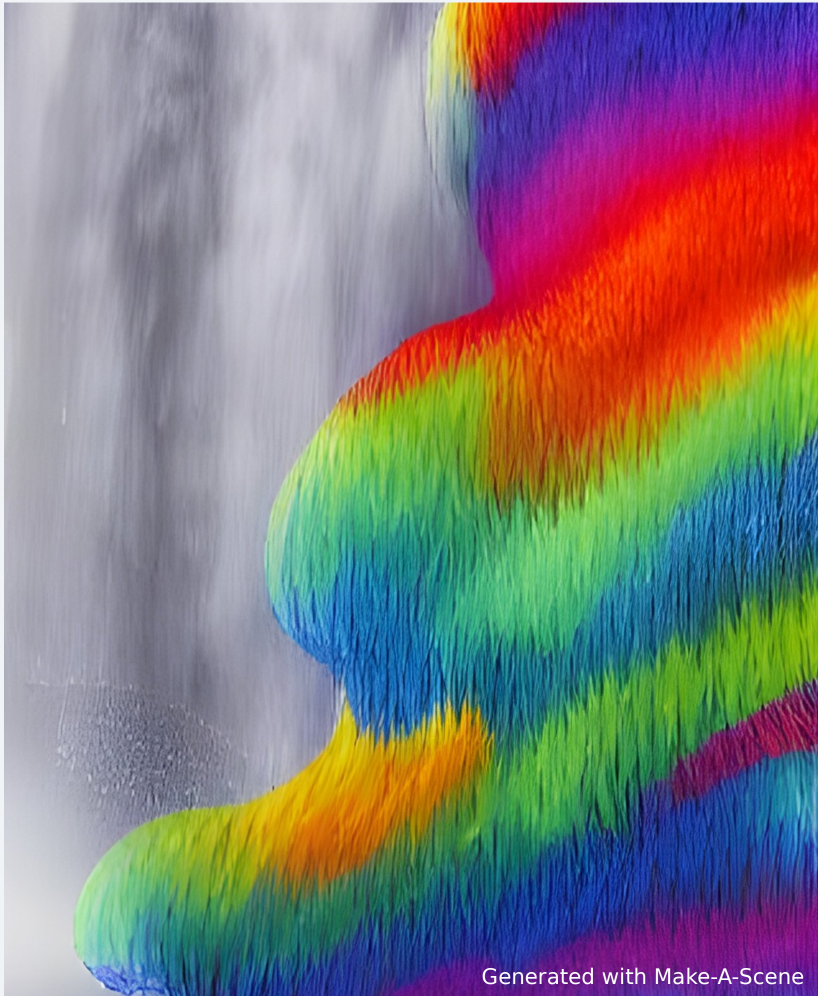
Do large language models need sensory grounding for meaning and understanding?

Spoiler: YES!

Yann LeCun
Courant Institute & Center for Data Science, NYU
Meta – Fundamental AI Research

NYU
2023-03-24

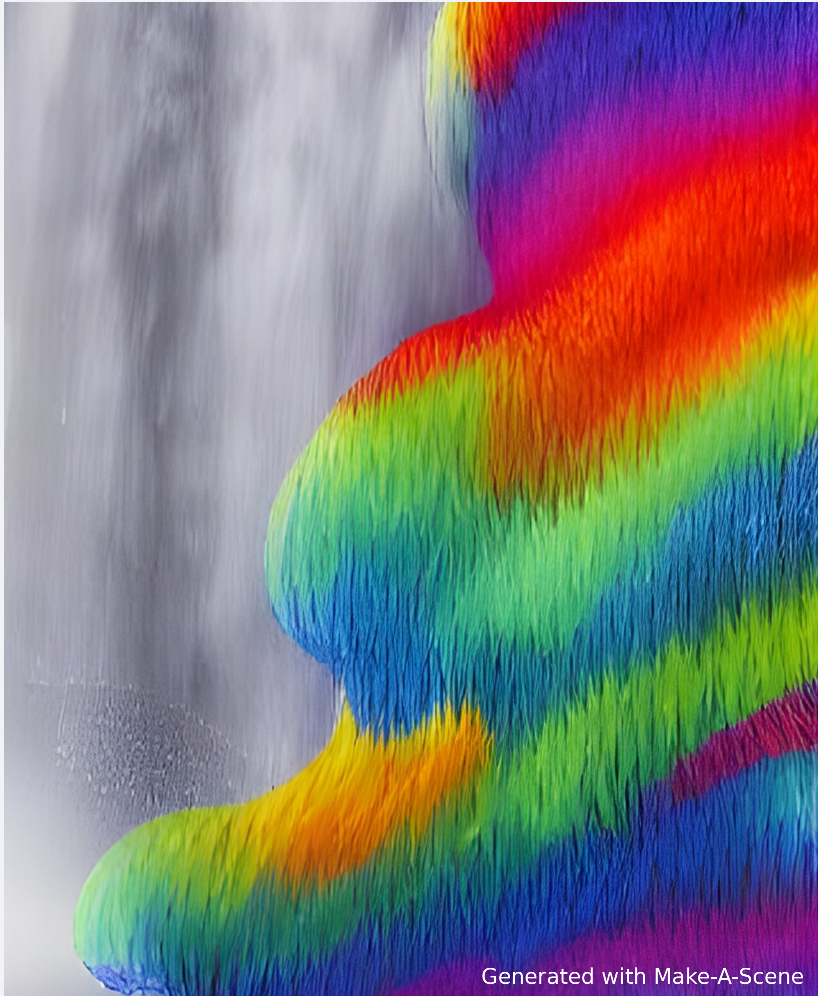

Generated with Make-A-Scene

Machine Learning sucks! (compared to humans and animals)

- **Supervised learning (SL) requires large numbers of labeled samples.**
- **Reinforcement learning (RL) requires insane amounts of trials.**
- **Self-Supervised Learning (SSL) requires large numbers of unlabeled samples.**
- **Most current ML-based AI systems:**
  - make stupid mistakes, do not reason nor plan
- **Animals and humans:**
  - Can learn new tasks **very** quickly.
  - Understand how the world works
  - Can reason and plan
- **Humans and animals have common sense**
- **current machines, not so much (it’s very superficial).**

Self-Supervised Learning has taken over the world

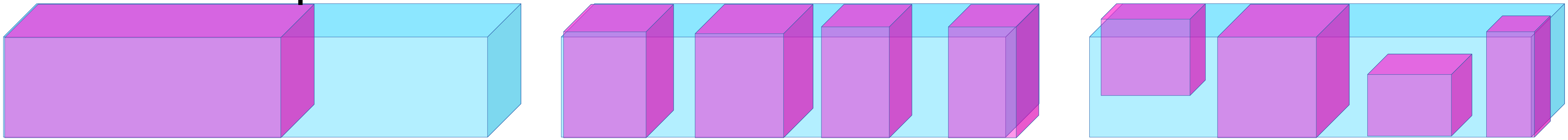
For understanding & generation of images, audio, text...



Self-Supervised Learning = Learning to Fill in the Blanks

- **Reconstruct the input or Predict missing parts of the input.**

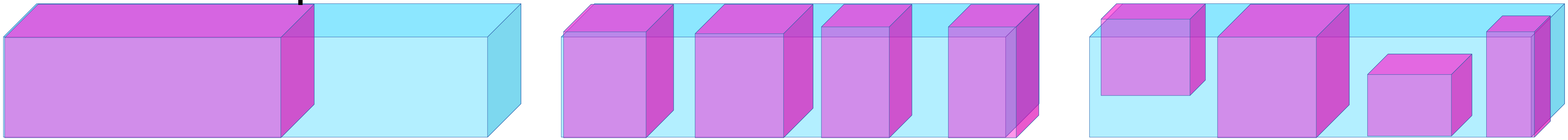
time or space →



Self-Supervised Learning = Learning to Fill in the Blanks

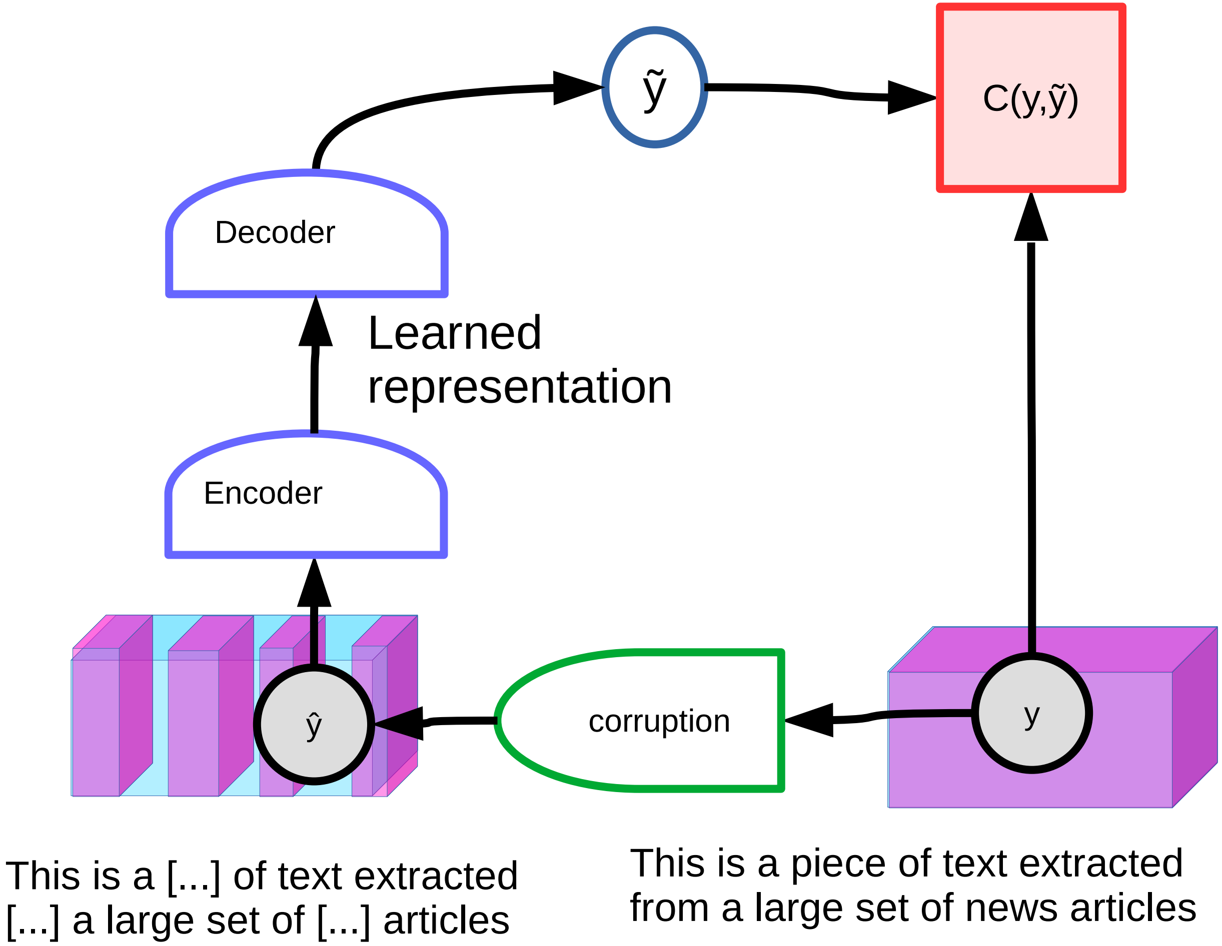
- **Reconstruct the input or Predict missing parts of the input.**

