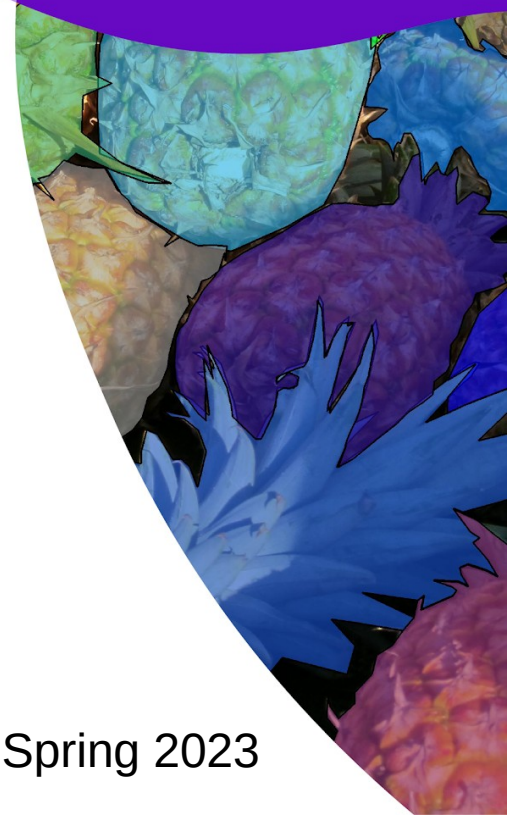
NEW YORK UNIVERSITY

A Path Towards Autonomous AI



Yann LeCun
NYU - Courant Institute & Center for Data Science
Facebook AI Research
http://yann.lecun.com

Deep Learning, NYU, Spring 2023

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.**
- **SL/RL-trained ML systems:**
  - are specialized and brittle
  - make “stupid” mistakes
  - do not reason nor plan
- **Animals and humans:**
  - Can learn new tasks **very** quickly.
  - Understand how the world works
  - Can reason an plan
- **Humans and animals have common sense, current machines don’t**

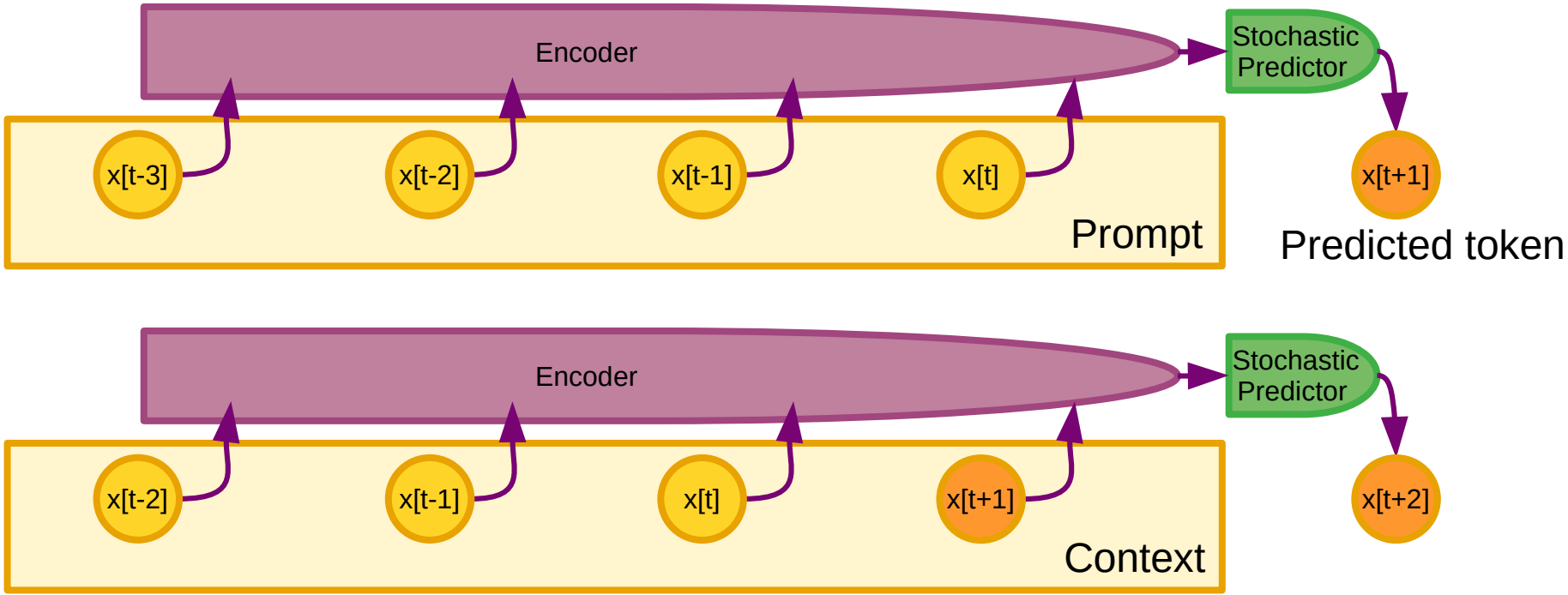
Machine Learning sucks! (plain ML/DL, at least)

- **Machine Learning systems (most of them anyway)**
  - Have a constant number of computational steps between input and output.
  - Do not reason.
  - Cannot plan.

- **Humans and some 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

Auto-Regressive Generative Architectures

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



Auto-Regressive Large Language Models (AR-LLMs)

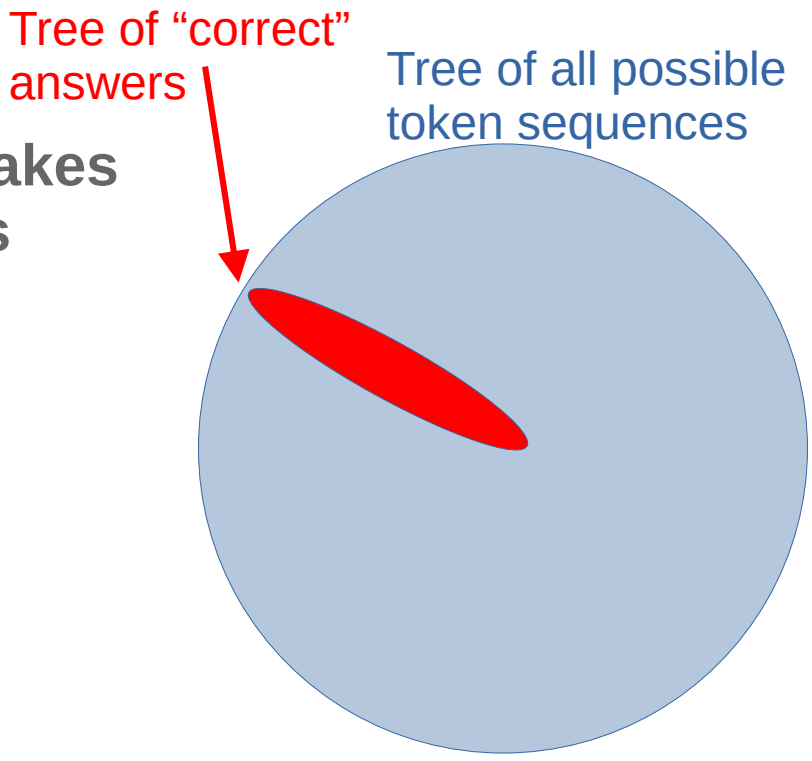
- **Outputs one text token after another**
- **Tokens may represent words or subwords**
- **Encoder/predictor is a transformer architecture**
  - With billions of parameters: typically from 1B to 500B
  - Training data: 1 to 2 trillion tokens
- **LLMs for dialog/text generation:**
  - BlenderBot, Galactica, 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

- **Pure Auto-Regressive LLMs will disappear.**
- **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. Diverge exponentially
  - Are not steerable/controllable. Behavior adjustment requires retraining.

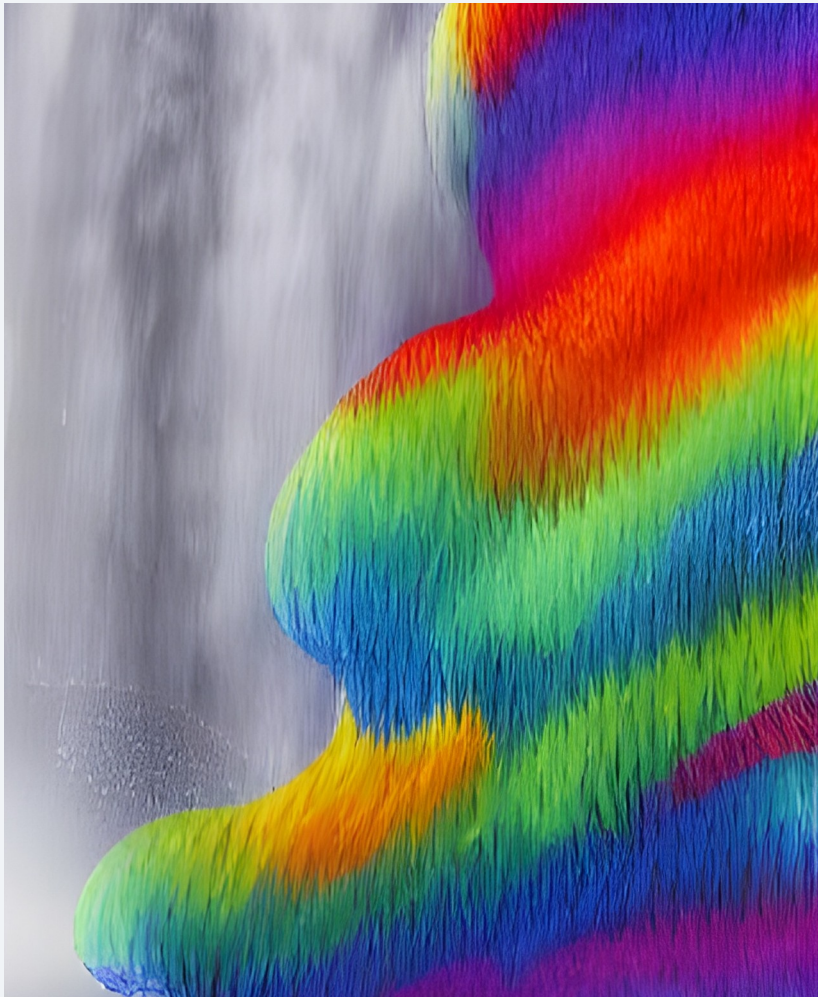
- **Humans and many animals**
  - Understand how the world works.
  - Can predict the consequences of their actions.
  - Can perform chains of reasoning with an unlimited number of steps.
  - Can plan complex tasks by decomposing it into sequences of subtasks

