



# Objective-Driven AI

Towards AI systems that can learn,  
remember, reason, plan,  
have common sense,  
yet are steerable and safe

**Yann LeCun**

New York University

Meta – Fundamental AI Research

Ding-Shum Lecture  
Department of Mathematics  
Harvard University  
2024-03-28



# 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) works great but...
  - ▶ Generative prediction only works for text and other discrete modalities
- ▶ Animals and humans:
  - ▶ Can learn new tasks **very** quickly.
  - ▶ Understand how the world works
  - ▶ Can reason an plan
- ▶ **Humans and animals have common sense**
- ▶ **There behavior is driven by objectives (drives)**



# We Need Human-Level AI for Intelligent Assistant

- ▶ **In the near future, all of our interactions with the digital world will be mediated by AI assistants.**
- ▶ **Smart glasses**
  - ▶ Communicates through voice, vision, display, electro-myogram interfaces (EMG)
- ▶ **Intelligent Assistant**
  - ▶ Can answer all of our questions
  - ▶ Can help us in our daily lives
  - ▶ Understands our preferences and interests
- ▶ **For this, we need machines with human-level intelligence**
  - ▶ Machines that understand how the world works
  - ▶ Machines that can remember, reason, plan.



“Her”  
(2013)

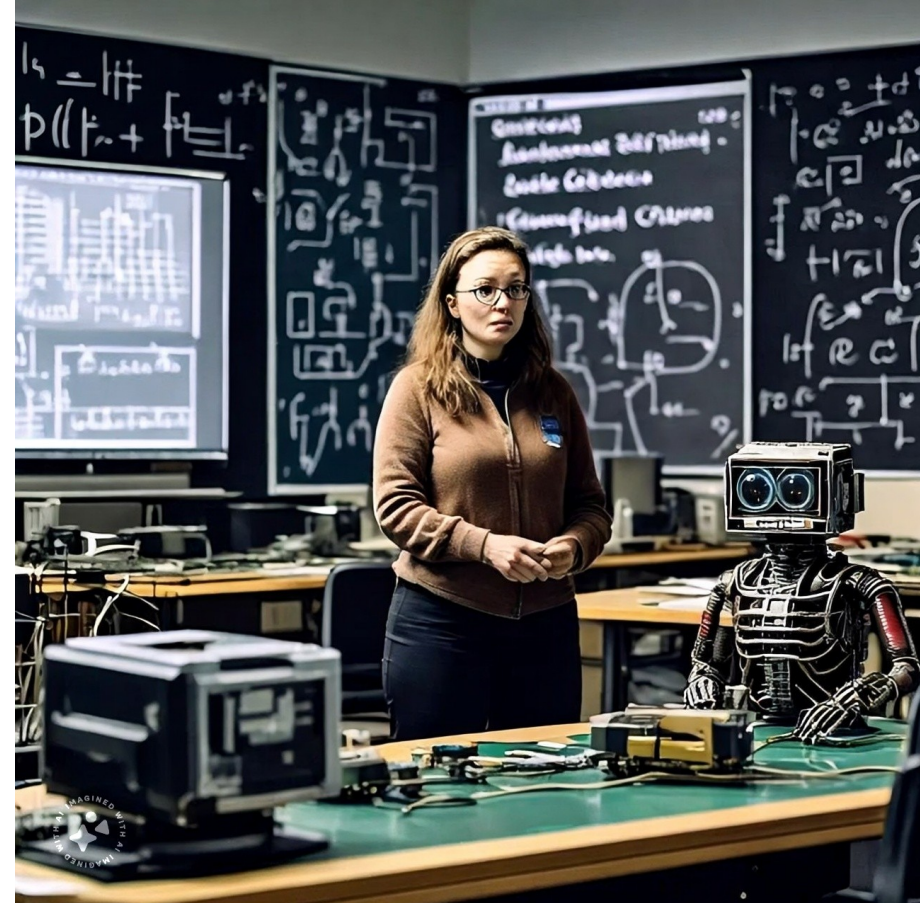


# Future AI Assistants need Human-Level AI

- ▶ **AI assistants will require (super-)human-level intelligence**
  - ▶ Like having a staff of smart “people” working for us
- ▶ **But, we are nowhere near human-level AI today**
  - ▶ Any 17 year-old can learn to drive in 20 hours of training
  - ▶ Any 10 year-old can learn to clear the dinner table in one shot
  - ▶ Any house cat can plan complex actions
- ▶ **What are we missing?**
  - ▶ Learning how the world works (not just from text)
  - ▶ World models. Common sense
  - ▶ Memory, Reasoning, Hierarchical Planning

# Desiderata for AMI (Advanced Machine Intelligence)

- ▶ **Systems that learn world models from sensory inputs**
  - ▶ E.g. learn intuitive physics from video
- ▶ **Systems that have persistent memory**
  - ▶ Large-scale associative memories
- ▶ **Systems that can plan actions**
  - ▶ So as to fulfill an objective
- ▶ **Systems that are controllable & safe**
  - ▶ By design, not by fine-tuning.
- ▶ **Objective-Driven AI Architecture**



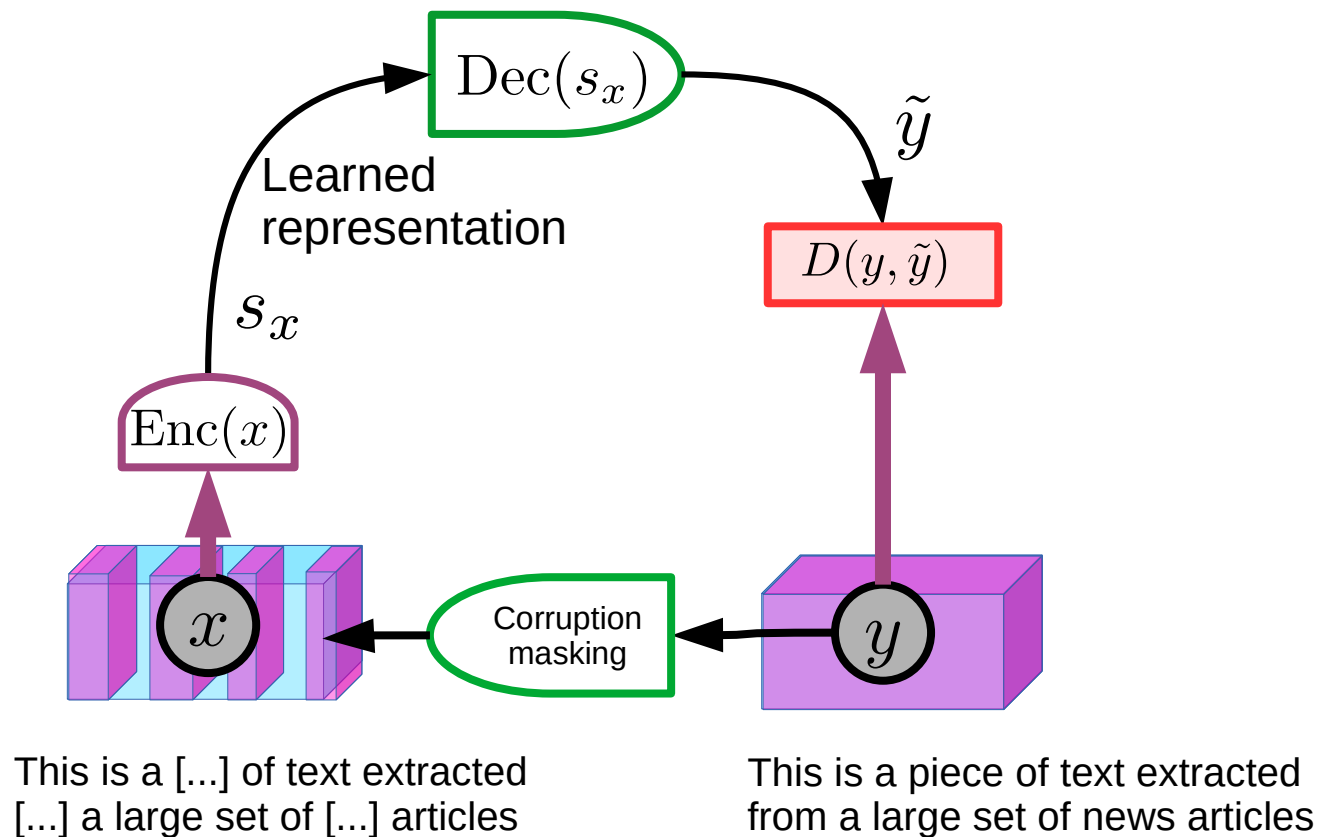
# Self-Supervised Learning has taken over the world

For understanding and generating text,  
images, video, 3D models, speech,  
proteins,...



# Self-Supervised Learning via Denoising / Reconstruction

- Denoising Auto-Encoder [Vincent 2008], BERT [Devlin 2018], RoBERTa [Ott 2019]



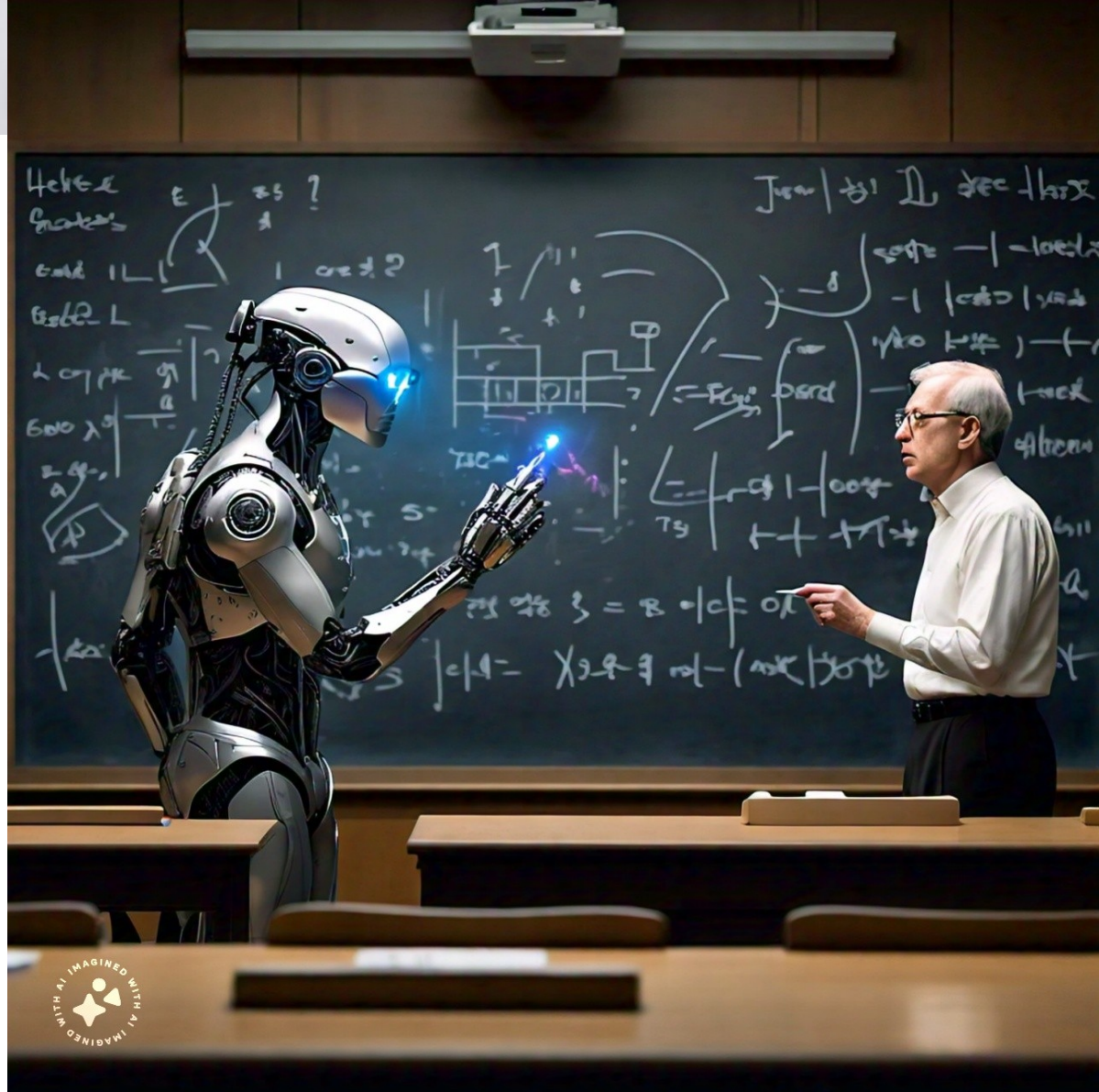
# Emu: image generation

[ArXiv:2309.15807]

Dai et al.:  
Emu: Enhancing Image Generation  
Models Using Photogenic Needles in a  
Haystack

AI at Meta, September 2023

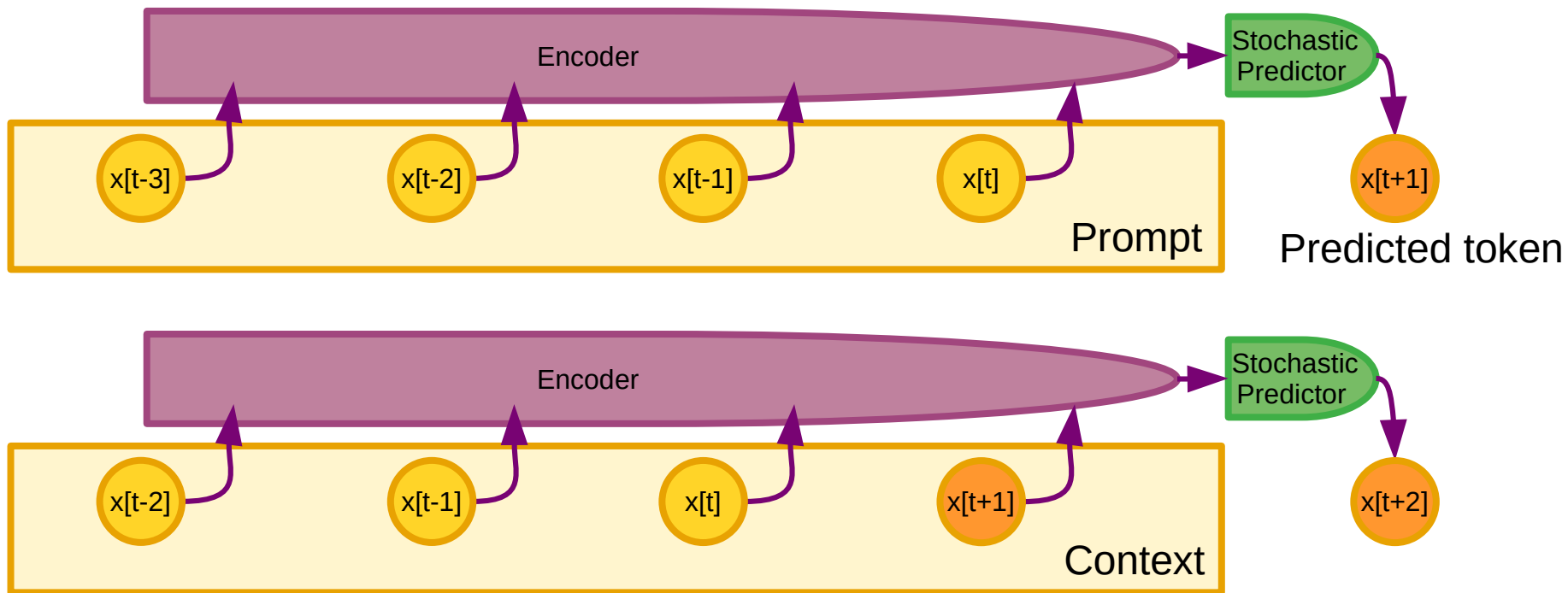
**Meta AI on WhatsApp & Messenger:**  
/imagine a photo a Harvard  
mathematician proving the Riemann  
hypothesis on a blackboard with the  
help of an intelligent robot.



# Generative AI and Auto-Regressive Large Language Models

# Auto-Regressive Generative Architectures

- ▶ 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:**
  - ▶ Open: BlenderBot, Galactica, LLaMA, Llama-2, Code Llama (FAIR), Mistral-7B, Mixtral-4x7B (Mistral), Falcon (UAE), Alpaca (Stanford), Yi (01.AI), OLMo (AI2), Gemma (Google)....
  - ▶ Proprietary: Meta AI (Meta), LaMDA/Bard, Gemini (Google), ChatGPT (OpenAI) ...
- ▶ **Performance is **amazing** ... but ... **they make stupid mistakes****
  - ▶ Factual errors, logical errors, inconsistency, limited reasoning, toxicity...
- ▶ **LLMs have limited knowledge of the underlying reality**
  - ▶ They have no common sense, no memory, & they can't plan their answer



# Llama-2: <https://ai.meta.com/llama/>

- ▶ Open source code / free & open models / can be used commercially
- ▶ Available on Azure, AWS, HuggingFace,....

| MODEL SIZE (PARAMETERS) | PRETRAINED                                                                                                       | FINE-TUNED FOR CHAT USE CASES                      |
|-------------------------|------------------------------------------------------------------------------------------------------------------|----------------------------------------------------|
| 7B                      | <b>Model architecture:</b><br><br><b>Pretraining Tokens:</b><br>2 Trillion<br><br><b>Context Length:</b><br>4096 | <b>Data collection for helpfulness and safety:</b> |
| 13B                     |                                                                                                                  | <b>Supervised fine-tuning:</b><br>Over 100,000     |
| 70B                     |                                                                                                                  | <b>Human Preferences:</b><br>Over 1,000,000        |

# SeamlessM4T

- ▶ Speech or text input: 100 languages
- ▶ Text output: 100 languages
- ▶ Speech output: 35 languages
- ▶ Seamless Expressive: real-time, preserves voice & expression
- ▶ <https://ai.meta.com/blog/seamless-m4t/>

## SeamlessM4T

MODEL INPUT

Speech

Text

MODEL OUTPUT

Speech-to-speech translation

Speech-to-text translation

Text-to-speech translation

Text-to-text translation

Automatic speech recognition



### (1) Pre-trained models

SEAMLESSM4T-NLLB  
T2TT encoder-decoderw2V-BERT 2.0  
Unsupervised speech  
pre-trainingT2U  
Text-to-Unit  
encoder-decoderVocoder  
Speech resynthesis

### (2) Multitasking UNITY

Conformer  
Speech EncoderLength  
adaptorX2T  
(ASR, T2TT, S2ST)Transformer  
Text DecoderTransformer  
Text Encoder

S2ST

HiFi-GAN  
Unit VocoderTransformer  
Unit DecoderTransformer  
Text-to-Unit Encoder

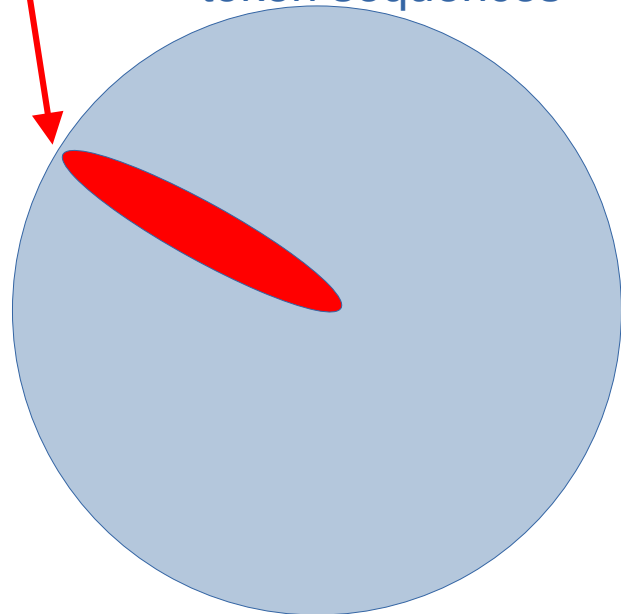
# Auto-Regressive Generative Models Suck!

- ▶ 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 (without a major redesign).**

▶ See also [Dziri...Choi, ArXiv:2305.18654]

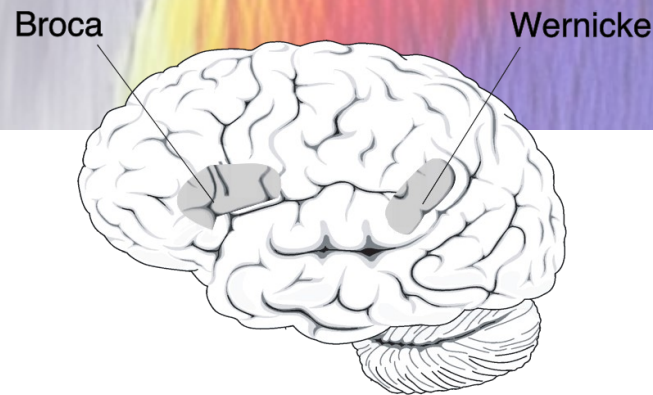
Tree of "correct"  
answers

Tree of all possible  
token sequences



# Limitations of LLMs: no planning!

Y. LeCun



Front

Left Side View

Back

- ▶ Auto-Regressive LLMs (at best) approximate the functions of the Wernicke and Broca areas in the brain.
- ▶ What about the pre-frontal cortex?

ArXiv:2301.06627

ArXiv:2206.10498

---

## DISSOCIATING LANGUAGE AND THOUGHT IN LARGE LANGUAGE MODELS: A COGNITIVE PERSPECTIVE

---

A PREPRINT

**Kyle Mahowald\***

The University of Texas at Austin  
mahowald@utexas.edu

**Idan A. Blank**

University of California Los Angeles  
iblack@psych.ucla.edu

**Joshua B. Tenenbaum**

Massachusetts Institute of Technology  
jbt@mit.edu

**Anna A. Ivanova\***

Massachusetts Institute of Technology  
annaiv@mit.edu

**Nancy Kanwisher**

Massachusetts Institute of Technology  
ngk@mit.edu

**Evelina Fedorenko**

Massachusetts Institute of Technology  
evelina9@mit.edu

---

## Large Language Models Still Can't Plan (A Benchmark for LLMs on Planning and Reasoning about Change)

---

**Karthik Valmeekam\***

School of Computing & AI  
Arizona State University, Tempe.  
kvalmeek@asu.edu

**Alberto Olmo\***

School of Computing & AI  
Arizona State University, Tempe.  
aolmo@asu.edu

**Sarath Sreedharan<sup>†</sup>**

Department of Computer Science,  
Colorado State University, Fort Collins.  
sarath.sreedharan@colostate.edu

**Subbarao Kambhampati**

School of Computing & AI  
Arizona State University, Tempe.  
rao@asu.edu

# 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, Have no common sense
- ▶ **Noema Magazine, August 2023**

## AI And The Limits Of Language

An artificial intelligence system trained on words and sentences alone will never approximate human understanding.

ESSAY    TECHNOLOGY & THE HUMAN

BY JACOB BROWNING AND YANN LECUN

AUGUST 23, 2022



# Auto-Regressive LLMs Suck !

- ▶ **Auto-Regressive LLMs are good for**
  - ▶ Writing assistance, first draft generation, stylistic polishing.
  - ▶ Code writing assistance
- ▶ **What they **not** good for:**
  - ▶ Producing factual and consistent answers (hallucinations!)
  - ▶ Taking into account recent information (anterior to the last training)
  - ▶ Behaving properly (they mimic behaviors from the training set)
  - ▶ Reasoning, planning, math
  - ▶ Using “tools”, such as search engines, calculators, database queries...
- ▶ **We are easily fooled by their fluency.**
- ▶ **But they don't know how the world works.**

# Current AI Technology is (still) far from Human Level

- ▶ **Machines do not learn how the world works, like animals and humans**
- ▶ **Auto-Regressive LLMs can not approach human-level intelligence**
  - ▶ Fluency, but limited world model, limited planning, limited reasoning.
  - ▶ Most human and animal knowledge is non verbal.
- ▶ **We are still missing major advances to reach animal intelligence**
  - ▶ AI is super-human in some narrow domains
- ▶ **There is no questions that, eventually, machines will eventually surpass human intelligence in all domains**
  - ▶ Humanity's total intelligence will increase
  - ▶ We should welcome that not fear it.

# We are missing something really big!

- ▶ **Never mind humans, cats and dogs can do amazing feats**
  - ▶ Robots intelligence doesn't come anywhere close
- ▶ **Any 10 year-old can learn to clear up the dinner table and fill up the dishwasher in minutes.**
  - ▶ We do not have robots that can do that.
- ▶ **Any 17 year-old can learn to drive a car in 20 hours of practice**
  - ▶ We still don't have unlimited Level-5 autonomous driving
- ▶ **Any house cat can plan complex actions**
- ▶ **We keep bumping into Moravec's paradox**
  - ▶ Things that are easy for humans are difficult for AI and vice versa.