time or space →



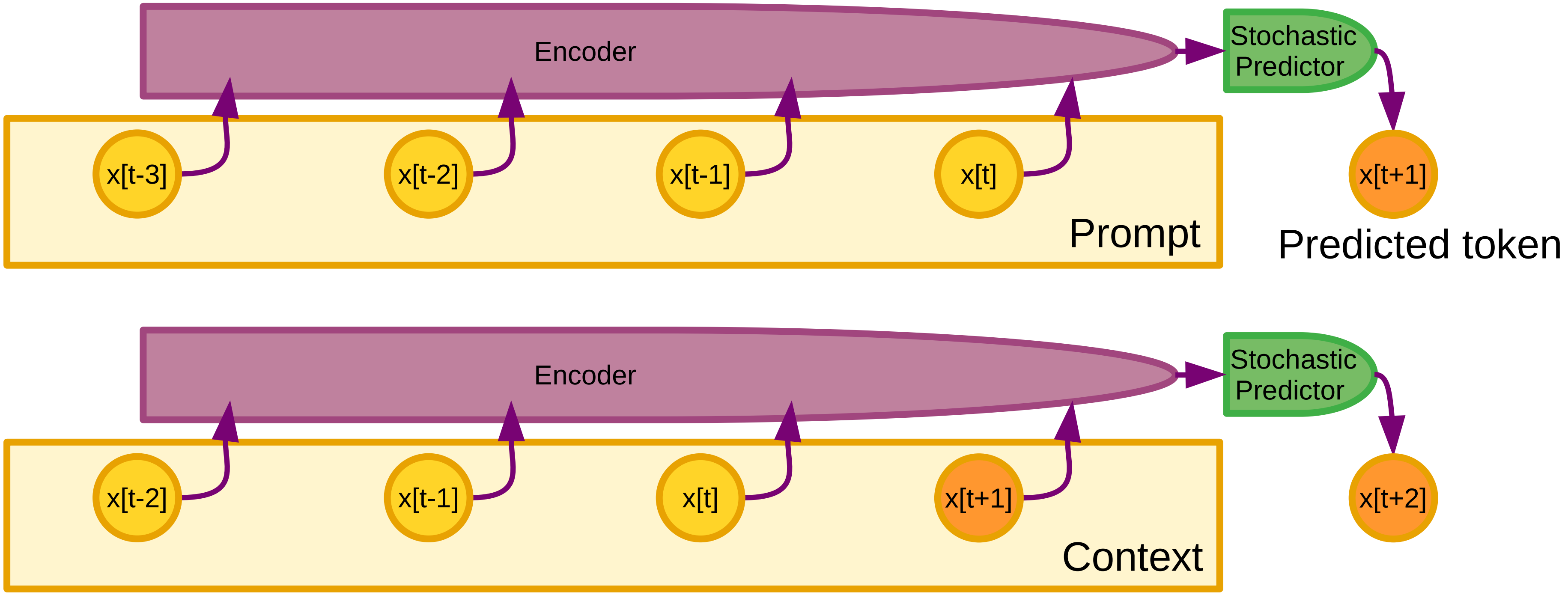
SSL via Denoising Auto-Encoder / Masked Auto-Encoder

- **BERT** [Devlin 2018]
- **RoBERTa** [Ott 2019]
- ....



Auto-Regressive Generative Models

- Outputs one “token” after another
- Tokens may represent words, image patches, speech segments...

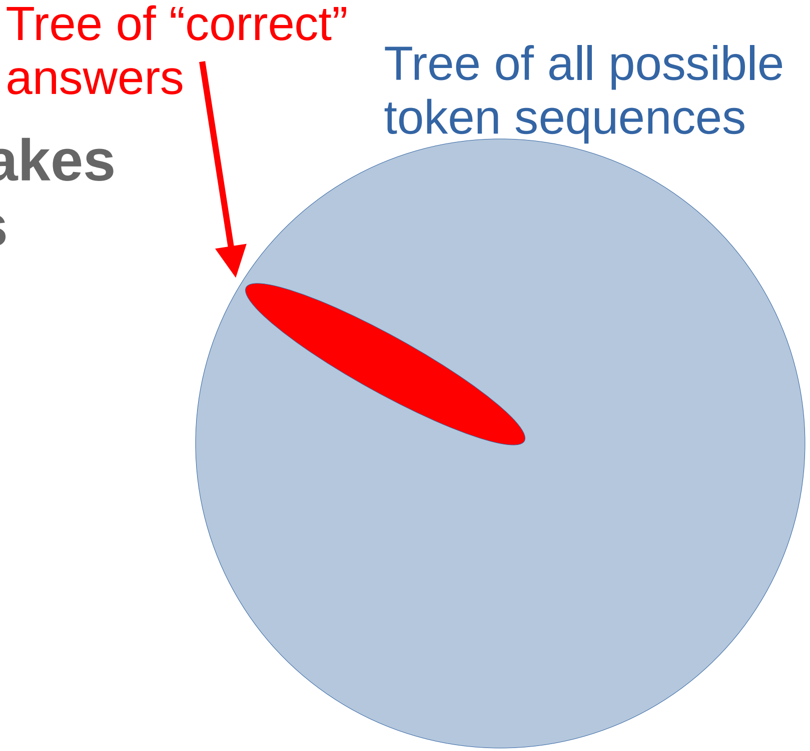


Auto-Regressive Large Language Models (AR-LLMs)

- **Outputs one text token after another**
- **Tokens may represent words or subwords**
- **Encoder/predictor is a transformer architecture**
  - With billions of parameters: typically from 1B to 500B
  - Training data: 1 to 2 trillion tokens
- **LLMs for dialog/text generation:**
  - BlenderBot, Galactica, LLaMA (FAIR), Alpaca (Stanford), LaMDA/Bard (Google), Chinchilla (DeepMind), ChatGPT (OpenAI), GPT-4 ??…
- **Performance is amazing … but … they make stupid mistakes**
  - Factual errors, logical errors, inconsistency, limited reasoning, toxicity...
- **LLMs have no knowledge of the underlying reality**
  - They have no common sense & they can't plan their answer

Unpopular Opinion about AR-LLMs

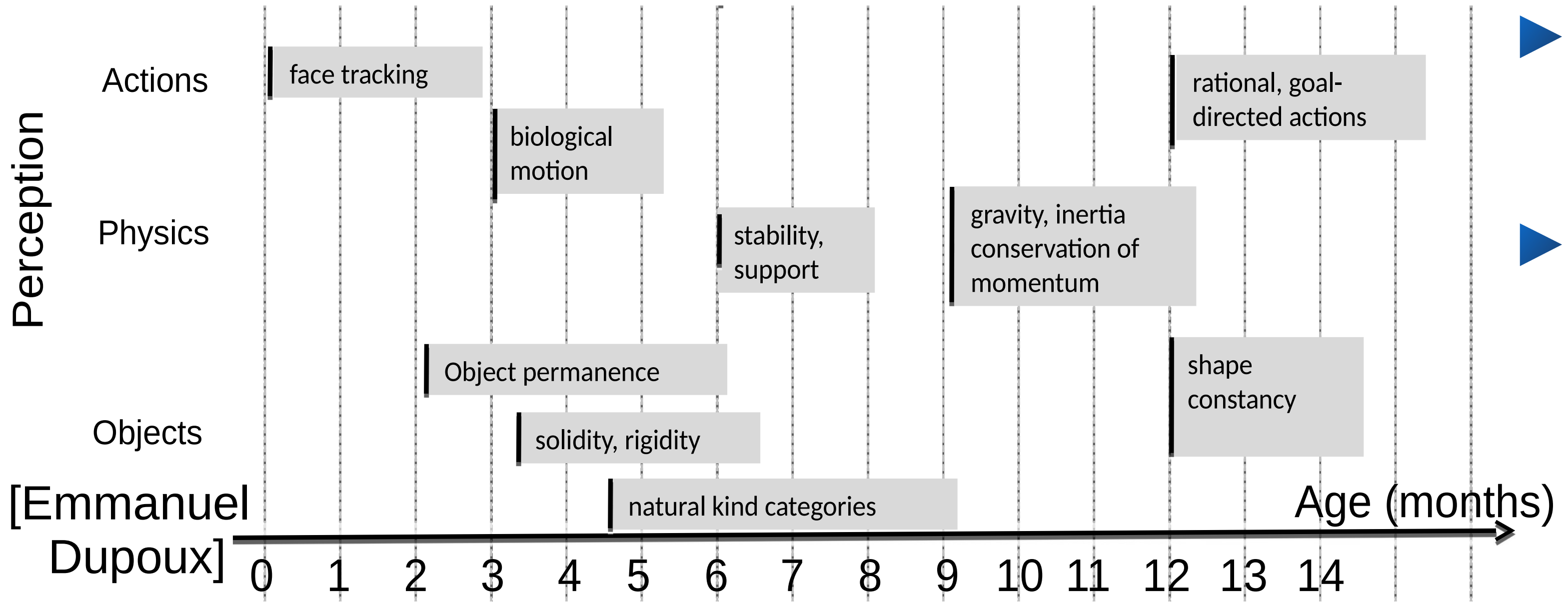
- **Auto-Regressive LLMs are doomed.**
- **They cannot be made factual, non-toxic, etc.**
- **They are not controllable**
- **Probability e that any produced token takes us outside of the set of correct answers**
- **Probability that answer of length n is correct:**
  - $P(\text{correct}) = (1-e)^n$
- **This diverges exponentially.**
- **It's not fixable.**



Auto-Regressive Generative Models Suck!

- **AR-LLMs**
  - Have a constant number of computational steps between input and output. Weak representational power.
  - Do not really reason. Do not really plan
- **Humans and many animals**
  - Understand how the world works.
  - Can predict the consequences of their actions.
  - Can perform chains of reasoning with an unlimited number of steps.
  - Can plan complex tasks by decomposing it into sequences of subtasks

How could machines learn like animals and humans?



- How can babies learn how the world works?
- How can teenagers learn to drive with 20h of practice?

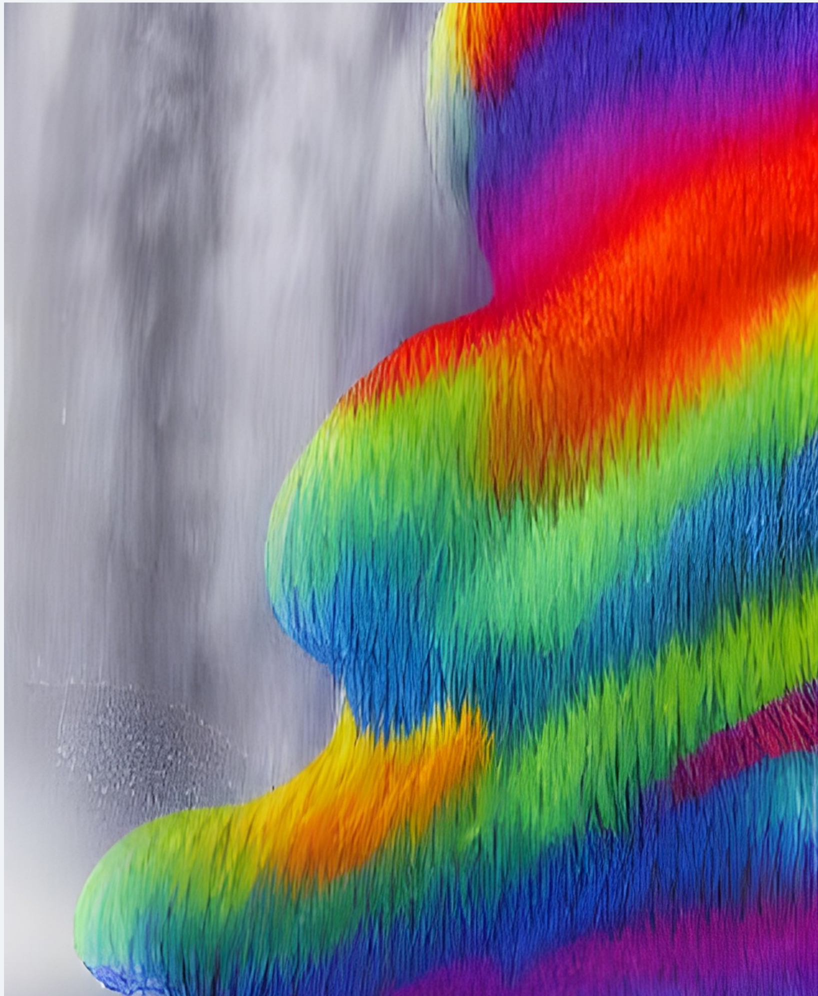
Three challenges for AI & Machine Learning

- **1. Learning representations and predictive models of the world**
  - Supervised and reinforcement learning require too many samples/trials
  - **Self-supervised learning** / learning dependencies / to fill in the blanks
    - learning to represent the world in a non task-specific way
    - Learning predictive models for planning and control
- **2. Learning to reason,** like Daniel Kahneman's "System 2"
  - Beyond feed-forward, System 1 subconscious computation.
  - Making reasoning compatible with learning.
    - Reasoning and planning as energy minimization.
- **3. Learning to plan complex action sequences**
  - Learning hierarchical representations of action plans