NEW YORK UNIVERSITY

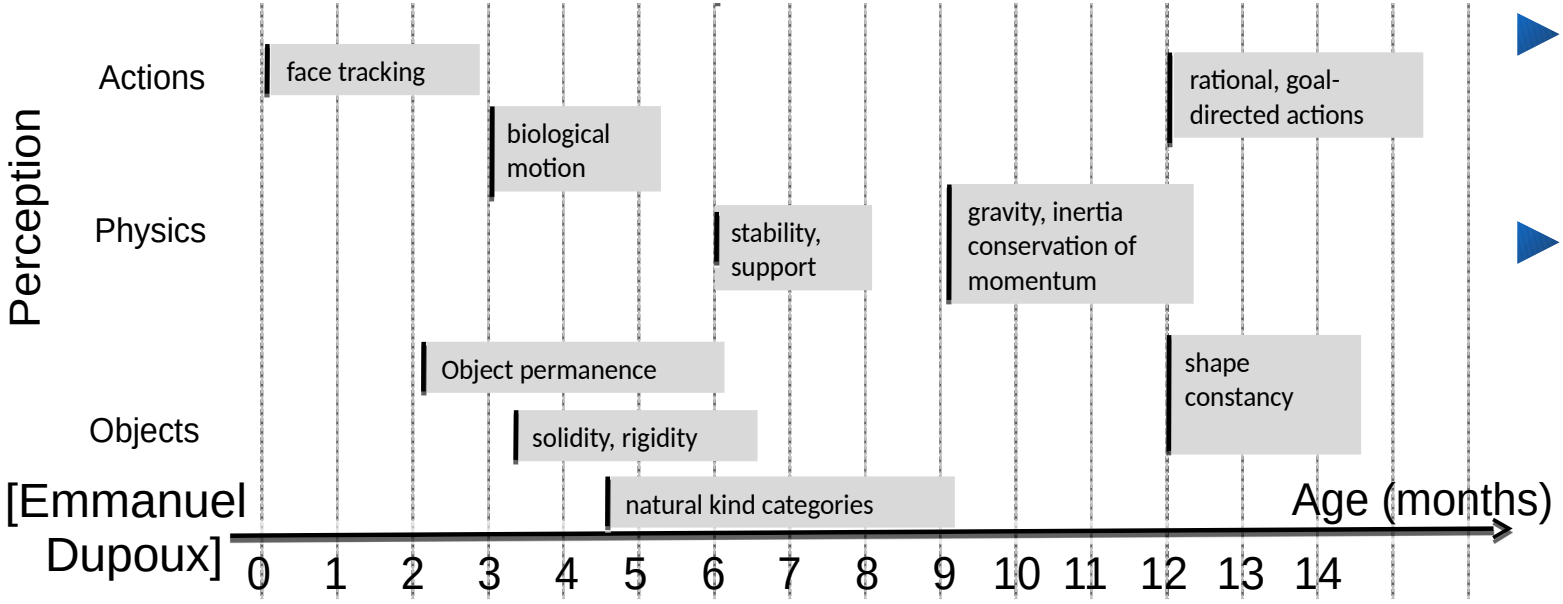
Meta AI

How do humans and animals learn so quickly?

Not supervised.
Not Reinforced.
At least not much.



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?

How do Human and Animal Babies Learn?

- **How do they learn how the world works?**
- **Largely by observation, with remarkably little interaction (initially).**
- **They accumulate enormous amounts of background knowledge**
  - About the structure of the world, like intuitive physics.
- **Perhaps common sense emerges from this knowledge?**

Photos courtesy of Emmanuel Dupoux

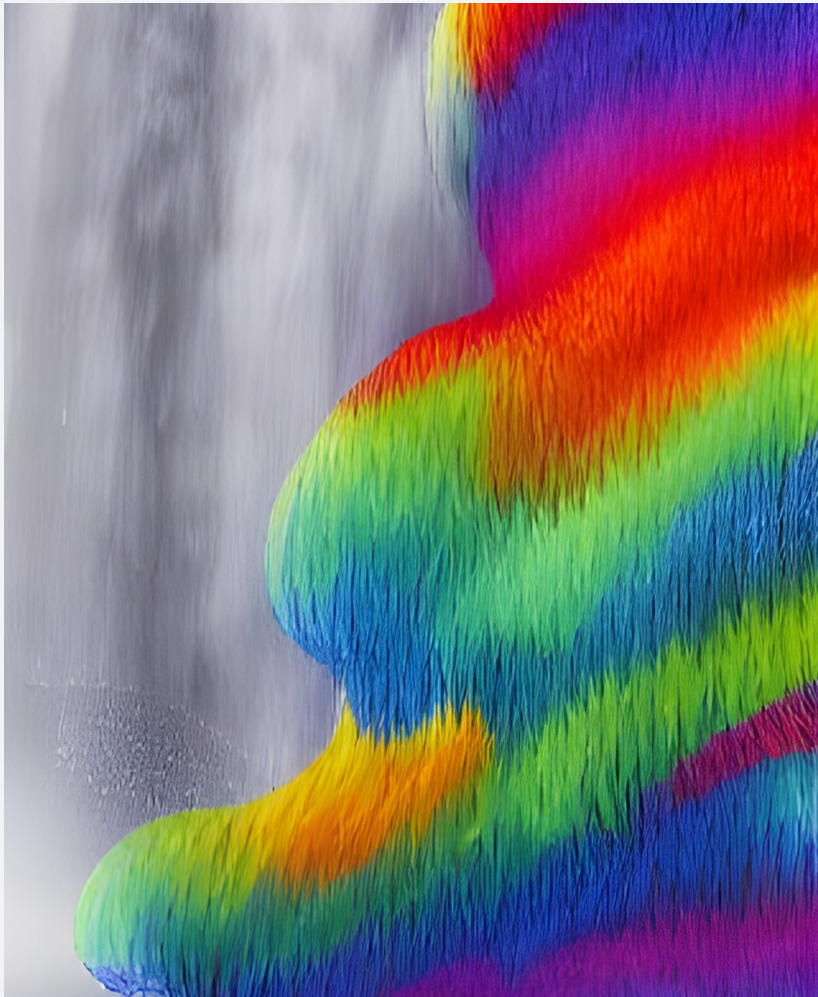
Three challenges for AI & Machine Learning

- **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

NEW YORK UNIVERSITY

Meta AI

A Cognitive Architecture capable of reasoning & planning

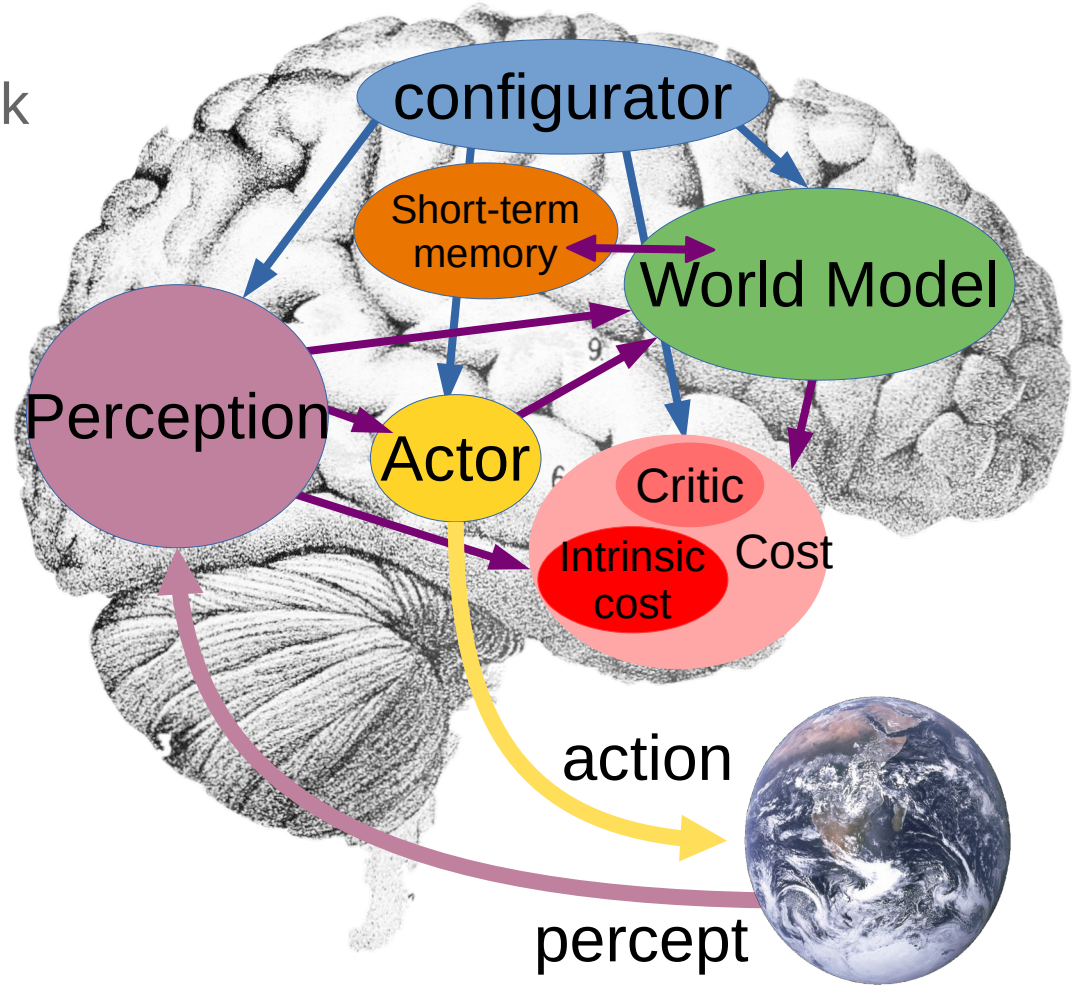


Position paper:
“A path towards autonomous machine intelligence”
https://openreview.net/forum?id=BZ5a1r-kVsf