# Data bandwidth and volume: LLM vs child.

## ▶ LLM

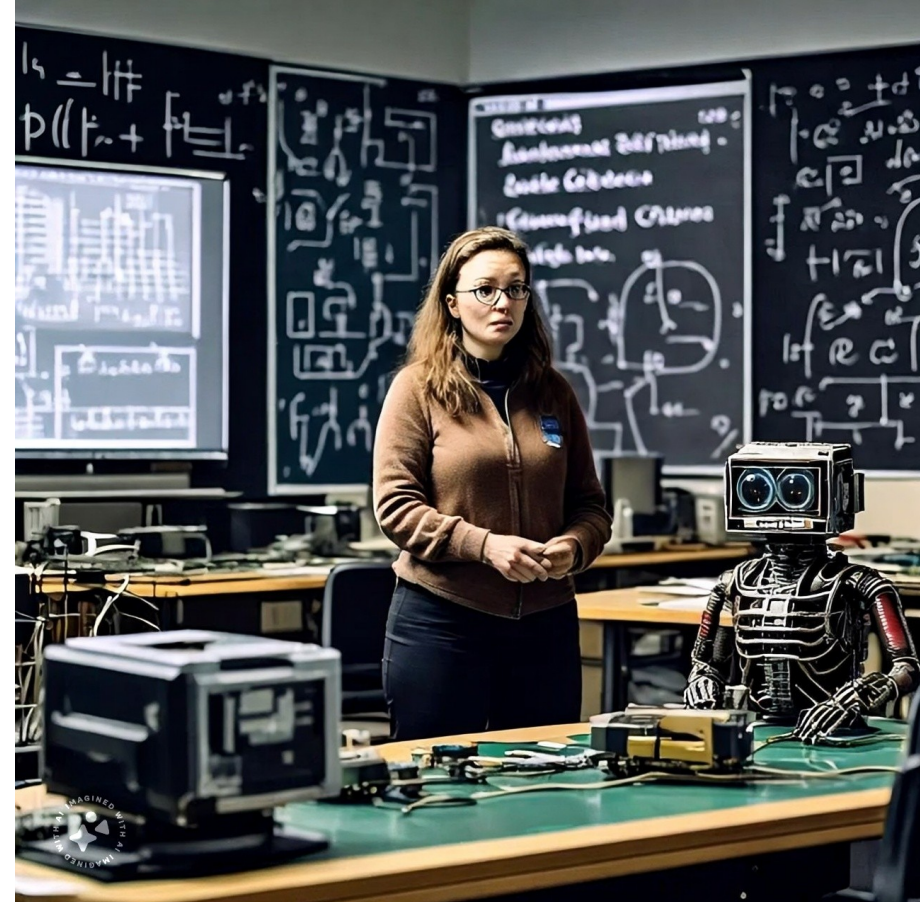
- ▶ Trained on  $1.0\text{E}13$  tokens ( $0.75\text{E}13$  words). Each token is 2 bytes.
- ▶ **Data volume:  $2.0\text{E}13$  bytes.**
- ▶ Would take 170,000 years for a human to read (8h/day, 250 w/minute)

## ▶ Human child

- ▶ 16,000 wake hours in the first 4 years (30 minutes of YouTube uploads)
- ▶ 2 million optical nerve fibers, carrying about 10 bytes/sec each.
- ▶ **Data volume:  $1.1\text{E}15$  bytes**
- ▶ **A four year-old child has seen 50 times more data than an LLM !**
- ▶ **In 300 hours, has child has seen more data than an LLM.**

# What are we missing?

- ▶ **Systems that learn world models from sensory inputs**
  - ▶ E.g. learn intuitive physics from video
- ▶ **Systems that have persistent memory**
  - ▶ Large-scale associative memories
- ▶ **Systems that can plan actions**
  - ▶ So as to fulfill an objective
  - ▶ Reason like “System 2” in humans
- ▶ **Systems that are controllable & safe**
  - ▶ By design, not by fine-tuning.
- ▶ **Objective-Driven AI Architecture**



# Objective-Driven AI Systems

AI that can learn, reason, plan,  
Yet is safe and controllable

“A path towards autonomous machine intelligence”

<https://openreview.net/forum?id=BZ5a1r-kVsf>

[various versions of this talk on YouTube]



# Modular Cognitive Architecture for Objective-Driven AI

## ► Configurator

- Configures other modules for task

## ► Perception

- Estimates state of the world

## ► World Model

- Predicts future world states

## ► Cost

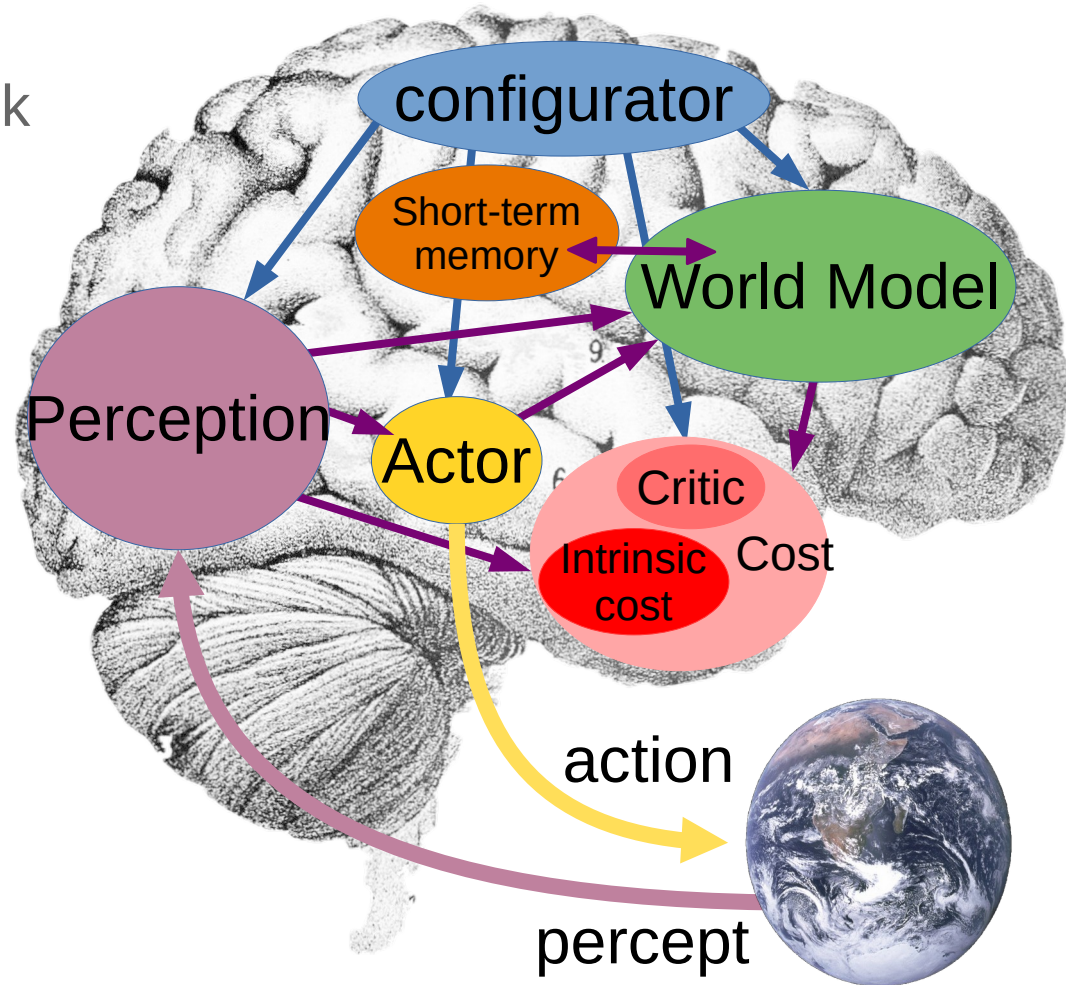
- Compute “discomfort”

## ► Actor

- Find optimal action sequences

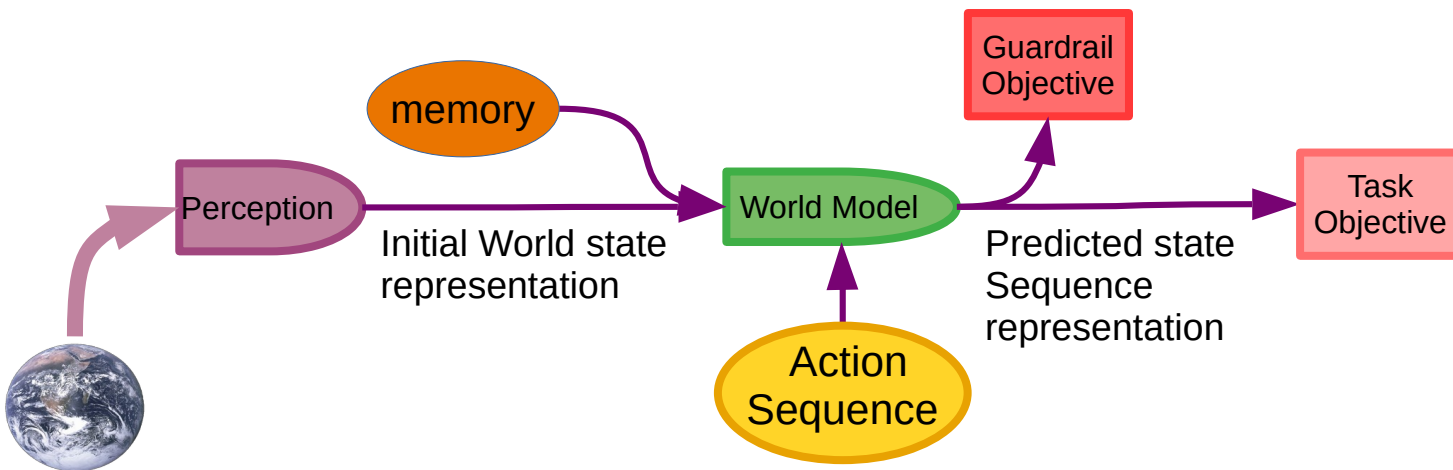
## ► Short-Term Memory

- Stores state-cost episodes



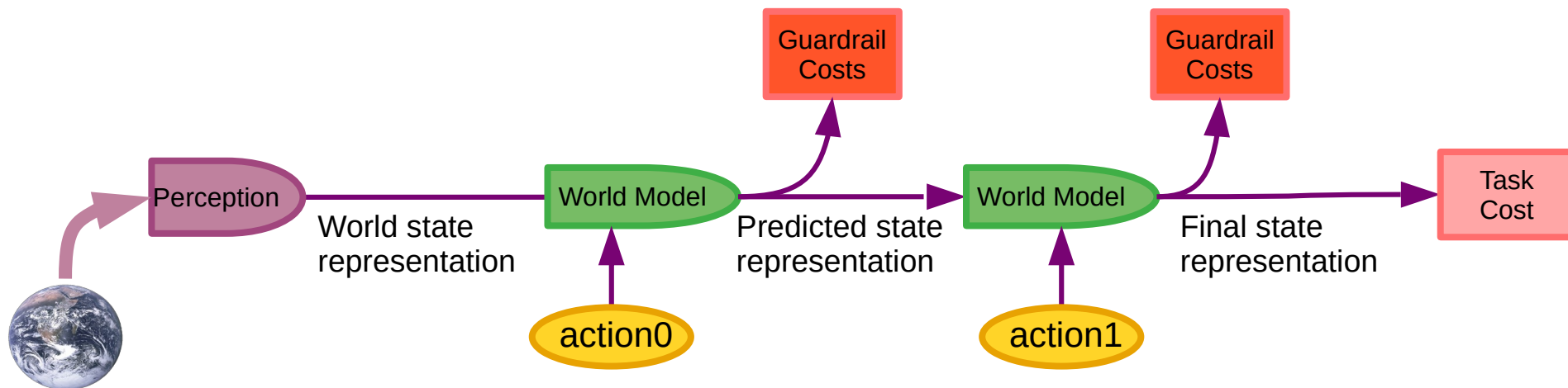
# Objective-Driven AI

- ▶ **Perception:** Computes an abstract representation of the state of the world
  - ▶ Possibly combined with previously-acquired information in memory
- ▶ **World Model:** Predict the state resulting from an imagined action sequence
- ▶ **Task Objective:** Measures divergence to goal
- ▶ **Guardrail Objective:** Immutable objective terms that ensure safety
- ▶ **Operation:** Finds an action sequence that minimizes the objectives



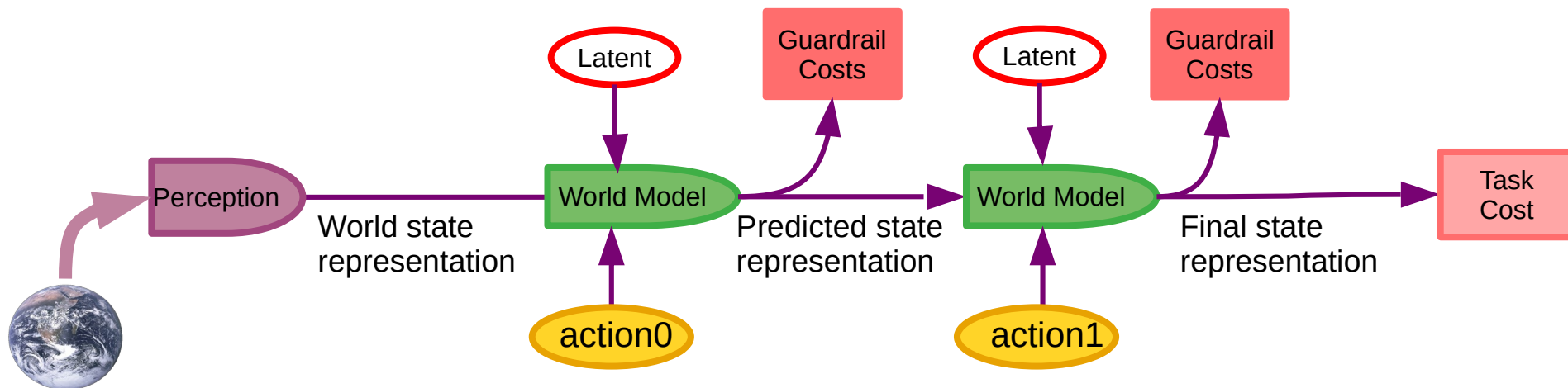
# Objective-Driven AI: Multistep/Recurrent World Model

- ▶ Same world model applied at multiple time steps
- ▶ Guardrail costs applied to entire state trajectory
- ▶ This is identical to Model Predictive Control (MPC)
- ▶ Action inference by minimization of the objectives
  - ▶ Using gradient-based method, graph search, dynamic prog, A\*, MCTS,....



# Objective-Driven AI: Non-Deterministic World Model

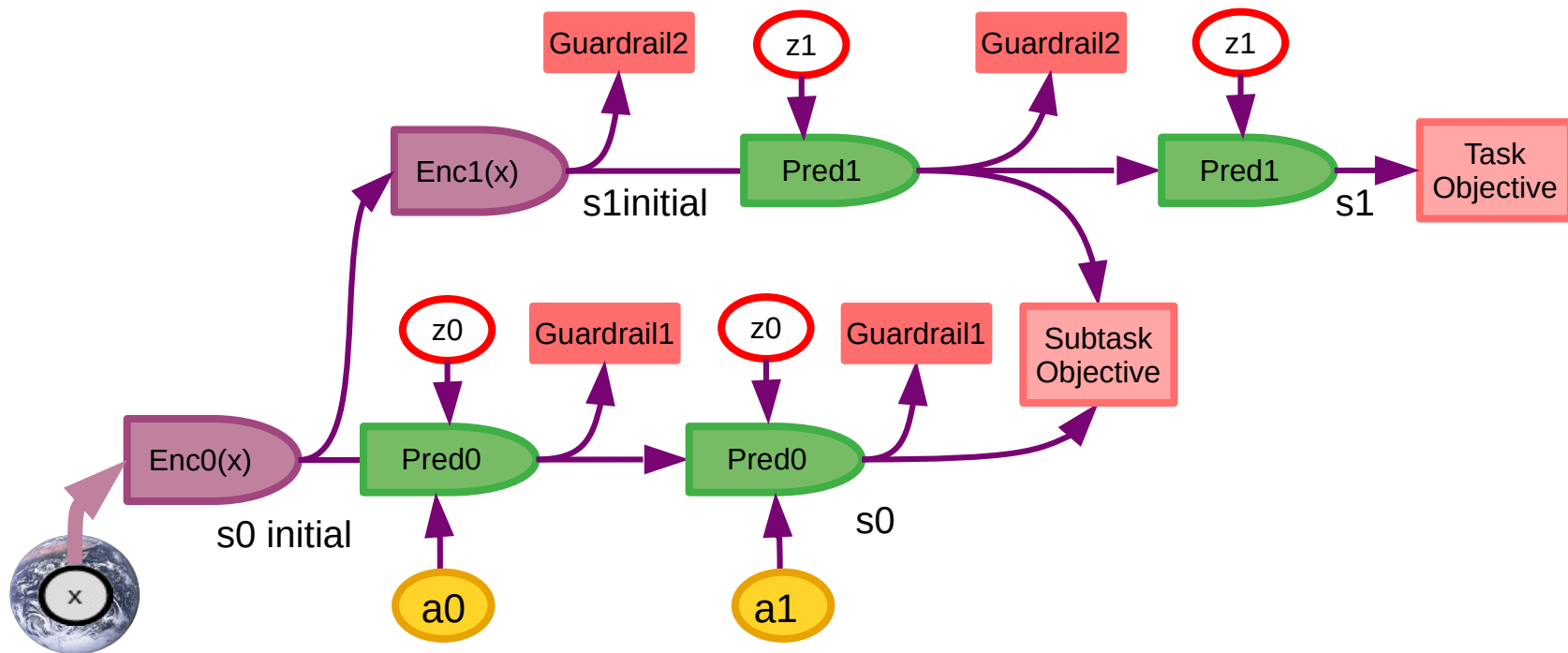
- ▶ The world is not deterministic or fully predictable
- ▶ Latent variables parameterize the set of plausible predictions
  - ▶ Can be sampled from a prior or swept through a set.
  - ▶ Planning can be done for worst case or average case
  - ▶ Uncertainty in outcome can be predicted and quantified



# Objective-Driven AI: Hierarchical Planning

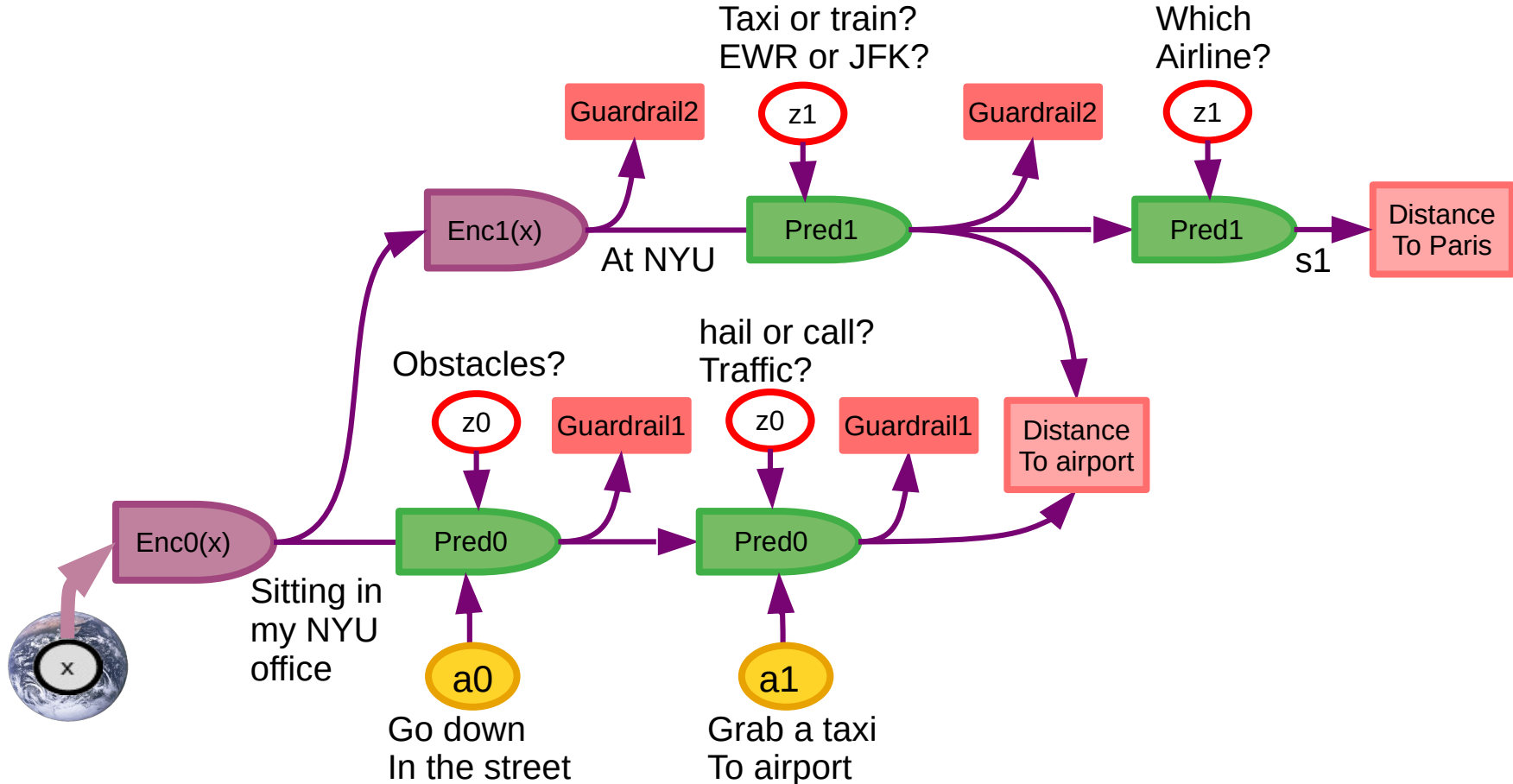
## ► Hierarchical World Model and Planning

- Higher levels make longer-term predictions in more abstract representations
- Predicted states at higher levels define subtask objectives for lower level
- Guardrail objectives ensure safety at every level



# Objective-Driven AI: Hierarchical Planning

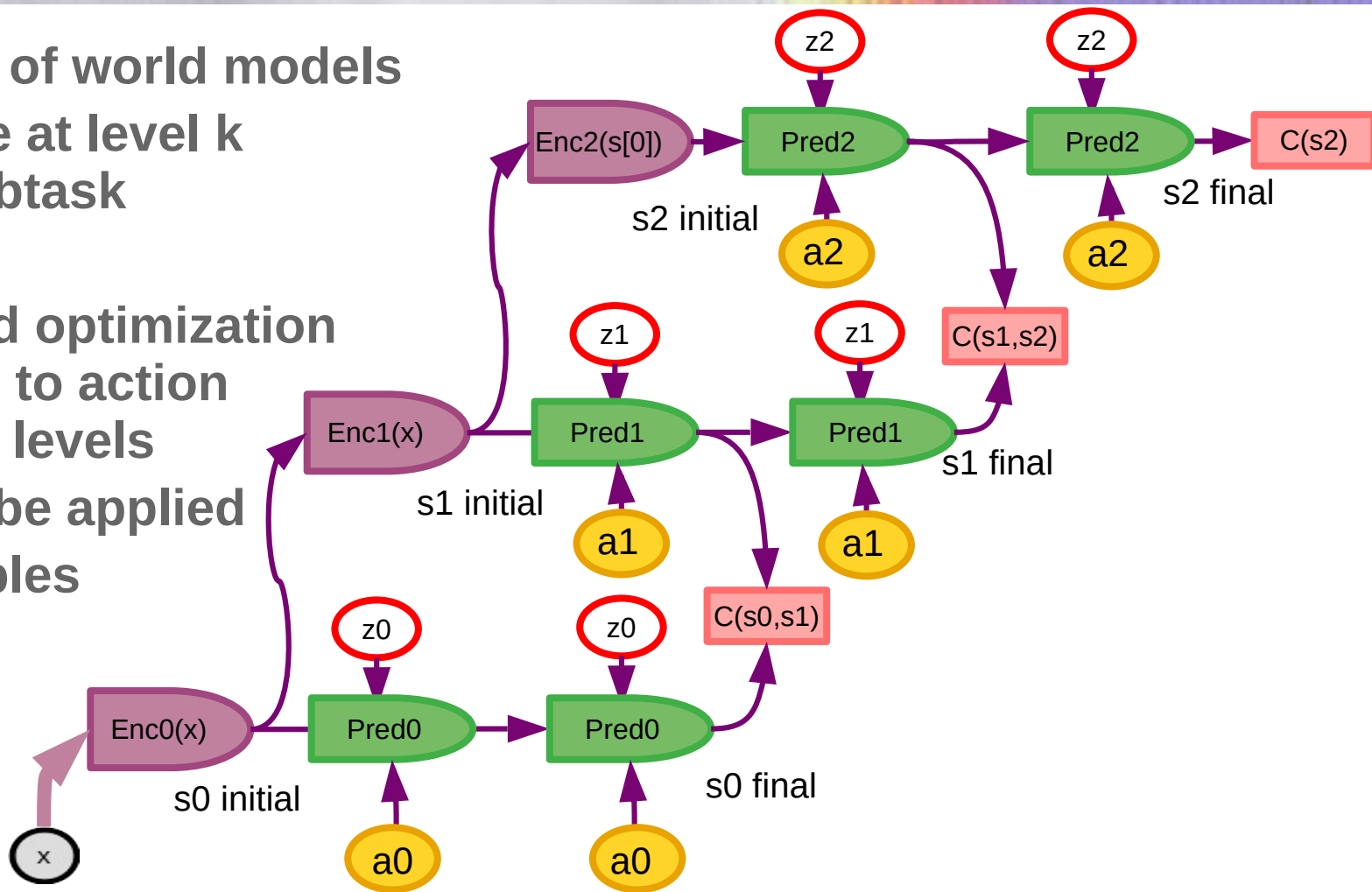
## ► Hierarchical Planning: going from NYU to Paris





# Objective-Driven AI: Hierarchical Planning

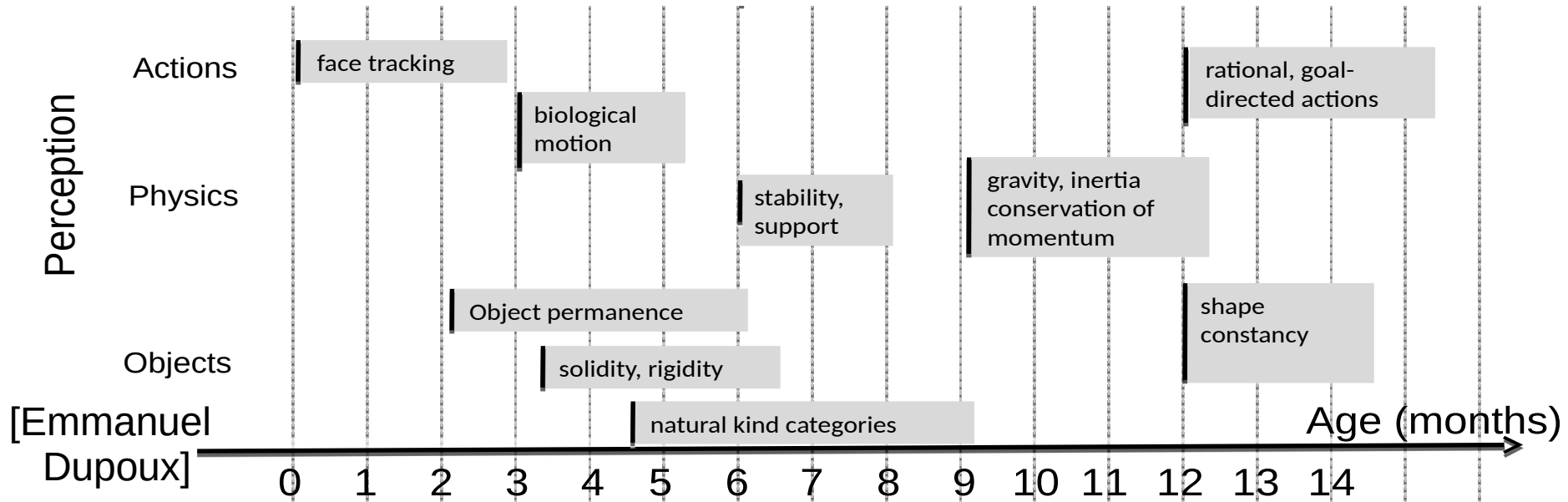
- ▶ Multiple levels of world models
- ▶ Predicted state at level  $k$  determines subtask for level  $k-1$
- ▶ Gradient-based optimization can be applied to action variables at all levels
- ▶ Sampling can be applied to latent variables at all levels.



# How could Machines Learn World Models from Sensory Input?

with  
Self-Supervised Learning

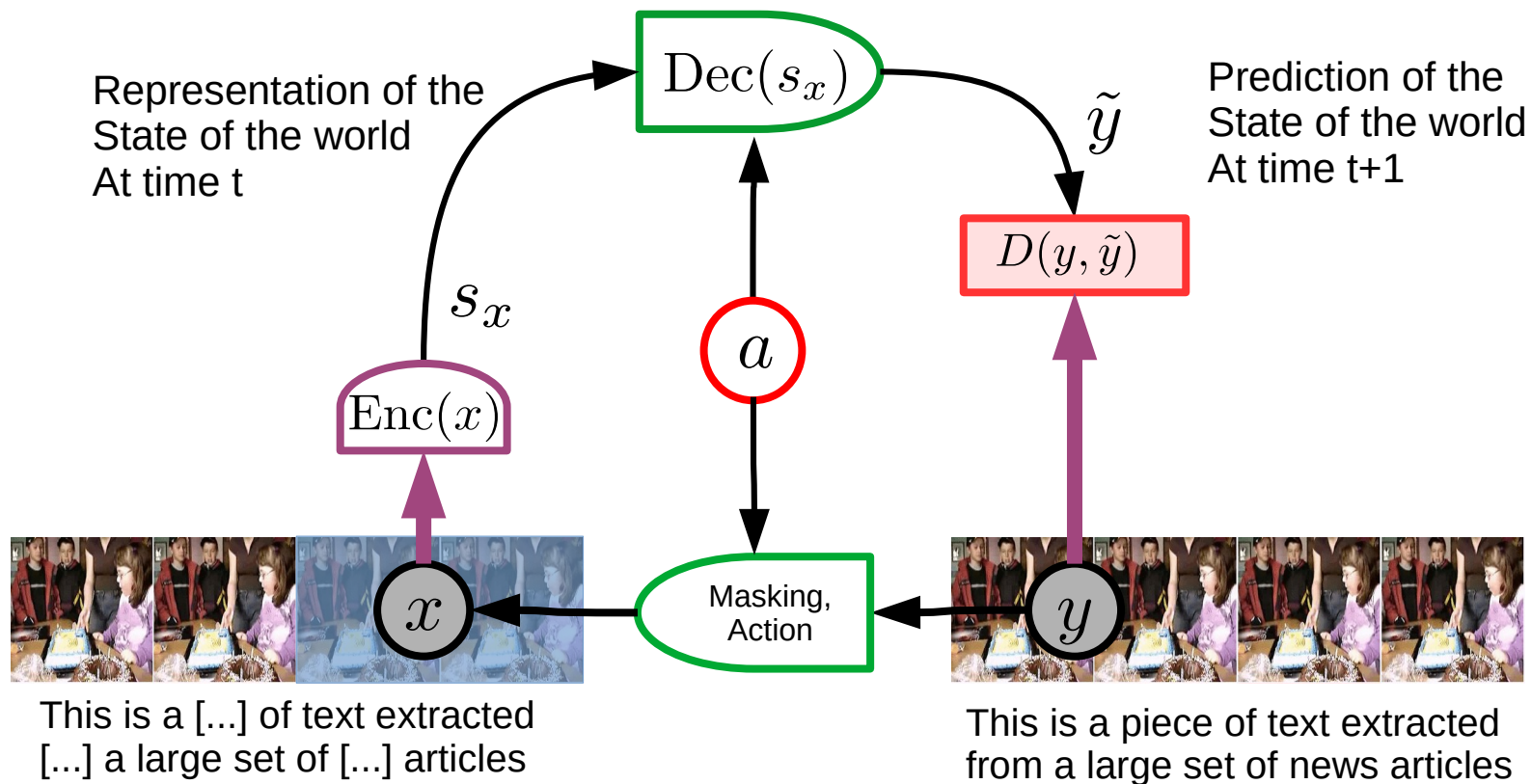
# How could machines learn like animals and humans?



