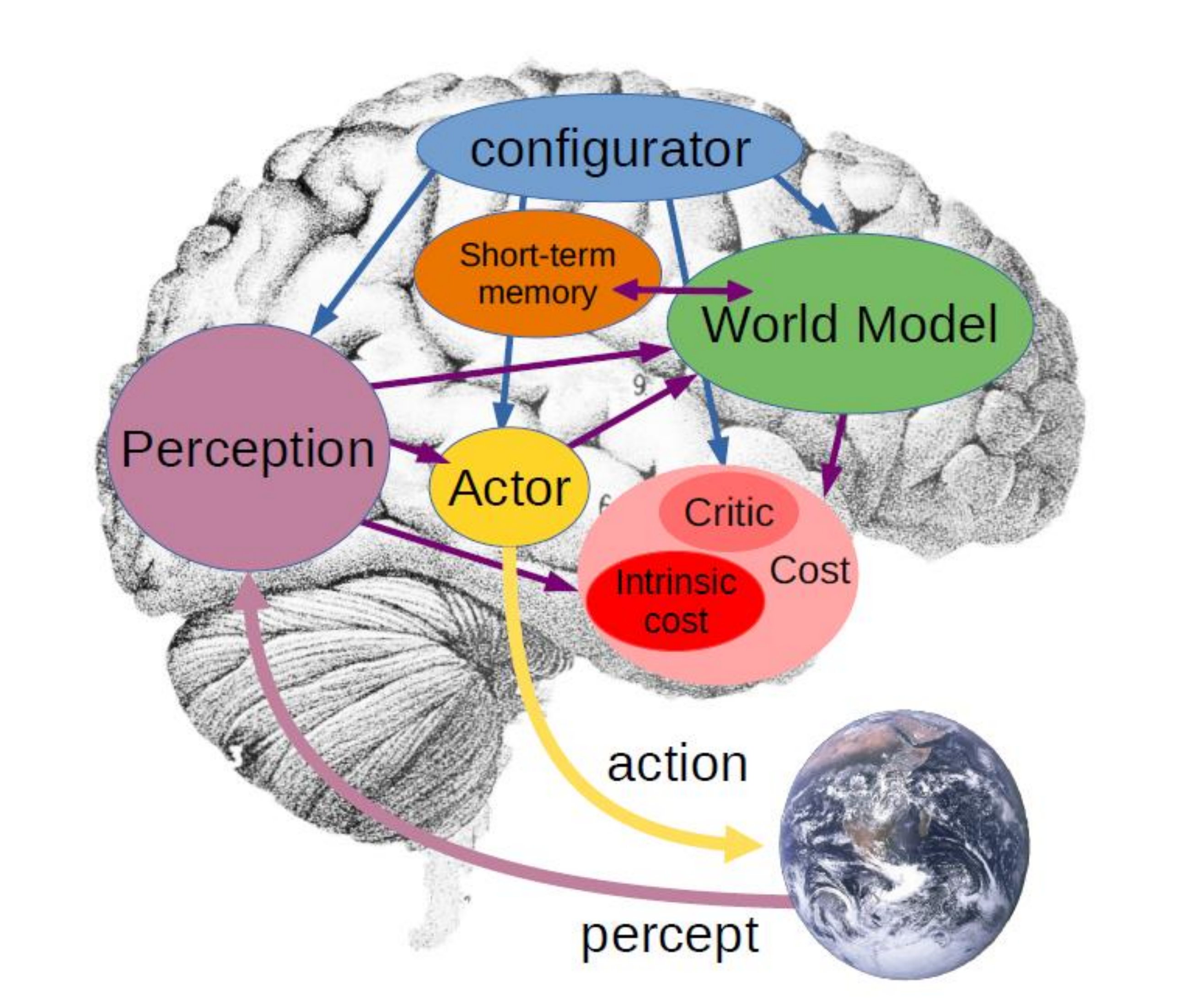
- The idea that humans, animals, and intelligent systems use world models goes back a long time in psychology.

- *Common sense* can be seen as a collection of models of the world that can tell an agent what is likely, what is plausible, and what is impossible. Using such *world models*, animals can learn new skills with very few trials.