Longer talk: search “LeCun Berkeley” on YouTube

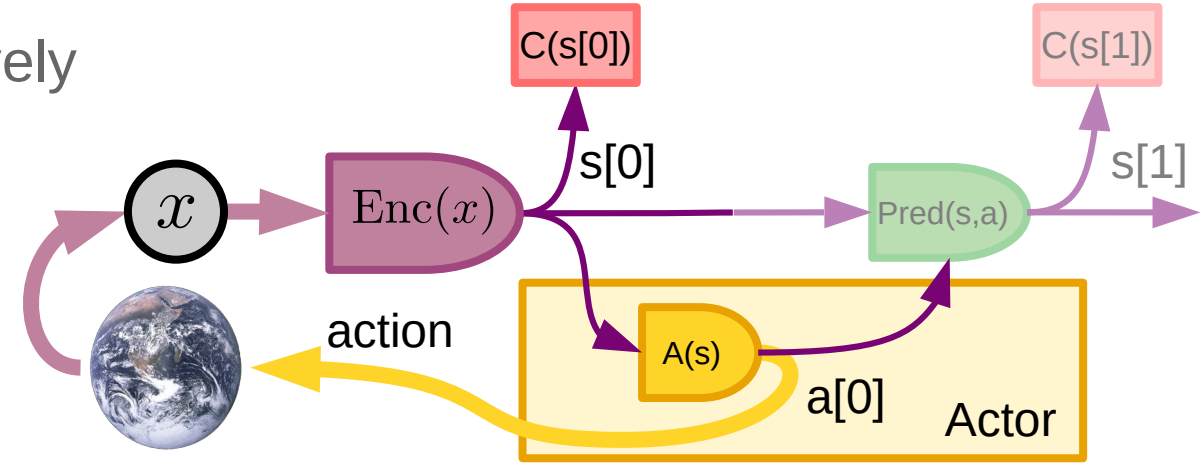
Modular Architecture for Autonomous AI

- **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



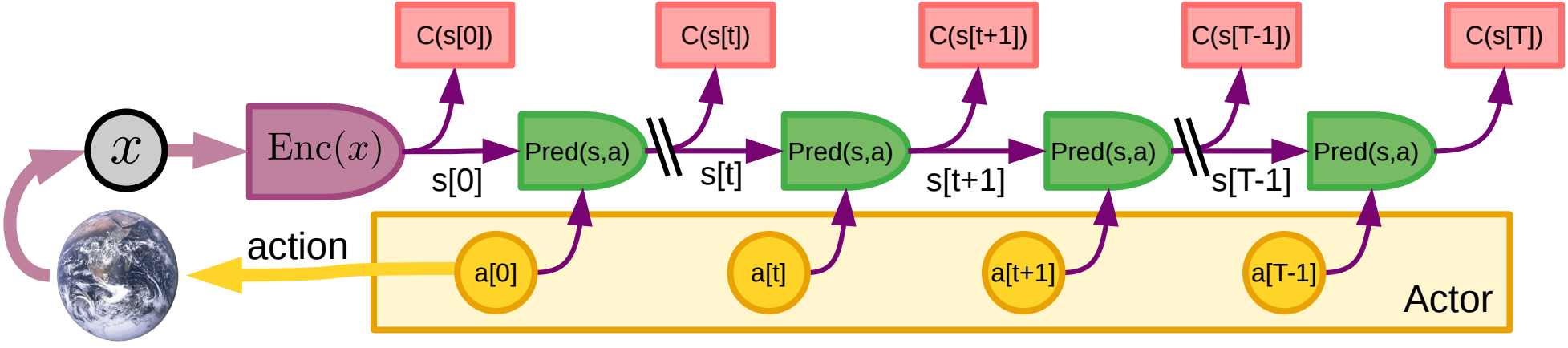
Mode-1 Perception-Action Cycle

- **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

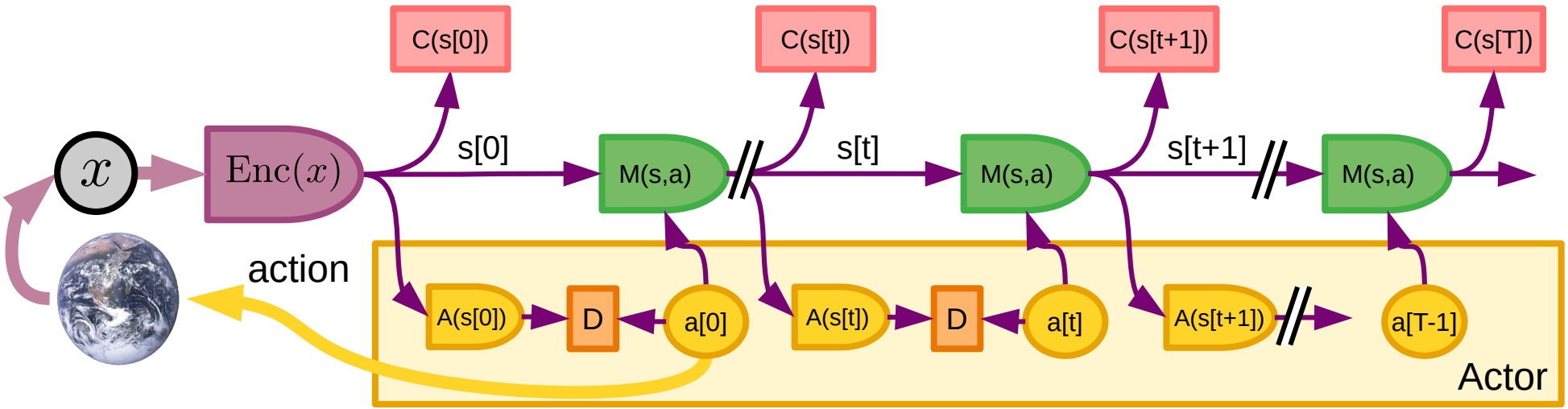
- **Akin to classical Model-Predictive Control (MPC)**
- **Actor proposes an ation sequence**
- **World Model predicts outcome**
- **Actor optimizes action sequence to minimize cost**
  - e.g. using gradient descent, dynamic programming, MC tree search…
- **Actor sends first action(s) to effectors**



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

Compiling Mode-2 into Mode-1

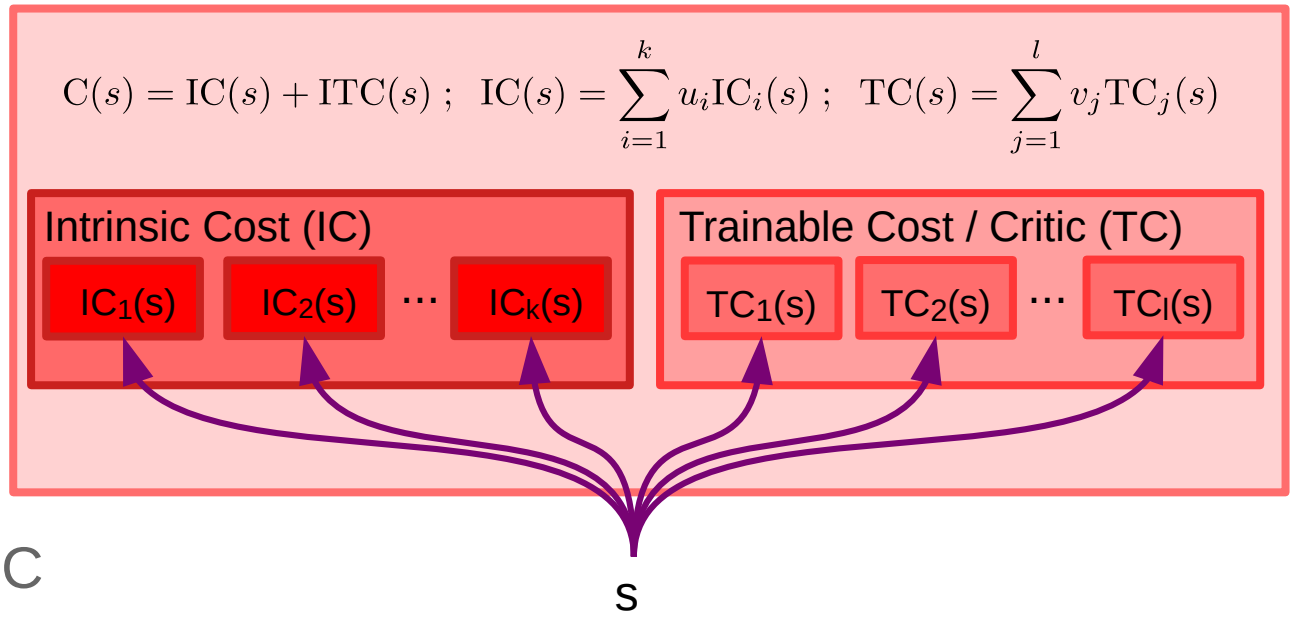
- Akin to Amortized Inference
- System performs Mode-2 cycle to get optimal action sequence.
- Optimal actions used as targets to train the policy module A(s)
- Policy module can be used for Mode-1 or to initialize Mode-2.