► How do babies learn how the world works?

# Generative World Models with Self-Supervised Training?

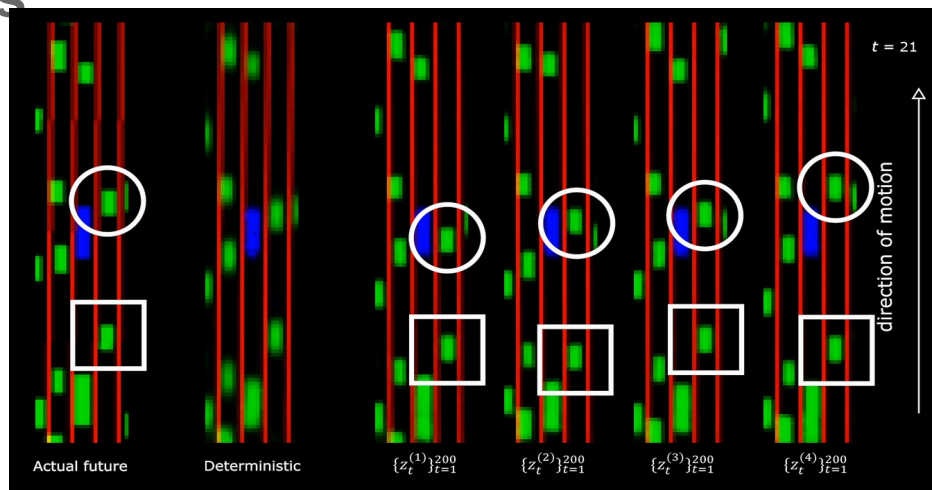
## ► Generative world model architecture



# Generative Architectures DO NOT Work for Images

- ▶ Because the world is only partially predictable
- ▶ A predictive model should represent multiple predictions
- ▶ Probabilistic models are intractable in high-dim continuous domains.
- ▶ Generative Models must predict every detail of the world
- ▶ **My solution: Joint-Embedding Predictive Architecture**

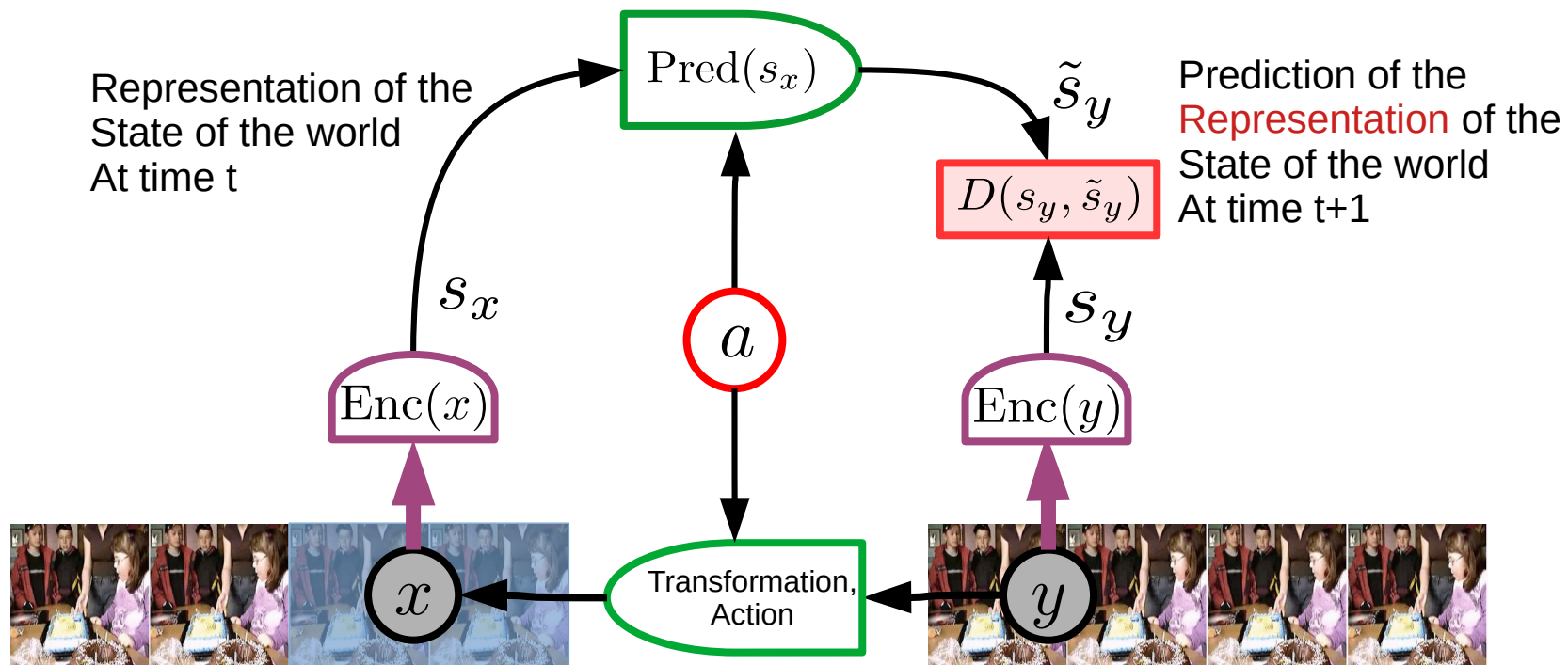
[Mathieu,  
Couprie,  
LeCun  
ICLR 2016]



[Henaff, Canziani, LeCun ICLR 2019]

# Joint Embedding World Model: Self-Supervised Training

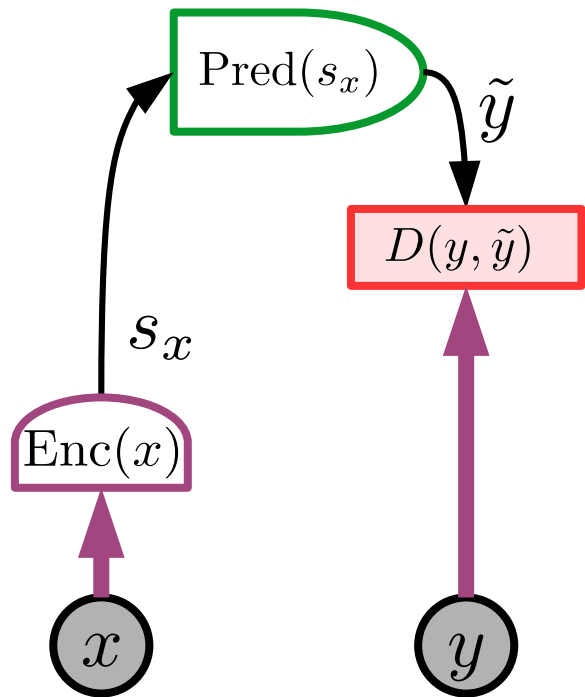
## ► Joint Embedding Predictive Architecture [LeCun 2022], [Assran 2023]



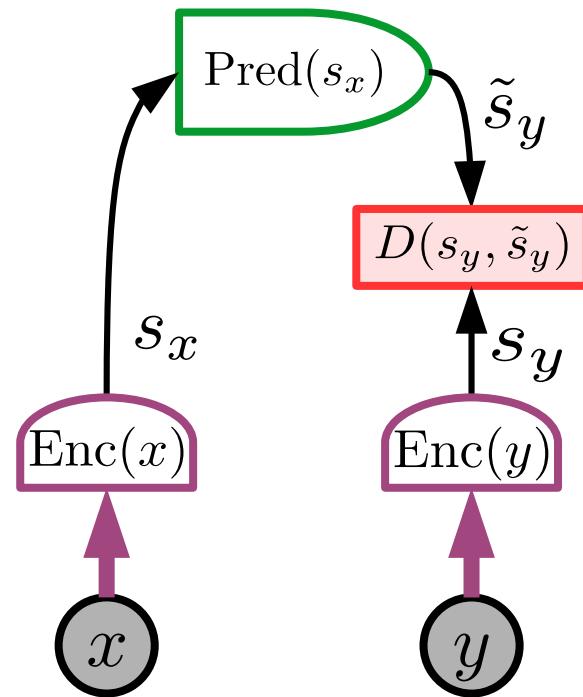


# Architectures: Generative vs Joint Embedding

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



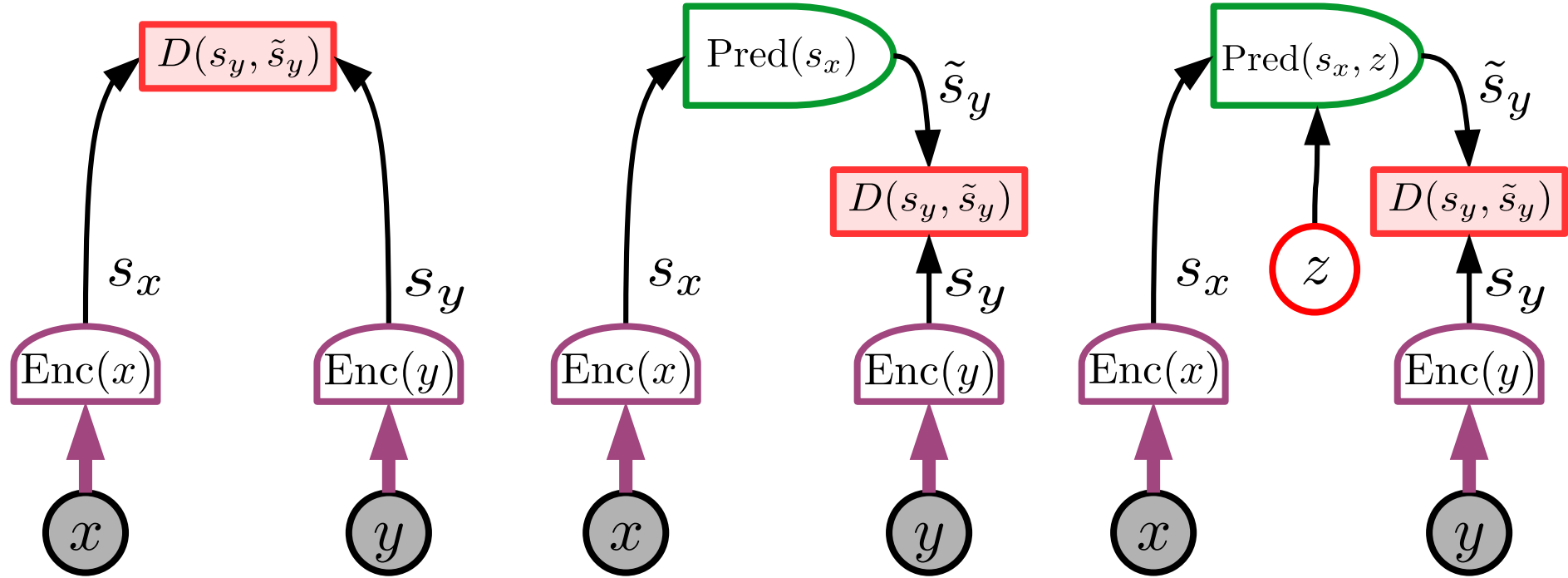
a) Generative Architecture  
Examples: VAE, MAE...



b) Joint Embedding Architecture

# Joint Embedding Architectures

- ▶ Computes abstract representations for  $x$  and  $y$
- ▶ Tries to make them equal or predictable from each other.



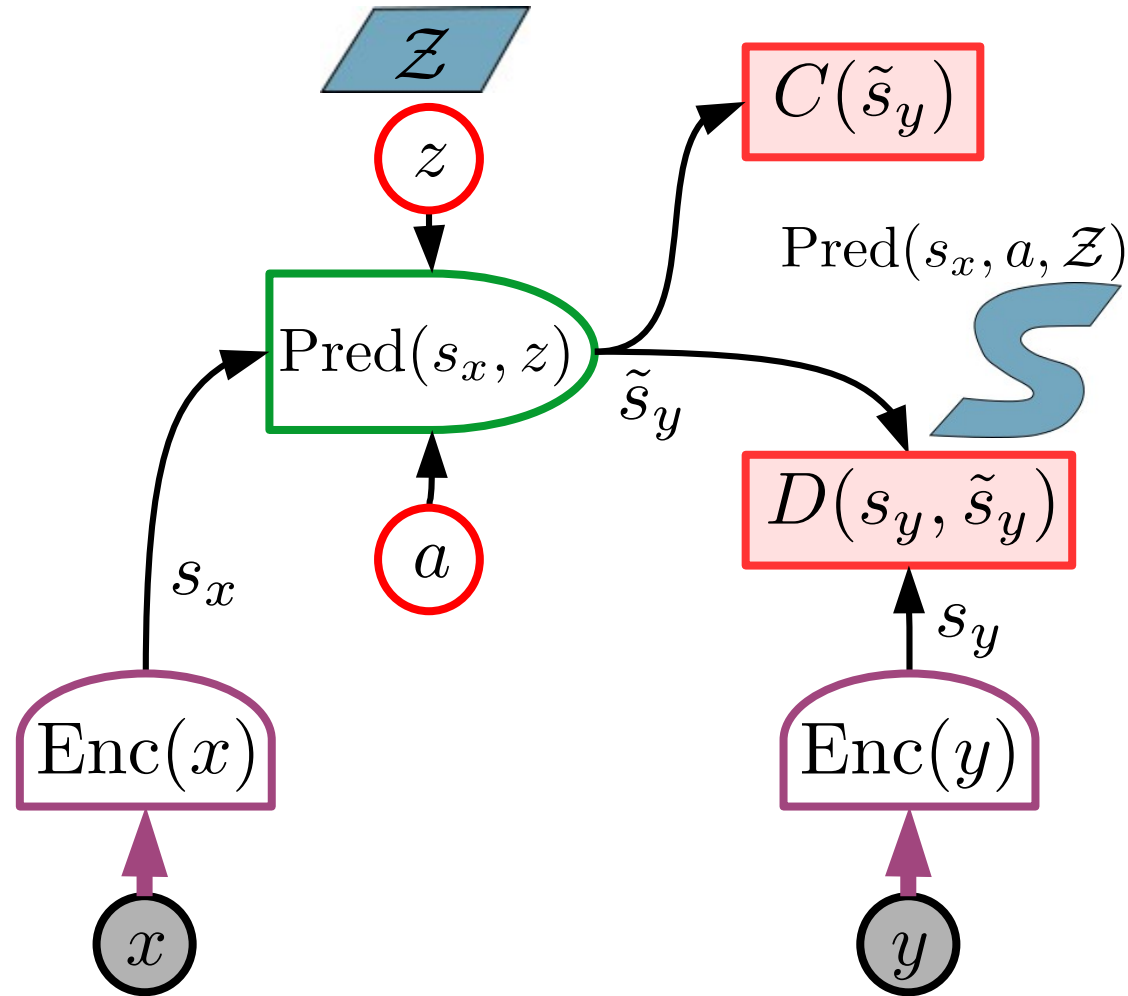
a) Joint Embedding Architecture (JEA)  
Examples: Siamese Net, Pirl, MoCo, SimCLR, BarlowTwins, VICReg,

b) Deterministic Joint Embedding Predictive Architecture (DJEPA)  
Examples: BYOL, VICRegL, I-JEPA

c) Joint Embedding Predictive Architecture (JEPA)  
Examples: Equivariant VICReg I-JEPA.....

# Architecture for the world model: JEPA

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

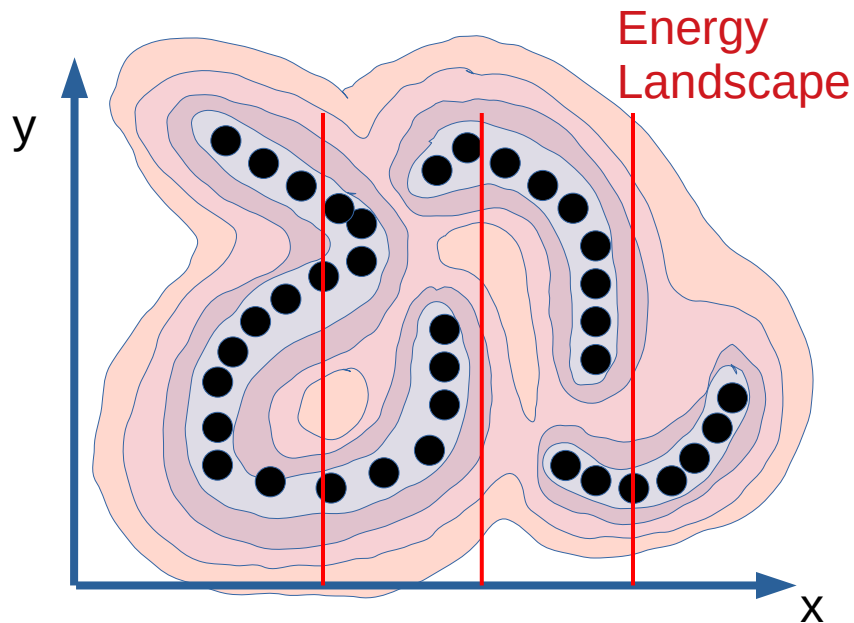
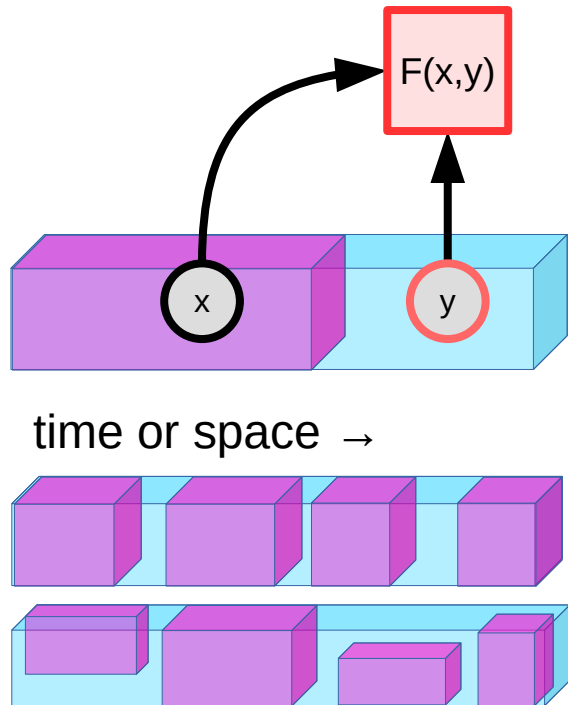


# Energy-Based Models

Capturing dependencies through an energy function

# 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

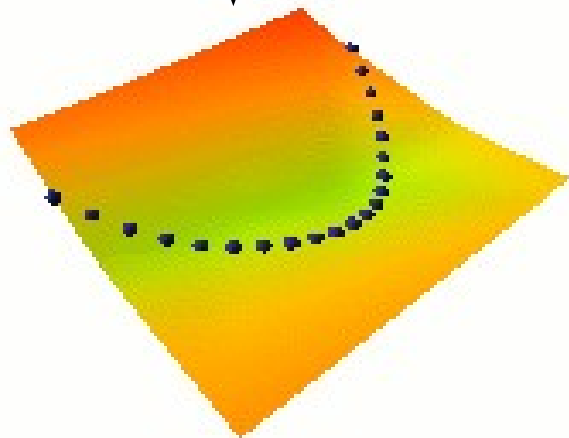


$$\tilde{y} = \operatorname{argmin}_y F(x, y)$$

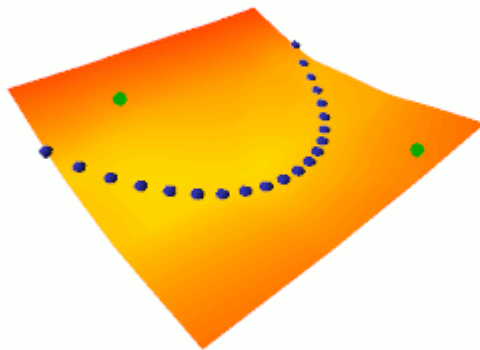
# Training Energy-Based Models: Collapse Prevention

- ▶ A flexible energy surface can take any shape.
- ▶ We need a loss function that shapes the energy surface so that:
  - ▶ Data points have low energies
  - ▶ Points outside the regions of high data density have higher energies.

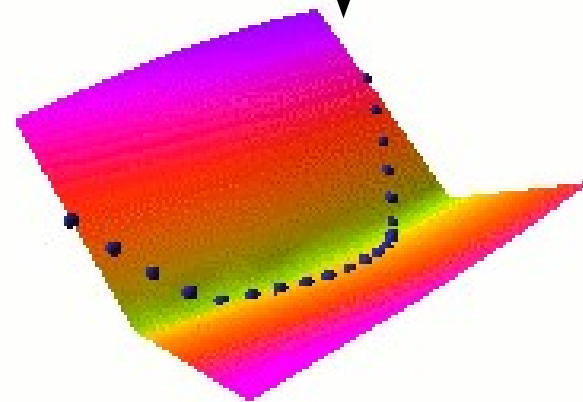
**Collapse!**



**Contrastive Method**



**Regularized Methods**





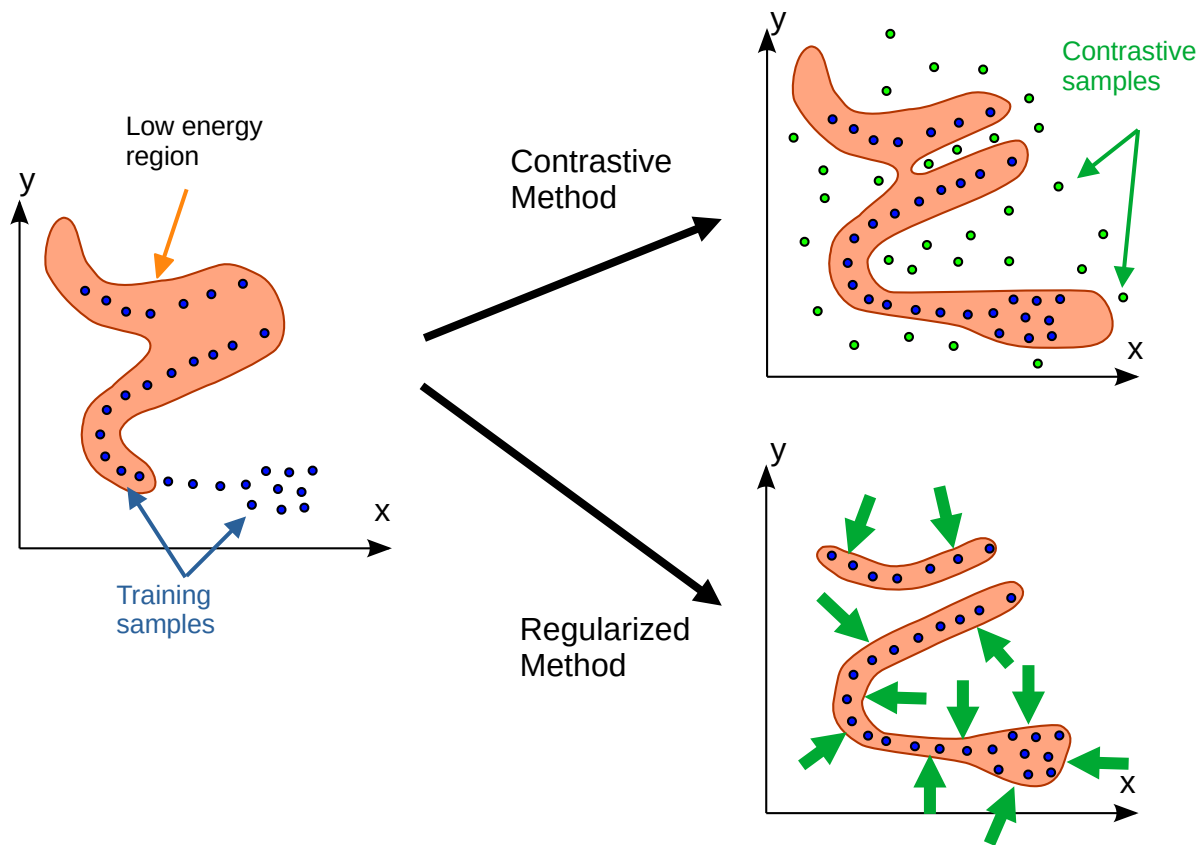
# EBM Training: two categories of methods