ARCHITECTURE FOR AUTONOMOUS INTELLIGENCE

The Model Architecture

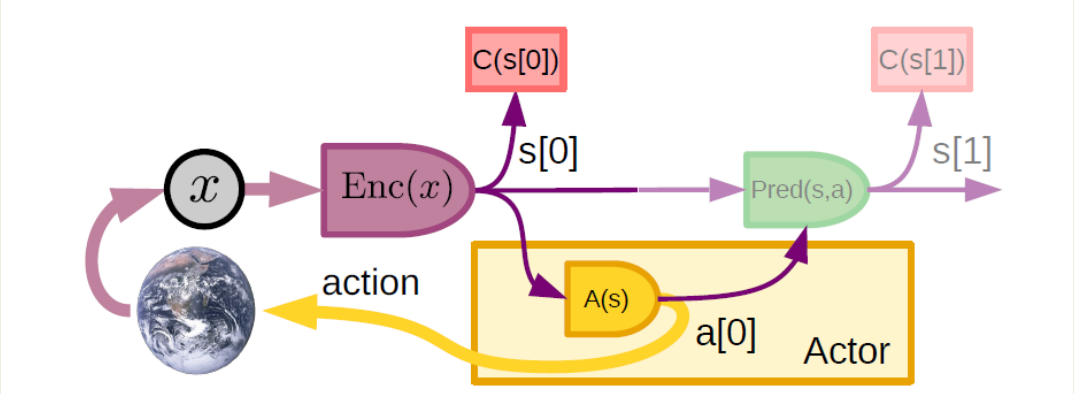
- **Configurator**: configures other modules
- **Perception**: estimates the current state of the world
- **World Model**: predicts possible future world states
- **Cost**: measures the level of discomfort of the agent. Intrinsic Cost (hard-wired) and Critic (trainable).
- **Short-term memory**: keeps track of the current and predicted world states
- **Actor**: find optimal action sequences



Typical Perception-Action Loops
Mode-1: Reactive behavior

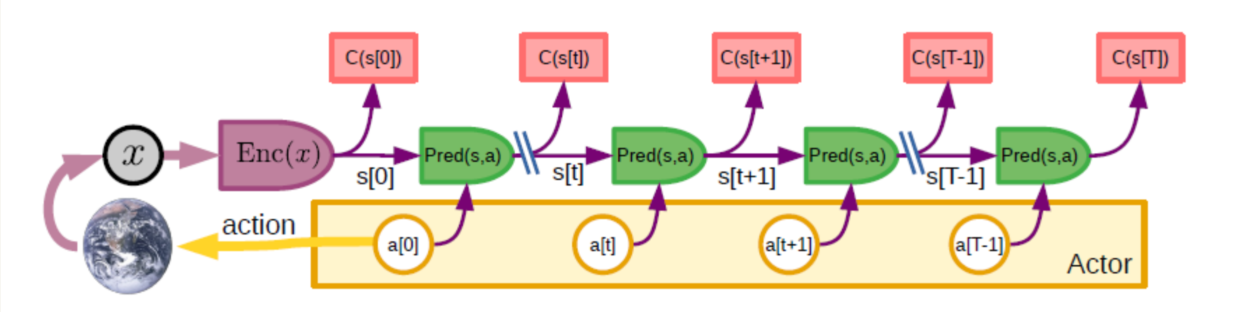
- Perception module: estimates the state of the world $s[0] = Enc(x)$.
- Actor module: directly computes an action, or a short sequence of actions, through a policy module $a[0] = A(s[0])$.
- Cost module: computes the energy of the initial state $f[0] = C(s[0])$ and stores the pairs $(s[0], f[0])$ in the short-term memory.

- Optionally, it may also predict the next state using the World Model $s[1] = Pred(s[0], a[0])$



Typical Perception-Action Loops
Mode-2: reasoning and planning using the world model

- Perception: estimates the state of the world $s[0] = Enc(x)$.
- Action Proposal: the actor proposes an initial sequence of actions to be fed to the world model for evaluation $(a[0], \ldots, a[t], \ldots, a[T])$.
- Simulation: the world model predicts one or several likely sequence of world state representations resulting from the proposed action sequence $(s[1], \ldots, s[t], \ldots, s[T])$.
- Evaluation: estimates a total cost from the predicted state sequence.
- Planning: the actor proposes a new action sequence with lower cost.
- Acting: after converging on a low-cost action sequence, the actor sends the first action in the low-cost sequence to the effectors.
- Memory: after every action, the states and associated costs from the intrinsic cost and the critic are stored in the short-term memory.