[Henaff et al. ICLR 2019] [Schrittwieser et al. MuZero 2020]

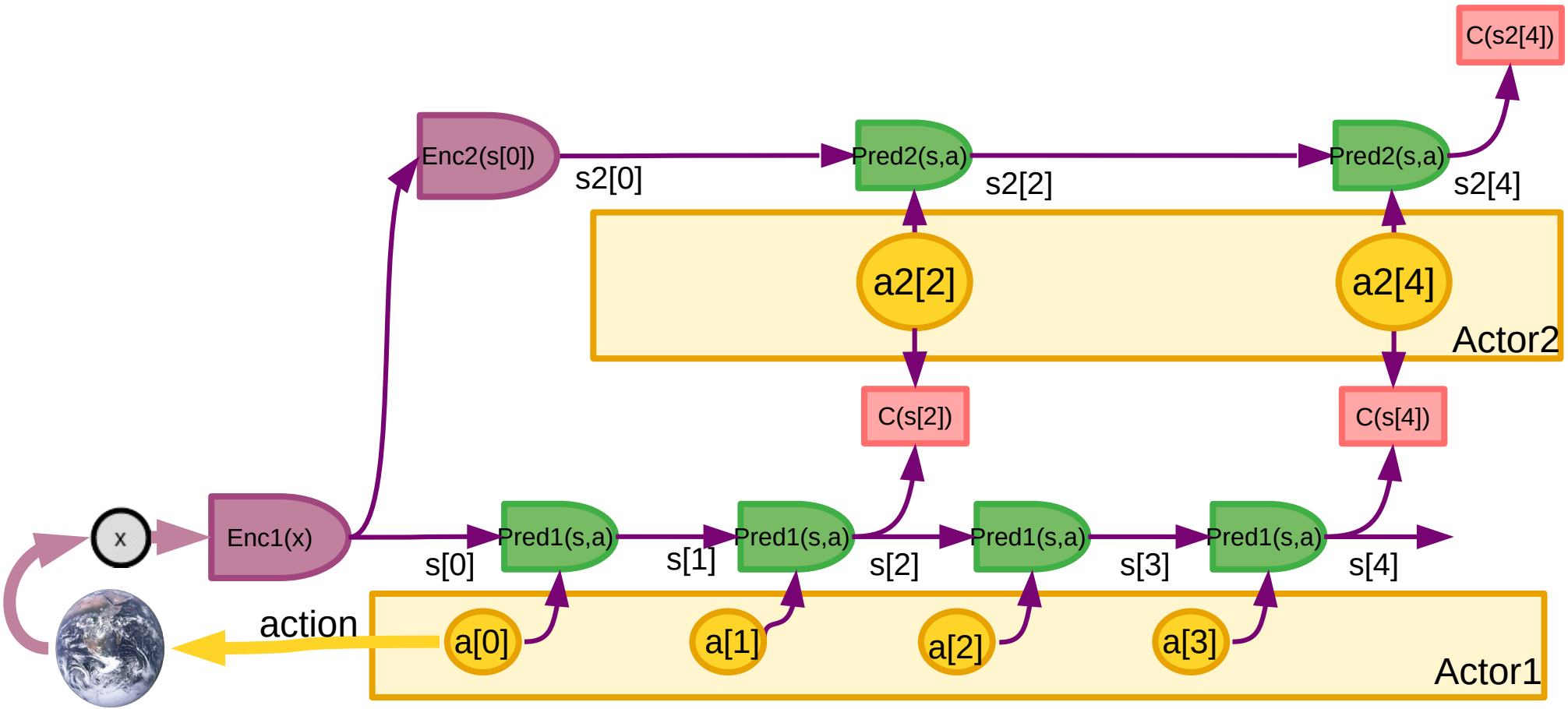
Cost Module

- **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**



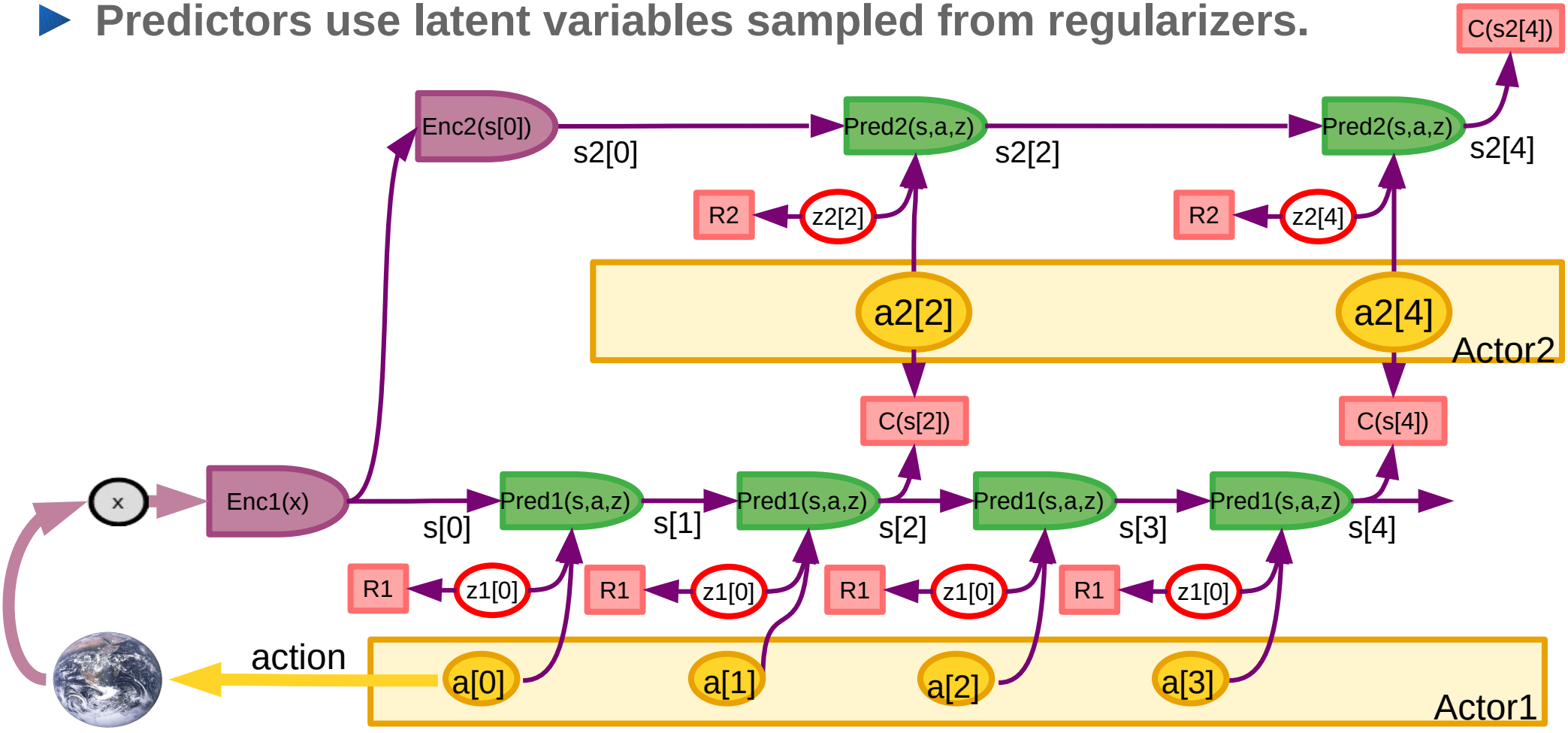
Hierarchical Planning

- **High level sets objectives (costs) for the lower level(s) to satisfy.**



Hierarchical Planning with Uncertainty

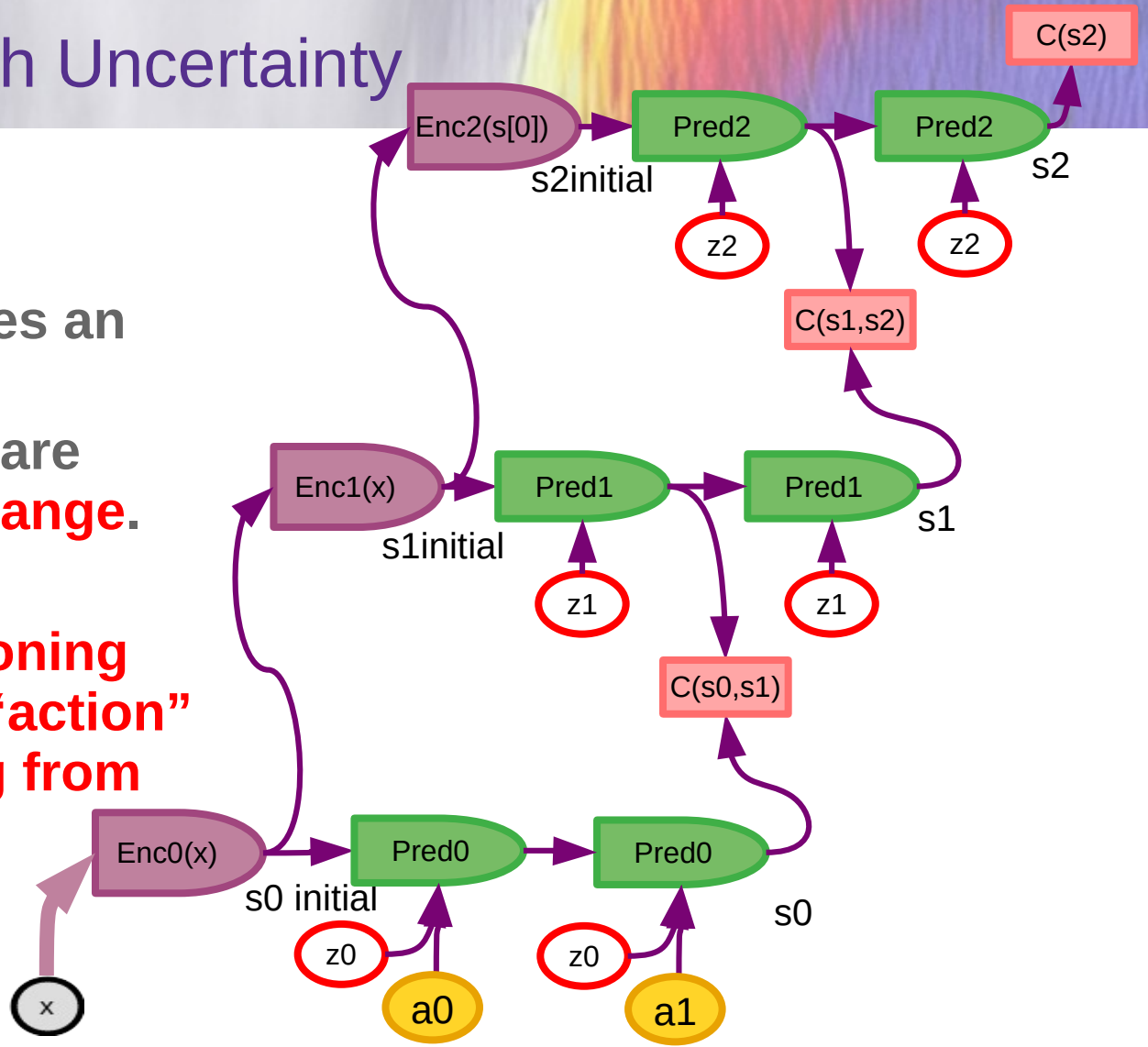
- **Predictors use latent variables sampled from regularizers.**