A Cognitive Architecture capable of reasoning & planning

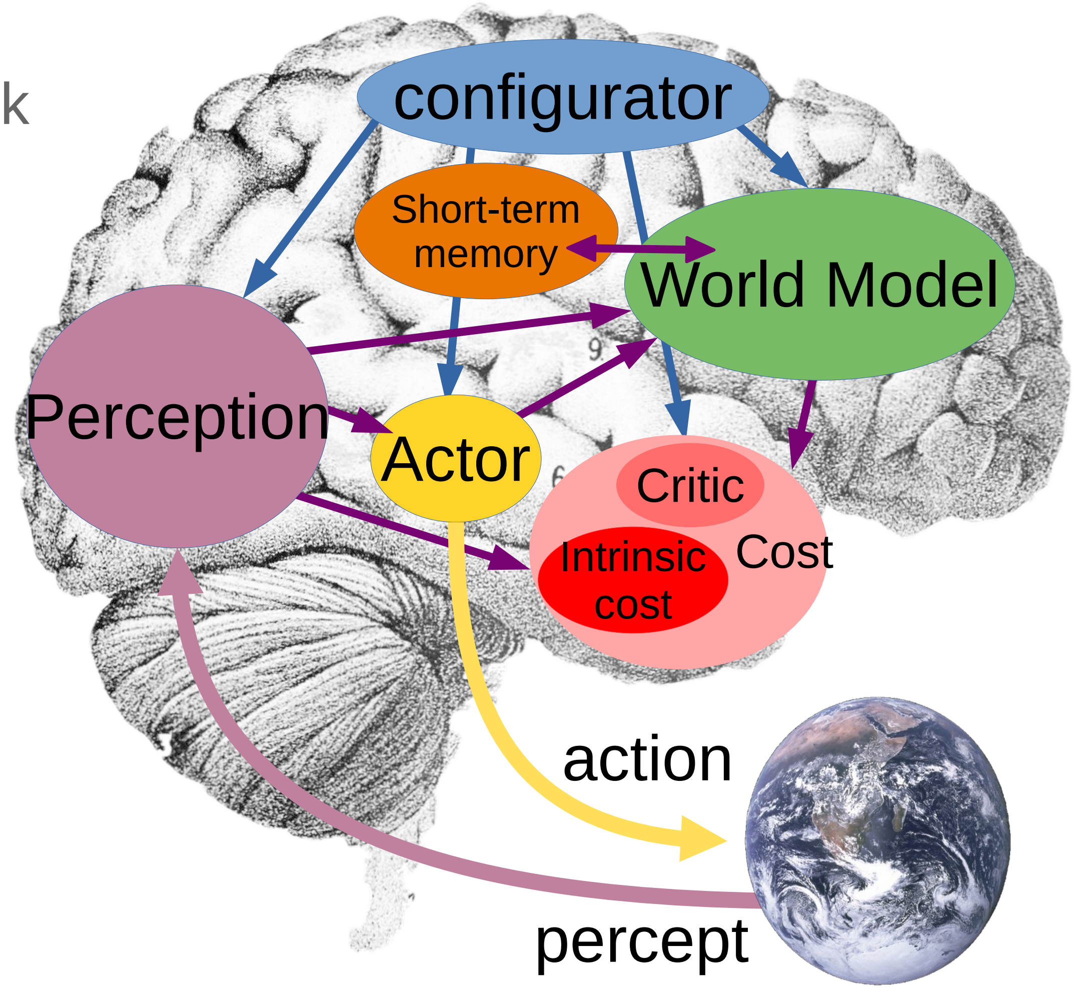
Position paper:
“A path towards autonomous machine intelligence”
https://openreview.net/forum?id=BZ5a1r-kVsf

Longer talk: search “LeCun Berkeley” on YouTube



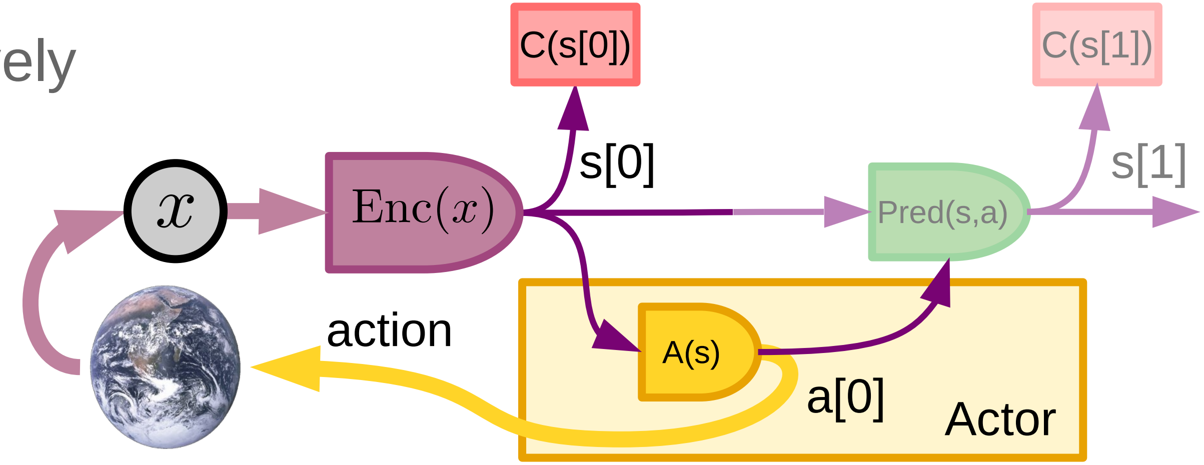
Modular Architecture for Autonomous AI

- **Configurator**
  - Configures other modules for task
- **Perception**
  - Estimates state of the world
- **World Model**
  - Predicts future world states
- **Cost**
  - Compute “discomfort”
- **Actor**
  - Find optimal action sequences
- **Short-Term Memory**
  - Stores state-cost episodes



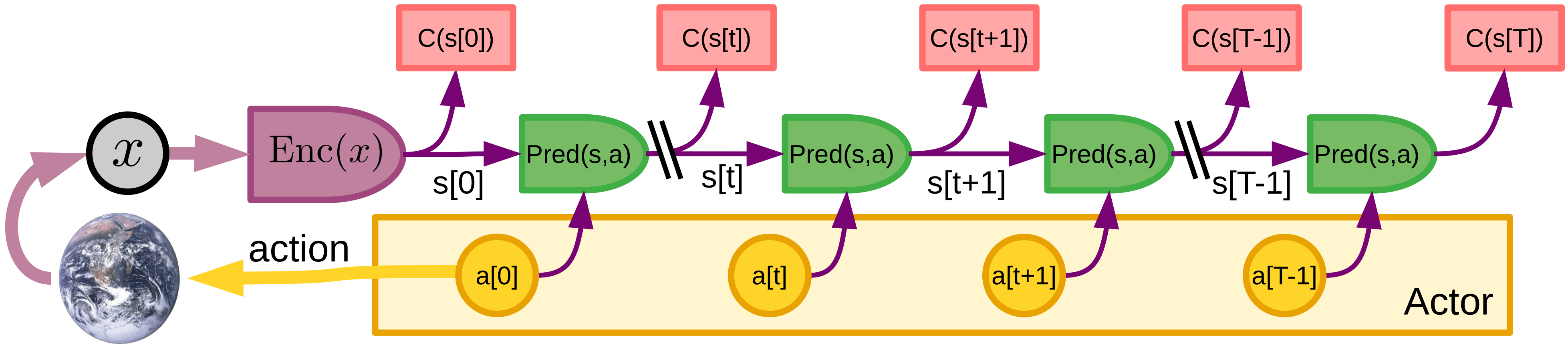
Mode-1 Perception-Action Cycle

- **Perception module s[0]=Enc(x)**
  - Extract representation of the world
- **Policy module A(s[0])**
  - Computes an action reactively
- **Cost module C(s[0])**
  - Computes cost of state
- **Optionally:**
  - World Model Pred(s,a)
  - Predicts future state
  - Stores states and costs in short-term memory



Mode-2 Perception-Planning-Action Cycle

- **Akin to classical Model-Predictive Control (MPC)**
- **Actor proposes an ation sequence**
- **World Model predicts outcome**
- **Actor optimizes action sequence to minimize cost**
  - e.g. using gradient descent, dynamic programming, MC tree search…
- **Actor sends first action(s) to effectors**

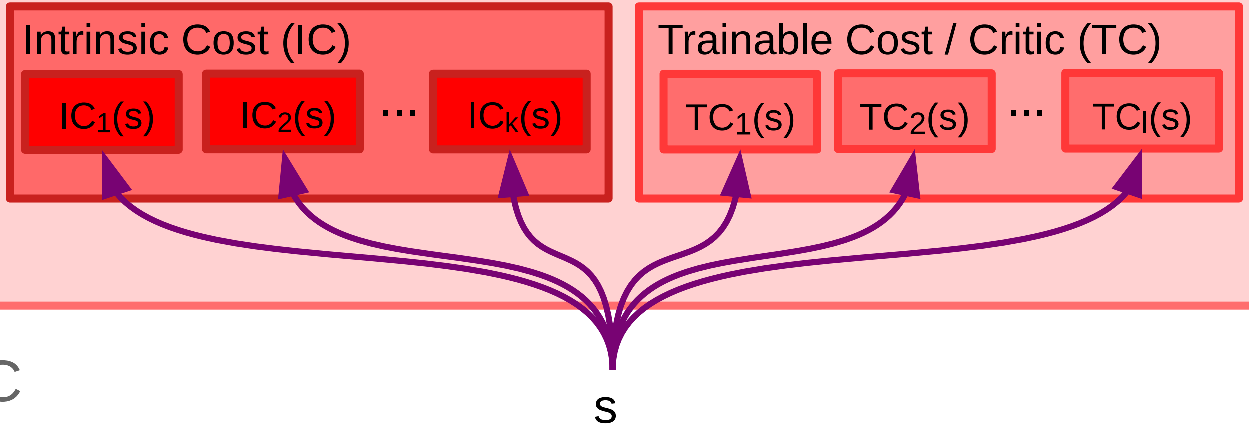


[Henaff et al ICLR 19],[Hafner et al. ICML 19],[Chaplot et al. ICML 21],[Escontrela CoRL 22],...

Cost Module

- **Intrinsic Cost (IC)**
  - Immutable cost modules.
  - Hard-wired drives.
- **Trainable Cost (TC)**
  - Trainable
  - Predicts future values of IC
  - Equivalent to a critic in RL
  - Implements subgoals
  - Configurable
- **All are differentiable**

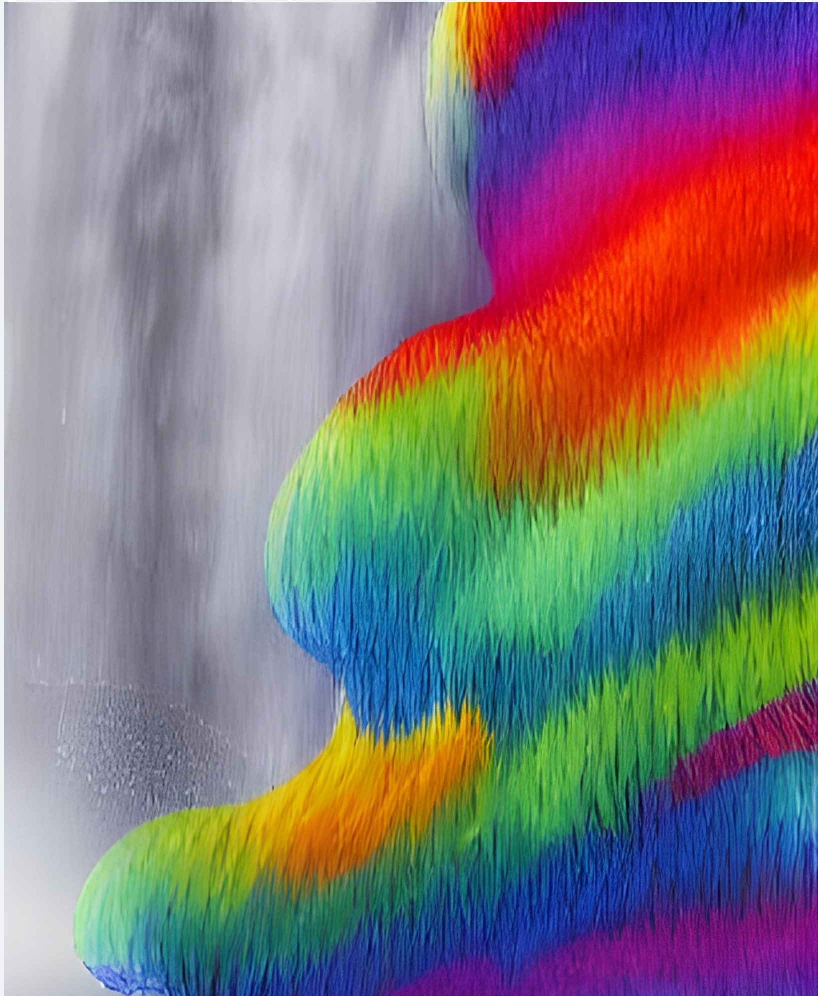
$$\mathrm{C}(s) = \mathrm{IC}(s) + \mathrm{ITC}(s) \; ; \;\; \mathrm{IC}(s) = \sum_{i=1}^{k} u_i \mathrm{IC}_i(s) \; ; \;\; \mathrm{TC}(s) = \sum_{j=1}^{l} v_j \mathrm{TC}_j(s)$$