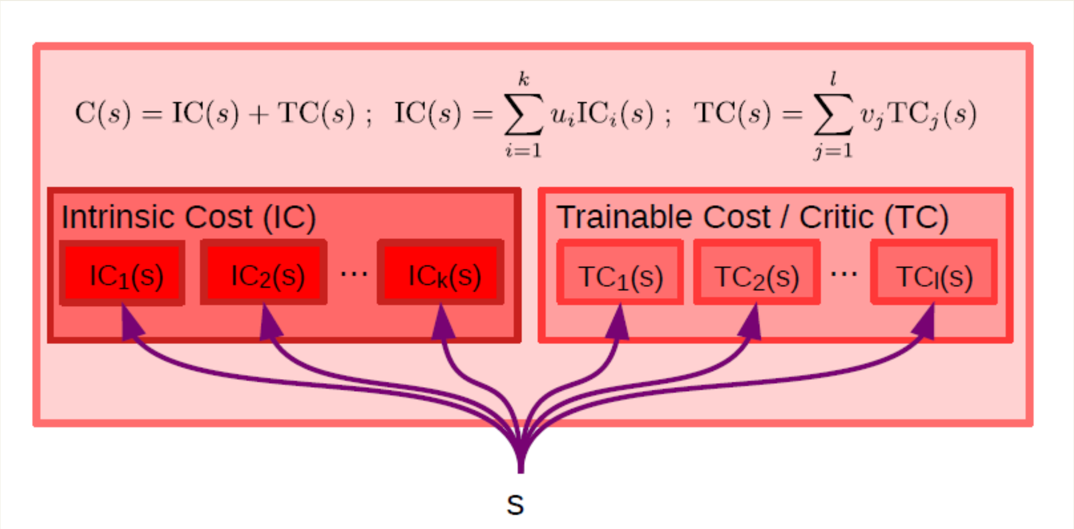


Typical Perception-Action Loops

- From Mode-2 to Mode-1: Learning New Skills
  - *Mode-2 is onerous. The agent can only focus on one complex task at a time.*
  - *Mode-1 is considerably less onerous, since it only requires a single pass through a policy module.*
  - *The system is run on Mode-2. Once properly trained, the policy module can be used to directly produce an action in Mode-1.*
- Reasoning as Energy Minimization
  - *The process of elaborating a suitable action sequence in Mode-2 can be seen as a form of reasoning. This form of reasoning is based on* ***simulation*** *using the world model, and* ***optimization*** *of the energy with respect to action sequences.*

Cost Module as the Driver of Behavior

- Intrinsic Cost module (IC)
  - *Hard-wired*
  - *Define the basic behavioral nature of the agent.*
  - *Like "pain", "hunger", etc.*
- Critic (TC)
  - *Trainable.*
  - *The principal role is to predict future values of the intrinsic energy.*



DESIGNING AND TRAINING THE WORLD MODEL

Main Issues

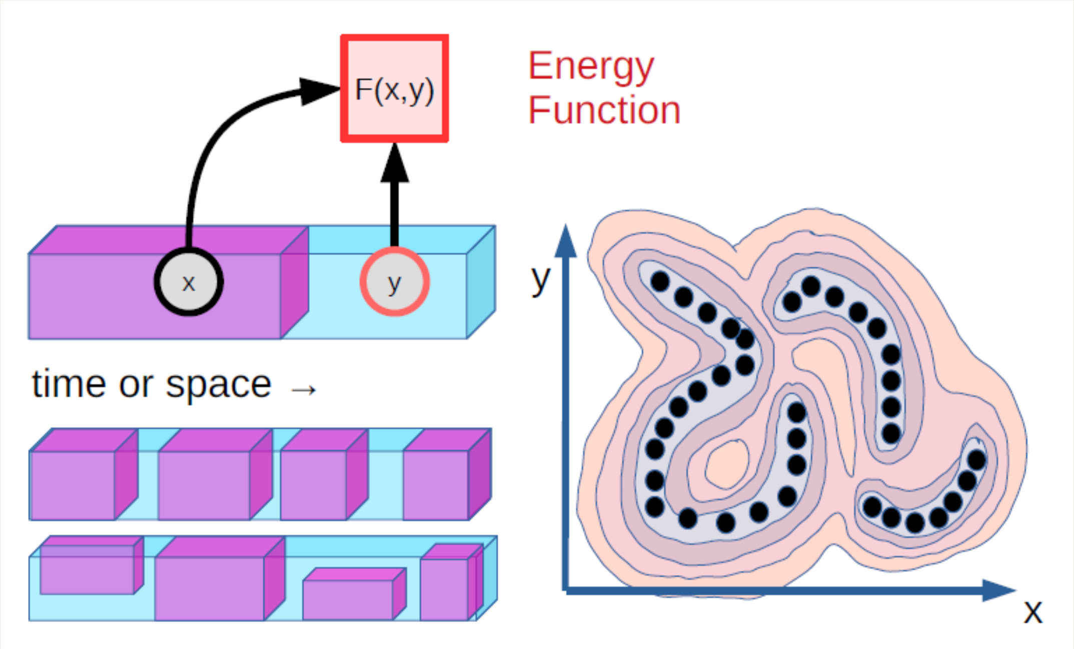
- The quality of the world model will greatly depend on the diversity of state sequences.