Hierarchical Planning with Uncertainty

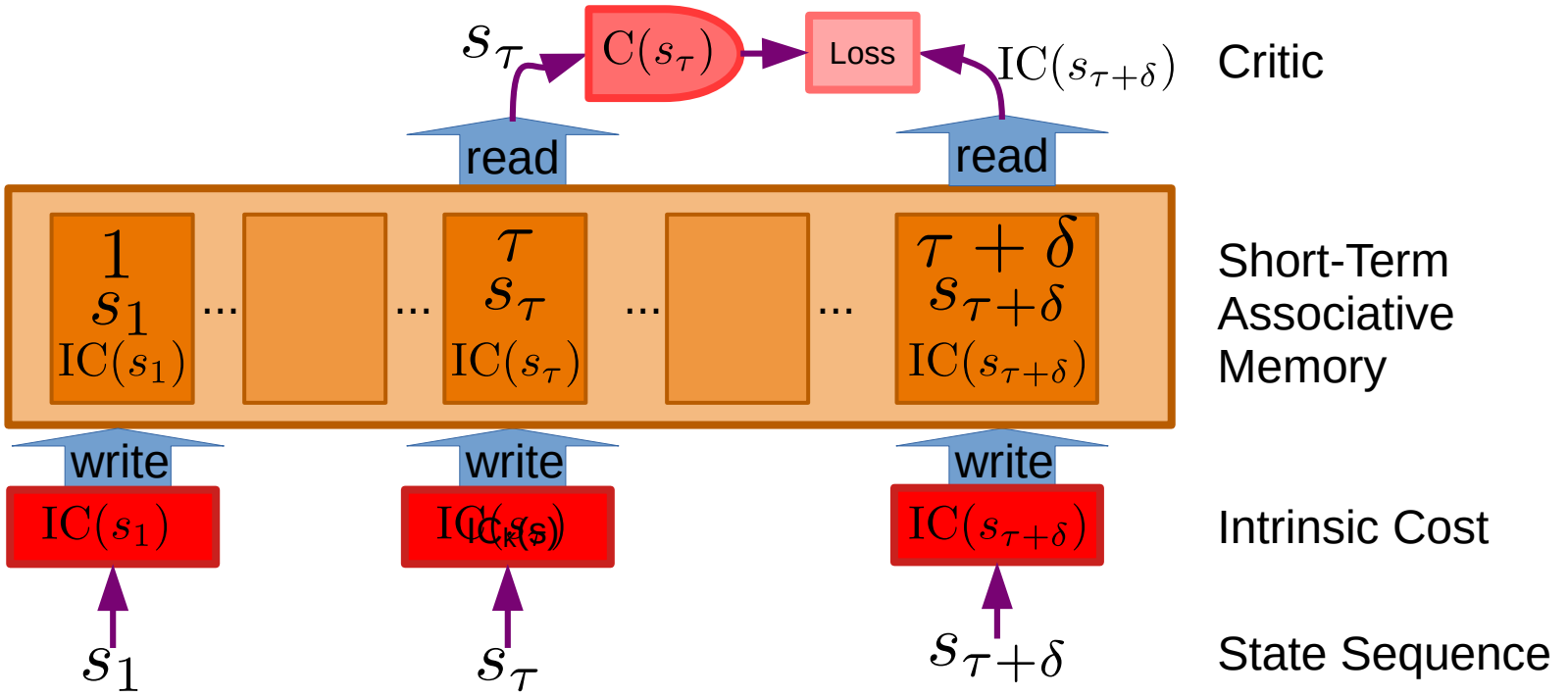
- Hierarchical world model
- Hierarchical planning
- An **action** at level k specifies an **objective** for level k-1
- Prediction in higher levels are more **abstract** and **longer-range**.

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



Training the Critic

- **Critic is trained to predict future values of the intrinsic cost from the current state**
  - Uses the short term memory to produce training pairs.

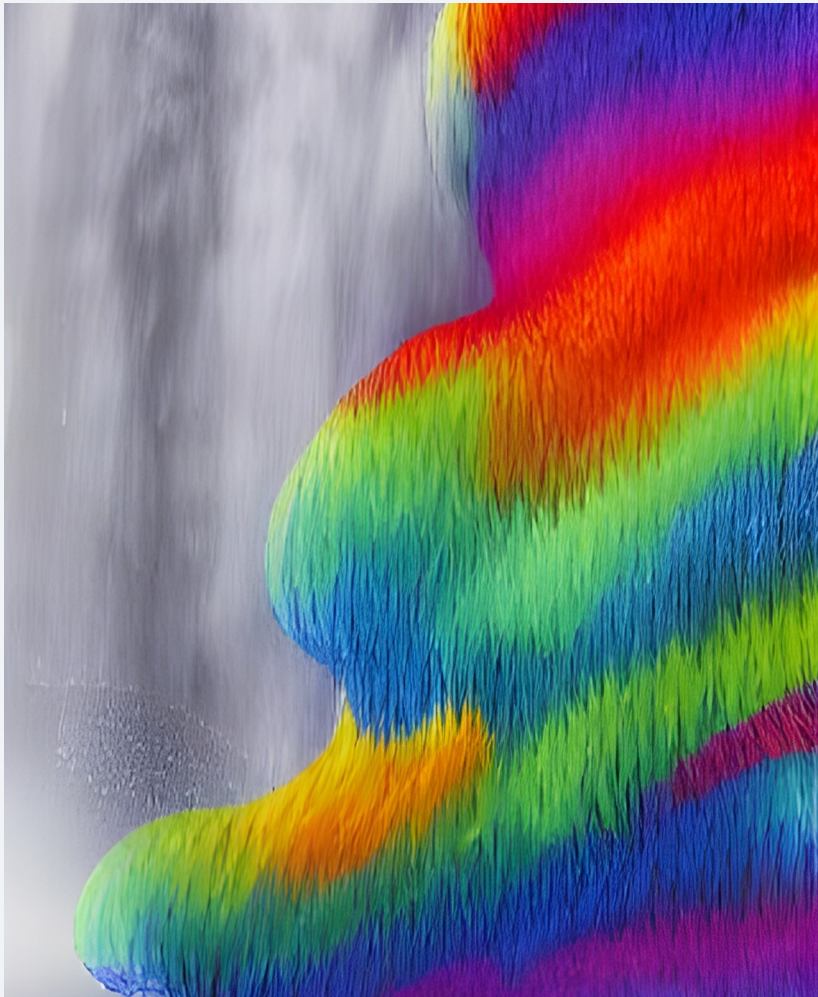


NEW YORK UNIVERSITY

Meta AI

Building & Training the World Model

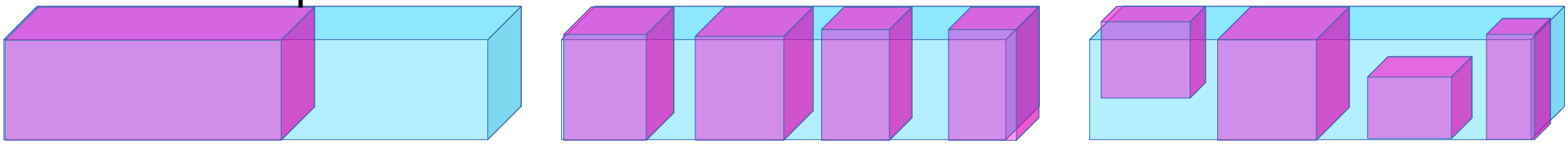
Energy-Based Models
Joint-Embedding Architecture