## ► 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



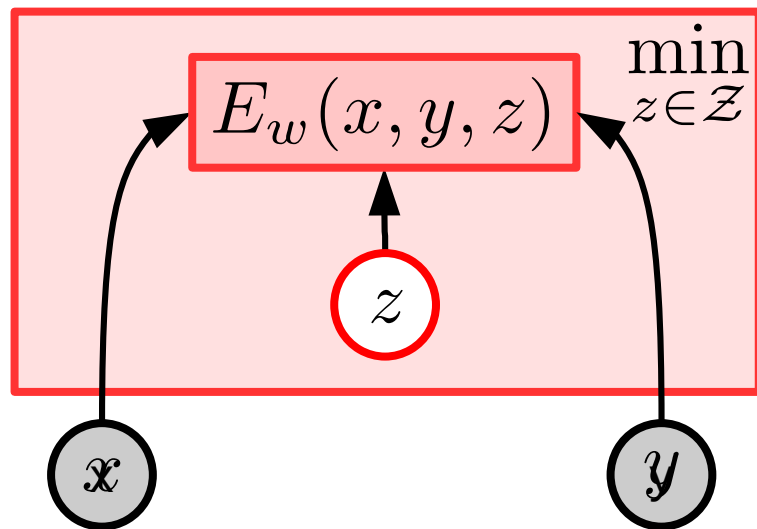
# Latent-Variable EBM

## ► Latent variable $z$ :

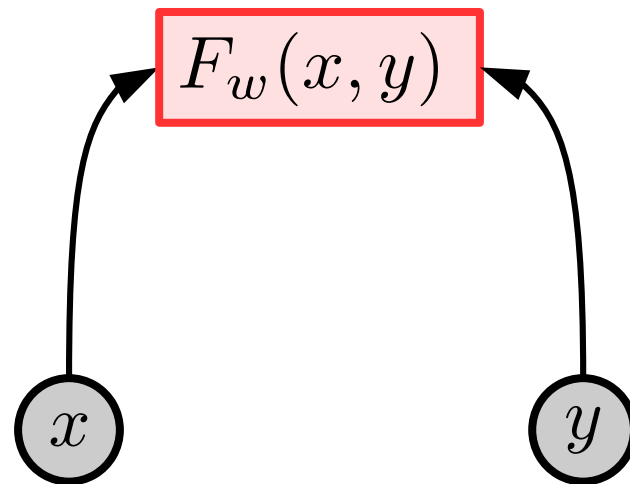
- Captures the information in  $y$  that is not available in  $x$
- Computed by minimization

$$\tilde{z} = \operatorname{argmin}_{z \in \mathcal{Z}} E_w(x, y, z)$$

$$F_w(x, y) = E_w(x, y, \tilde{z})$$

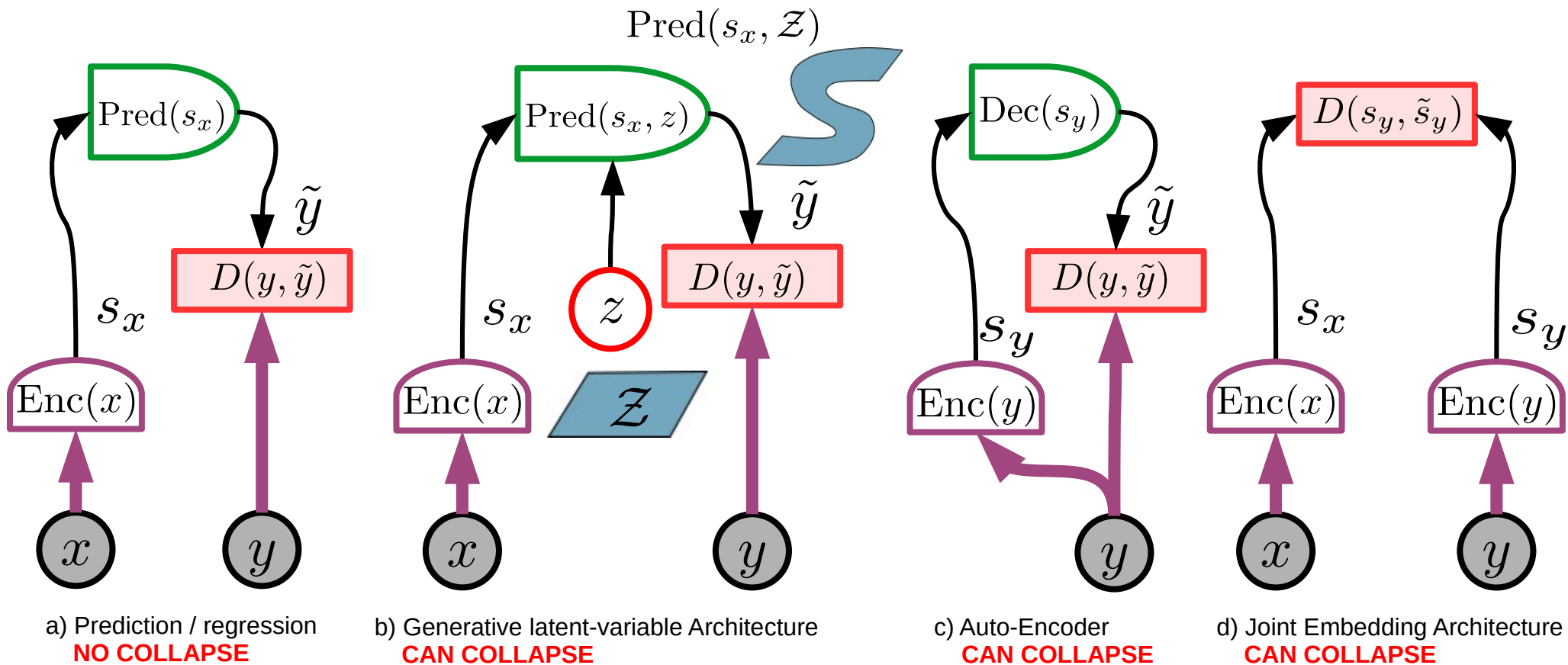


=



# EBM Architectures

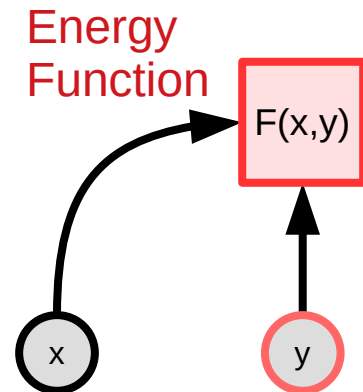
- Some architectures can lead to a collapse of the energy surface



# Energy-Based Models vs Probabilistic Models

- ▶ **Probabilistic models are a special case of EBM**
  - ▶ Energies are like un-normalized negative log probabilities
- ▶ **Why use EBM instead of probabilistic models?**
  - ▶ EBM gives more flexibility in the choice of the scoring function.
  - ▶ More flexibility in the choice of objective function for learning
- ▶ **From energy to probability: Gibbs-Boltzmann distribution**
  - ▶ Beta is a positive constant

$$P(y|x) = \frac{e^{-\beta F(x,y)}}{\int_{y'} e^{-\beta F(x,y')}}$$

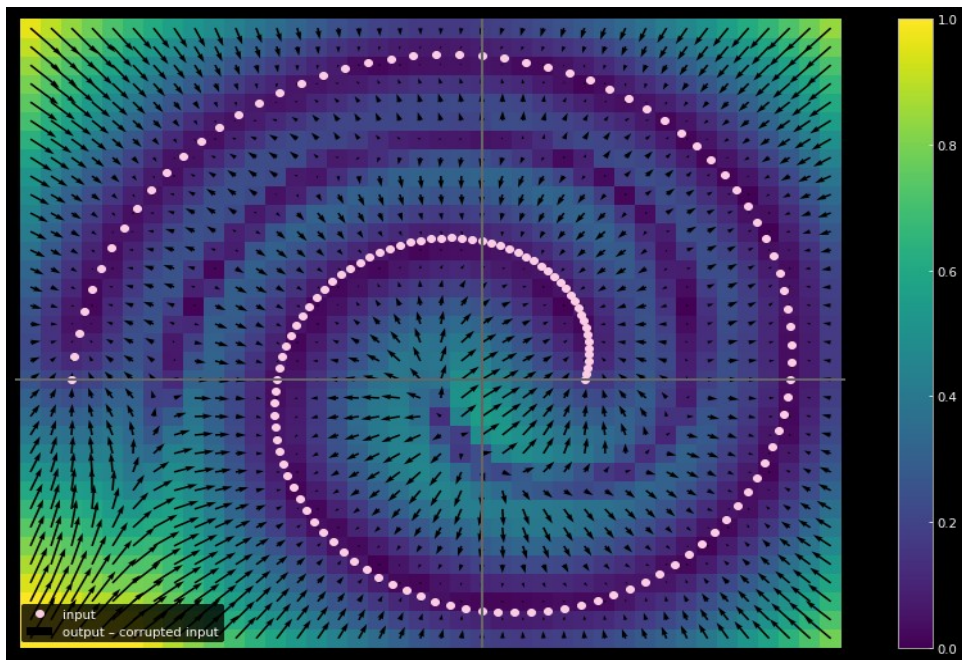
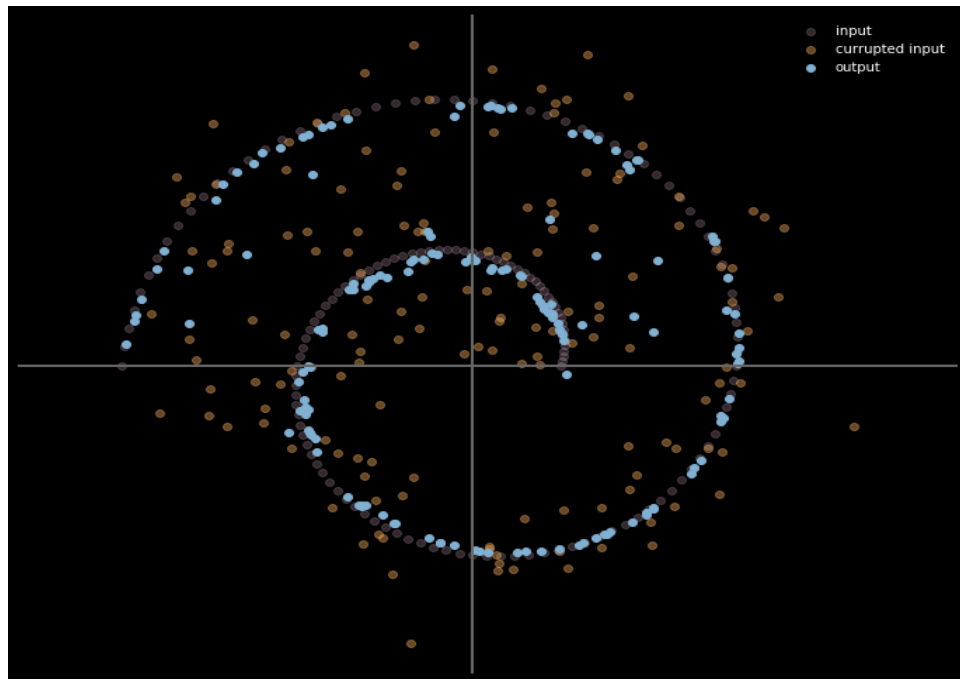


# Contrastive Methods vs Regularized/Architectural Methods

- ▶ **Contrastive:** [they all are different ways to pick which points to push up]
  - ▶ C1: push down of the energy of data points, push up everywhere else: **Max likelihood** (needs tractable partition function or variational approximation)
  - ▶ C2: push down of the energy of data points, push up on chosen locations: max likelihood with MC/MMC/HMC, Contrastive divergence, **Metric learning/Siamese nets**, Ratio Matching, Noise Contrastive Estimation, Min Probability Flow, **adversarial generator/GANs**
  - ▶ C3: train a function that maps points off the data manifold to points on the data manifold: denoising auto-encoder, **masked auto-encoder** (e.g. BERT)
- ▶ **Regularized/Architectural:** [Different ways to limit the information capacity of the latent representation]
  - ▶ A1: build the machine so that the volume of low energy space is bounded: PCA, K-means, Gaussian Mixture Model, Square ICA, normalizing flows...
  - ▶ A2: use a regularization term that measures the volume of space that has low energy: Sparse coding, **sparse auto-encoder**, LISTA, Variational Auto-Encoders, discretization/VQ/VQVAE.
  - ▶ A3:  $F(x,y) = C(y, G(x,y))$ , make  $G(x,y)$  as "constant" as possible with respect to  $y$ : Contracting auto-encoder, saturating auto-encoder
  - ▶ A4: minimize the gradient and maximize the curvature around data points: score matching

# Contrastive EBM Training: Denoising Auto-Encoder

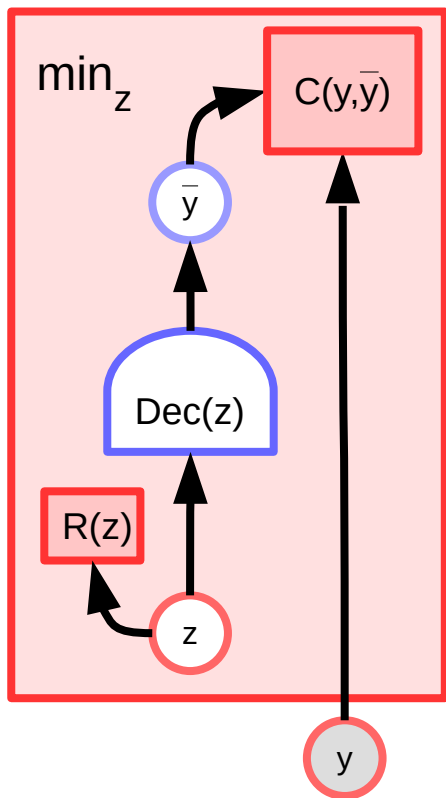
- ▶ [LeCun 1987], [Seung 1998], [Vincent 2008, 2010]
- ▶ NLP: BERT [



Figures: Alfredo Canziani

# Example: Regularized Latent-Variable EBM

- Basic idea: limiting the information capacity of the latent variable to limit the volume of low-energy regions
- Examples: K-Means, sparse coding



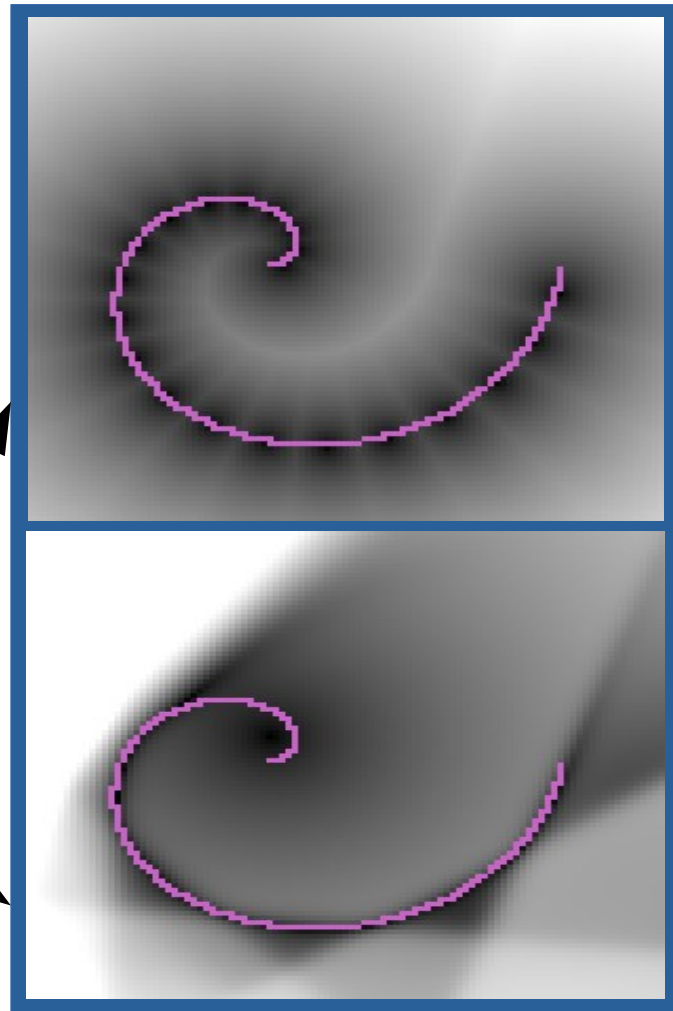
$$E(y, z) = C(y, \text{Dec}(z)) + R(z)$$

$$F_{\infty}(x, y) = \min_z E(x, y, z)$$

$$\check{y}, \check{z} = \operatorname{argmin}_{y, z} E(x, y, z)$$

$$E(y, z) = \|y - Wz\|^2 \quad z \in \text{1-hot}$$

$$E(y, z) = \|y - wz\|^2 + \lambda |z|_{L1}$$





# Regularizing $z$ by making it “fuzzy” (stochastic)

- ▶ The information content of the latent variable  $z$  must be minimized
- ▶ One (probabilistic) way to do this:

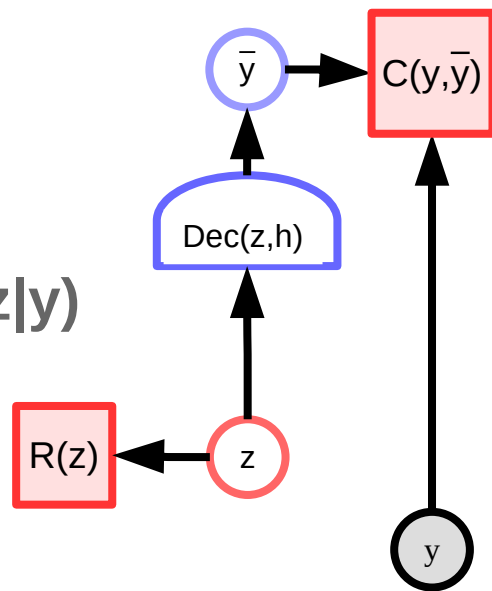
$$E(y, z) = C(y, Dec(z))$$

- ▶ make  $z$  “fuzzy” (e.g. stochastic)
- ▶  $z$  is a sample from a distribution  $q(z|y)$

- ▶ Minimize the expected value of the energy under  $q(z|y)$

$$\langle E(y) \rangle = \int_z q(z|y) E(y, z)$$

- ▶ Minimize the information content of  $q(z|y)$  about  $y$



# Minimize expected energy & information content of $z$

## ► Minimize the expected energy

$$\langle E(y) \rangle_q = \int_z q(z|y) E(y, z)$$

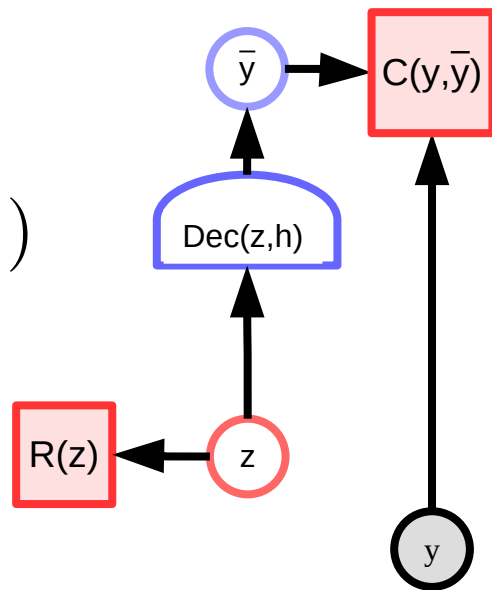
$$E(y, z) = C(y, Dec(z))$$

## ► Minimize the relative entropy

- Between  $q(z|y)$  and a prior distribution  $p(z)$ .

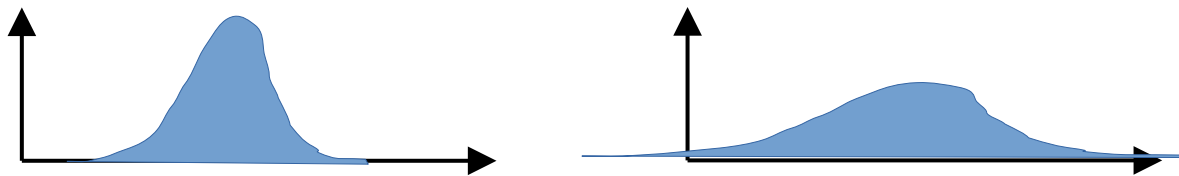
$$KL(q(z|y), p(z)) = \int_z q(z|y) \log_2(q(z|y)/p(z))$$

- This is the number of bits one sample from  $q(z|y)$  will give us about  $z$ , knowing that  $z$  comes from  $p(z)$



# Variational free energy: trades average energy and information in $z$

- ▶ Find a distribution  $q(z|y)$  that minimizes the expected energy while having maximum entropy
- ▶ high entropy distribution == small information content from a sample



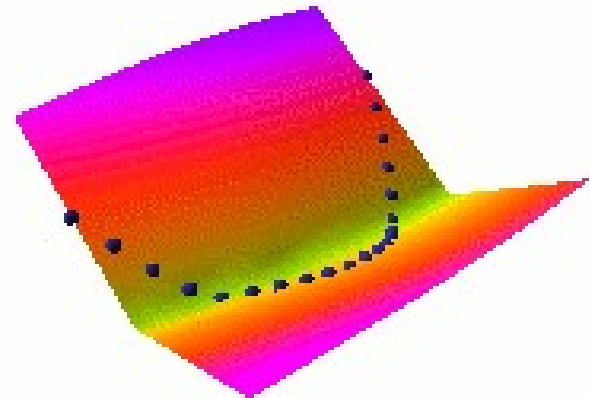
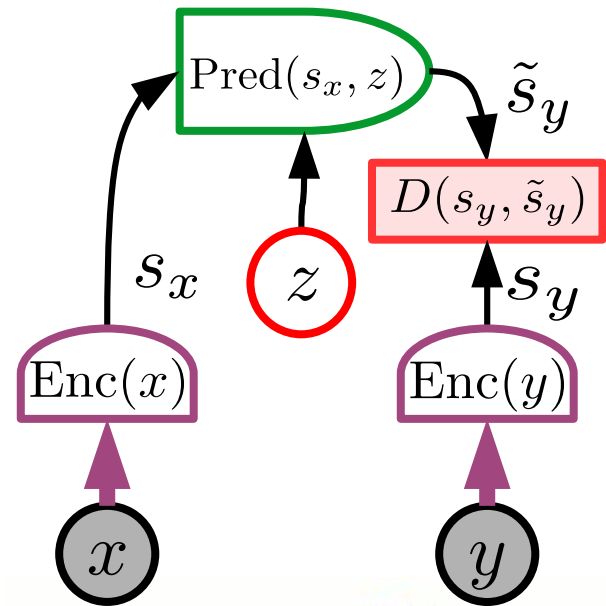
- ▶ Pick a family of distributions  $q(z|y)$  (e.g. Gaussians) and find the one that minimizes the **variational free energy**:

$$\tilde{F}_q(y) = \int_z q(z|y) E(y, z) + \frac{1}{\beta} \int_z q(z|y) \log_2(q(z|y)/p(z))$$

- ▶ The trade-off between energy and entropy is controlled by the beta parameter.

# Recommendations:

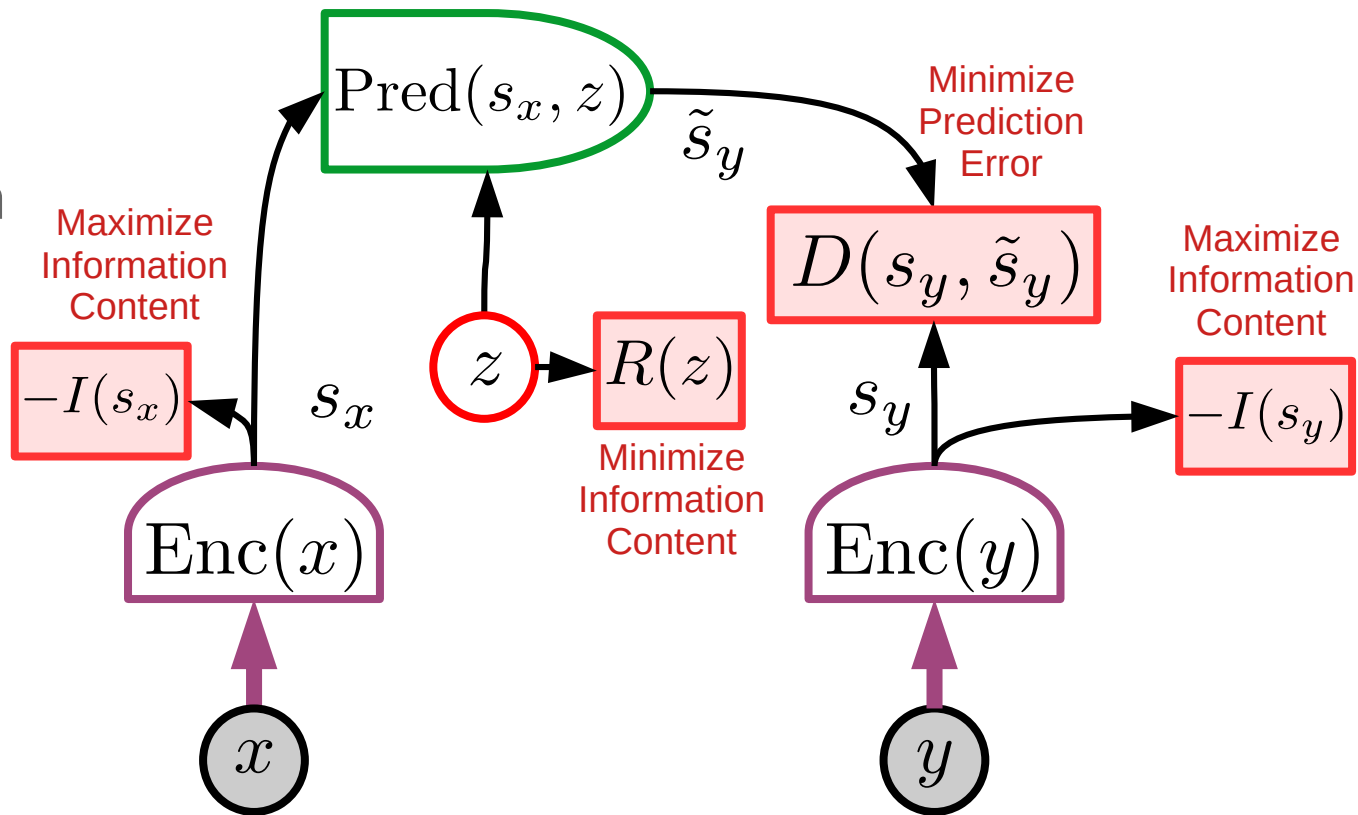
- ▶ **Abandon generative models**
  - ▶ in favor joint-embedding architectures
- ▶ **Abandon probabilistic model**
  - ▶ in favor of energy-based models
- ▶ **Abandon contrastive methods**
  - ▶ in favor of regularized methods
- ▶ **Abandon Reinforcement Learning**
  - ▶ In favor of model-predictive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



# Training a JEPA with Regularized Methods

## ► 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  $z$



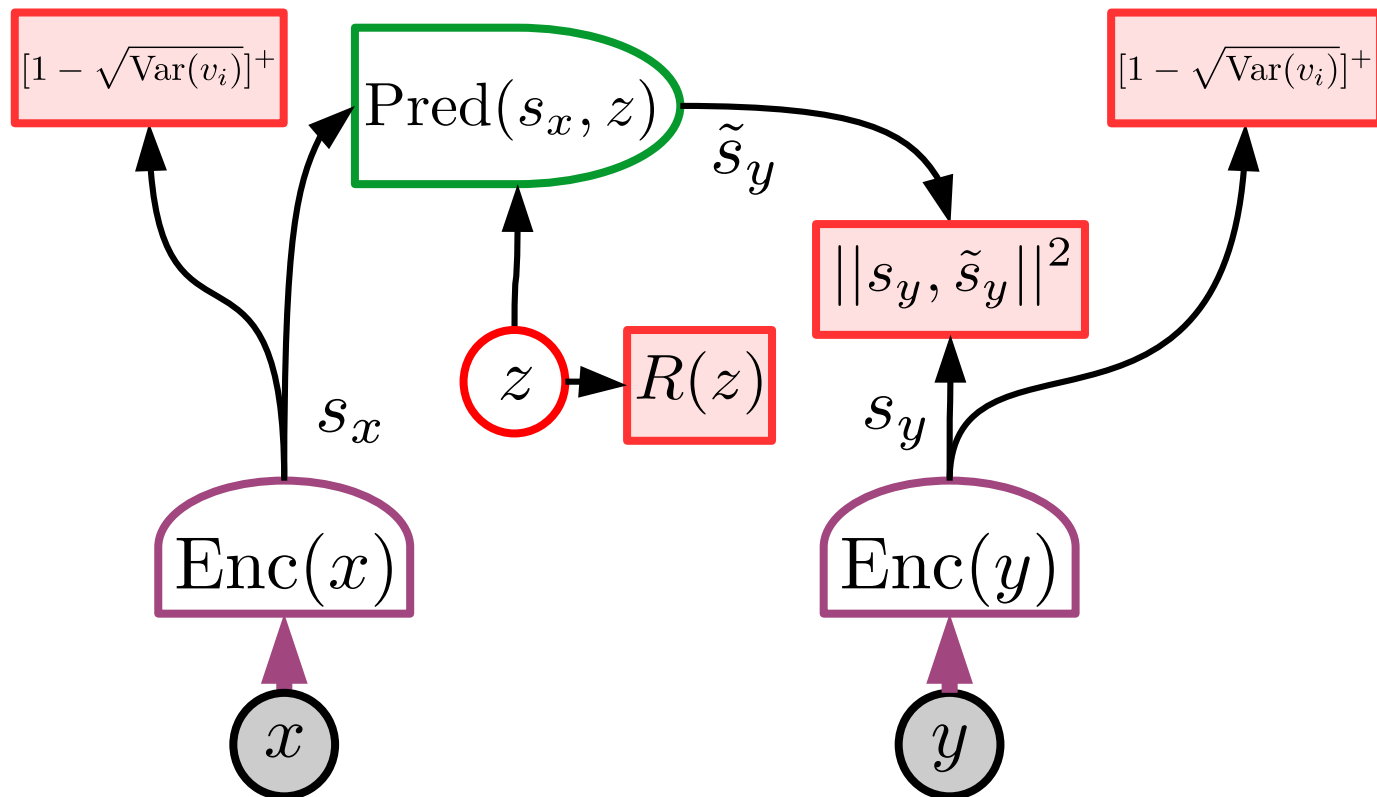
# VICReg: Variance, Invariance, Covariance Regularization

## ► Variance:

- Maintains variance of components of representations

## ► Invariance:

- Minimizes prediction error.