- The world is not entirely predictable, there may be multiple plausible world state representations.

- The world model must be able to make predictions at different time scales and different levels of abstraction.
  - *High-level goals and sub-goals.*

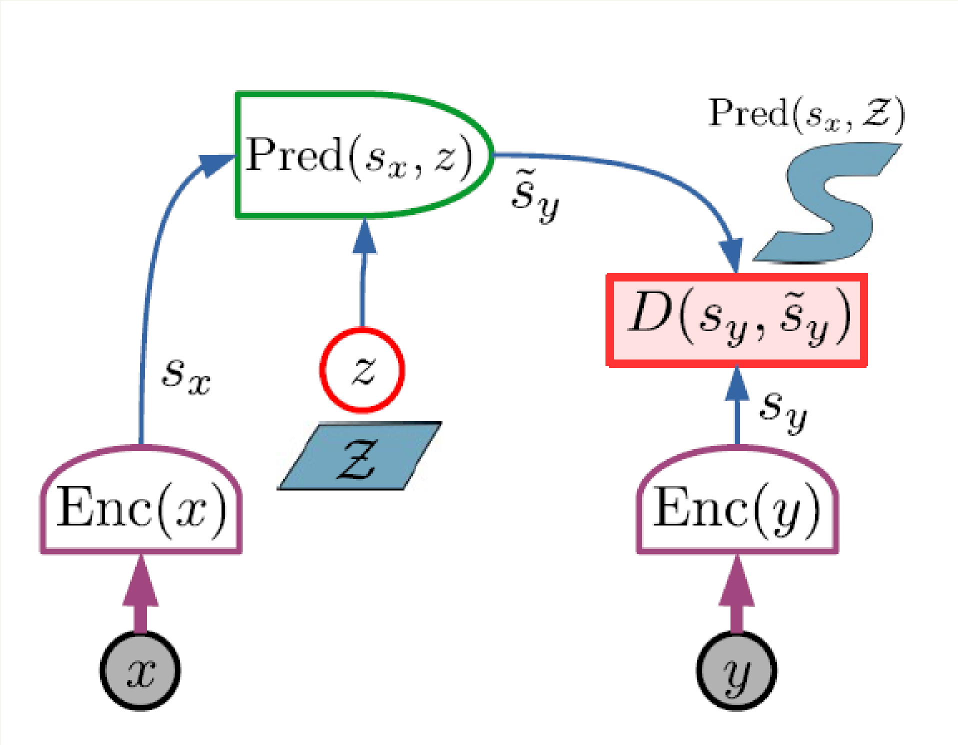
Self-Supervised Learning (SSL) and Energy-Based Model (EBM)

- SSL is a paradigm in which a learning system is trained to capture the mutual dependencies between its inputs.
- The system of EBM is a scalar-valued function $F(x, y)$ that produces low energy values when x and y are compatible and higher values when they are not.
  - *x: observed part;*
  - *y: possibly-unobserved part*
- We do not impose that the model be able to predict y from x, because there may be an infinite number of y that are compatible with a given x.