Self-Supervised Learning = Learning to Fill in the Blanks

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

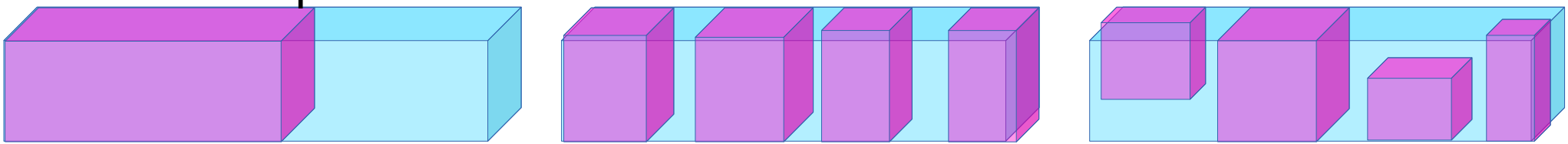
time or space →



Self-Supervised Learning = Learning to Fill in the Blanks

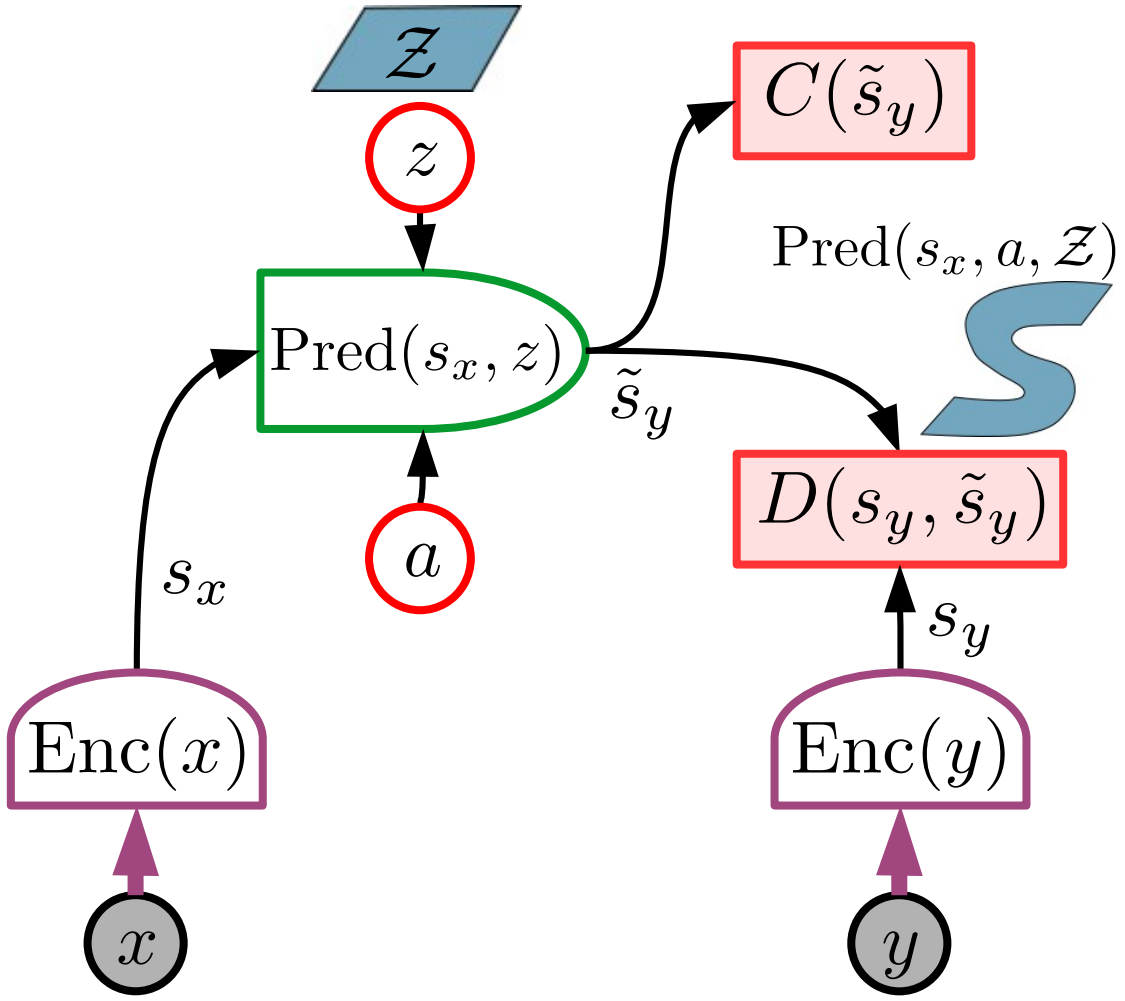
- **Reconstruct the input or Predict missing parts of the input.**

time or space →



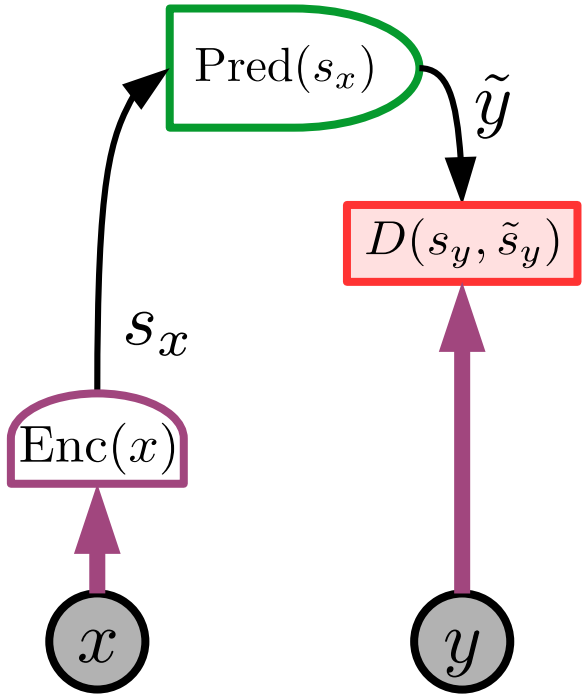
Architecture for the world model: JEPA

- **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 S_y from a representation of the past and present S_x

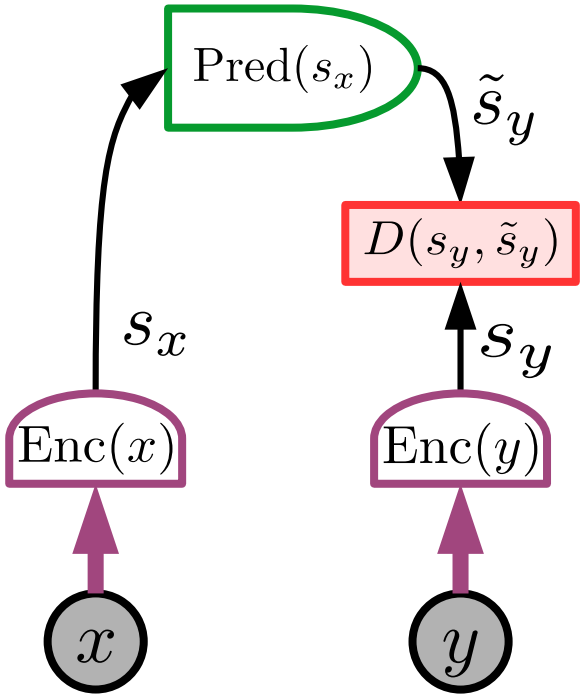


Architectures: Generative vs Joint Embedding

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



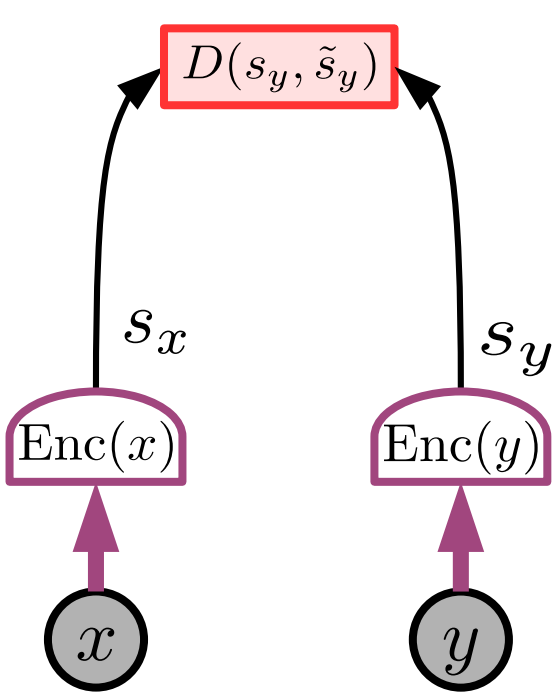
a) Generative Architecture
Examples: VAE, MAE...



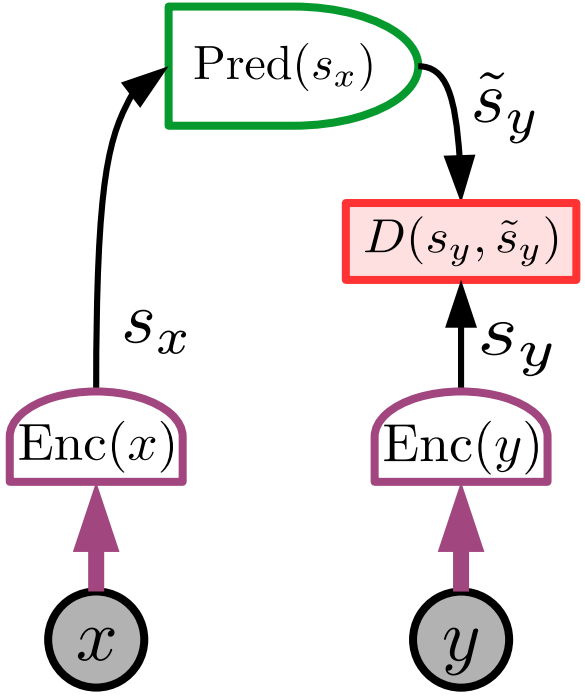
b) Joint Embedding Architecture

Joint Embedding Architectures

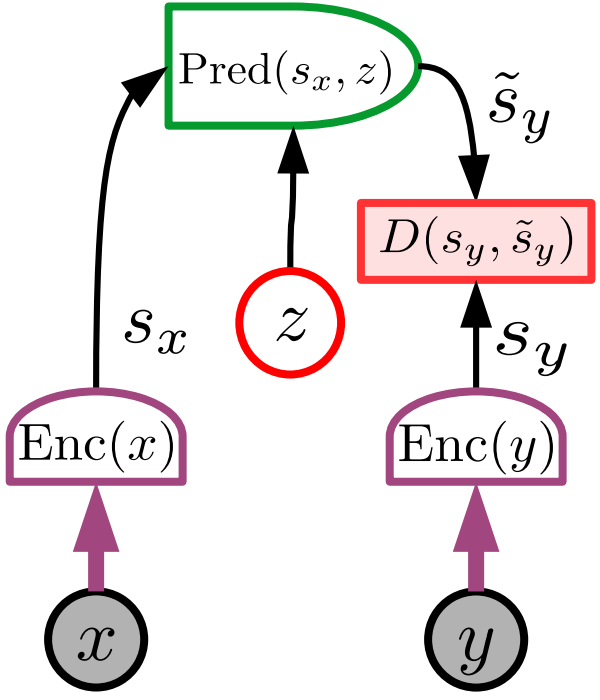
- Computes abstract representations for x and y
- Tries to make them equal or predictable from each other.