# VICReg: Variance, Invariance, Covariance Regularization

## ► Variance:

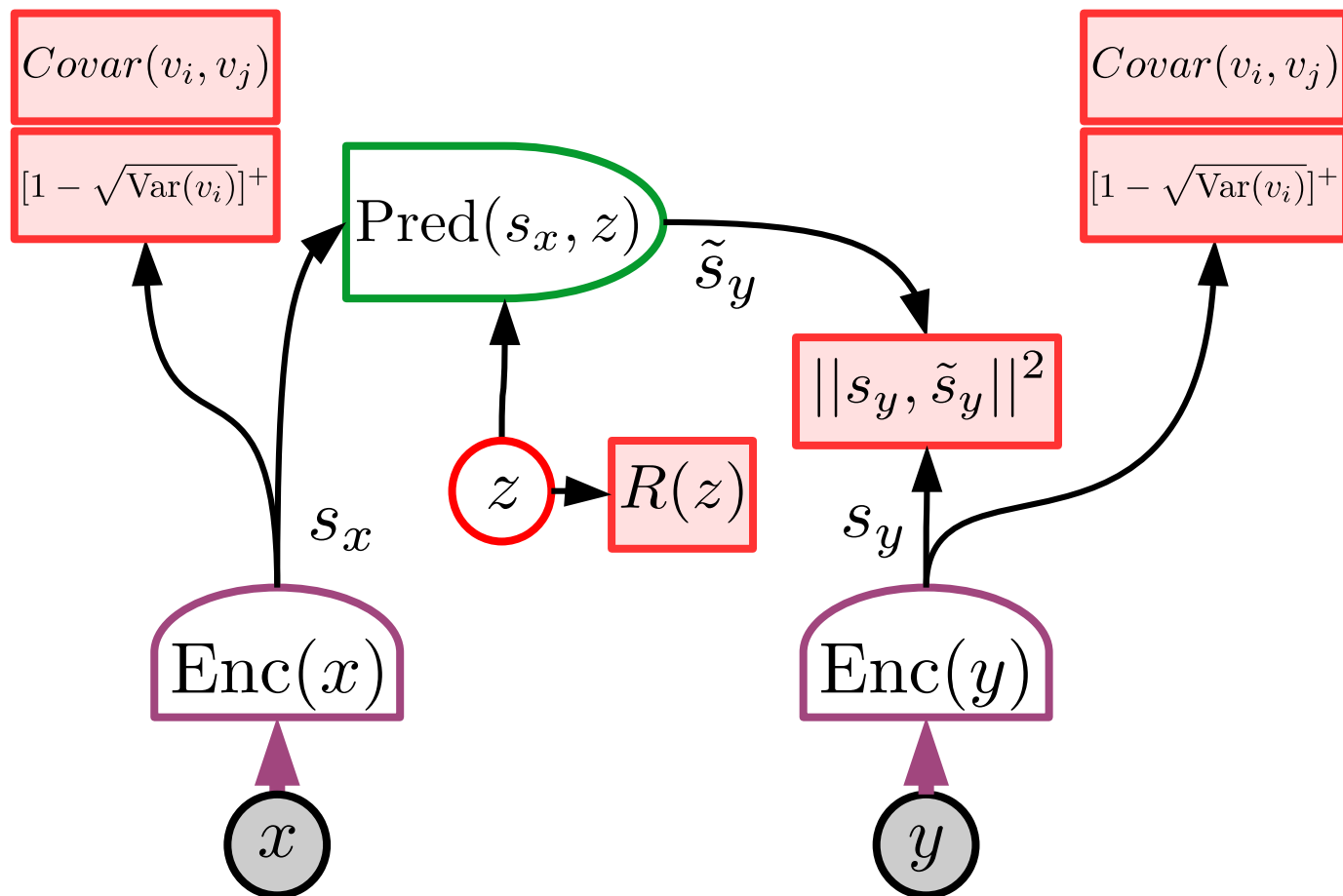
- Maintains variance of components of representations

## ► Covariance:

- Decorrelates components of covariance matrix of representations

## ► Invariance:

- Minimizes prediction error.





# VICReg: Variance, Invariance, Covariance Regularization

## ► Variance:

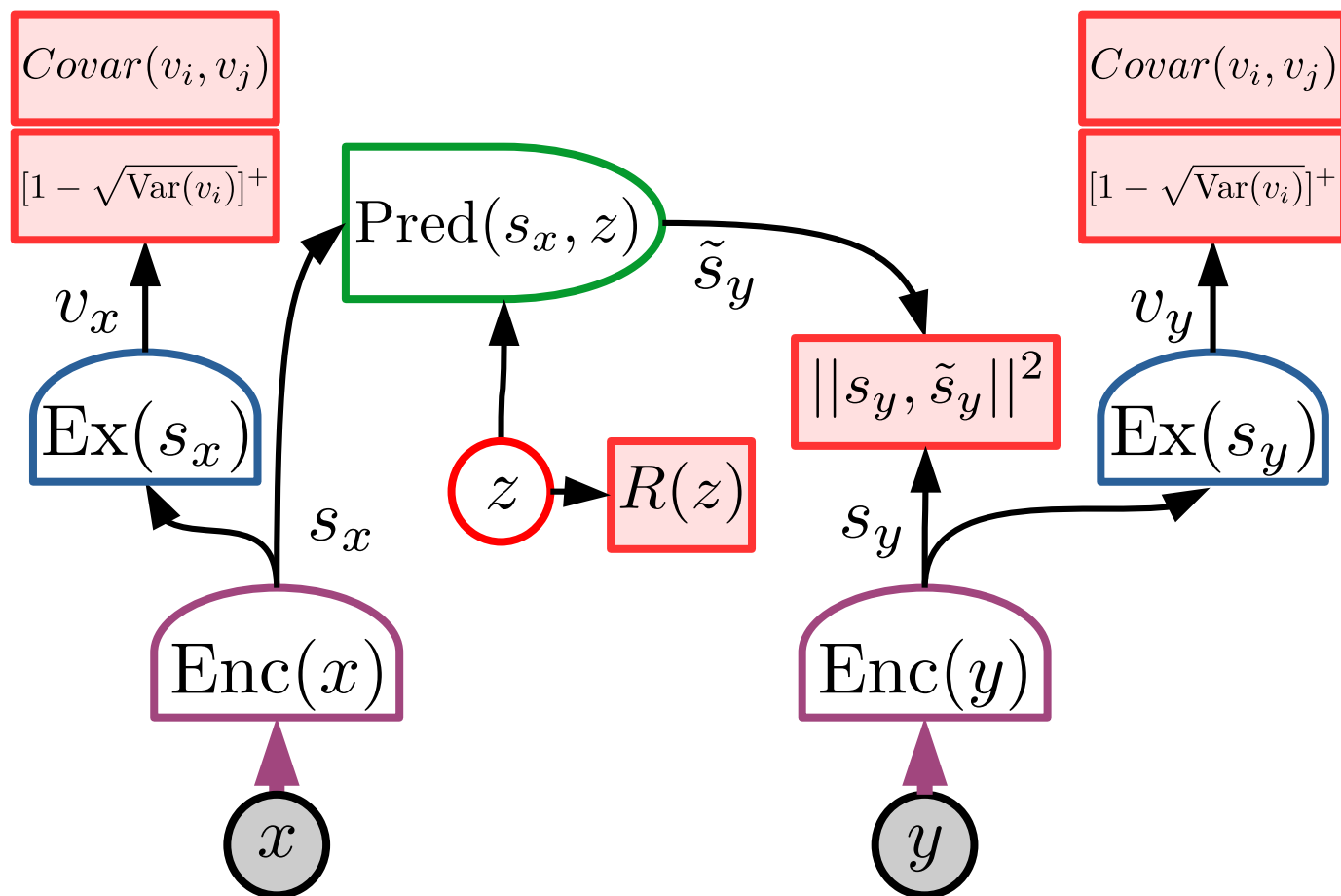
- Maintains variance of components of representations

## ► Covariance:

- Decorrelates components of covariance matrix of representations

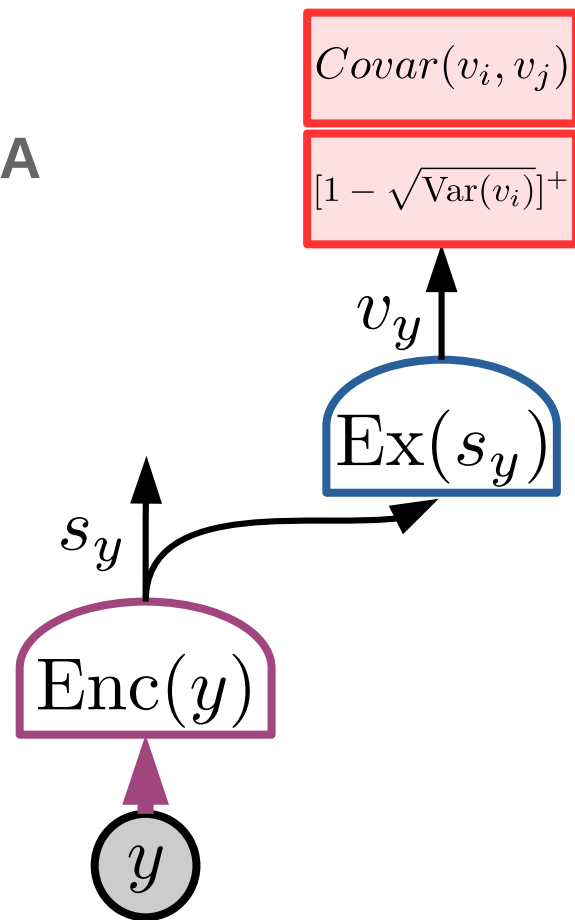
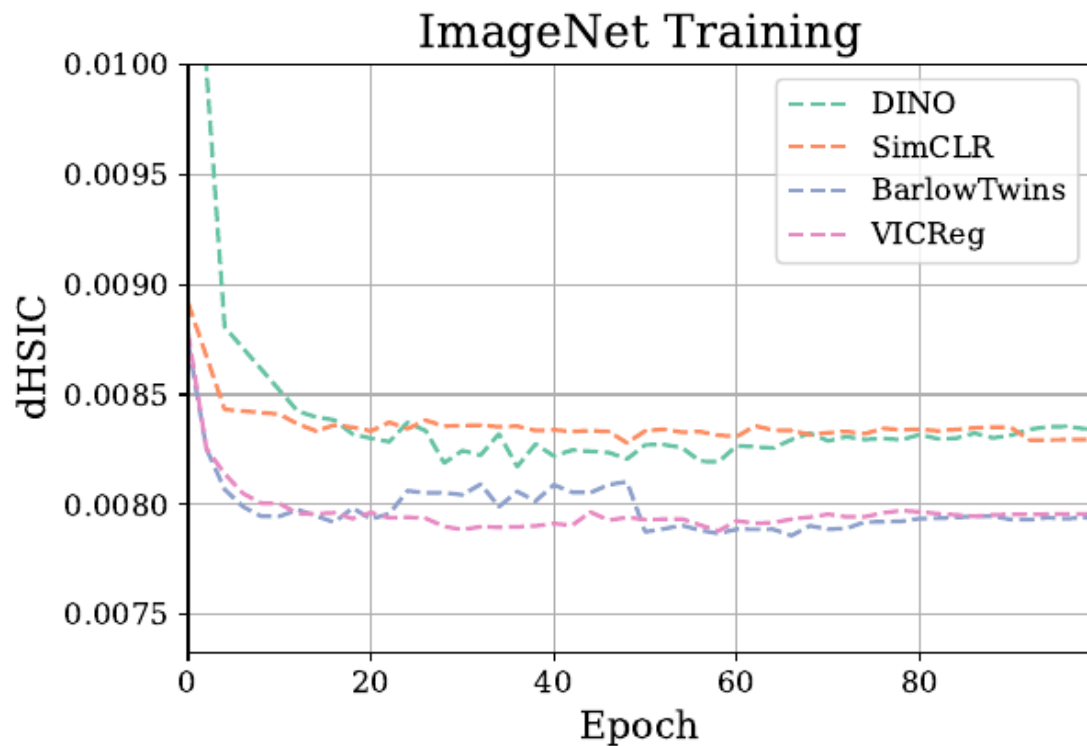
## ► Invariance:

- Minimizes prediction error.



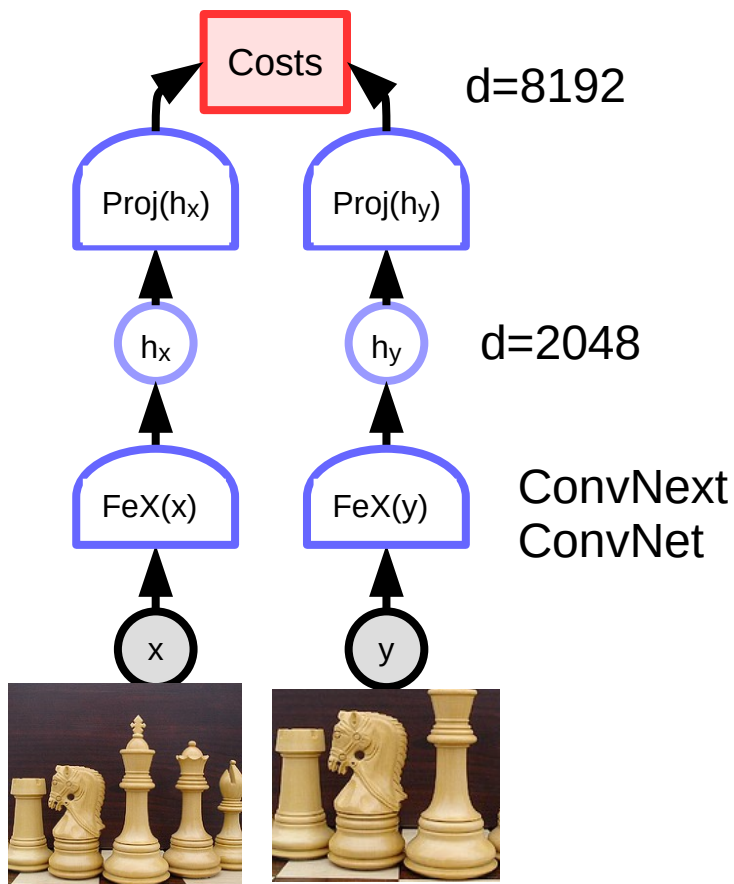
# VICReg: expander makes variables pairwise independent

- [Mialon, Balestrieri, LeCun arxiv:2209.14905]
- VC criterion can be used for source separation / ICA

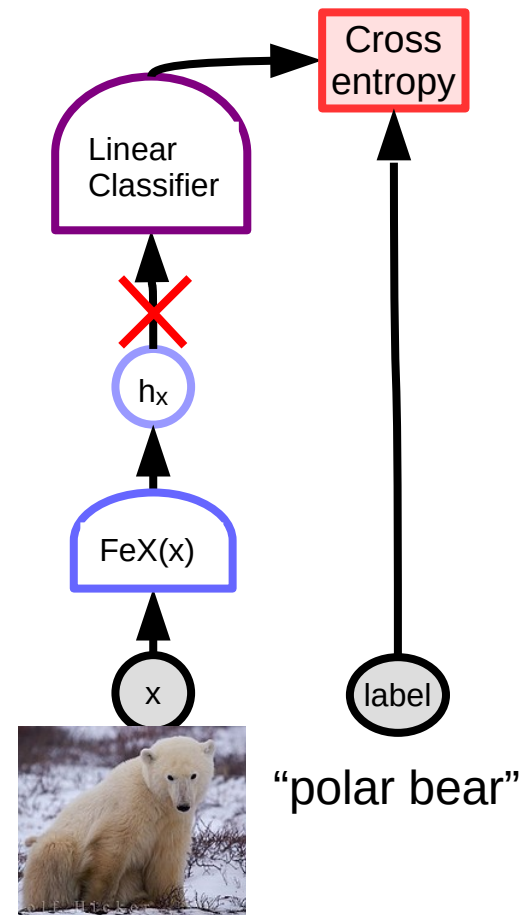


# SSL-Pretrained Joint Embedding for Image Recognition

JEA pretrained with VICReg



Training a supervised linear head



# VICReg: Results with linear head and semi-supervised.

| Method                                                   | Linear      |             | Semi-supervised |             |             |             |
|----------------------------------------------------------|-------------|-------------|-----------------|-------------|-------------|-------------|
|                                                          | Top-1       | Top-5       | Top-1           |             | Top-5       |             |
|                                                          |             |             | 1%              | 10%         | 1%          | 10%         |
| Supervised                                               | 76.5        | -           | 25.4            | 56.4        | 48.4        | 80.4        |
| MoCo <a href="#">He et al. (2020)</a>                    | 60.6        | -           | -               | -           | -           | -           |
| PIRL <a href="#">Misra &amp; Maaten (2020)</a>           | 63.6        | -           | -               | -           | 57.2        | 83.8        |
| CPC v2 <a href="#">Hénaff et al. (2019)</a>              | 63.8        | -           | -               | -           | -           | -           |
| CMC <a href="#">Tian et al. (2019)</a>                   | 66.2        | -           | -               | -           | -           | -           |
| SimCLR <a href="#">Chen et al. (2020a)</a>               | 69.3        | 89.0        | 48.3            | 65.6        | 75.5        | 87.8        |
| MoCo v2 <a href="#">Chen et al. (2020c)</a>              | 71.1        | -           | -               | -           | -           | -           |
| SimSiam <a href="#">Chen &amp; He (2020)</a>             | 71.3        | -           | -               | -           | -           | -           |
| SwAV <a href="#">Caron et al. (2020)</a>                 | 71.8        | -           | -               | -           | -           | -           |
| InfoMin Aug <a href="#">Tian et al. (2020)</a>           | 73.0        | <u>91.1</u> | -               | -           | -           | -           |
| OBoW <a href="#">Gidaris et al. (2021)</a>               | <u>73.8</u> | -           | -               | -           | <u>82.9</u> | <u>90.7</u> |
| BYOL <a href="#">Grill et al. (2020)</a>                 | <u>74.3</u> | <u>91.6</u> | 53.2            | 68.8        | <u>78.4</u> | <u>89.0</u> |
| SwAV (w/ multi-crop) <a href="#">Caron et al. (2020)</a> | <u>75.3</u> | -           | <u>53.9</u>     | <u>70.2</u> | <u>78.5</u> | <u>89.9</u> |
| Barlow Twins <a href="#">Zbontar et al. (2021)</a>       | 73.2        | 91.0        | <u>55.0</u>     | <u>69.7</u> | <u>79.2</u> | <u>89.3</u> |
| VICReg (ours)                                            | 73.2        | <u>91.1</u> | <u>54.8</u>     | <u>69.5</u> | <u>79.4</u> | <u>89.5</u> |

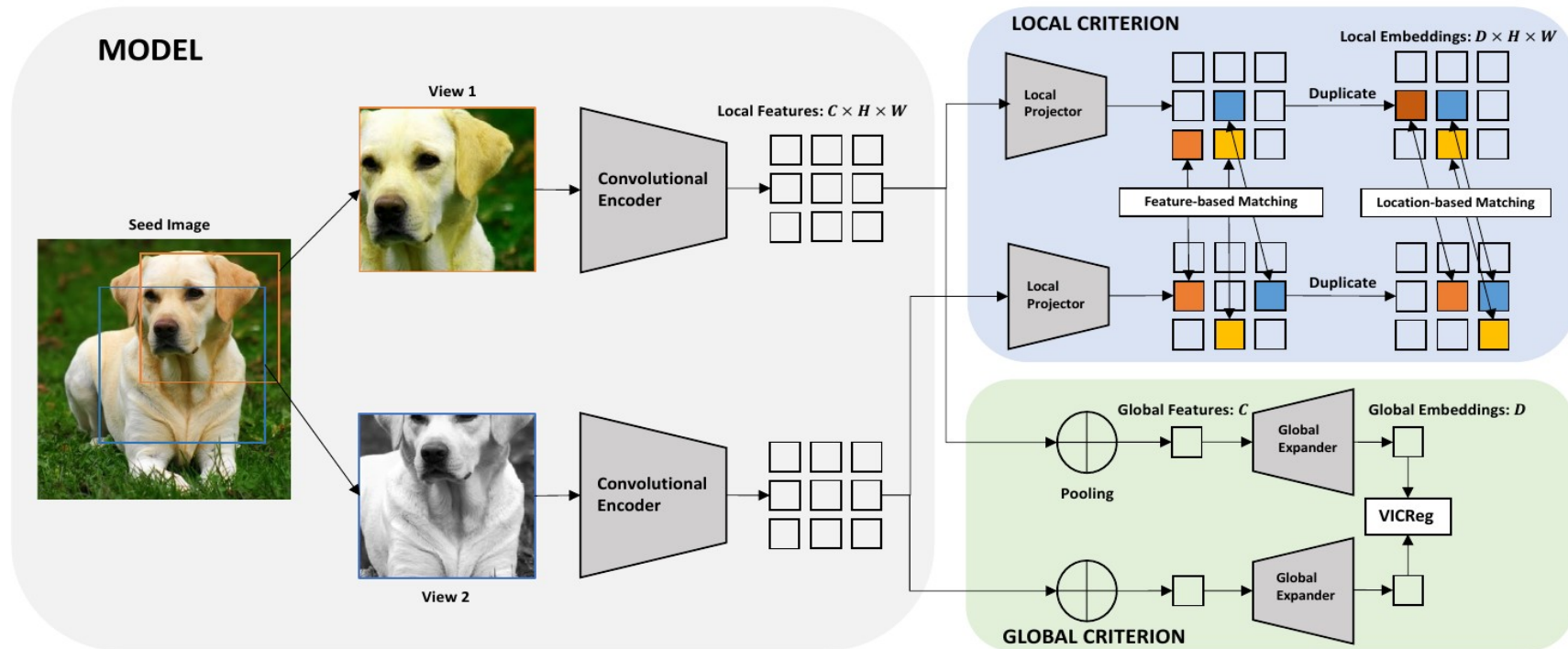
# VICReg: Results with transfer tasks.

| Method                                           | Linear Classification |             |             | Object Detection |                          |                          |
|--------------------------------------------------|-----------------------|-------------|-------------|------------------|--------------------------|--------------------------|
|                                                  | Places205             | VOC07       | iNat18      | VOC07+12         | COCO det                 | COCO seg                 |
| Supervised                                       | 53.2                  | 87.5        | 46.7        | 81.3             | 39.0                     | 35.4                     |
| MoCo <a href="#">He et al. (2020)</a>            | 46.9                  | 79.8        | 31.5        | -                | -                        | -                        |
| PIRL <a href="#">Misra &amp; Maaten (2020)</a>   | 49.8                  | 81.1        | 34.1        | -                | -                        | -                        |
| SimCLR <a href="#">Chen et al. (2020a)</a>       | 52.5                  | 85.5        | 37.2        | -                | -                        | -                        |
| MoCo v2 <a href="#">Chen et al. (2020c)</a>      | 51.8                  | 86.4        | 38.6        | 82.5             | 39.8                     | 36.1                     |
| SimSiam <a href="#">Chen &amp; He (2020)</a>     | -                     | -           | -           | 82.4             | -                        | -                        |
| BYOL <a href="#">Grill et al. (2020)</a>         | 54.0                  | <u>86.6</u> | <u>47.6</u> | -                | <u>40.4</u> <sup>†</sup> | <u>37.0</u> <sup>†</sup> |
| SwAV (m-c) <a href="#">Caron et al. (2020)</a>   | <u>56.7</u>           | <u>88.9</u> | <u>48.6</u> | <u>82.6</u>      | <u>41.6</u>              | <u>37.8</u>              |
| OBoW <a href="#">Gidaris et al. (2021)</a>       | <u>56.8</u>           | <u>89.3</u> | -           | <u>82.9</u>      | -                        | -                        |
| Barlow Twins <a href="#">Grill et al. (2020)</a> | 54.1                  | 86.2        | 46.5        | <u>82.6</u>      | <u>40.0</u> <sup>†</sup> | <u>36.7</u> <sup>†</sup> |
| VICReg (ours)                                    | <u>54.3</u>           | <u>86.6</u> | <u>47.0</u> | 82.4             | 39.4                     | 36.4                     |

# VICRegL: local matching latent variable for segmentation

## ► Latent variable optimization:

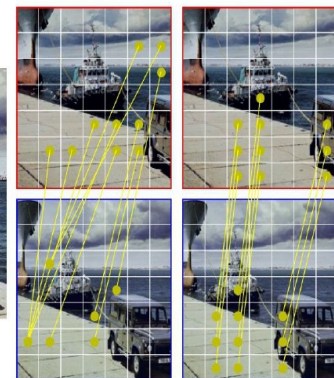
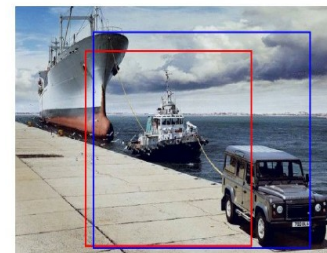
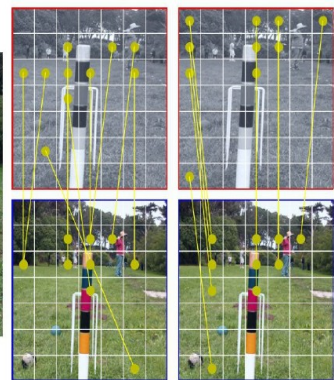
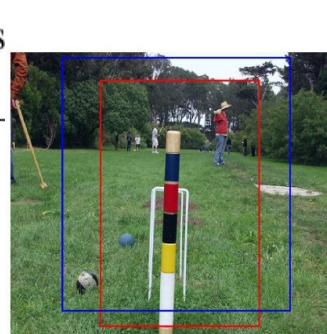
- Finds a pairing between local feature vectors of the two images
- [Bardes, Ponce, LeCun, NeurIPS 2022, arXiv:2210.01571]





# VICRegL: local matching latent variable for segmentation

| Method                              | Epochs | Linear Cls. (%)    | Linear Seg. (mIoU)   |                          |                      |
|-------------------------------------|--------|--------------------|----------------------|--------------------------|----------------------|
|                                     |        | ImageNet<br>Frozen | Pascal VOC<br>Frozen | Pascal VOC<br>Fine-Tuned | Cityscapes<br>Frozen |
| <i>Global features</i>              |        |                    |                      |                          |                      |
| MoCo v2 [Chen et al., 2020b]        | 200    | 67.5               | 35.6                 | 64.8                     | 14.3                 |
| SimCLR [Chen et al., 2020a]         | 400    | 68.2               | 45.9                 | 65.4                     | 17.9                 |
| BYOL [Grill et al., 2020]           | 300    | <b>72.3</b>        | 47.1                 | 65.7                     | 22.6                 |
| VICReg [Bardes et al., 2022]        | 300    | 71.5               | 47.8                 | 65.5                     | 23.5                 |
| <i>Local features</i>               |        |                    |                      |                          |                      |
| PixPro [Xie et al., 2021]           | 400    | 60.6               | 52.8                 | 67.5                     | 22.6                 |
| DenseCL [Wang et al., 2021]         | 200    | 65.0               | 45.3                 | 66.8                     | 11.2                 |
| DetCon [Hénaff et al., 2021]        | 1000   | 66.3               | 53.6                 | 67.4                     | 16.2                 |
| InsLoc [Yang et al., 2022]          | 400    | 45.0               | 24.1                 | 64.4                     | 7.0                  |
| CP <sup>2</sup> [Wang et al., 2022] | 820    | 53.1               | 21.7                 | 65.2                     | 8.4                  |
| ReSim [Xiao et al., 2021]           | 400    | 59.5               | 51.9                 | 67.3                     | 12.3                 |
| <i>Ours</i>                         |        |                    |                      |                          |                      |
| VICRegL $\alpha = 0.9$              | 300    | 71.2               | 54.0                 | 66.6                     | 25.1                 |
| VICRegL $\alpha = 0.75$             | 300    | 70.4               | <b>55.9</b>          | <b>67.6</b>              | <b>25.2</b>          |