NEW YORK UNIVERSITY Meta AI

Building & Training the World Model

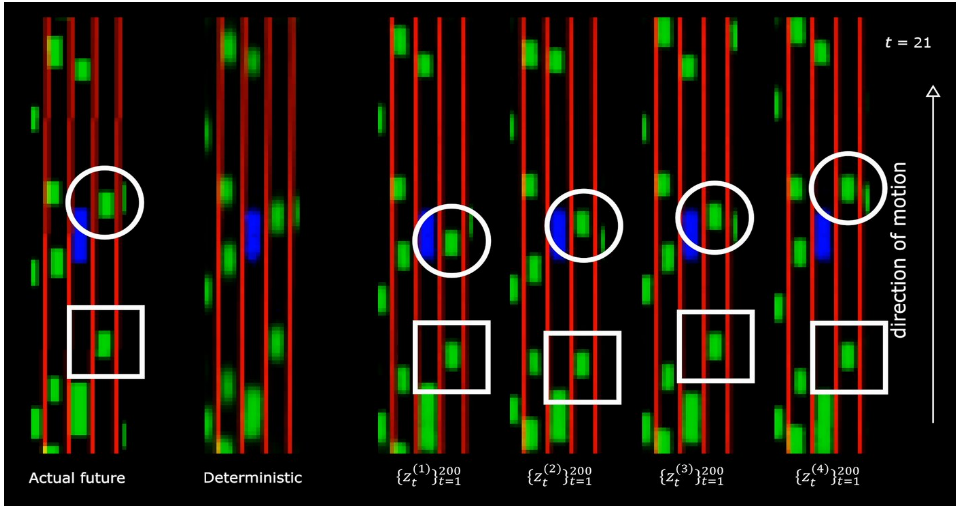
Energy-Based Models
Joint-Embedding Architecture



How do we represent uncertainty in the predictions?

- **The world is only partially predictable**
- **How can a predictive model represent multiple predictions?**
- **Probabilistic models are intractable in continuous domains.**
- **Generative Models must predict every detail of the world**
- **My solution: Joint-Embedding Predictive Architecture**

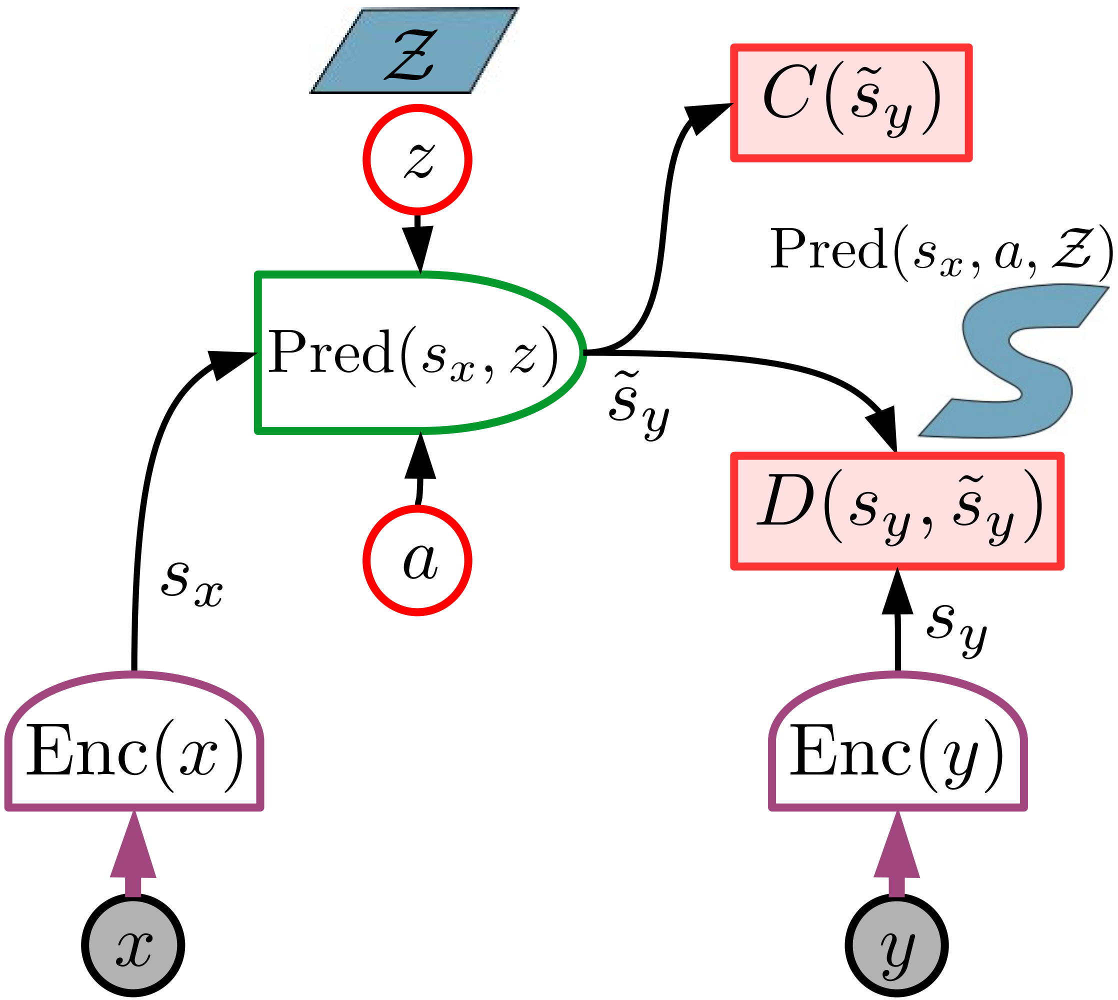
[Mathieu, Couprie, LeCun ICLR 2016]



[Henaff, Canziani, LeCun ICLR 2019]

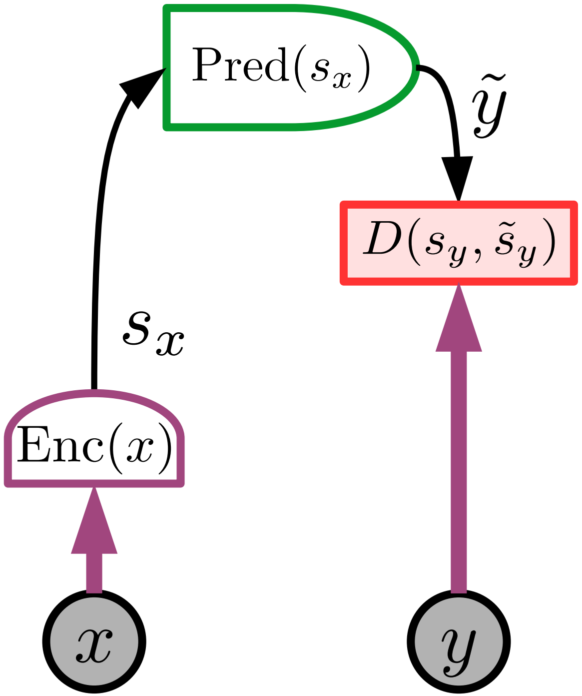
Architecture for the world model: JEPA

- **JEPA: Joint Embedding Predictive Architecture.**
  - x: observed past and present
  - y: future
  - a: action
  - z: latent variable (unknown)
  - D( ): prediction cost
  - C( ): surrogate cost
  - JEPA predicts a representation of the future Sy from a representation of the past and present Sx

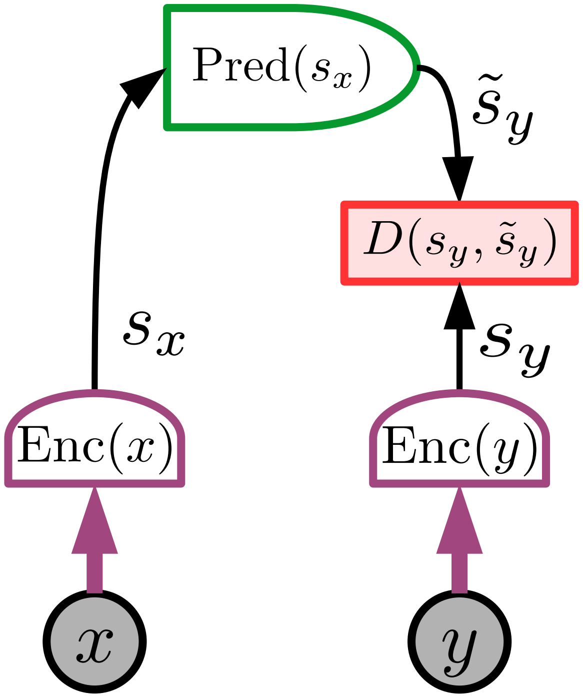


Architectures: Generative vs Joint Embedding

- **Generative: predicts y** (with all the details, including irrelevant ones)
- **Joint Embedding: predicts an abstract representation of y**



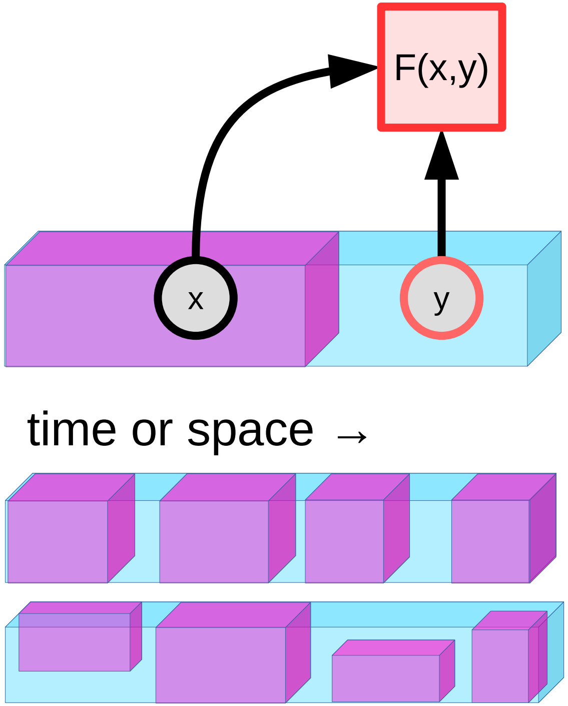
a) Generative Architecture
Examples: VAE, MAE...