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



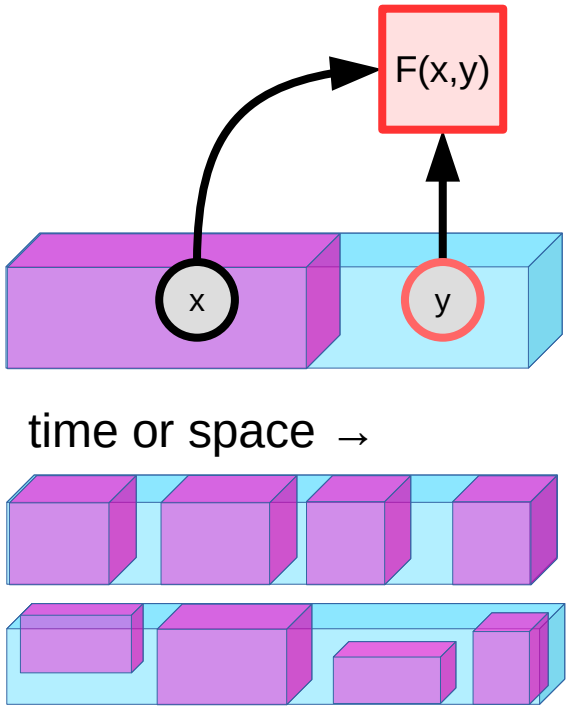
b) Deterministic Joint Embedding Predictive Architecture (DJEPA)
Examples: BYOL, VICRegL, I-JEPA

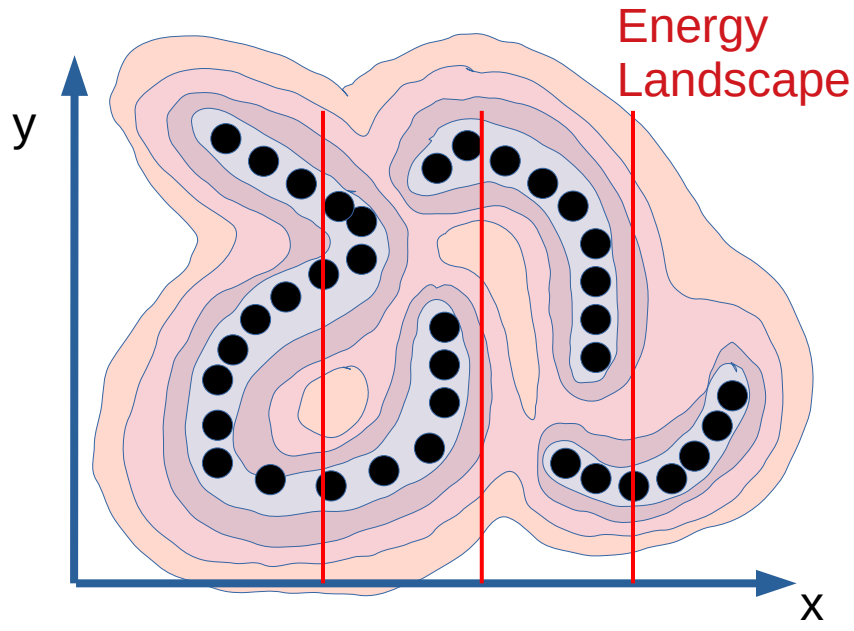


c) Joint Embedding Predictive Architecture (JEPA)
Examples: VICRegL, Equivariant VICReg I-JEPA…..

Energy-Based Models: Implicit function

- Gives low energy for compatible pairs of x and y
- Gives higher energy for incompatible pairs



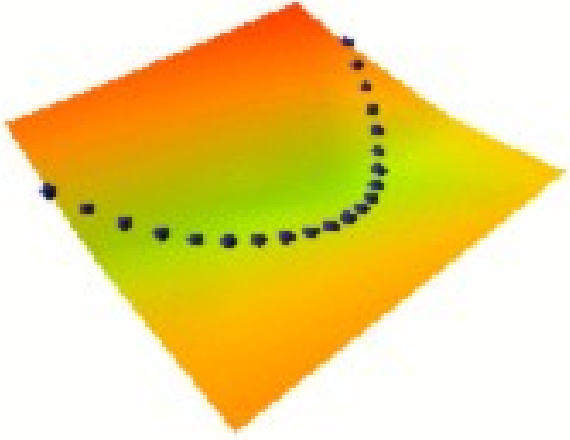


$$\check{y} = \text{argmin}_y F(x, y)$$

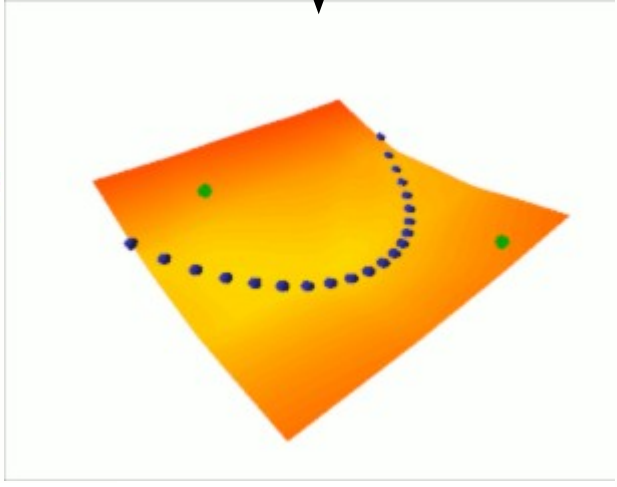
Training Energy-Based Models: Collapse Prevention

- **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.

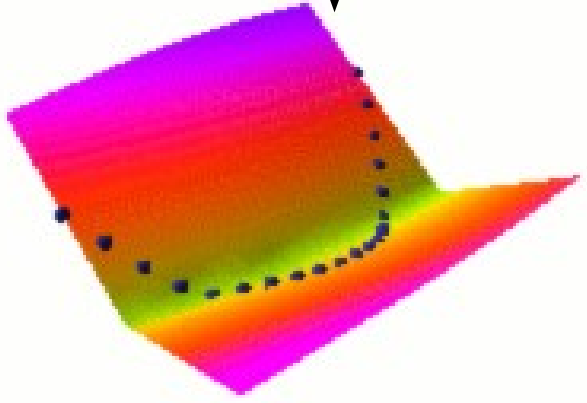
Collapse!