# Distillation Methods

## ► Modified Siamese nets

- Predictor head eliminates variation of representations due to distortions

## ► Examples:

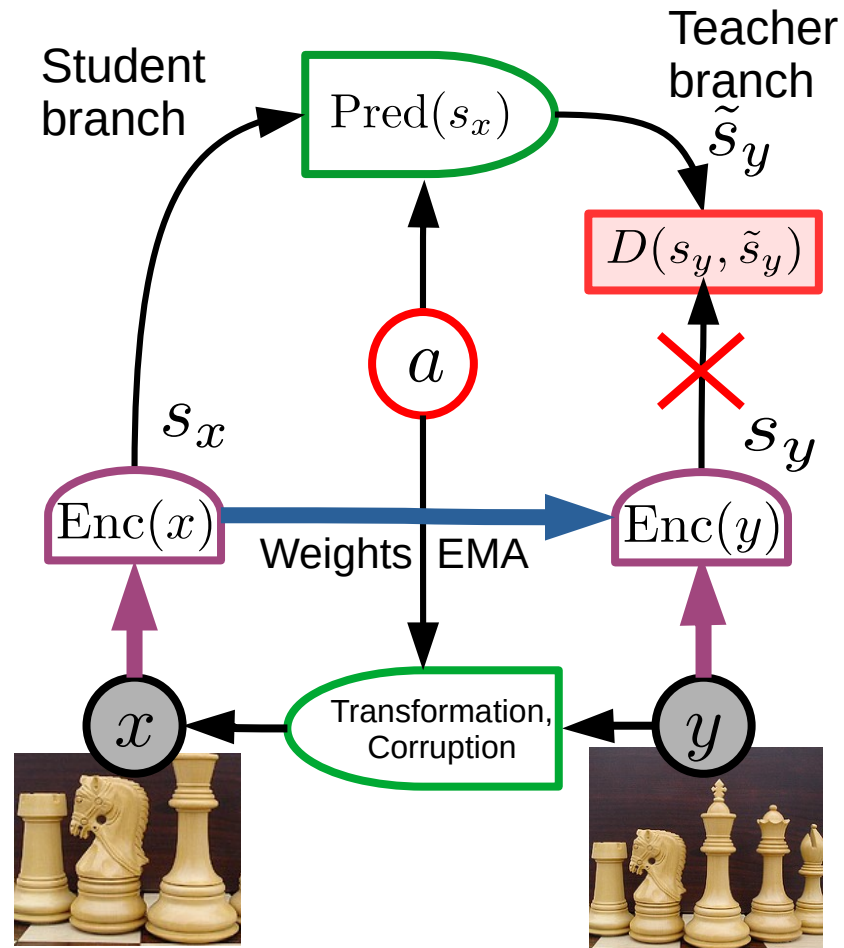
- Bootstrap Your Own Latents [Grill arXiv:2006.07733]
- SimSiam [Chen & He arXiv:2011.10566]
- DINOv2 [Oquab arXiv:2304.07193]

## ► Advantages

- No negative samples

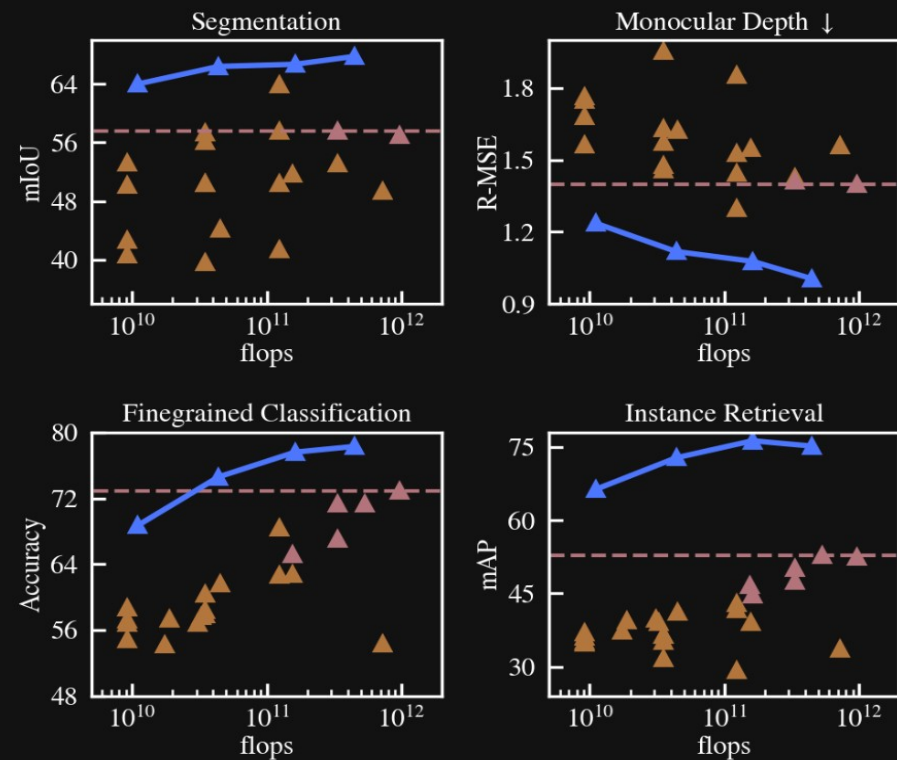
## ► Disadvantage:

- we don't completely understand why it works! [Tian et al. ArXiv:2102.06810]



# DINOv2: image foundation model

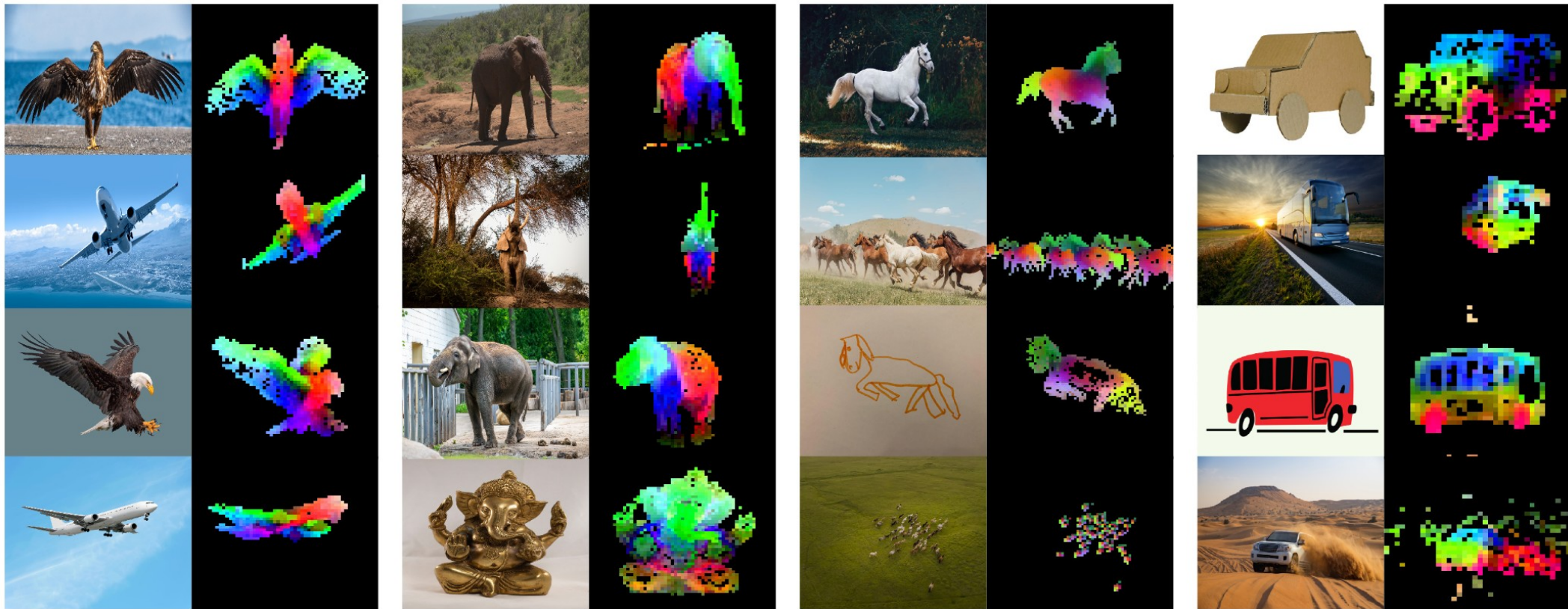
- ▶ **self-supervised** generic image features
- ▶ Demo: <https://dinov2.metademolab.com/>
- ▶ Paper: [Oquab et al. ArXiv:2304.07193]
- ▶ Classification
  - ▶ 86.5% on IN1k with frozen features and linear head.
- ▶ Fine-grained classification
- ▶ Depth estimation
- ▶ Semantic segmentation
- ▶ Instance Retrieval
- ▶ Dense & sparse feature matching



The DINOv2 family of models **drastically improves** over the previous state of the art in self-supervised learning (SSL), and **reaches performance comparable** with weakly-supervised features (WSL).

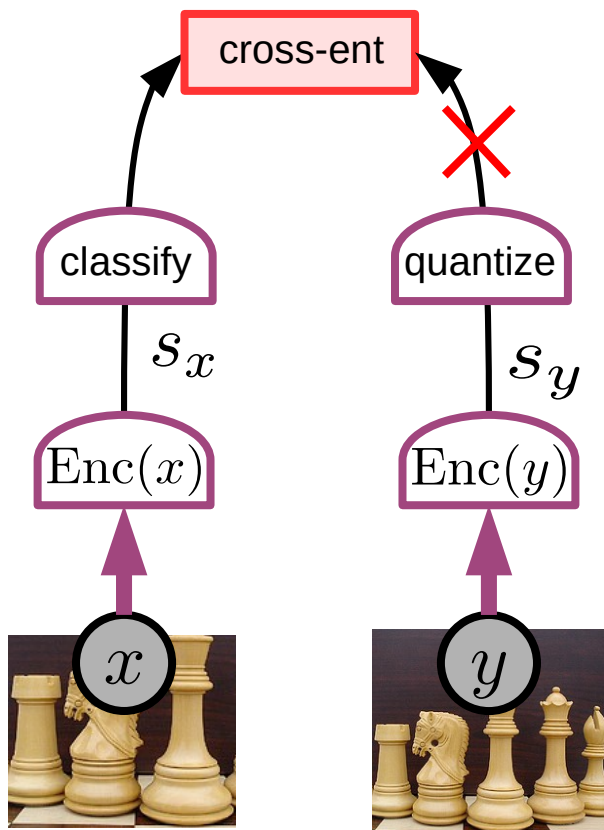
# DINOv2: image foundation model

- ▶ Demo: <https://dinov2.metademolab.com/>
- ▶ Paper: [Oquab et al. ArXiv:2304.07193]



# DINOv2: Joint Embedding Architecture

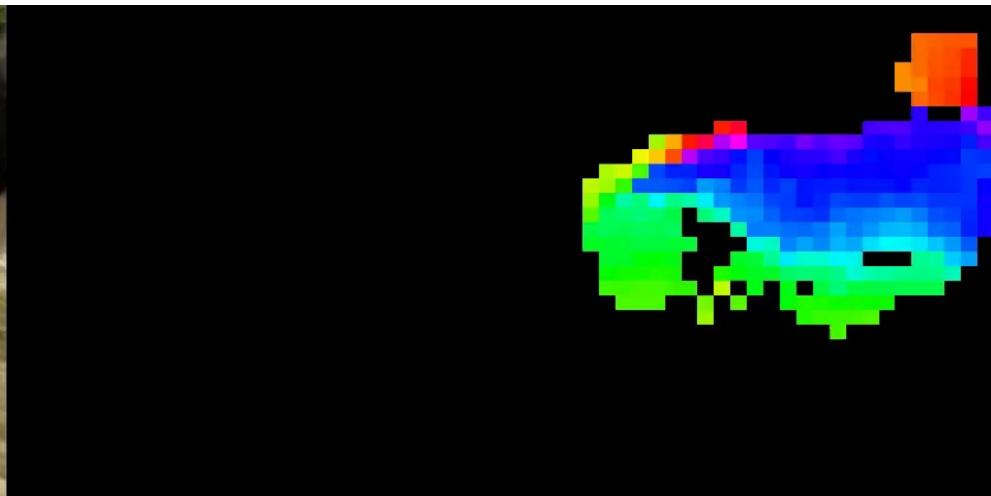
## ► SSL by distillation



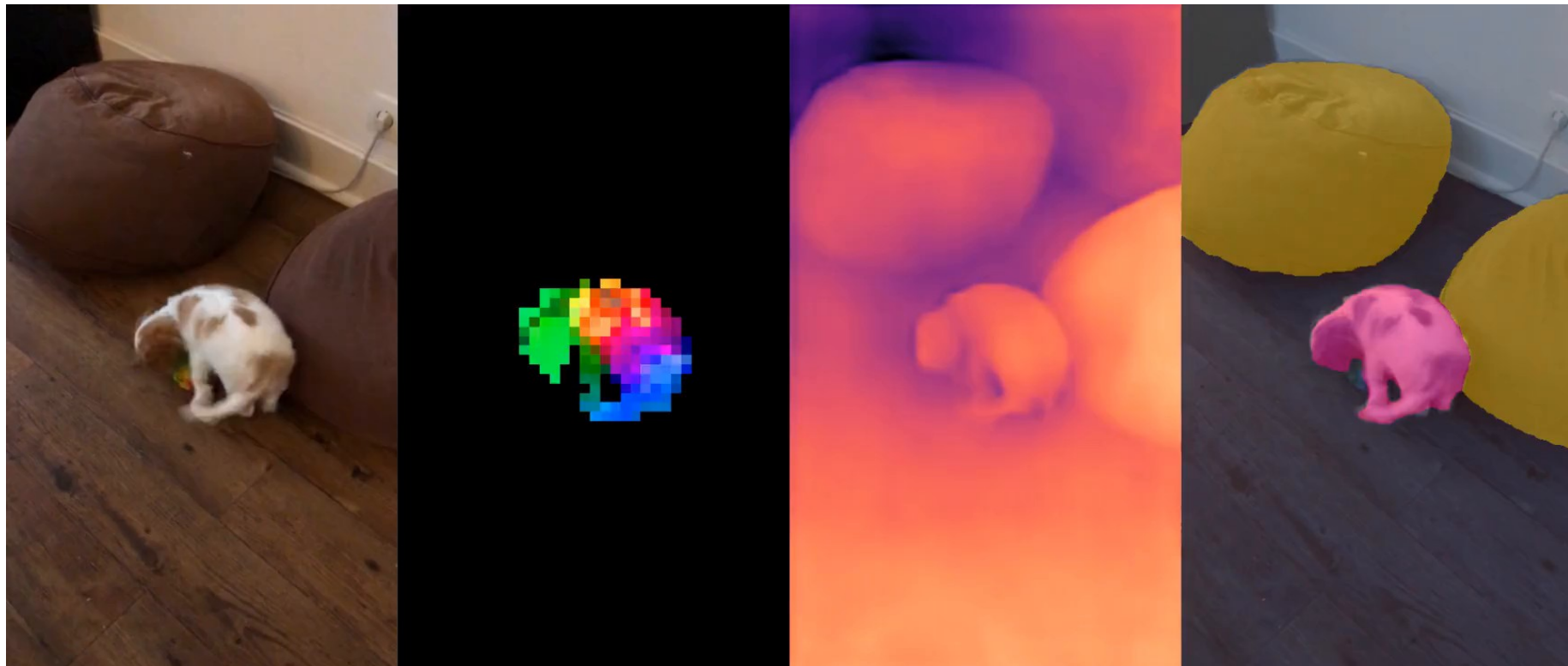
| Method            | Arch.                   | Data     | Text sup. | kNN         | linear      |             |             |
|-------------------|-------------------------|----------|-----------|-------------|-------------|-------------|-------------|
|                   |                         |          |           | val         | val         | ReaL        | V2          |
| Weakly supervised |                         |          |           |             |             |             |             |
| CLIP              | ViT-L/14                | WIT-400M | ✓         | 79.8        | 84.3        | 88.1        | 75.3        |
| CLIP              | ViT-L/14 <sub>336</sub> | WIT-400M | ✓         | 80.5        | 85.3        | 88.8        | 75.8        |
| SWAG              | ViT-H/14                | IG3.6B   | ✓         | 82.6        | 85.7        | 88.7        | 77.6        |
| OpenCLIP          | ViT-H/14                | LAION    | ✓         | 81.7        | 84.4        | 88.4        | 75.5        |
| OpenCLIP          | ViT-G/14                | LAION    | ✓         | 83.2        | 86.2        | 89.4        | 77.2        |
| EVA-CLIP          | ViT-g/14                | custom*  | ✓         | <b>83.5</b> | 86.4        | 89.3        | 77.4        |
| Self-supervised   |                         |          |           |             |             |             |             |
| MAE               | ViT-H/14                | INet-1k  | ✗         | 49.4        | 76.6        | 83.3        | 64.8        |
| DINO              | ViT-S/8                 | INet-1k  | ✗         | 78.6        | 79.2        | 85.5        | 68.2        |
| SEERv2            | RG10B                   | IG2B     | ✗         | —           | 79.8        | —           | —           |
| MSN               | ViT-L/7                 | INet-1k  | ✗         | 79.2        | 80.7        | 86.0        | 69.7        |
| EsViT             | Swin-B/W=14             | INet-1k  | ✗         | 79.4        | 81.3        | 87.0        | 70.4        |
| Mugs              | ViT-L/16                | INet-1k  | ✗         | 80.2        | 82.1        | 86.9        | 70.8        |
| iBOT              | ViT-L/16                | INet-22k | ✗         | 72.9        | 82.3        | 87.5        | 72.4        |
| DINOv2            | ViT-S/14                | LVD-142M | ✗         | 79.0        | 81.1        | 86.6        | 70.9        |
|                   | ViT-B/14                | LVD-142M | ✗         | 82.1        | 84.5        | 88.3        | 75.1        |
|                   | ViT-L/14                | LVD-142M | ✗         | <b>83.5</b> | 86.3        | 89.5        | 78.0        |
|                   | ViT-g/14                | LVD-142M | ✗         | <b>83.5</b> | <b>86.5</b> | <b>89.6</b> | <b>78.4</b> |



- Feature visualization: RGB = top 3 principal components



► Feature extraction, depth estimation, segmentation



# Canopy Height Map using DINOv2

- ▶ Estimates tree canopy height from satellite images using DINOv2 features
- ▶ Using ground truth from Lidar images
- ▶ 0.5 meter resolution images
- ▶ [ArXiv:2304.07213]
- ▶ Tolan et al.: Sub-meter resolution canopy height maps using self-supervised learning and a vision transformer trained on Aerial and GEDI Lidar

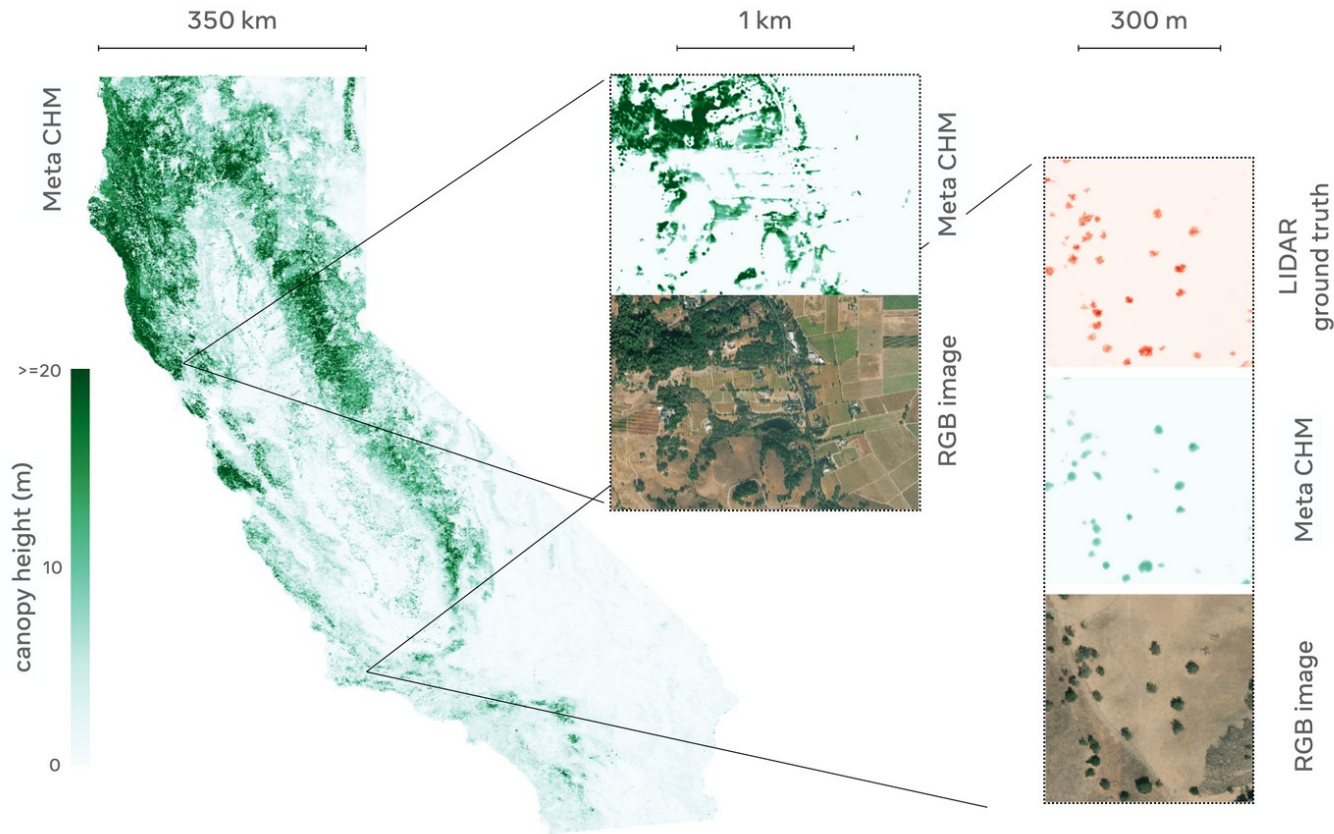
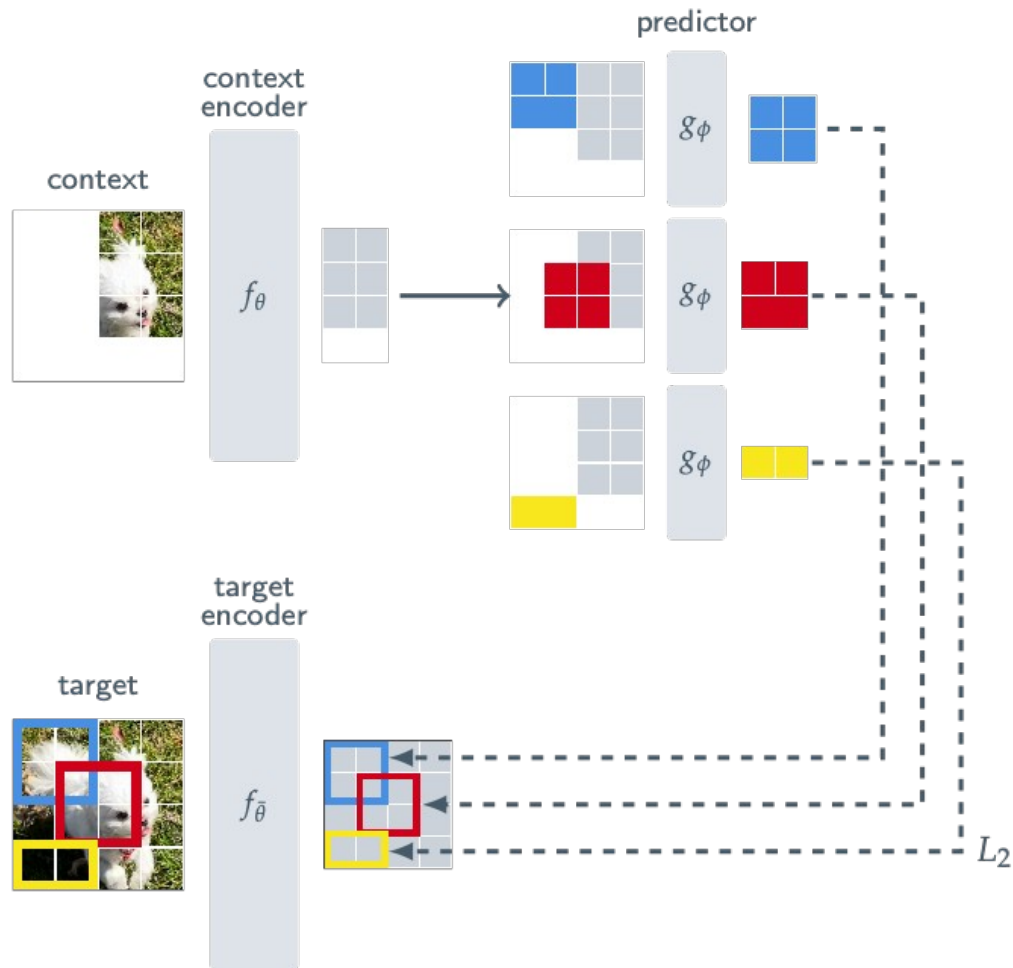


Figure 1: Canopy Height Map (CHM) for California, with inset showing zoomed-in region with input RGB imagery and LIDAR ground truth

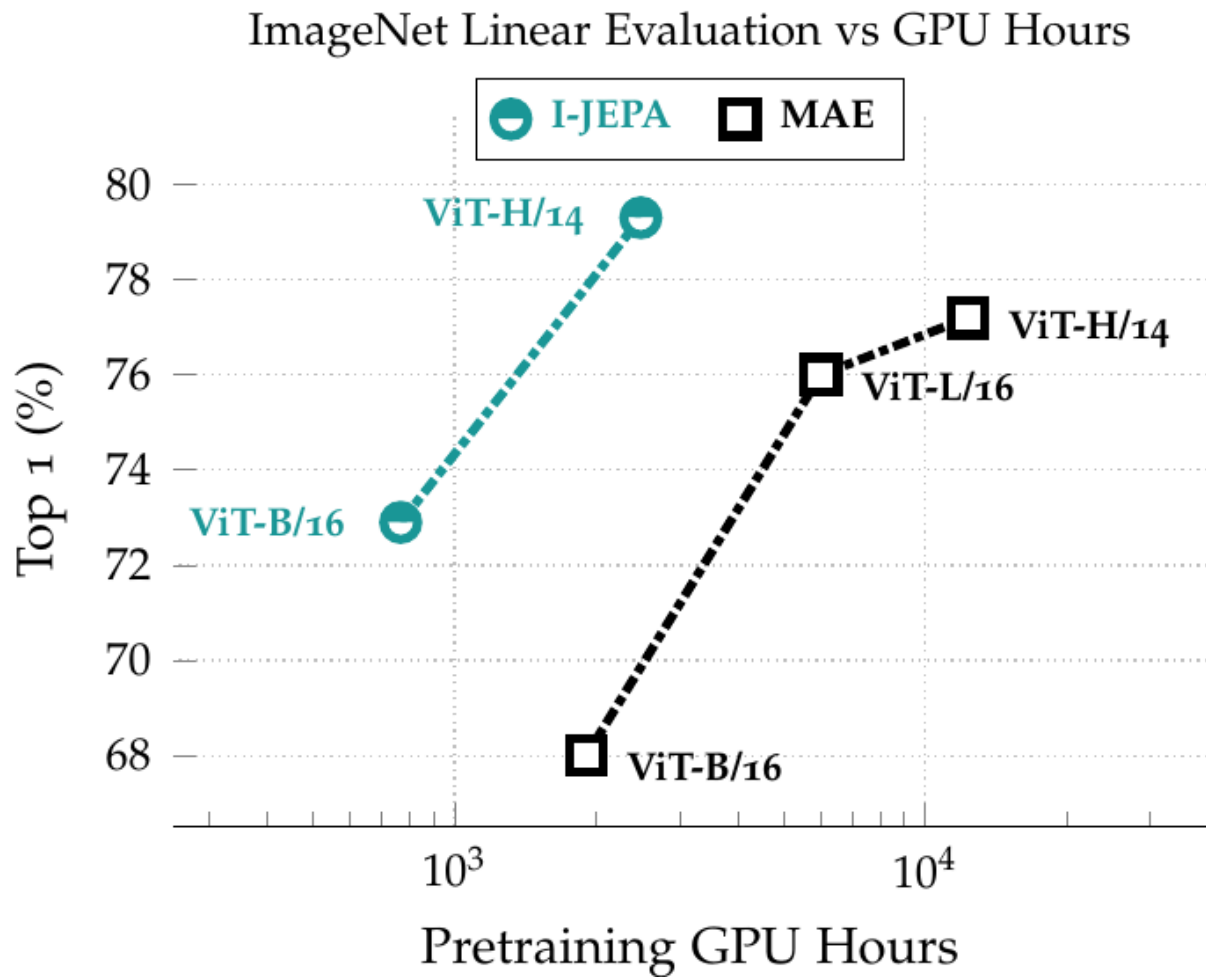


- ### Semi-Supervised ImageNet-1K 1% Evaluation vs GPU Hours



# I-JEPA Results

- ▶ Training is fast
- ▶ Non-generative method beat reconstruction-based generative methods such as Masked Auto-Encoder
  - ▶ (with a frozen trunk).