Joint Embedding Predictive Architecture (JEPA)

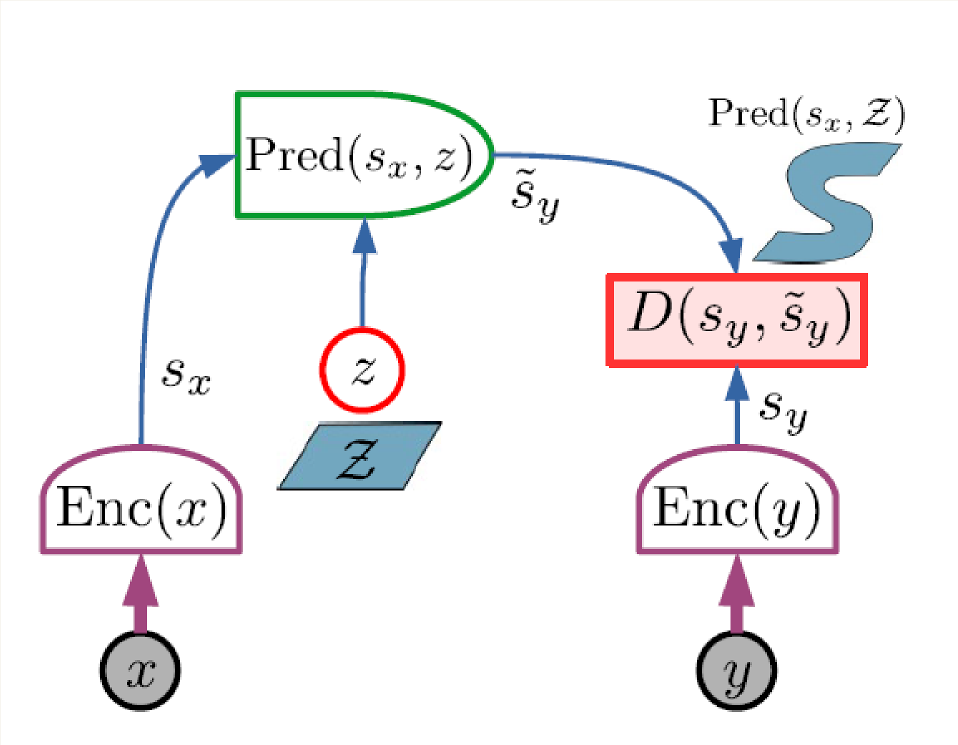
- JEPA consists of two encoding branches.
- A predictor module predicts s_y from s_x with the possible help of a latent variable z.
  - *Using **latent variable**, the model can present multiple predictions.*
  - *A latent variable is an input variable whose value is not observed but inferred.*
  - *In a temporal prediction scenario, the latent variable represents what cannot be predicted about y (the future) solely from x and from past observations (the past).*
- The energy is the prediction error.



Joint Embedding Predictive Architecture (JEPA)

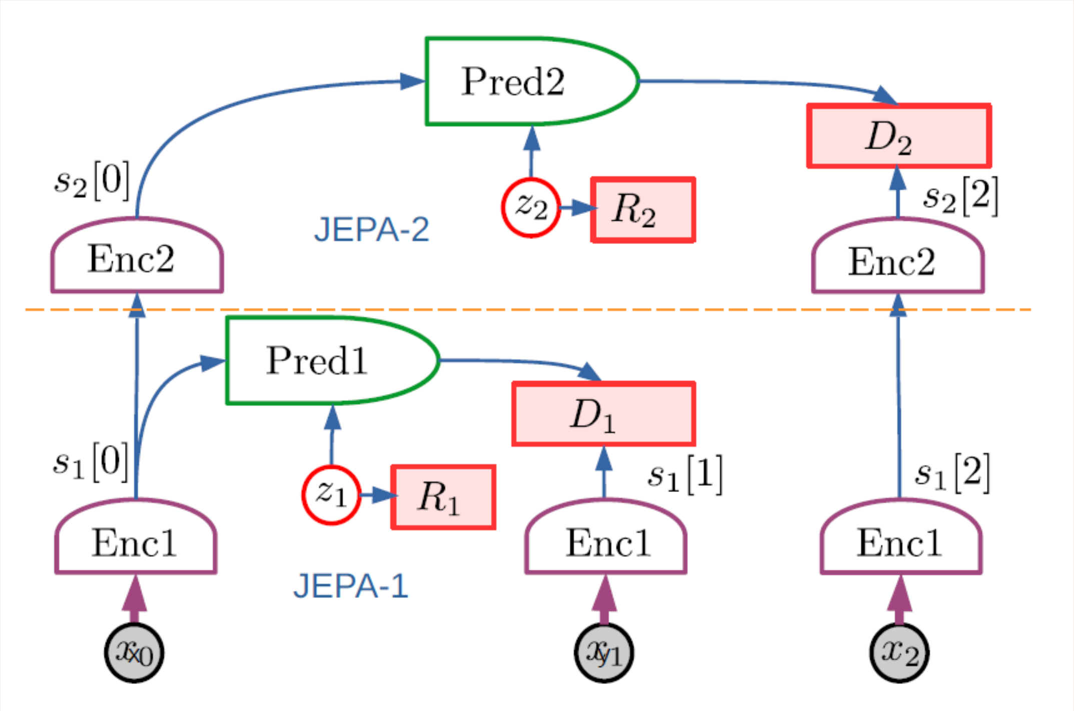
- The main advantage of JEPA is that it performs predictions in representation space, eschewing the need to predict every detail of y. This is enabled by the fact that the encoder of y may choose to produce an abstract representation from which irrelevant details have been eliminated.