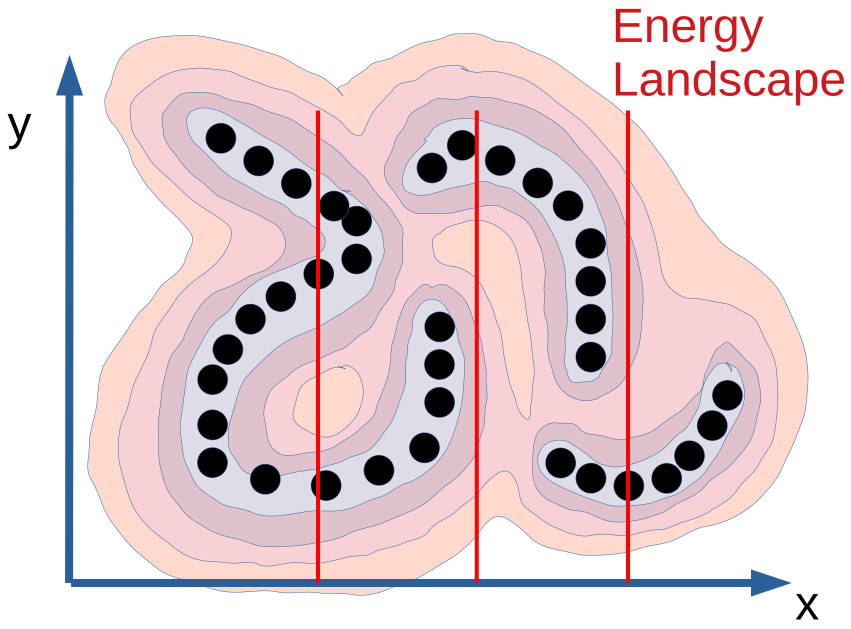


b) Joint Embedding Architecture

Energy-Based Models: Implicit function

- **The only way to formalize & understand all model types**
  - Gives low energy to compatible pairs of x and y
  - Gives higher energy to incompatible pairs

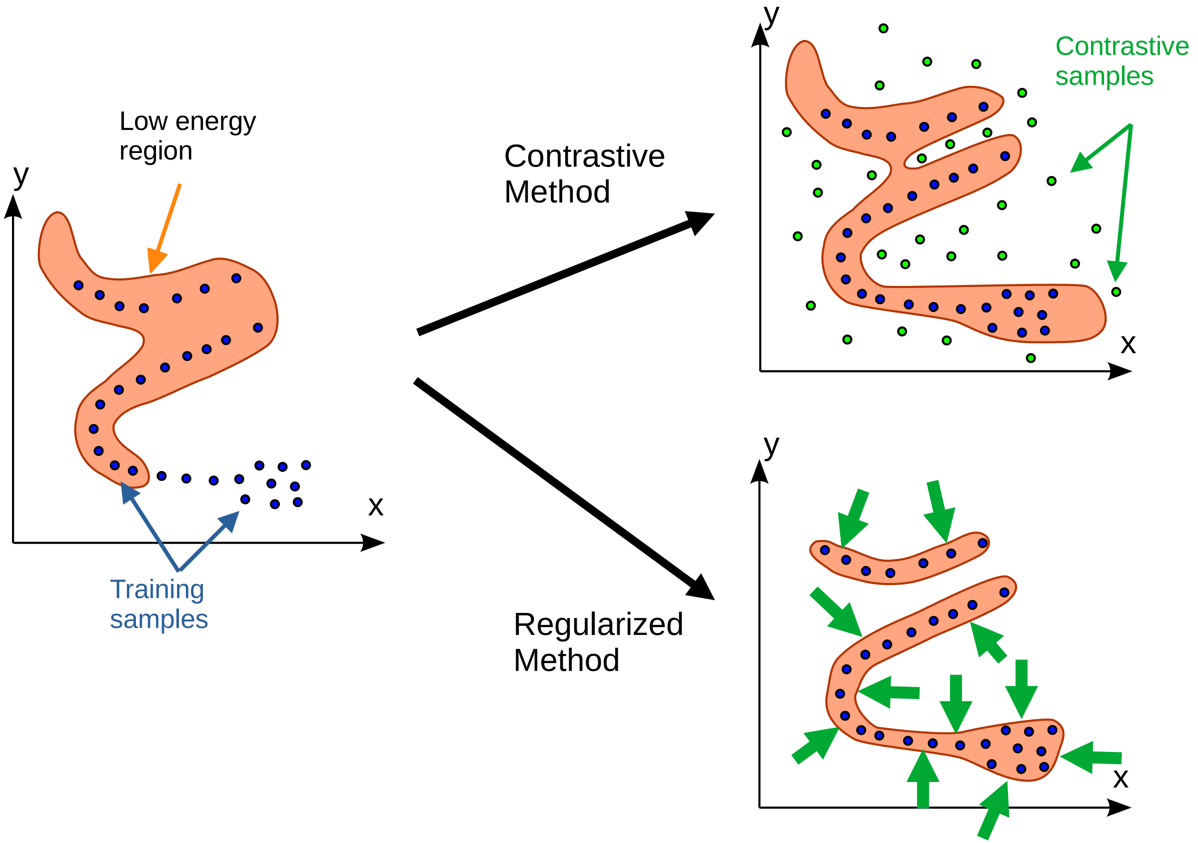




$$\check{y} = \mathrm{argmin}_y F(x, y)$$

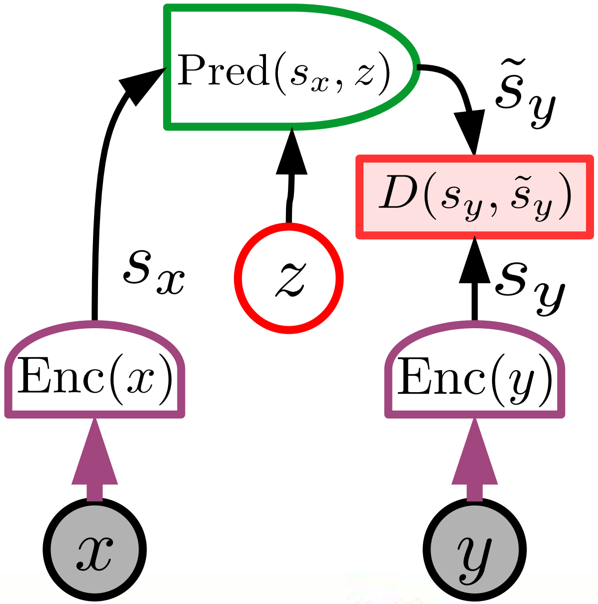
EBM Training: two categories of methods

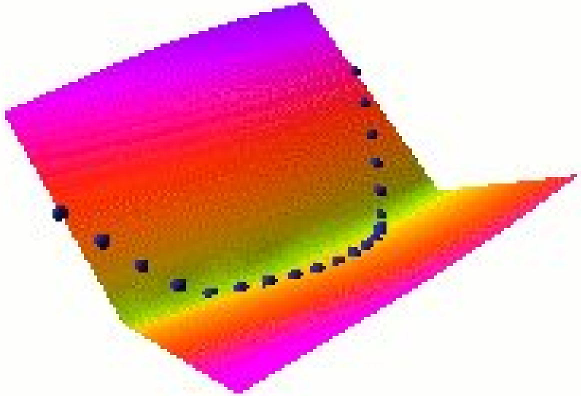
- **Contrastive methods**
  - Push down on energy of training samples
  - Pull up on energy of suitably-generated contrastive samples
  - Scales very badly with dimension
- **Regularized Methods**
  - Regularizer minimizes the volume of space that can take low energy



Recommendations:

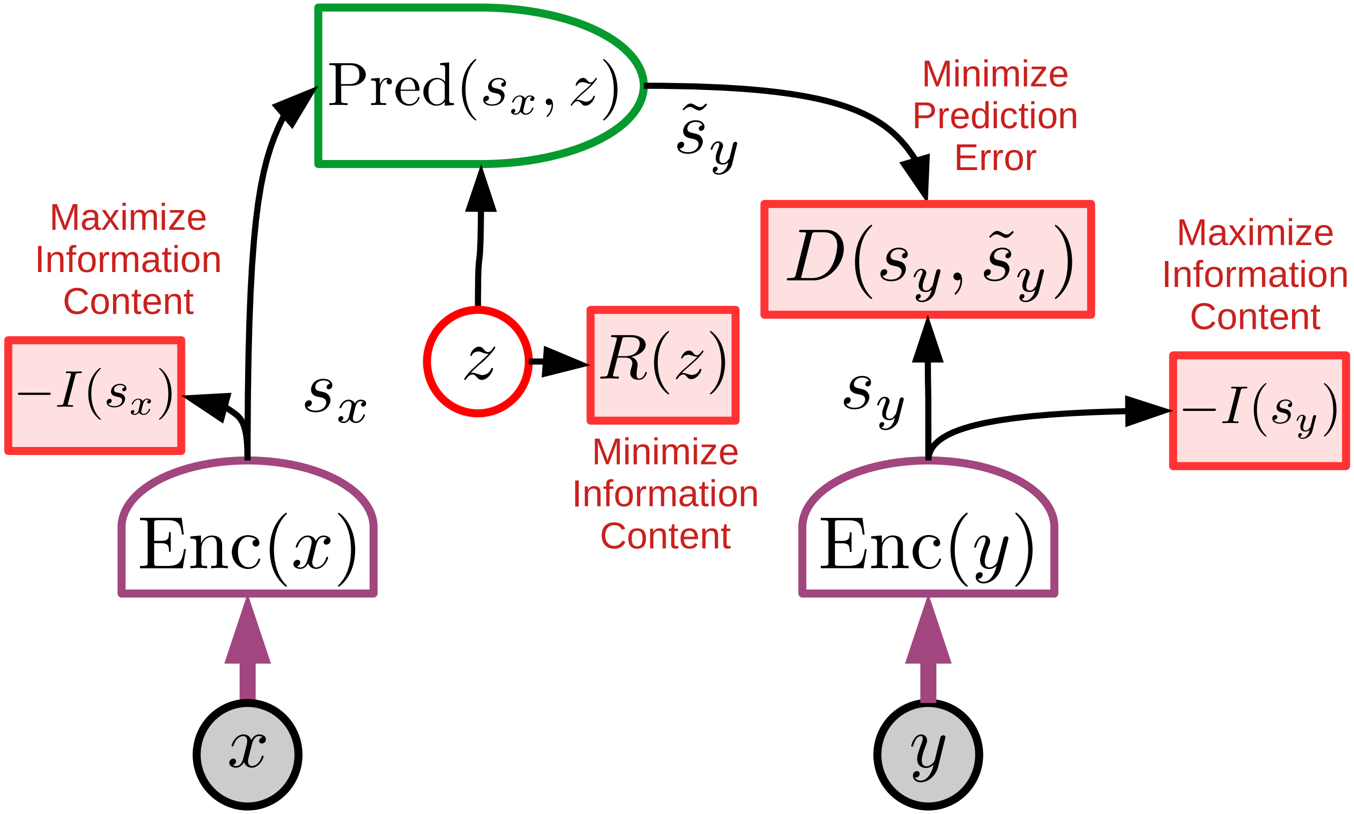
- **Abandon generative models**
  - in favor joint-embedding architectures
  - Abandon Auto-Regressive generation
- **Abandon probabilistic model**
  - in favor of energy-based models
- **Abandon contrastive methods**
  - in favor of regularized methods
- **Abandon Reinforcement Learning**
  - In favor of model-predictive control
  - Use RL only when planning doesn't yield the predicted outcome, to adjust the world model or the critic.