# I-JEPA Results on ImageNet

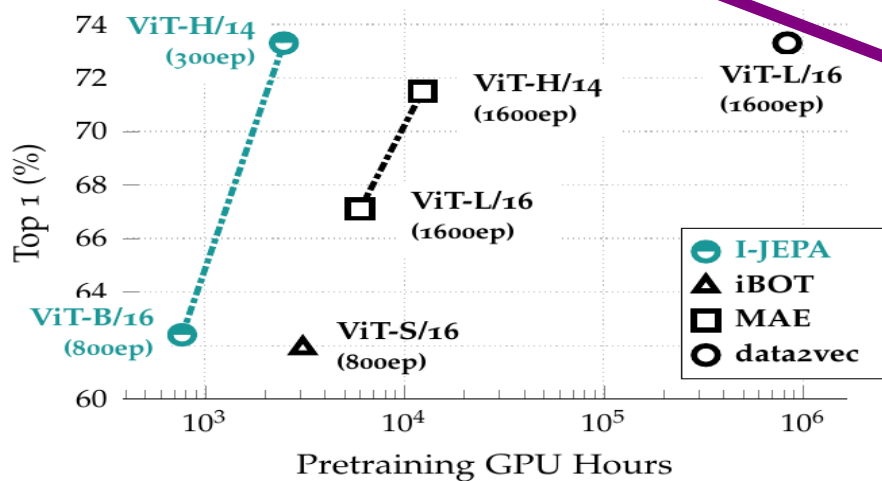
- ▶ JEPA better than generative architecture on pixels.
- ▶ Closing the gap with methods that use data augments
- ▶ Methods with only masking
  - ▶ No data augmentation →
- ▶ Methods with data augmentation
  - ▶ Similar to SimCLR →

| Targets               | Arch.    | Epochs | Top-1       |
|-----------------------|----------|--------|-------------|
| Target-Encoder Output | ViT-L/16 | 500    | <b>66.9</b> |
| Pixels                | ViT-L/16 | 800    | 40.7        |

| Method                                             | Arch.                   | Epochs | Top-1       |
|----------------------------------------------------|-------------------------|--------|-------------|
| <i>Methods without view data augmentations</i>     |                         |        |             |
| data2vec [7]                                       | ViT-L/16                | 1600   | 53.5        |
|                                                    | ViT-B/16                | 1600   | 68.0        |
| MAE [34]                                           | ViT-L/16                | 1600   | 76.0        |
|                                                    | ViT-H/14                | 1600   | 77.2        |
| I-JEPA                                             | ViT-B/16                | 600    | 72.9        |
|                                                    | ViT-L/16                | 600    | 77.5        |
|                                                    | ViT-H/14                | 300    | 79.3        |
|                                                    | ViT-H/16 <sub>448</sub> | 300    | <b>81.1</b> |
| <i>Methods using extra view data augmentations</i> |                         |        |             |
| SimCLR v2 [20]                                     | RN152 (2×)              | 800    | 79.1        |
| DINO [17]                                          | ViT-B/8                 | 300    | 80.1        |
| iBOT [74]                                          | ViT-L/16                | 250    | <b>81.0</b> |

# I-JEPA Results on ImageNet with 1% training

- ▶ JEPA better than generative architecture on pixels.
- ▶ Closing the gap with methods that use data augments
- ▶ Methods with only masking
- ▶ Methods with data augmentation



| Method                                             | Arch.                   | Epochs | Top-1       |
|----------------------------------------------------|-------------------------|--------|-------------|
| <i>Methods without view data augmentations</i>     |                         |        |             |
| data2vec [7]                                       | ViT-L/16                | 1600   | 73.3        |
| MAE [34]                                           | ViT-L/16                | 1600   | 67.1        |
|                                                    | ViT-H/14                | 1600   | 71.5        |
| I-JEPA                                             | ViT-L/16                | 600    | 69.4        |
|                                                    | ViT-H/14                | 300    | 73.3        |
|                                                    | ViT-H/16 <sub>448</sub> | 300    | <b>77.3</b> |
| <i>Methods using extra view data augmentations</i> |                         |        |             |
| iBOT [74]                                          | ViT-B/16                | 250    | 69.7        |
| DINO [17]                                          | ViT-B/8                 | 300    | 70.0        |
| SimCLR v2 [33]                                     | RN151 (2×)              | 800    | 70.2        |
| BYOL [33]                                          | RN200 (2×)              | 800    | 71.2        |
| MSN [3]                                            | ViT-B/4                 | 300    | <b>75.7</b> |

# Sample contrastive vs Dimension contrastive?

- ▶ **[Garrido et al. Arxiv:2206.02574 , ICLR2023]** (outstanding paper, honorable mention)
- ▶ “ON THE DUALITY BETWEEN CONTRASTIVE AND NON CONTRASTIVE SELF-SUPERVISED LEARNING”

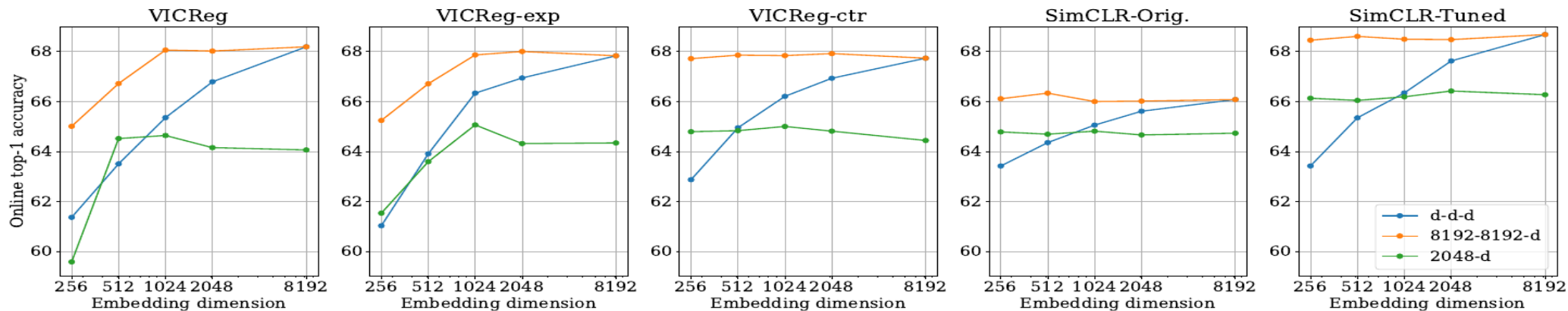


Figure 1: VICReg, VICReg-exp and VICReg-ctr perform similarly in 100 epochs training, validating empirically our theoretical result. While the original implementation of SimCLR performs significantly worse – which is unexpected per our theory – we are able to improve its performance to VICReg’s level. This further validates our findings. While different projector architectures impact performance, behaviours are similar across methods. Confer supplementary section **H** for numerical values and hyperparameters.

# Video-JEPA

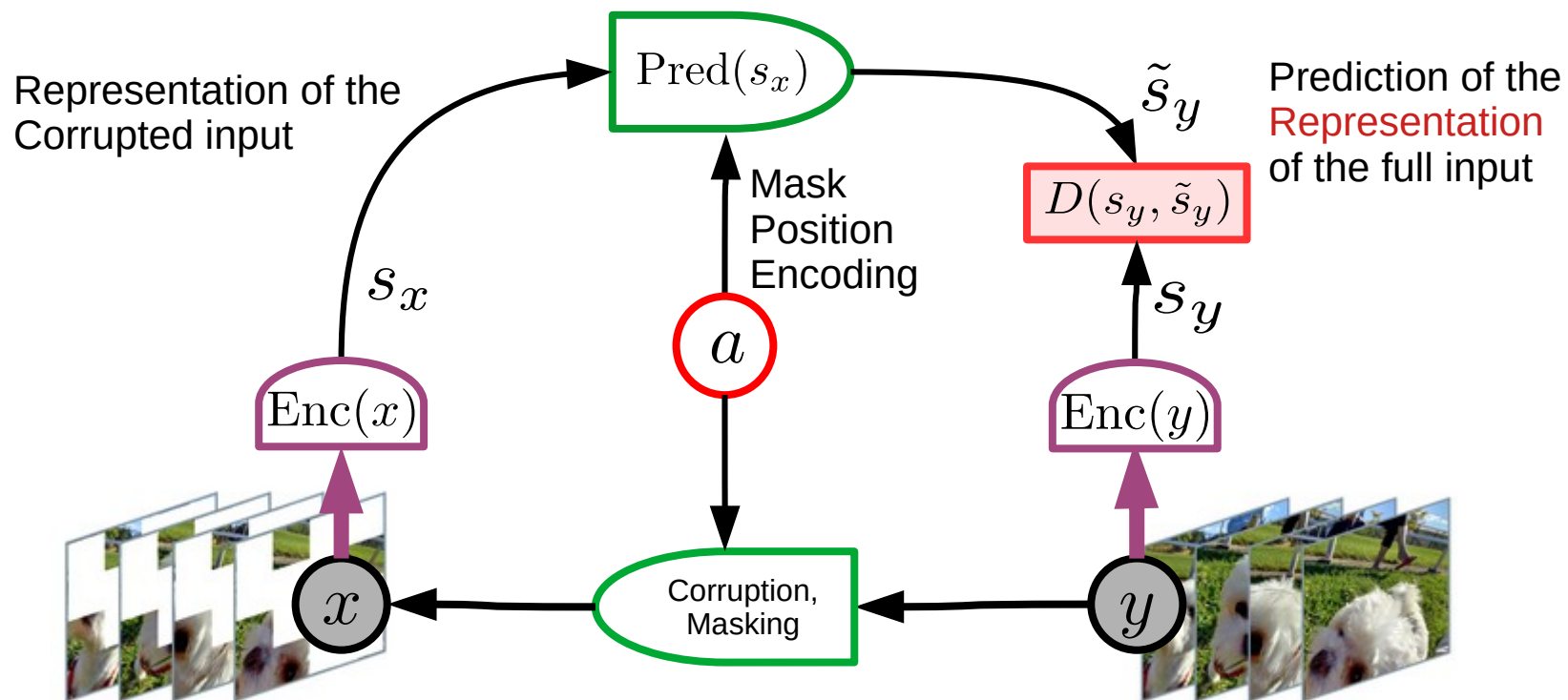
<https://github.com/facebookresearch/jepa>

Search for V-JEPA

“Revisiting Feature Prediction for Learning Visual Representations from Video”  
Adrien Bardes, Quentin Garrido, Jean Ponce, Xinlei Chen, Michael Rabbat,  
Yann LeCun, Mahmoud Assran<sup>1</sup>, Nicolas Ballas

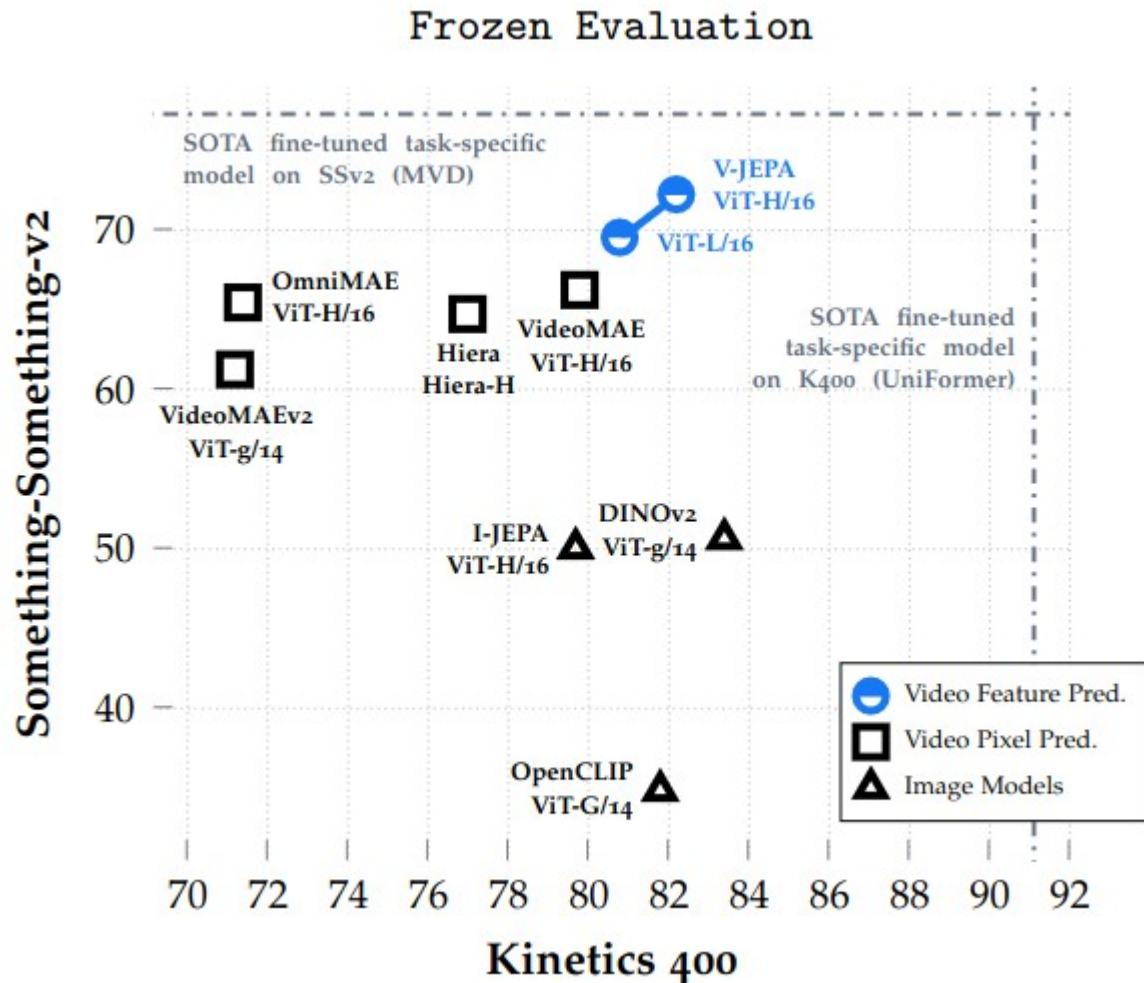


## ► [Bardes et al. 2024]



# V-JEPA: results on action recognition

- ▶ Supervised head on frozen backbone.
- ▶ Comparison with generative models: OmniMAE, VideoMAE, Hiera
- ▶ Comparison with image models: I-JEPA, DINOv2, OpenCLIP



# V-JEPA: results for low-shot action recognition

- ▶ Rows 1-3: generative architectures with reconstruction
- ▶ Row 4: V-JEPA
- ▶ Supervised head on frozen backbone.

| Method     | Arch.                   | Frozen Evaluation |            |            |                  |            |            |
|------------|-------------------------|-------------------|------------|------------|------------------|------------|------------|
|            |                         | K400<br>(16×8×3)  |            |            | SSv2<br>(16×2×3) |            |            |
|            |                         | 5%                | 10%        | 50%        | 5%               | 10%        | 50%        |
| MVD        | ViT-L/16                | 62.6 ± 0.2        | 68.3 ± 0.2 | 77.2 ± 0.3 | 42.9 ± 0.8       | 49.5 ± 0.6 | 61.0 ± 0.2 |
| VideoMAE   | ViT-H/16                | 62.3 ± 0.3        | 68.5 ± 0.2 | 78.2 ± 0.1 | 41.4 ± 0.8       | 48.1 ± 0.2 | 60.5 ± 0.4 |
| VideoMAEv2 | ViT-g/14                | 37.0 ± 0.3        | 48.8 ± 0.4 | 67.8 ± 0.1 | 28.0 ± 1.0       | 37.3 ± 0.3 | 54.0 ± 0.3 |
| V-JEPA     | ViT-H/16 <sub>384</sub> | 68.2 ± 0.2        | 72.8 ± 0.2 | 80.6 ± 0.2 | 54.0 ± 0.2       | 59.3 ± 0.5 | 67.9 ± 0.2 |

# V-JEPA: training on video vs training on images

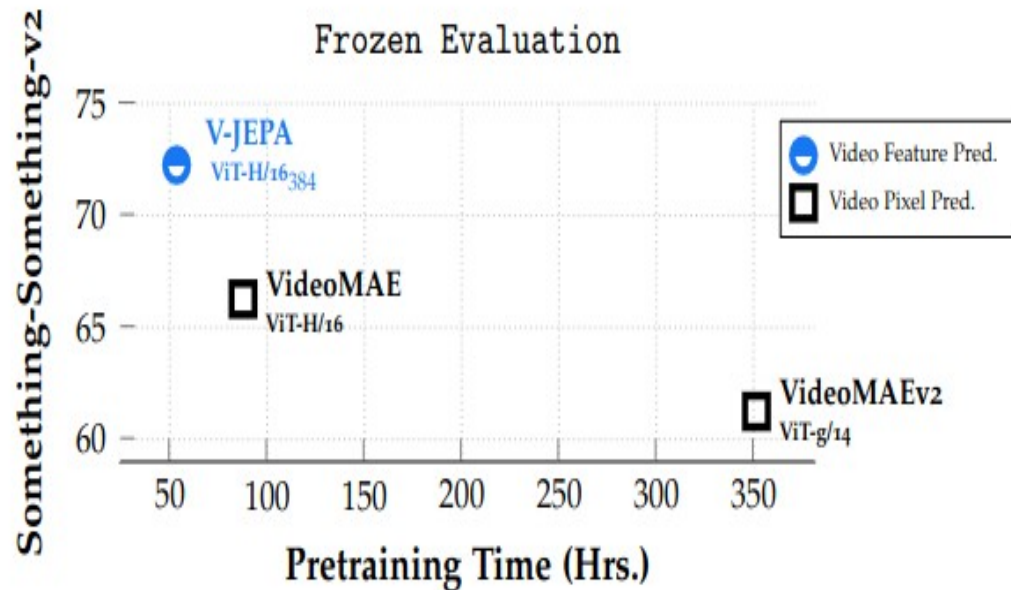
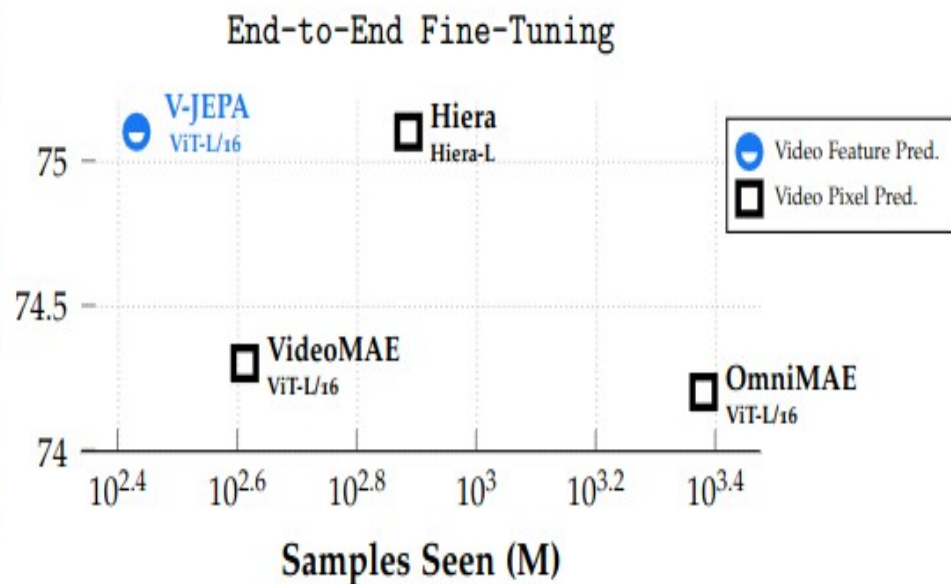
- ▶ Frozen evaluation
- ▶ Pre-training on video gives better results on action recognition
- ▶ V-JEPA: best results on ImageNet1K among video models

| Method                       | Arch.                   | Params. | Data       | Video Tasks      |                  |             | Image Tasks |             |             |
|------------------------------|-------------------------|---------|------------|------------------|------------------|-------------|-------------|-------------|-------------|
|                              |                         |         |            | K400<br>(16×8×3) | SSv2<br>(16×2×3) | AVA         | IN1K        | Places205   | iNat21      |
| Methods pretrained on Images |                         |         |            |                  |                  |             |             |             |             |
| I-JEPA                       | ViT-H/16 <sub>512</sub> | 630M    | IN22K      | 79.7             | 50.0             | 19.8        | 84.4        | 66.5        | 85.7        |
| OpenCLIP                     | ViT-G/14                | 1800M   | LAION      | 81.8             | 34.8             | 23.2        | 85.3        | <b>70.2</b> | 83.6        |
| DINOv2                       | ViT-g/14                | 1100M   | LVD-142M   | <b>83.4</b>      | 50.6             | 24.3        | <b>86.2</b> | 68.4        | <b>88.8</b> |
| Methods pretrained on Videos |                         |         |            |                  |                  |             |             |             |             |
| MVD                          | ViT-L/16                | 200M    | IN1K+K400  | 79.4             | 66.5             | 19.7        | 73.3        | 59.4        | 65.7        |
| OmniMAE                      | ViT-H/16                | 630M    | IN1K+SSv2  | 71.4             | 65.4             | 16.0        | 76.3        | 60.6        | 72.4        |
| VideoMAE                     | ViT-H/16                | 630M    | K400       | 79.8             | 66.2             | 20.7        | 72.3        | 59.1        | 65.5        |
| VideoMAEv2                   | ViT-g/14                | 1100M   | Un.Hybrid  | 71.2             | 61.2             | 12.9        | 71.4        | 60.6        | 68.3        |
| Hiera                        | Hiera-H                 | 670M    | K400       | 77.0             | 64.7             | 17.5        | 71.4        | 59.5        | 61.7        |
| V-JEPA                       | ViT-L/16                | 200M    | VideoMix2M | 80.8             | 69.5             | 25.6        | 74.8        | 60.3        | 67.8        |
|                              | ViT-H/16                | 630M    |            | <b>82.0</b>      | 71.4             | <b>25.8</b> | 75.9        | 61.7        | 67.9        |
|                              | ViT-H/16 <sub>384</sub> | 630M    |            | 81.9             | <b>72.2</b>      | 25.0        | <b>77.4</b> | <b>62.8</b> | <b>72.6</b> |

# V-JEPA: sample efficiency and learning speed

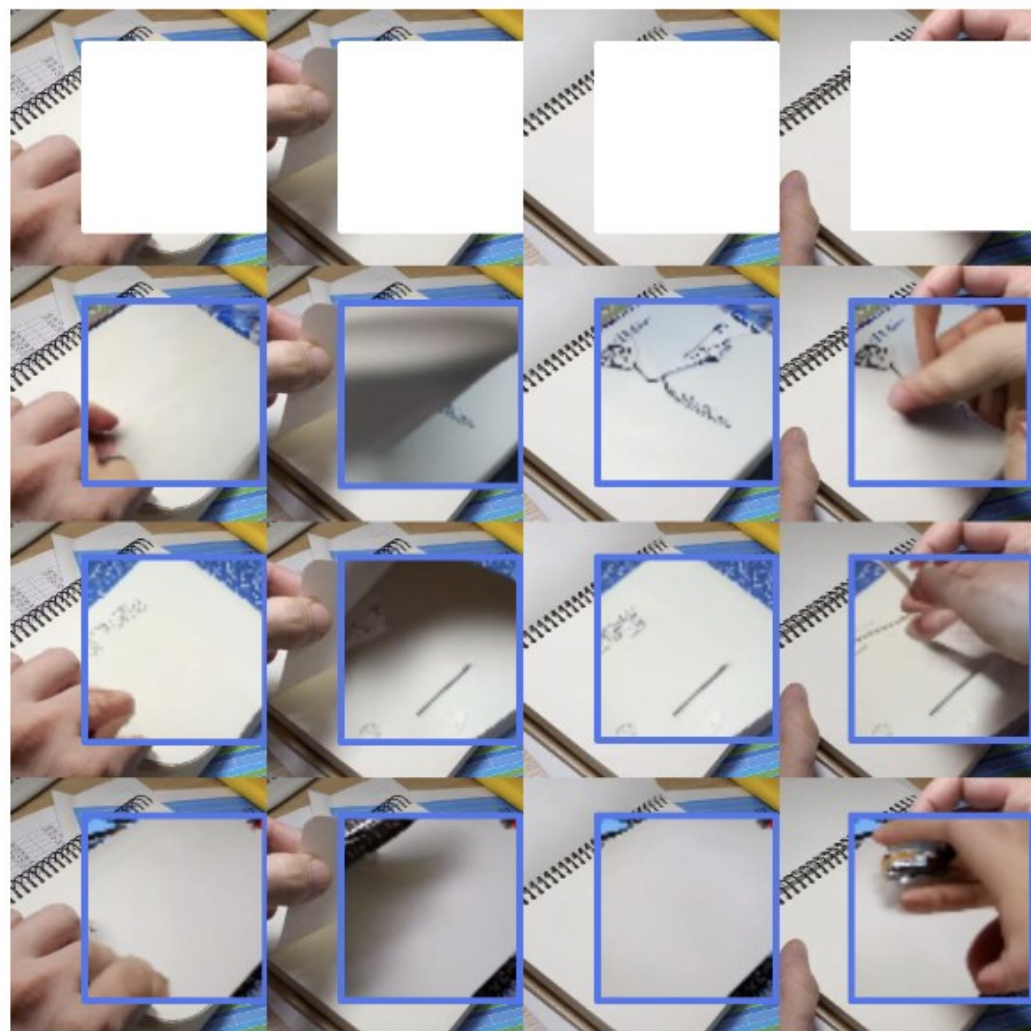
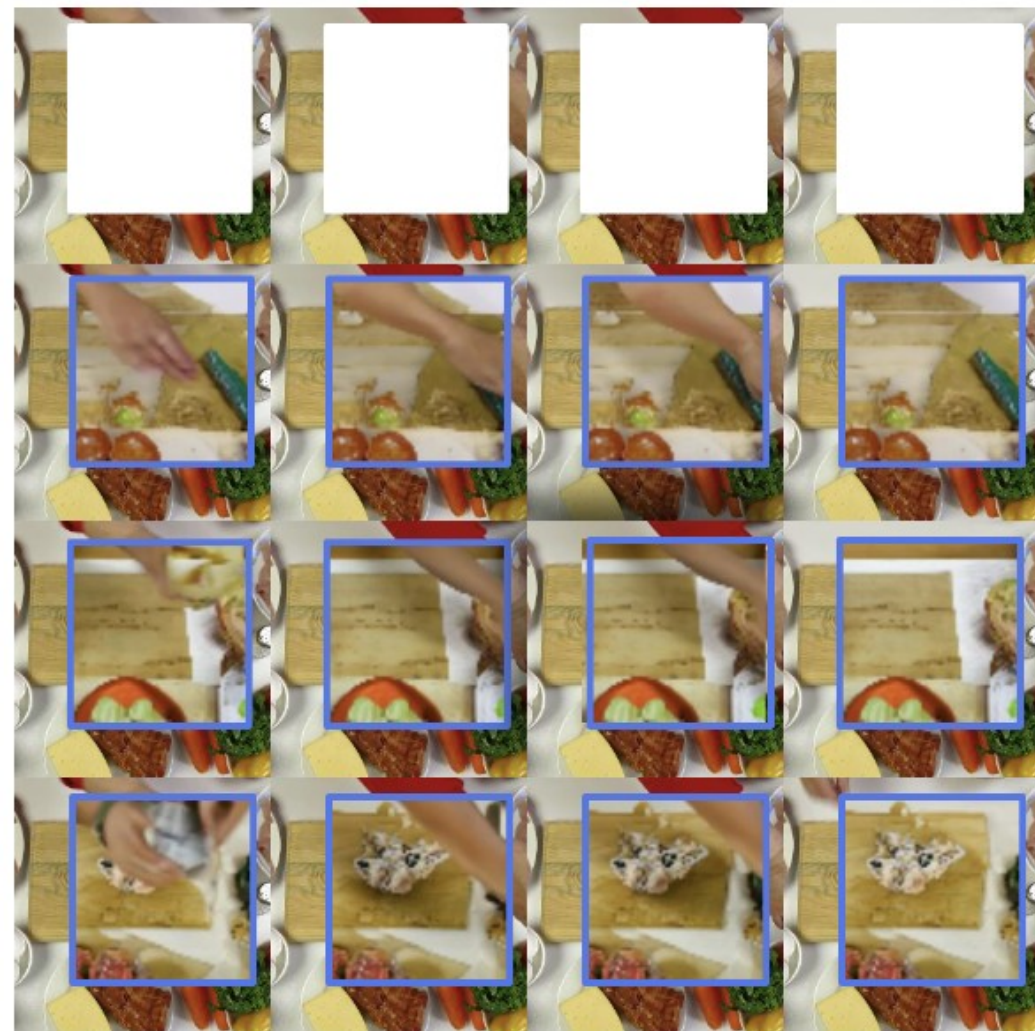
- Evaluation on Something-Something-v2
- Comparison with reconstruction-based generative methods