- There are two ways a JEPA may represent the multiplicity of values of y compatible with x.
  - *invariance properties of the y encoder,*
  - *the latent variable z.*



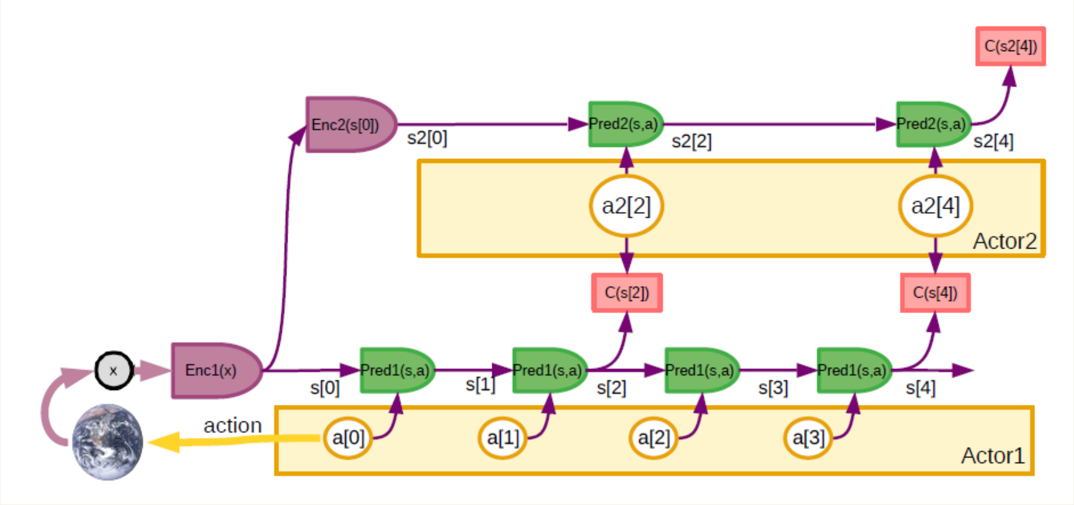
Hierarchical JEPA (H-JEPA)

- The ability of the JEPA to learn abstract representations in which accurate prediction can be performed allows hierarchical stacking.
- In this diagram, JEPA-1 extracts low-level representations and performs short-term predictions. JEPA-2 takes the representations extracted by JEPA-1 as inputs and extracts higher-level representations with which longer-term predictions can be performed.
- More abstract representations ignore details of the inputs that are difficult to predict in the long term, enabling them to perform longer-term predictions with coarser descriptions of the world state.
- The capacity of JEPA to learn abstractions suggests an extension of the architecture to handle prediction at multiple time scales and multiple levels of abstraction.