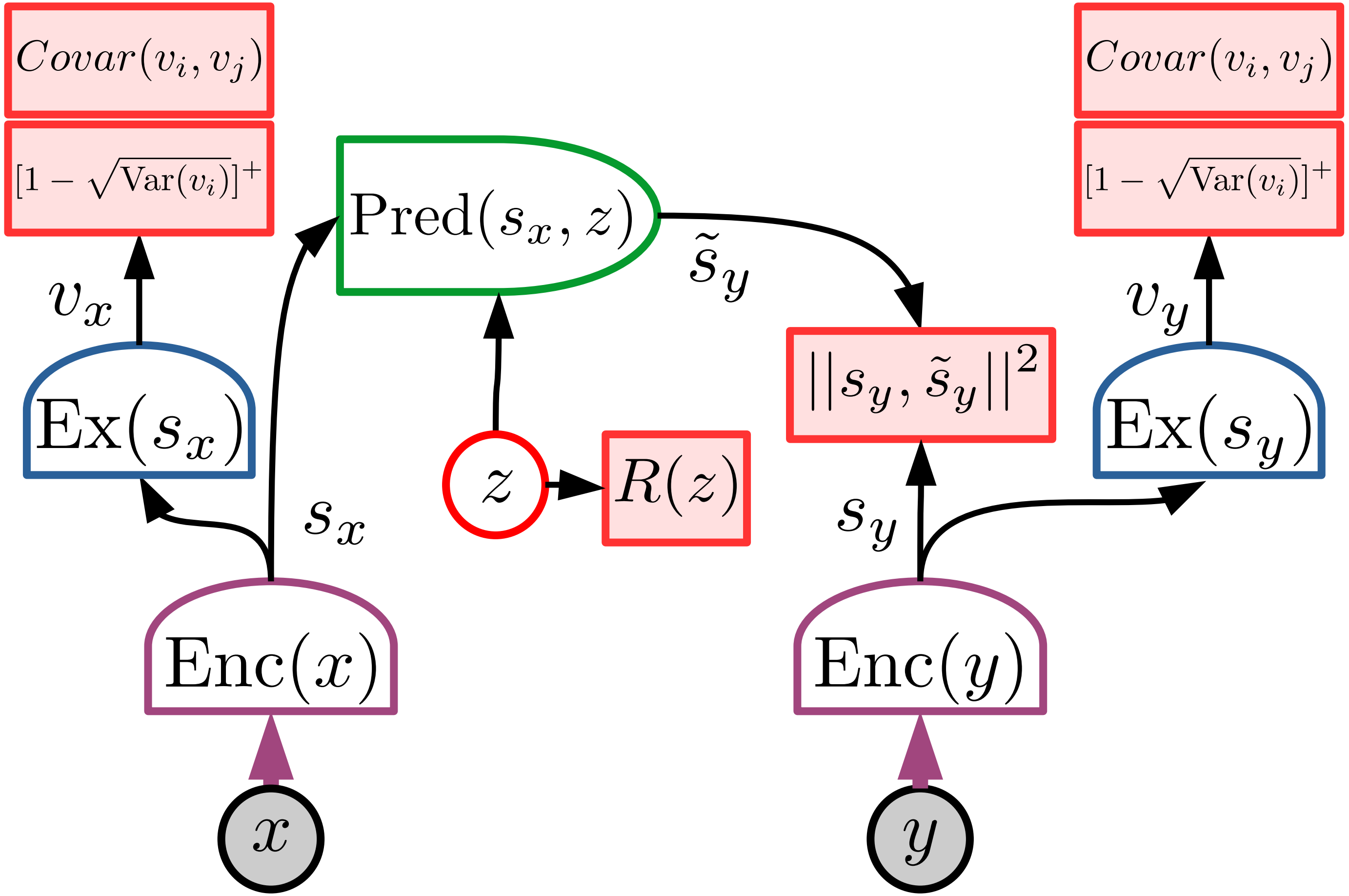
Training a JEPA non contrastively

- **Four terms in the cost**
  - Maximize information content in representation of x
  - Maximize information content in representation of y
  - Minimize Prediction error
  - Minimize information content of latent variable z



VICReg: Variance, Invariance, Covariance Regularization

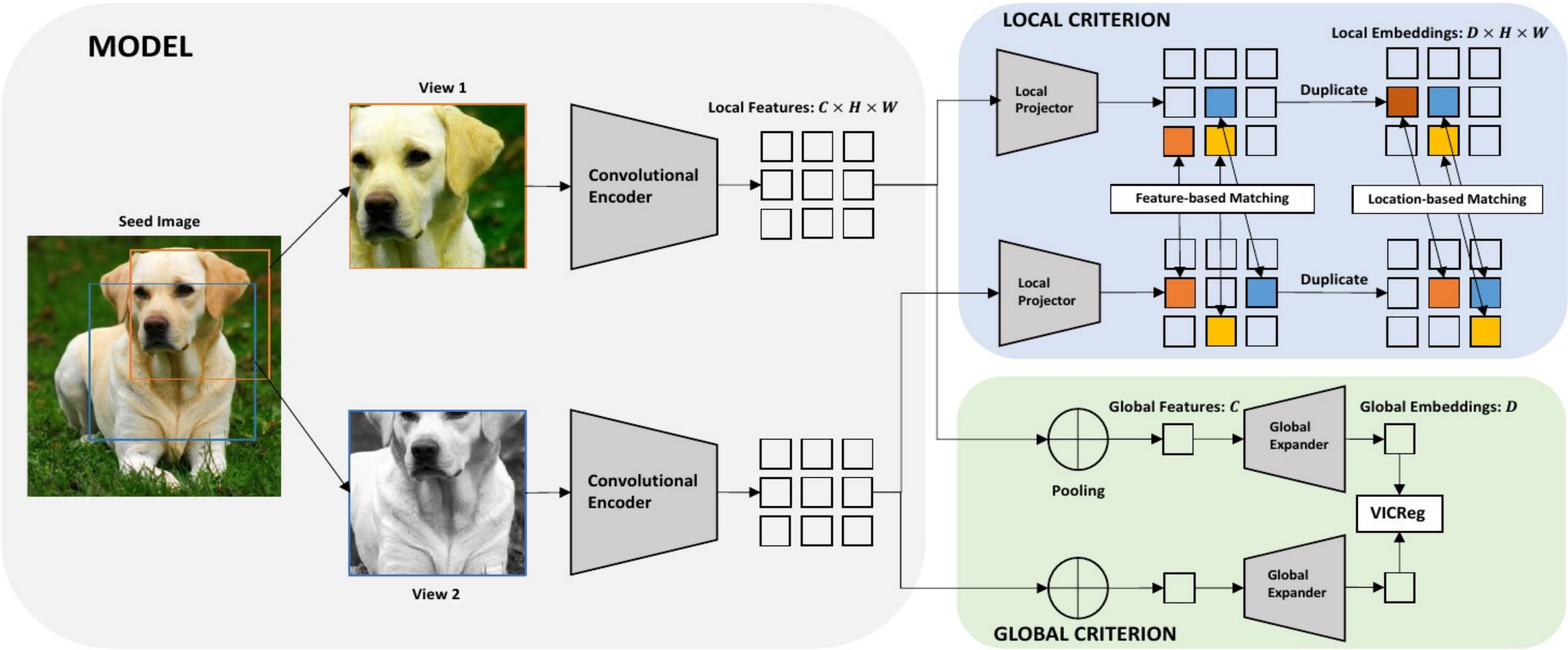
- **Variance:**
  - Maintains variance of components of representations
- **Covariance:**
  - Decorrelates components of covariance matrix of representations
- **Invariance:**
  - Minimizes prediction error.



Barlow Twins [Zbontar et al. ArXiv:2103.03230], VICReg [Bardes, Ponce, LeCun arXiv:2105.04906, ICLR 2022], VICRegL [Bardes et al. NeurIPS 2022], MCR2 [Yu et al. NeurIPS 2020][Ma, Tsao, Shum, 2022]

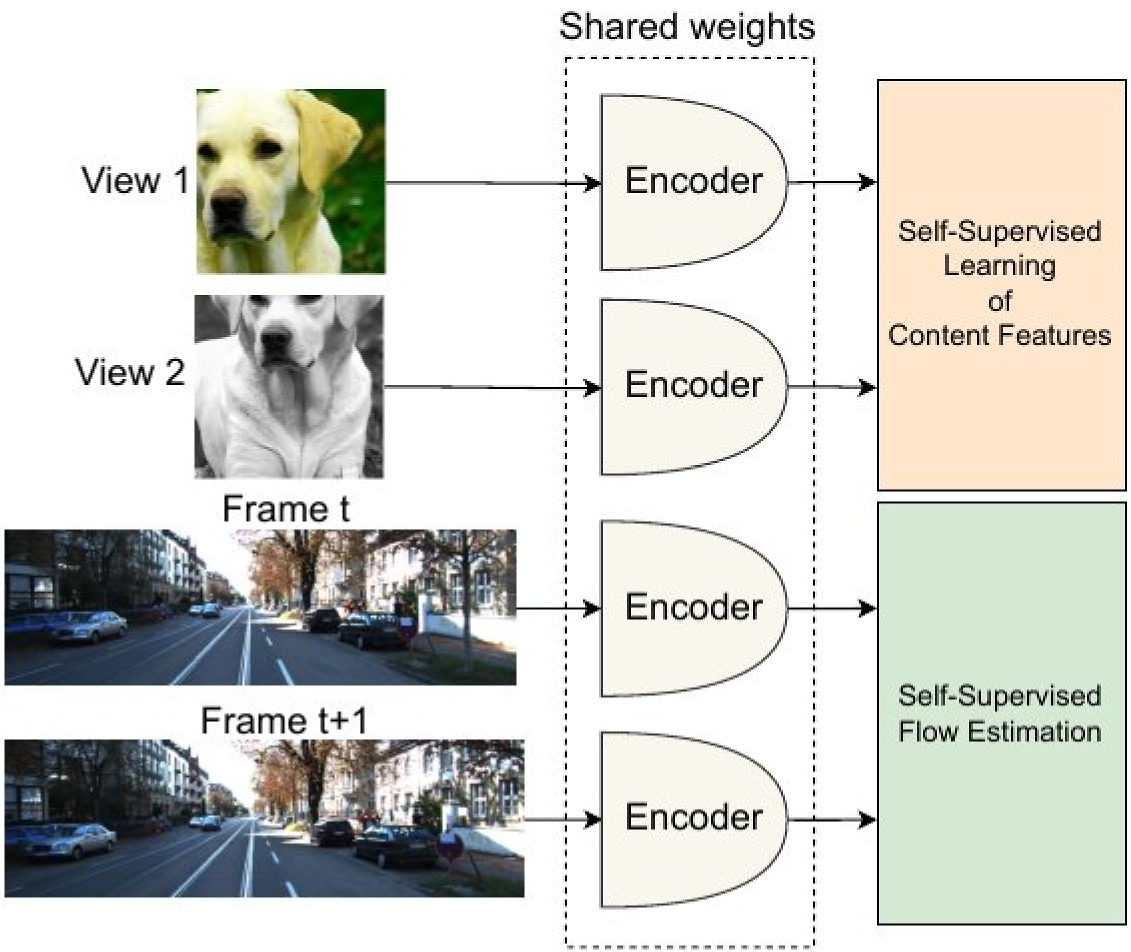
VICRegL: local matching latent variable for segmentation

- **Latent variable optimization:**
  - Finds a pairing between local feature vectors of the two images
  - [Bardes, Ponce, LeCun NeurIPS 2022, arXiv:2210.01571]



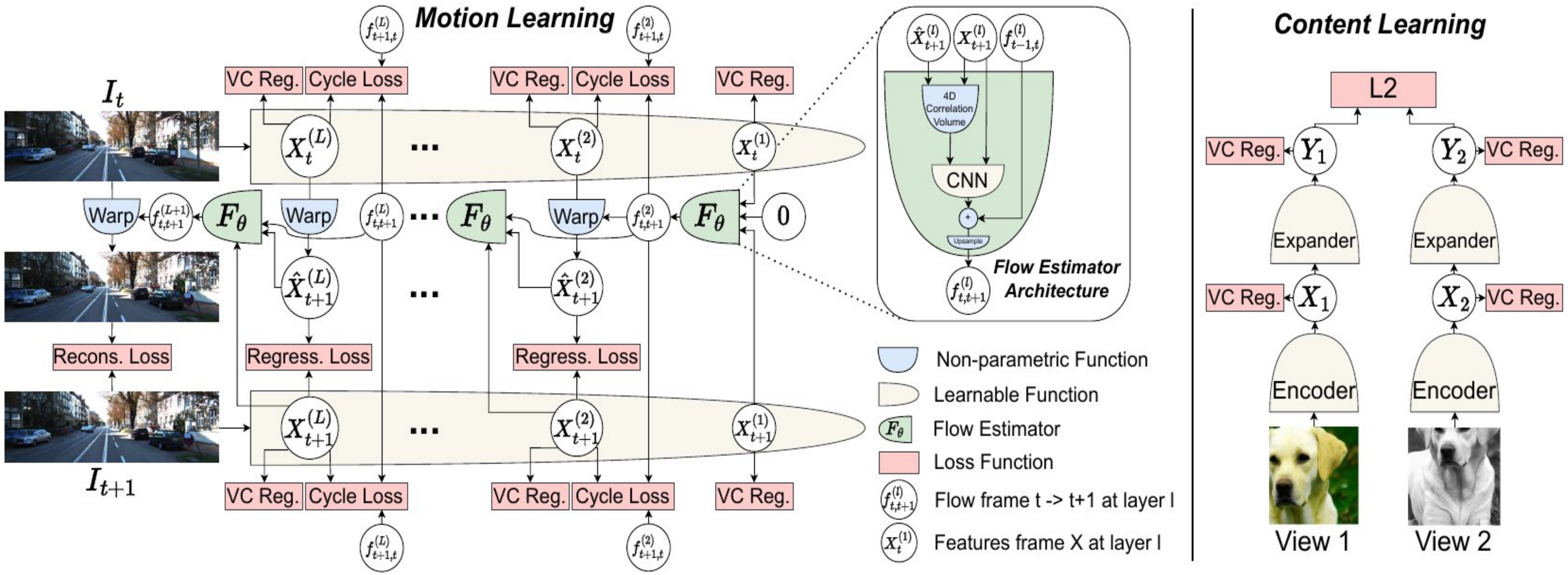
MC-JEPA: Motion & Content JEPA

- **Simultaneous SSL for**
  - Image recognition
  - Motion estimation
- **Trained on**
  - ImageNet 1k
  - Various video datasets
- **Uses VCReg to prevent collapse**
  - ConvNext-T backbone



MC-JEPA: Motion & Content JEPA

- Motion estimation architecture uses a top-down hierarchical predictor that “warp” feature maps.