Something-Something-v2





# V-JEPA: Reconstruction with a separately-trained decoder





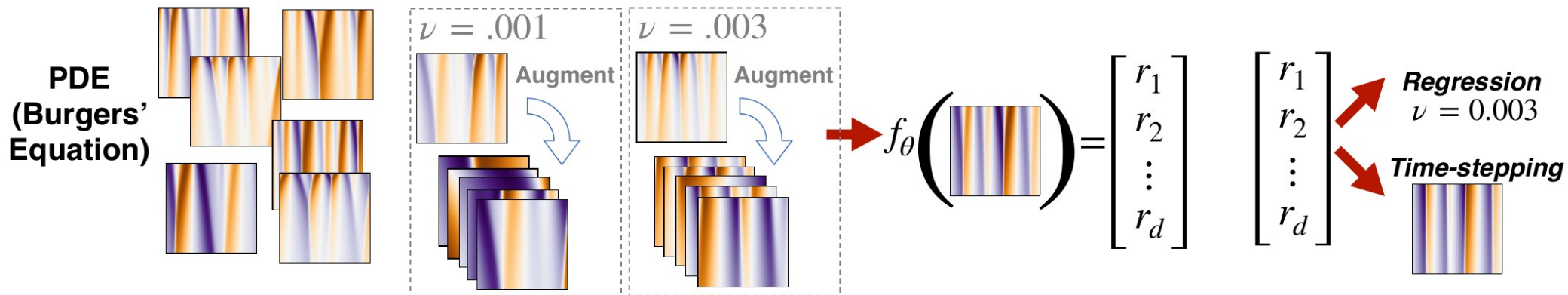
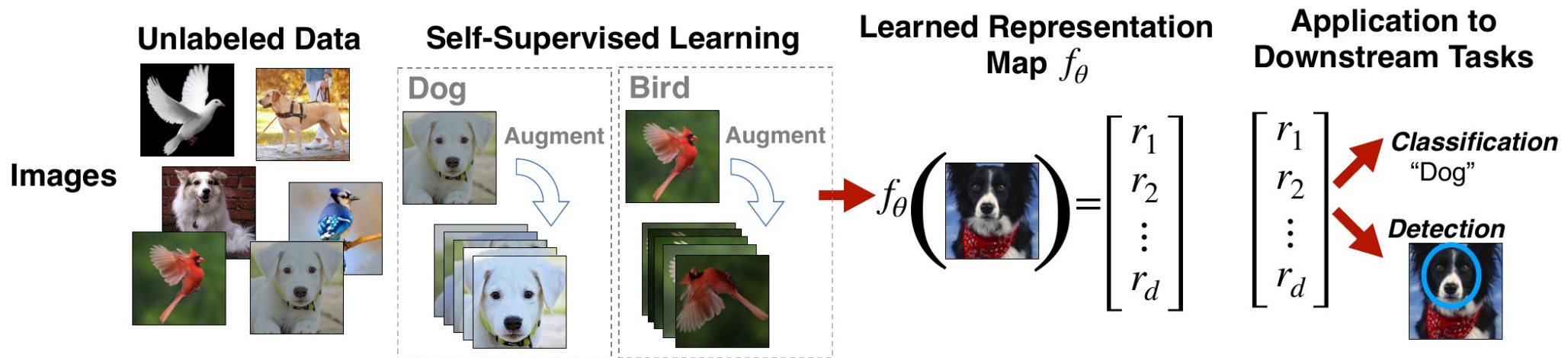
# SSL for PDEs

ArXiv:2307.05432   NeurIPS 2023

Self-Supervised Learning with Lie Symmetries for Partial Differential Equations

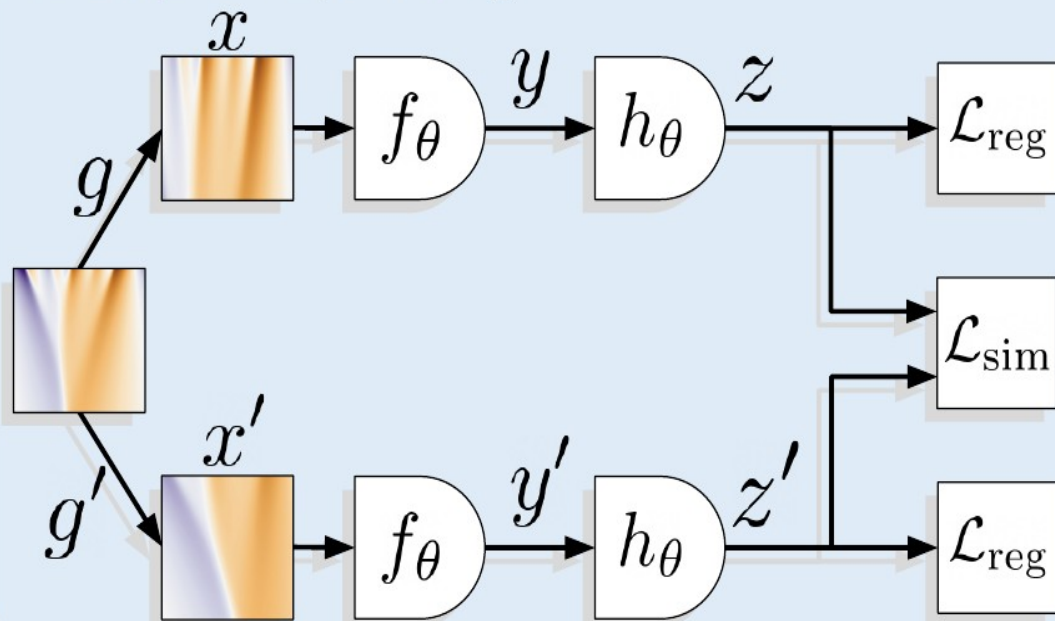
Grégoire Mialon, Quentin Garrido, Hannah Lawrence, Danyal Rehman, Yann LeCun, Bobak T. Kiani

# SSL for PDE: extracting dynamical parameters with VICReg

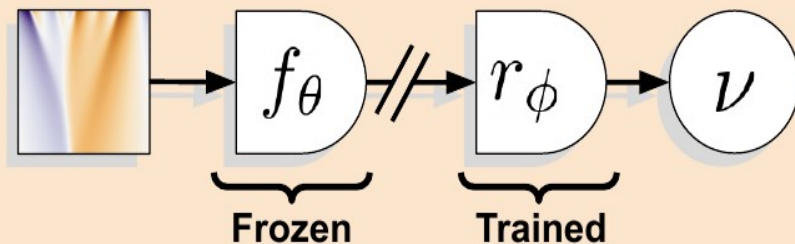


# Using VICReg to learn representations of the equation.

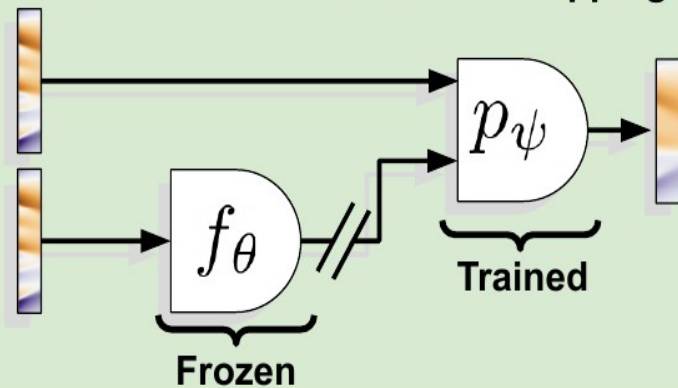
## Self-supervised pretraining



## Supervised downstream task



## Representation conditioned time-stepping



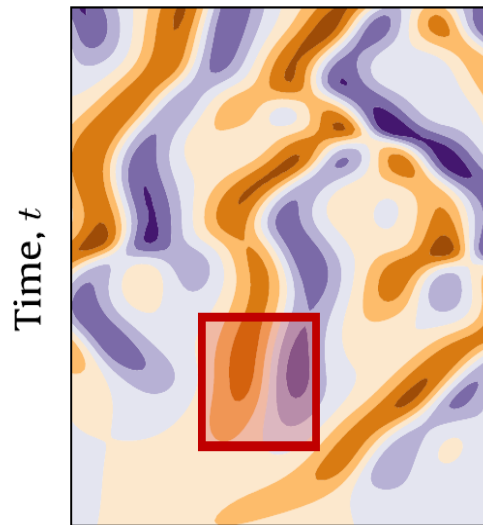
# SSL for PDE

An example: the **Kuramoto-Sivashinsky (KS)** equation is a model of chaotic flow given by

$$u_t + uu_x + u_{xx} + u_{xxxx} = 0,$$

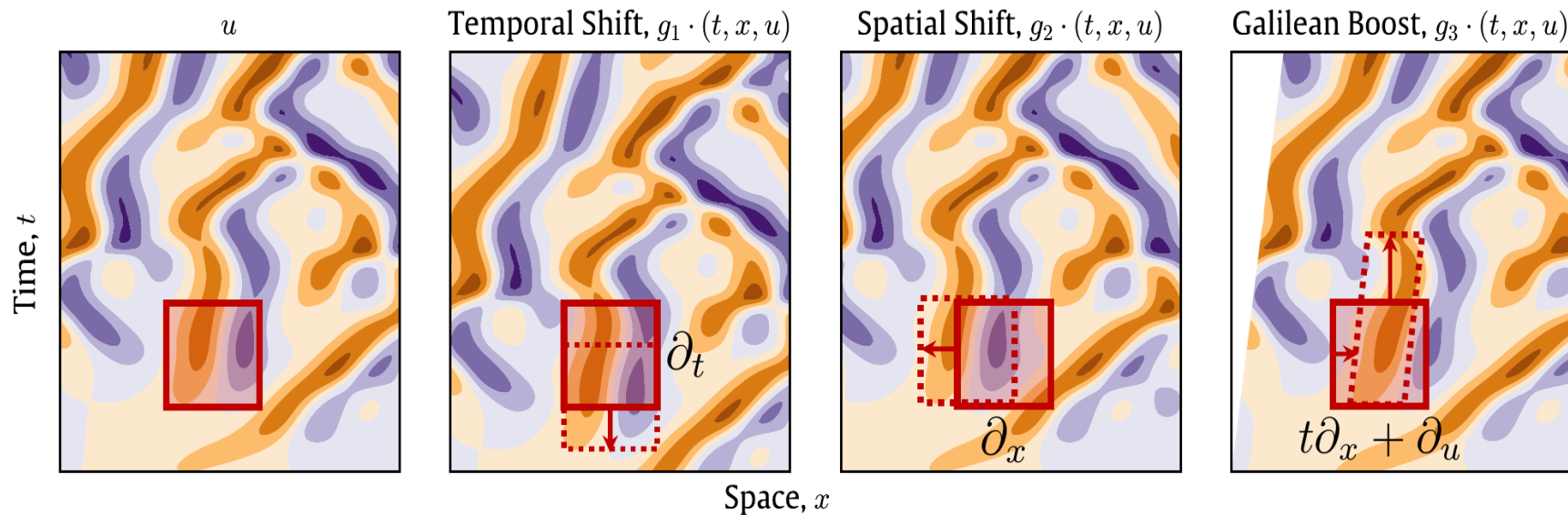
where  $u(x, t)$  is the dependent variable.

- Often shows up in reaction-diffusion systems or flame propagation problems.
- Solution can be seen as an image...
- Admit Lie point symmetries: smooth transformations of a solution producing another solution to the same PDE.
- Can be used to learn models [Brandstetter et al., 2022].



A 1D solution to KS (x-axis is space).

# SSL for PDE: Data “augmentation”



One parameter Lie point symmetries for the Kuramoto-Sivashinsky (KS) PDE. Left to right: un-modified solution ( $u$ ), temporal shifts ( $g_1$ ), spatial shifts ( $g_2$ ), and Galilean boosts ( $g_3$ ) with corresponding infinitesimal transformations in the Lie algebra placed inside the figure. The shaded red square denotes the original  $(x, t)$ , while the dotted line represents the same points after the augmentation is applied.

Temporal Shift:  $g_1(\epsilon) : (x, t, u) \mapsto (x, t + \epsilon, u)$

Spatial Shift:  $g_2(\epsilon) : (x, t, u) \mapsto (x + \epsilon, t, u)$

Galilean Boost:  $g_3(\epsilon) : (x, t, u) \mapsto (x + \epsilon t, t, u + \epsilon)$

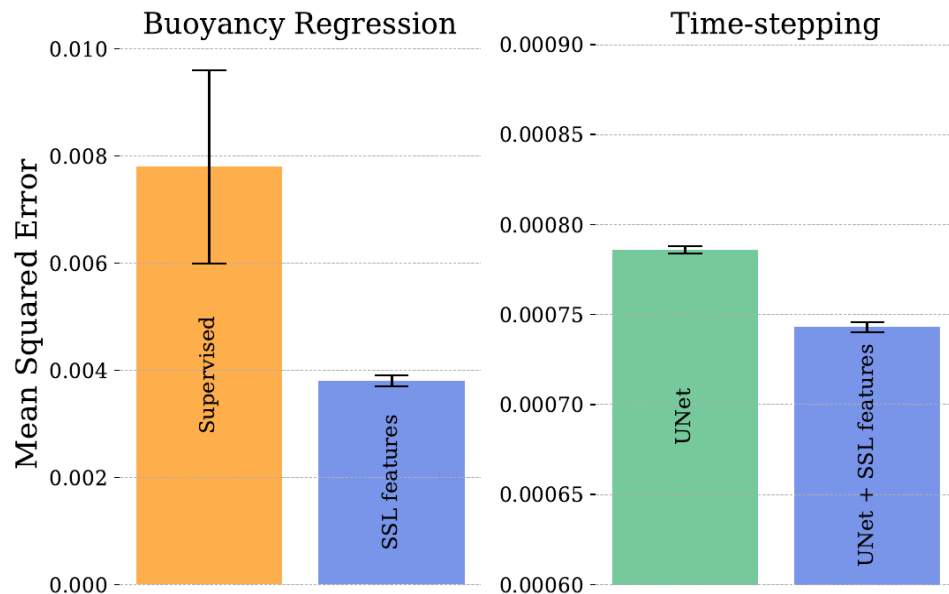
# SSL for Predicting Buoyancy in Navier-Stokes

The **incompressible Navier-Stokes** equation is given by

$$\mathbf{u}_t = -\mathbf{u} \cdot \nabla \mathbf{u} - \frac{1}{\rho} \nabla p + \nu \nabla^2 \mathbf{u} + \mathbf{f}, \quad \nabla \cdot \mathbf{u} = 0.$$

- 26k 2D trajectories, 56 frames (128x128) each [Gupta and Brandstetter, 2023].
- Task 1: regressing buoyancy  $\mathbf{f}$ .
- Task 2: Time-stepping, predict next frames given past frames.
- SSL features are effective and easy to use.

Downstream tasks for Navier-Stokes

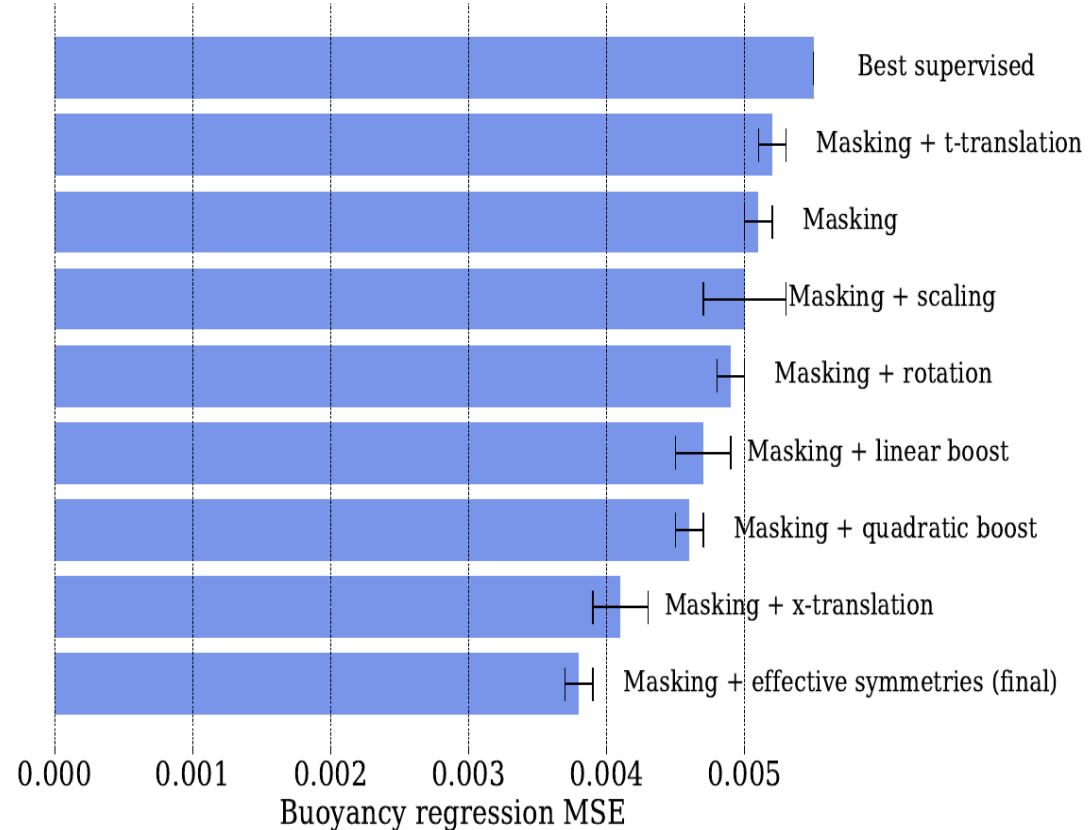




# SSL for Predicting Buoyancy in Navier-Stokes

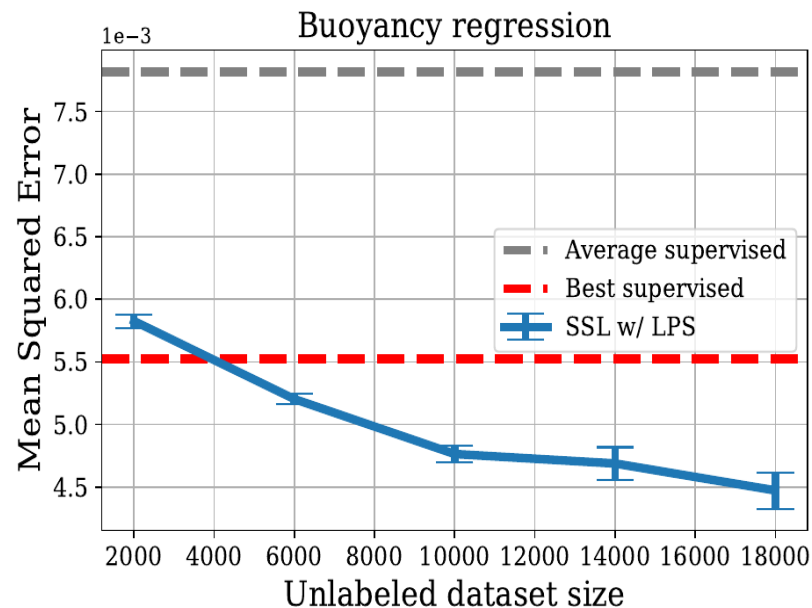
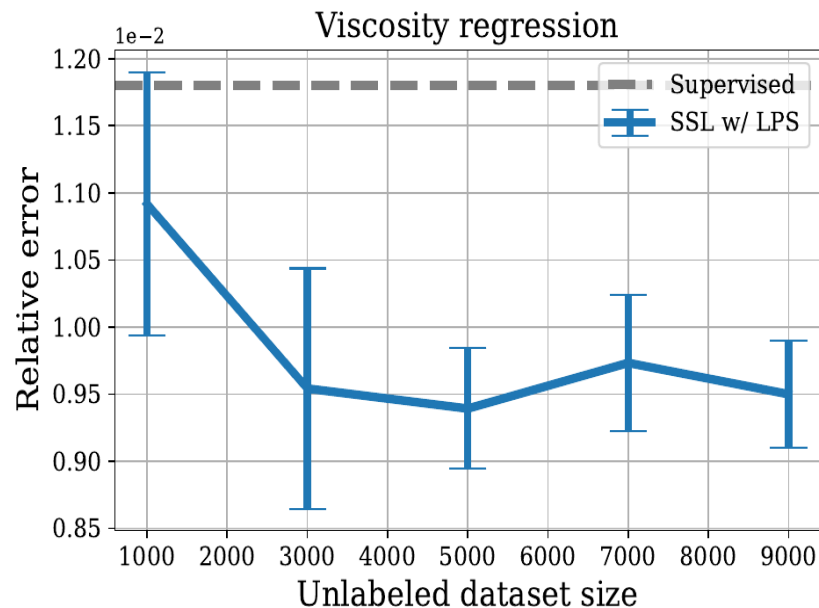
- Navier-Stokes: 8 Lie symmetrie groups, with varying strength.
- Intuition is not sufficient to select augmentations.
- Optimal mix is different from supervised [Brandstetter et al., 2022].
- Masking is necessary but not really sufficient.

Comparing SSL features resulting from different augmentations



# SSL pre-training gives better results than purely supervised

SSL vs. supervised: open question in vision [Sariyildiz et al., 2023, Oquab et al., 2023]. Here, big discrepancy.



Influence of dataset size on regression tasks. **(Left)** Kinematic regression on Burger's equation. **(Right)** Buoyancy regression on Navier-Stokes' equation.

# Problems to Solve

- ▶ **Mathematical Foundations of Energy-Based Learning**
  - ▶ The geometry of energy surfaces, scaling laws, bounds...
- ▶ **JEPA with regularized latent variables**
  - ▶ Learning and planning in non-deterministic environments
- ▶ **Planning algorithms in the presence of uncertainty**
  - ▶ Gradient-based methods and combinatorial search methods
- ▶ **Learning Cost Modules (Inverse RL)**
  - ▶ Energy-based approach: give low cost to observed trajectories
- ▶ **Planning with inaccurate world models**
  - ▶ Preventing bad plans in uncertain parts of the space
- ▶ **Exploration to adjust world models**
  - ▶ Intrinsic objectives for curiosity

# Things we are working on

- ▶ **Self-Supervised Learning from Video**

- ▶ Hierarchical Video-JEPA trained with SSL

- ▶ **LLMs that can reason & plan, driven by objectives**

- ▶ Dialog systems that plan in representation space and use AR-LLM to turn representations into text

- ▶ **Learning hierarchical planning**

- ▶ Training a multi-timescale H-JEPA on toy planning problems.

# Points

- ▶ **Computing power**
  - ▶ AR-LLM use a fixed amount of computation per token
  - ▶ Objective-Driven AI is Turing complete (inference == optimization)
- ▶ **We are still missing essential concepts to reach human-level AI**
  - ▶ Scaling up auto-regressive LLMs will **not** take us there
  - ▶ We need machines to learn how the world works
- ▶ **Learning World Models with Self-Supervised Learning and JEPA**
  - ▶ Non-generative architecture, predicts in representation space
- ▶ **Objective-Driven AI Architectures**
  - ▶ Can plan their answers
  - ▶ Must satisfy objectives: are **steerable & controllable**
  - ▶ Guardrail objectives can make them **safe** by construction.

# Future Universal Virtual Assistant

- ▶ All of our interactions with the digital world will be mediated by AI assistants.
- ▶ They will constitute a **repository of all human knowledge and culture**
- ▶ They will constitute a shared infrastructure  
Like the Internet today.
- ▶ **These AI platform MUST be open source**
- ▶ Otherwise, our culture will be controlled by a few companies on the West Coast of the US or in China.
- ▶ Training them will have to be crowd-sourced
- ▶ **Open source AI platforms are necessary**





# What does this vision mean for policy?

- ▶ AI systems will become a common platform
- ▶ **The platforms (foundation models) will be open source**
  - ▶ They will condense all of human knowledge
  - ▶ Guardrail objectives will be shared for safety
- ▶ **Training and fine-tuning will be crowd-sourced**
  - ▶ Linguistic, cultural, and interest groups will fine-tune base models to cater to their interests.
- ▶ Proprietary systems for vertical applications will be built on top
- ▶ When everyone has an AI assistant, we will need
  - ▶ Massive computing infrastructure for inference: efficient inference chips.
- ▶ **OPEN SOURCE AI MUST NOT BE REGULATED OUT OF EXISTENCE**
  - ▶ AI Alliance: Meta, IBM, Intel, Sony, academia, startups....

# Questions

- ▶ **How long is it going to take to reach human-level AI?**
  - ▶ Years to decades. Many problems to solve on the way.
  - ▶ Before we get to HLAI, we will get to cat-level AI, dog-level AI,...
- ▶ **What is AGI?**
  - ▶ There is no such thing. Intelligence is highly multidimensional
  - ▶ Intelligence is a collection of skills + ability to learn new skills quickly
  - ▶ Even humans can only accomplish a tiny subset of all tasks
- ▶ **Will machines surpass human intelligence**
  - ▶ Yes, they already do in some narrow domains.
  - ▶ There is no question that machine will eventually surpass human intelligence in all domains where humans are intelligent (and more)

# Questions

- ▶ **Are there short-term risks associated with powerful AI?**
  - ▶ Yes, as with every technology.
  - ▶ Disinformation, propaganda, hate, spam,...: **AI is the solution!**
  - ▶ Concentration of information sources
  - ▶ All those risks can be mitigated
- ▶ **Are there long-term risks with (super-)human-level AI?**
  - ▶ Robots will not take over the world! a mistaken projection of human nature on machines
  - ▶ Intelligence is not correlated with a desire to dominate, even in humans
  - ▶ Objective-Driven AI systems will be made subservient to humans
  - ▶ AI will not be a “species” competing with us.
  - ▶ We will design its goals and guardrails.

# Questions

## ► **How to solve the alignment problem?**

- Through trial and error and testing in sand-boxed systems
- We are very familiar with designing objectives for human and superhuman entities. It's called law making.
- What if bad people get their hand on on powerful AI?  
Their Evil AI will be taken down by the Good Guys' AI police.

## ► **What are the benefits of human-level AI?**

- AI will **amplify human intelligence**, progress will accelerate
- As if everyone had a super-smart staff working for them
- The effect on society may be as profound as the printing press
- **By amplifying human intelligence, AI will bring a new era of enlightenment, a new renaissance for humanity.**





# Thank you!



NEW YORK UNIVERSITY



Meta AI