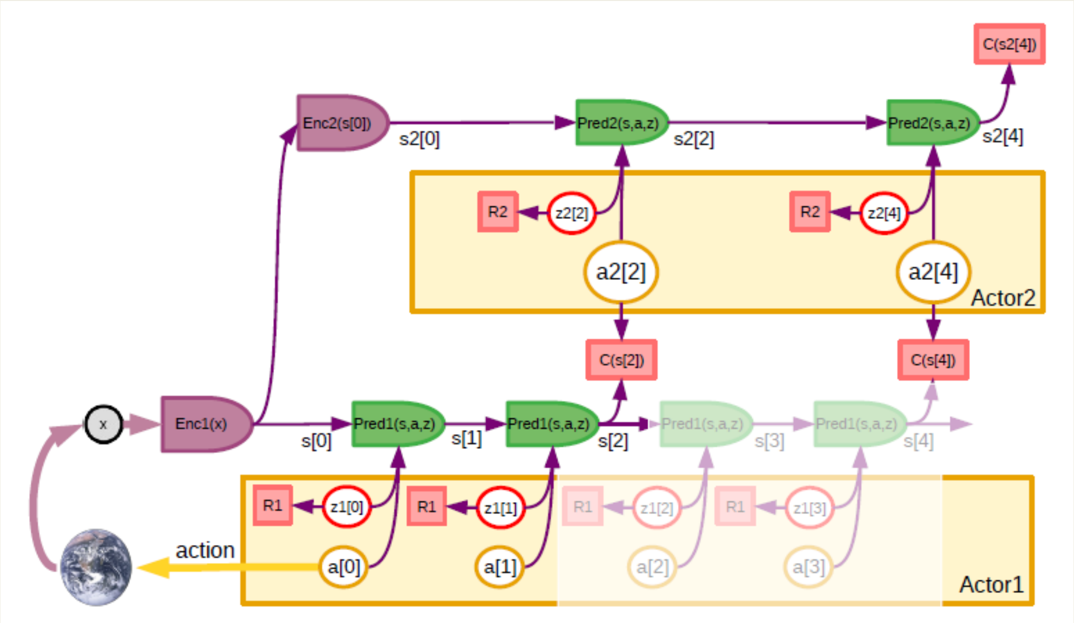
Hierarchical Planning

- Prediction takes place at all levels.
- Higher levels perform longer-term prediction, while lower levels perform shorter-term predictions. The overall task is defined by a high-level objective, depicted as $C(s_2[4])$ in the diagram.
- These high-level "actions" are not real actions but targets for the lower-level predicted states.



Handling uncertainty

- The real world is not entirely predictable. Uncertainty in predictions of future world states may be due to a number of reasons.
- We push the possible stochasticity of a predicted variable into a latent variable, which may be optimized, predicted, or sampled.
- As the prediction progresses, the number of generated state trajectories may grow exponentially. If each latent variable has k possible discrete values, the number of possible trajectories will grow as k^t, where t is the number of time steps. Directed search and pruning strategies must be employed.



Keeping track of the state of the world

- Traditionally, modules in deep learning architectures communicate states through vectors or multi-dimensional arrays. But this tends to be a very inefficient method when the state of the object being modeled only changes in minor ways from one time to the next.

- This suggests that the state of the world should be maintained in some sort of writable memory. Whenever an event occurs, only the part of the world-state memory affected by the event is to be updated, while the rest is to be left unchanged.

Data Streams

- The laws of motion of physical objects can, in principle, be derived from observation, without a need for intervention. But training a world model efficiently may require more active or “agentive” information gathering.
- One can list five modes of information gathering with which an agent can learn about how the world works:
  - ***passive observation:*** *the agent is being fed a sensor stream (e.g. video, audio, etc.)*
  - ***active foveation:*** *the focus of attention can be directed without affecting the environment.*
  - ***passive agency:*** *observe another agent acting on the environment, enabling the inference of causal effects of agent actions on the state of the environment.*
  - ***active egomotion:*** *the agent receives sensory streams from a real or virtual environment within which the position of the sensors can be modified without significantly affecting the environment.*
  - ***active agency:*** *sensory streams that are influenced by the agent's actions.*

Actor and Configurator

- The role of the actor module:
  - *inferring optimal action sequences that minimize the cost, given the predictions produced by the world model for Mode-2 actions.*
  - *producing multiple configurations of latent variables that represent the portion of the world state the agent does not know.*
  - *training policy networks for producing Mode-1 actions.*
- The configurator is the main controller of the agent. It takes input from all other modules and modulates their parameters and connection graphs.
  - *key ability: hardware reuse, and knowledge sharing*
  - *most important function: to set subgoals for the agent and to configure the cost module for this subgoal*

THANK YOU