MC-JEPA: Optical Flow Estimation Results

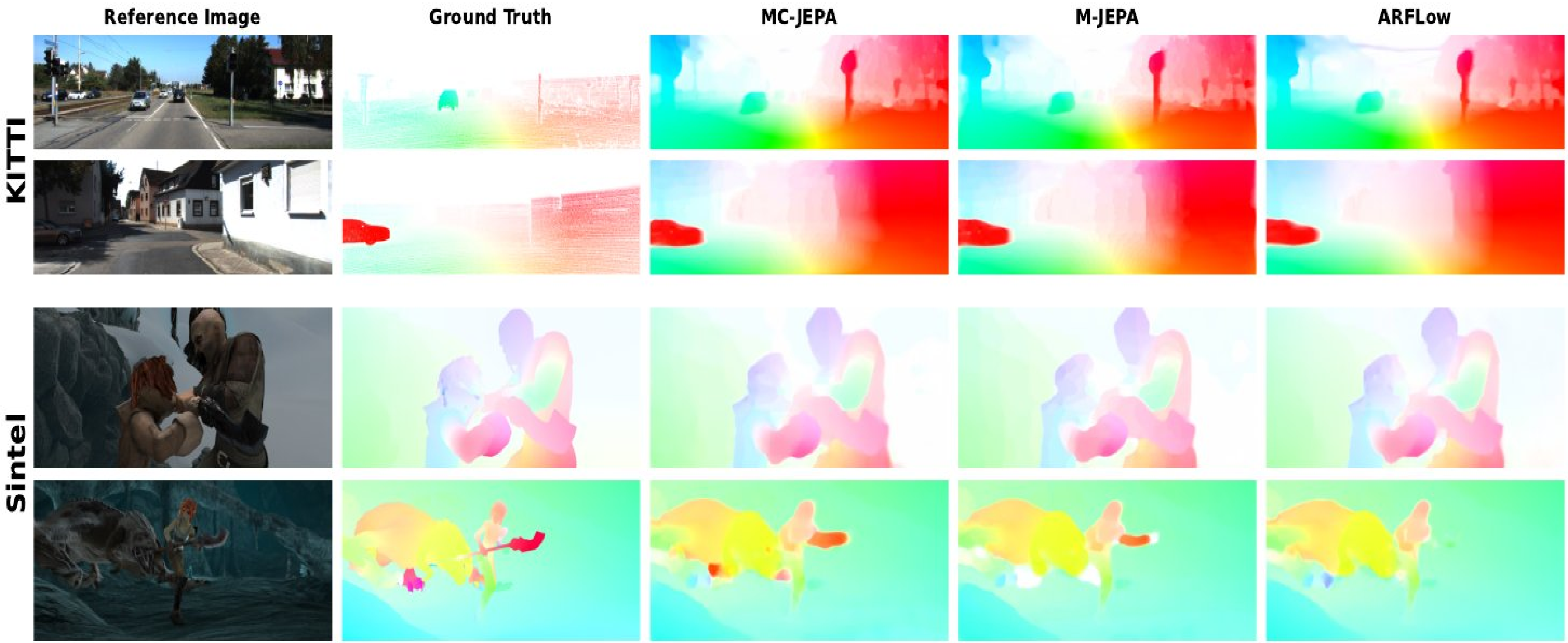
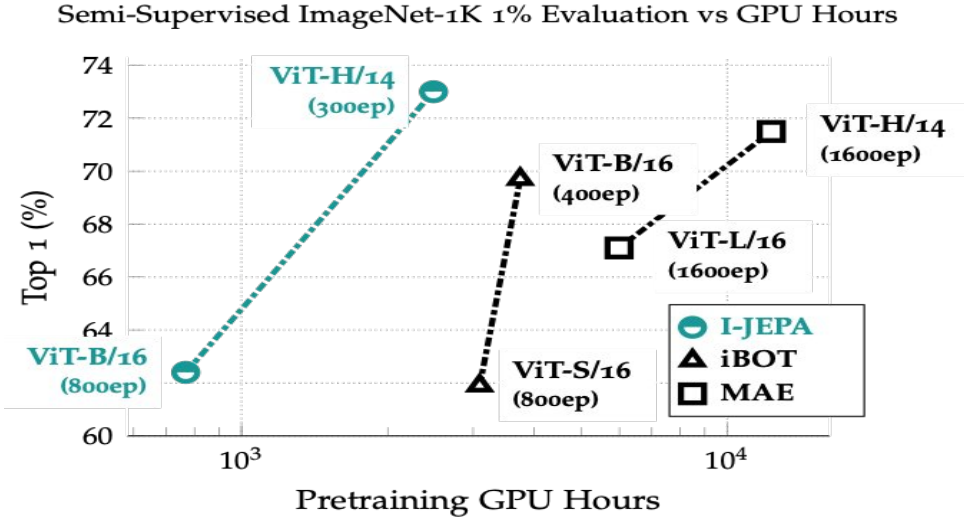
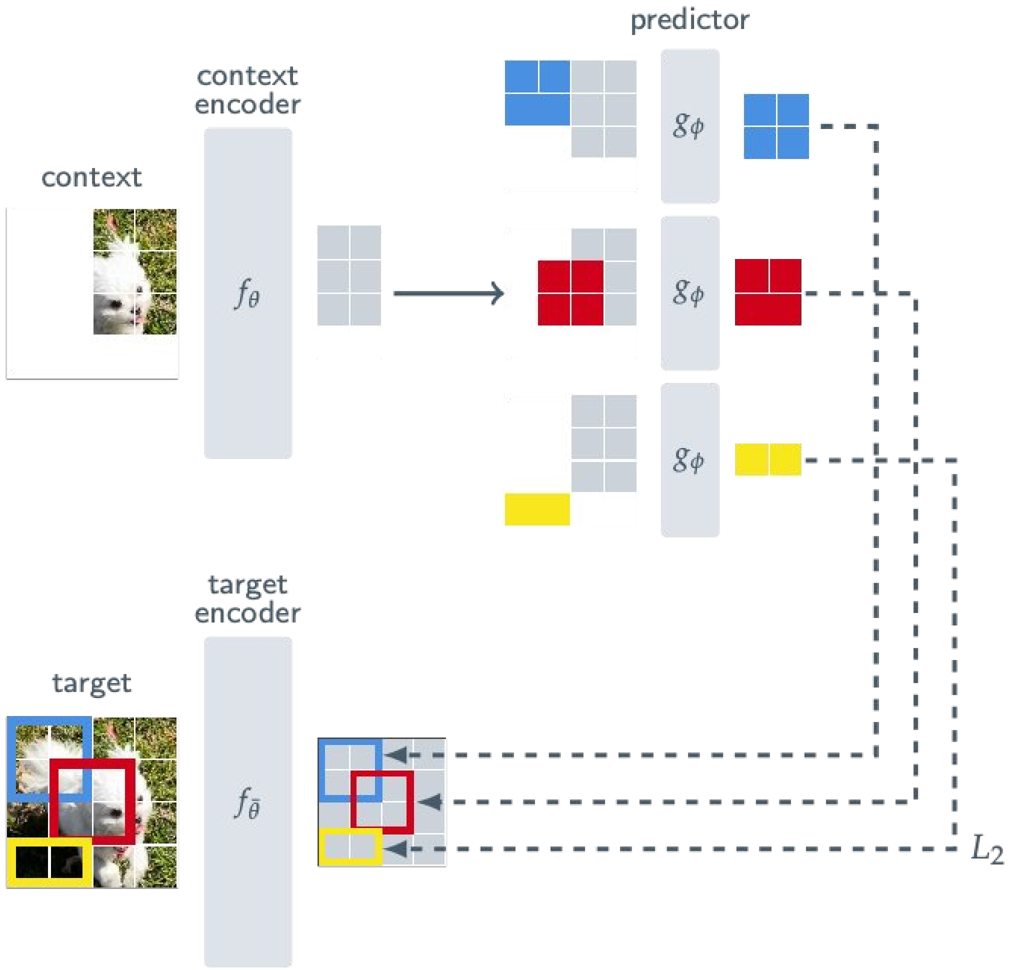


Image-JEPA: uses masking, transformer, EMA weights

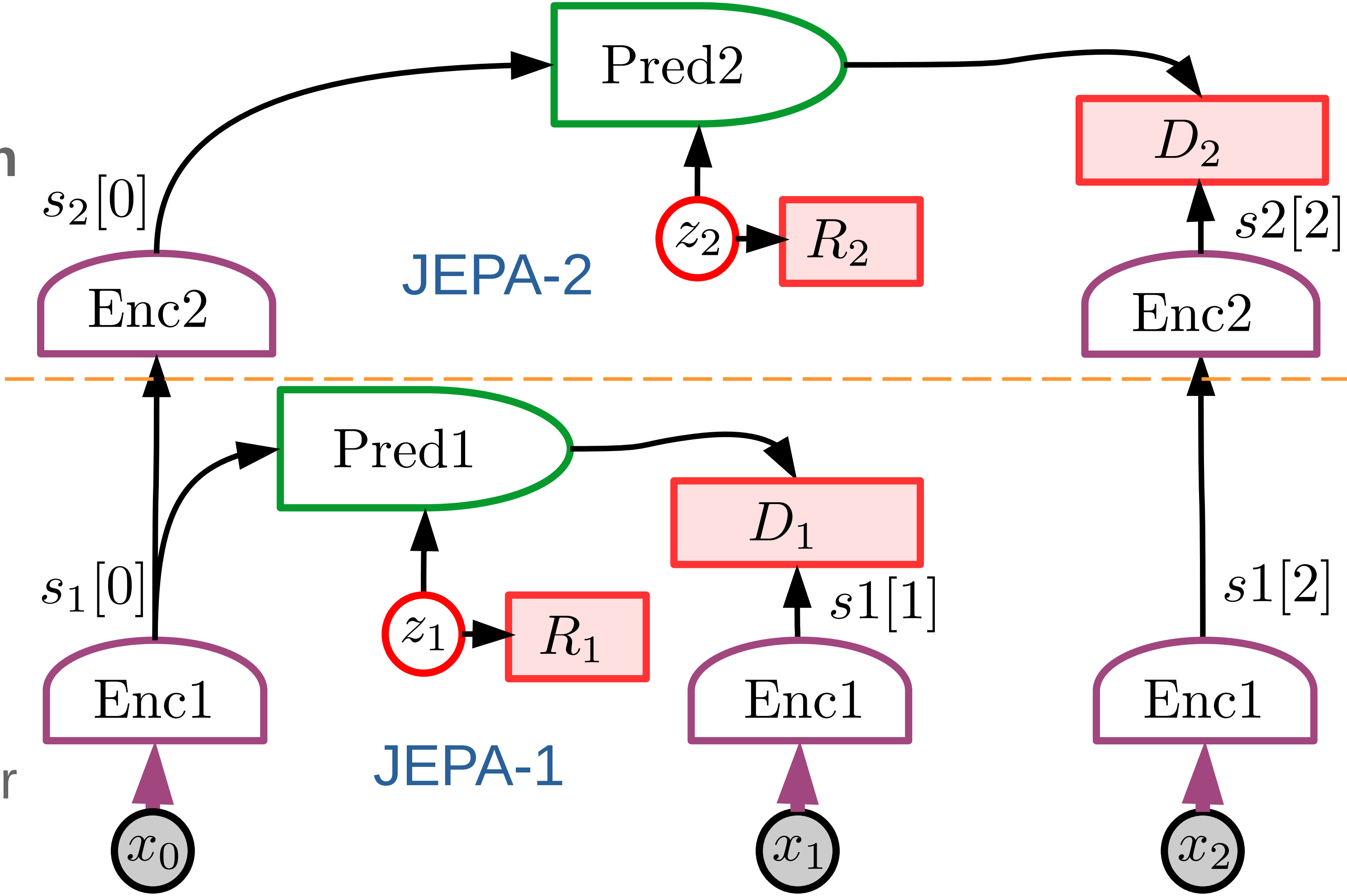
- **"SSL from images with a JEPA"**
  - M. Assran et al arxiv:2301.08243
- **Jointly embeds a context and a number of neighboring patches.**
  - Uses predictors
  - Uses only masking





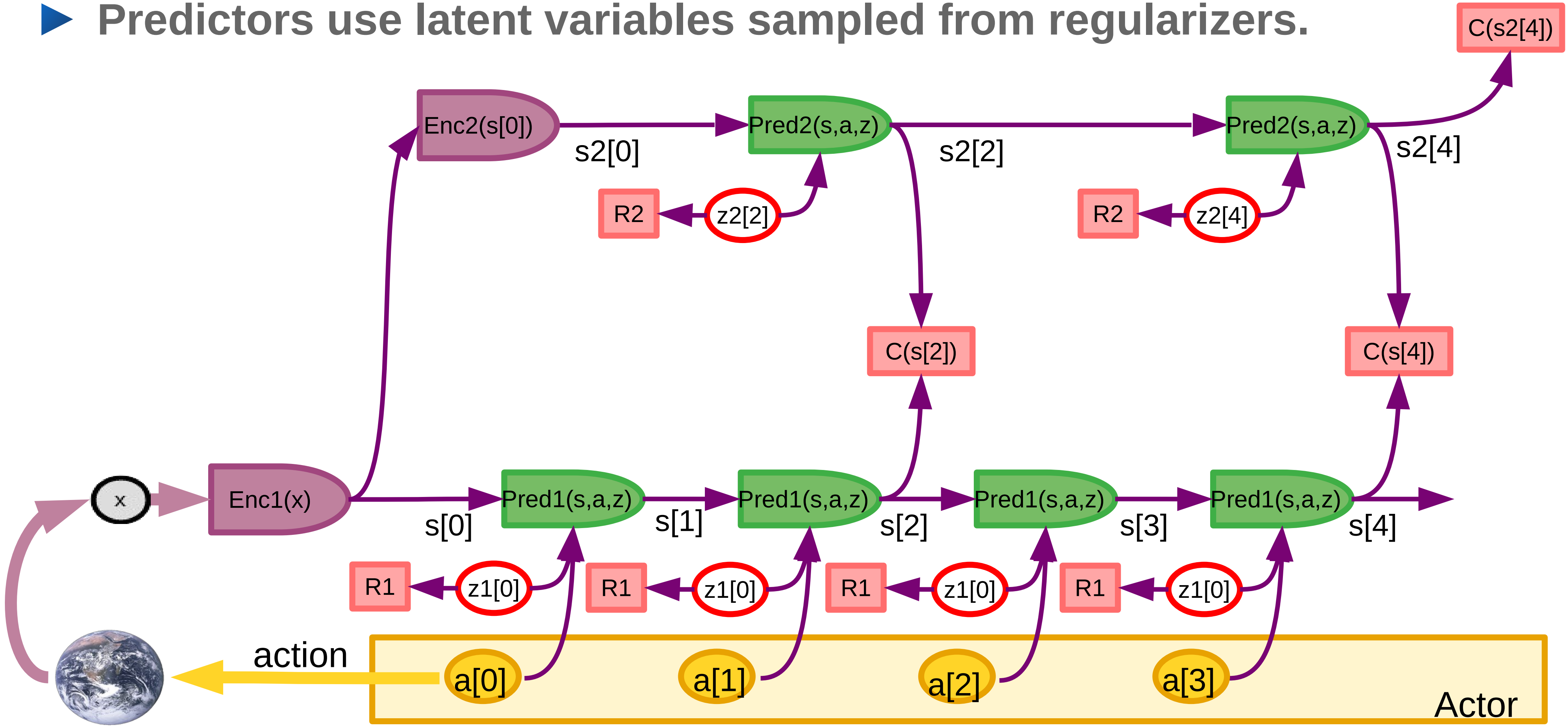
Hierarchical Prediction at Multiple Time-Scales & Abstraction Levels

- **Low-level representations can only predict in the short term.**
  - Too much details
  - Prediction is hard
- **Higher-level representations can predict in the longer term.**
  - Less details.
  - Prediction is easier



Hierarchical Planning with Uncertainty

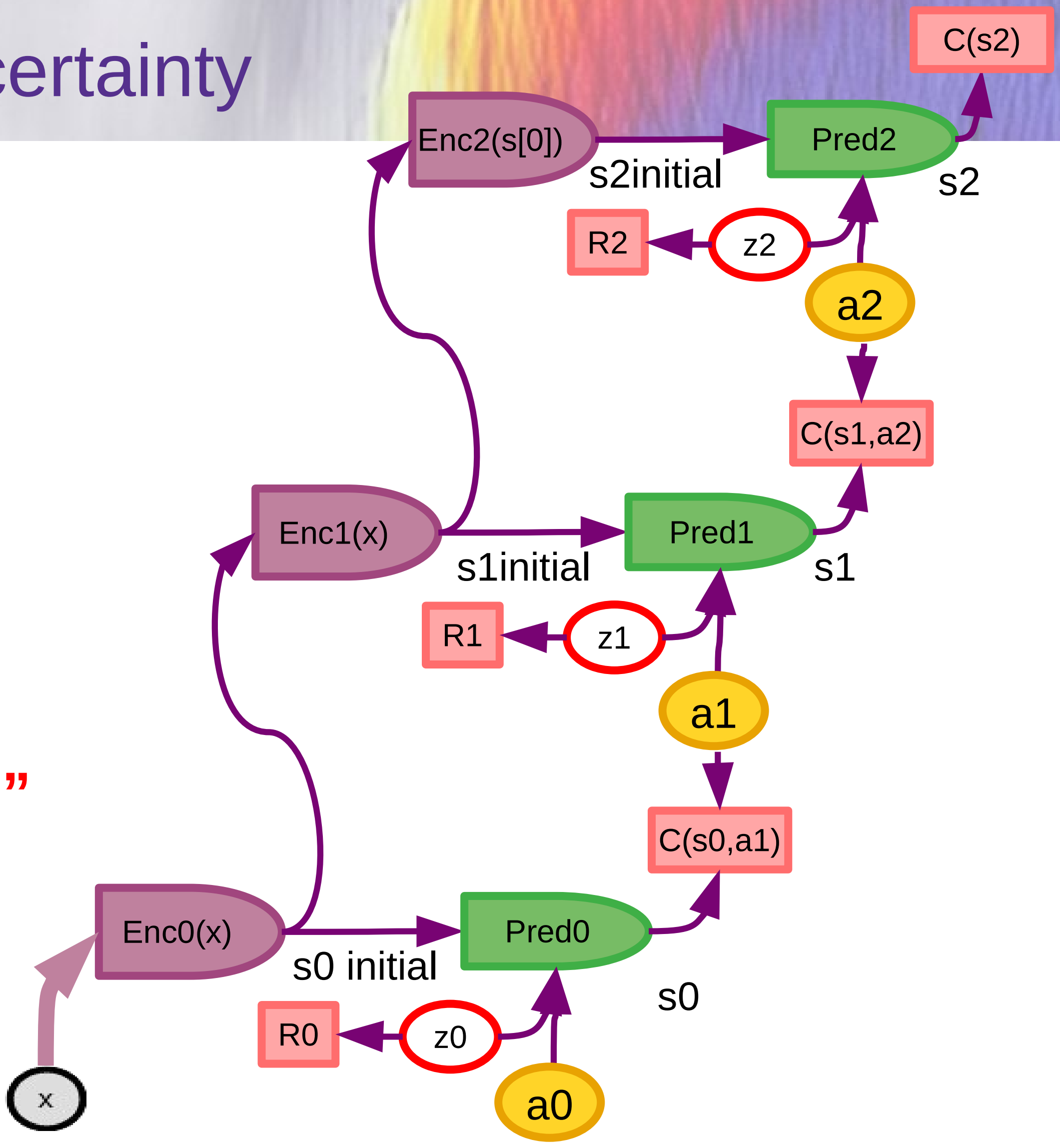
- Predictors use latent variables sampled from regularizers.



Hierarchical Planning with Uncertainty

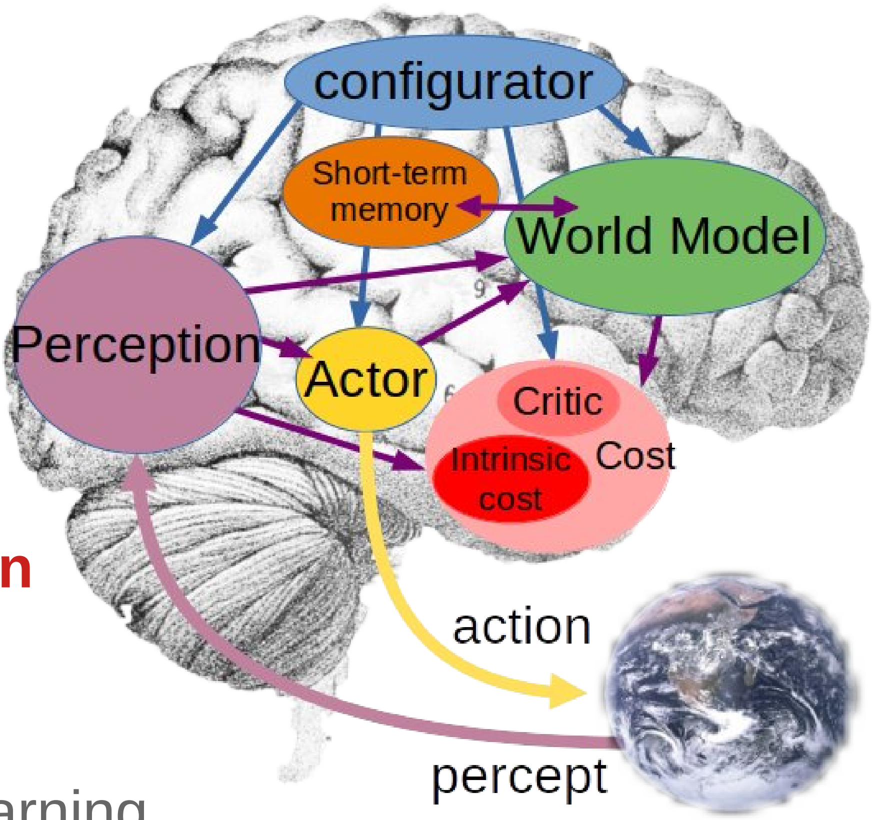
- **Hierarchical world model**
- **Hierarchical planning**
- **An action at level k specifies an objective for level k-1**
- **Prediction in higher levels are more abstract and longer-range.**

- **This type of planning/reasoning by minimizing a cost w.r.t "action" variables is what's missing from current architectures**
  - Including AR-LLMs, multimodal systems, learning robots,...



Steps towards Autonomous AI Systems

- **Self-Supervised Learning**
  - To learn representations of the world
  - To learn predictive models of the world
- **Handling uncertainty in predictions**
  - Joint-embedding predictive architectures
  - Energy-Based Model framework
- **Learning world models from observation**
  - Like animals and human babies?
- **Reasoning and planning**
  - That is compatible with gradient-based learning
  - No symbols, no logic → vectors & continuous functions



Positions / Conjectures

- **Prediction is the essence of intelligence**
  - Learning predictive models of the world is the basis of common sense

  Almost everything is learned through self-supervised learning
  - Low-level features, space, objects, physics, abstract representations…
  - Almost nothing is learned through reinforcement, supervision or imitation
- **Reasoning == simulation/prediction + optimization of objectives**
  - Computationally more powerful than auto-regressive generation.
- **H-JEPA with non-contrastive training is the thing**
  - Probabilistic generative models and contrastive methods are doomed.
- **Intrinsic cost & architecture drive behavior & determine what is learned**
- **Emotions are necessary for autonomous intelligence**
  - Anticipation of outcomes by the critic or world model+intrinsic cost.

Challenges for AI Research

- Finding a **general recipe** for training Hierarchical Joint Embedding Architectures-based **World Models** from video, image, audio, text…
- Designing **surrogate costs** to drive the H-JEPA to learn relevant representations (prediction is just one of them)
- Integrating an H-JEPA into an **agent capable of planning/reasoning**
- Devising **inference procedures for hierarchical planning** in the presence of uncertainty (gradient-based methods, beam search, MCTS,….)
- **Minimizing the use of RL** to situations where the model or the critic are inaccurate and lead to unforeseen outcomes.
- **Scaling**

Thank you!