Contrastive Method

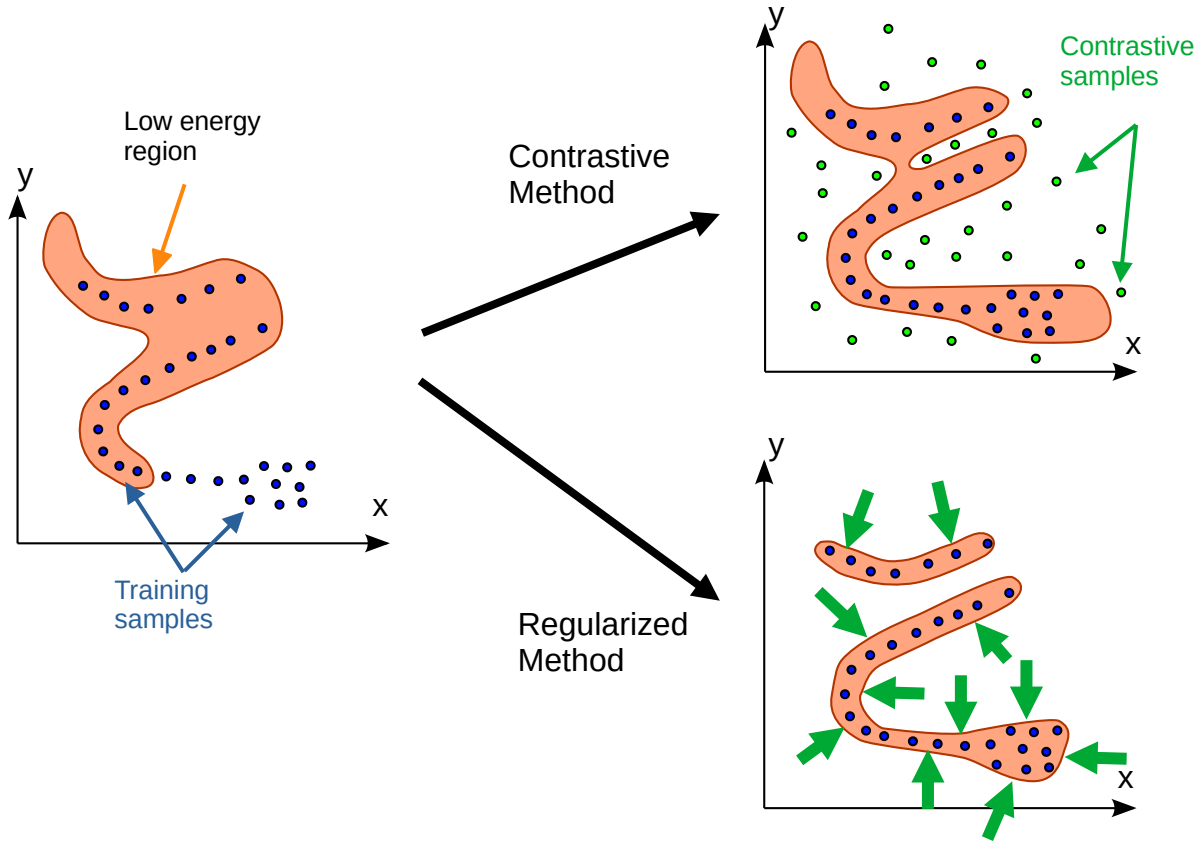


Regularized Methods



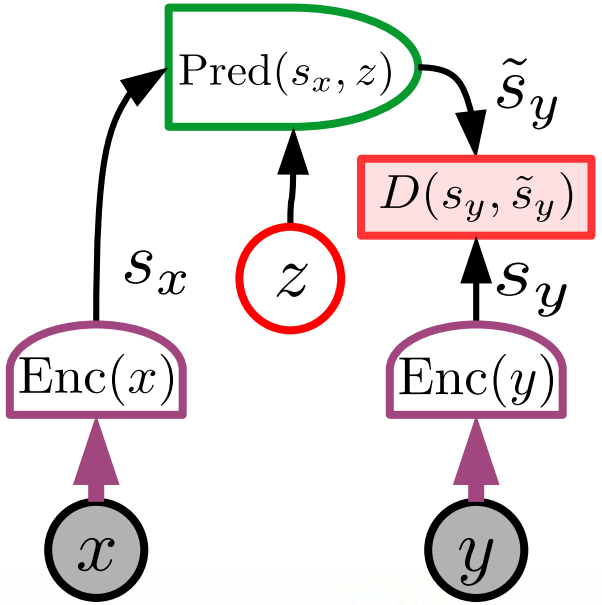
EBM Training: two categories of methods

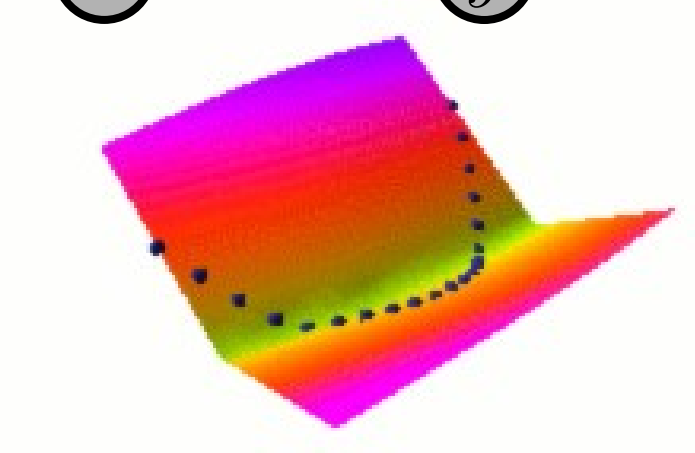
- **Contrastive methods**
  - Push down on energy of training samples
  - Pull up on energy of suitably-generated contrastive samples
  - Scales very badly with dimension
- **Regularized Methods**
  - Regularizer minimizes the volume of space that can take low energy



Recommendations:

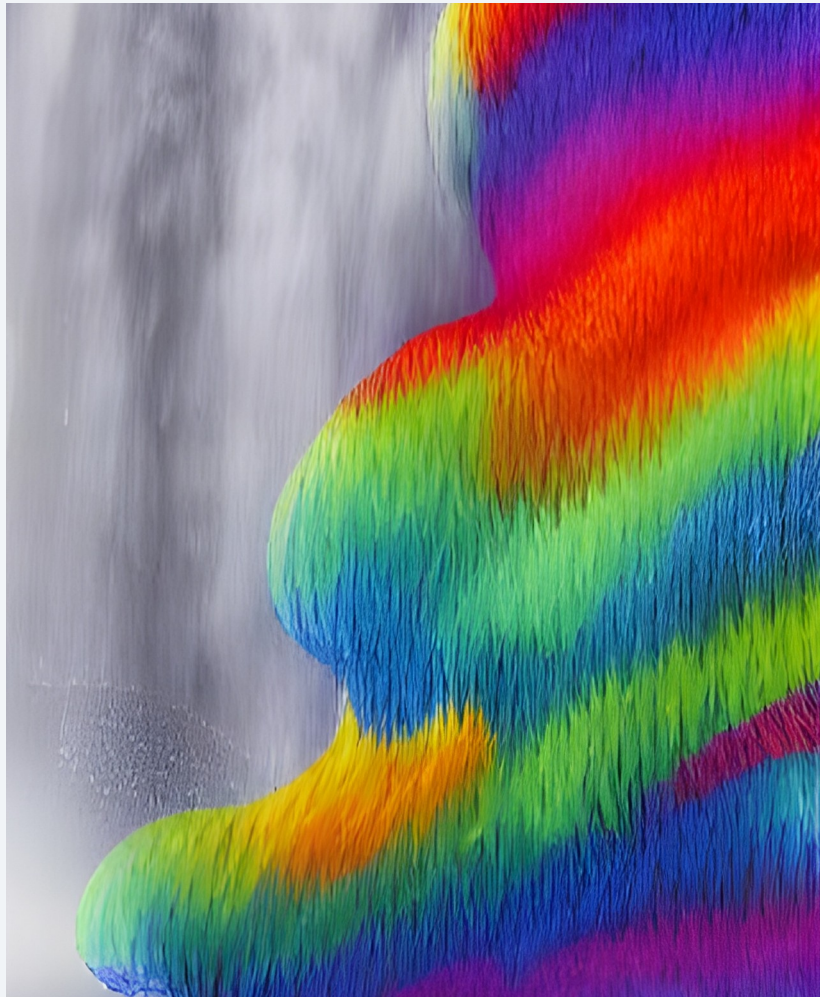
- **Abandon generative models**
  - in favor joint-embedding architectures
- **Abandon probabilistic model**
  - in favor of energy-based models
- **Abandon contrastive methods**
  - in favor of regularized methods
- **Abandon Reinforcement Learning**
  - In favor of model-predictive control
  - Use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





NEW YORK UNIVERSITY Meta AI

Regularized Methods for joint embedding architectures



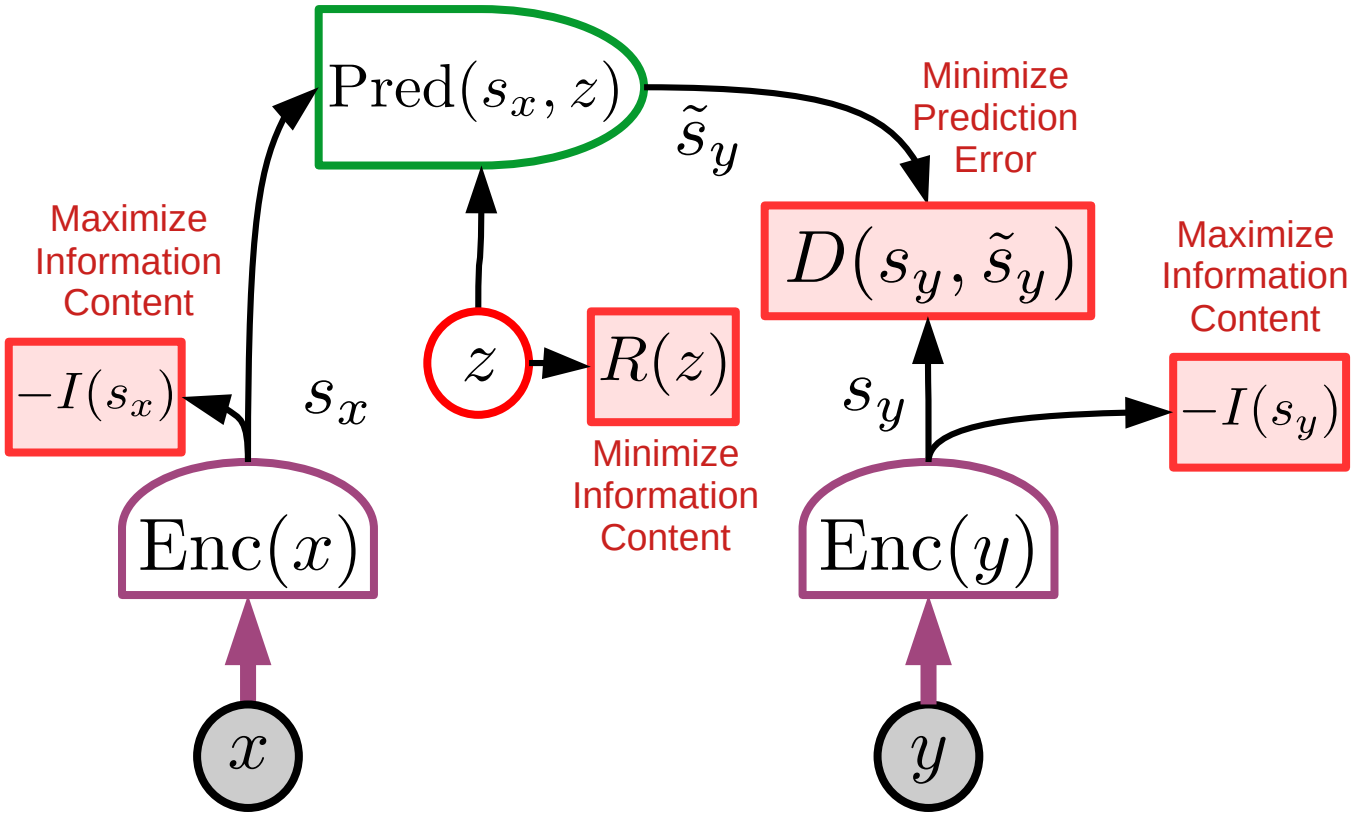
This is the cool stuff!

Push down on the energy of compatible sample pairs

Maximize the information capacity of representations

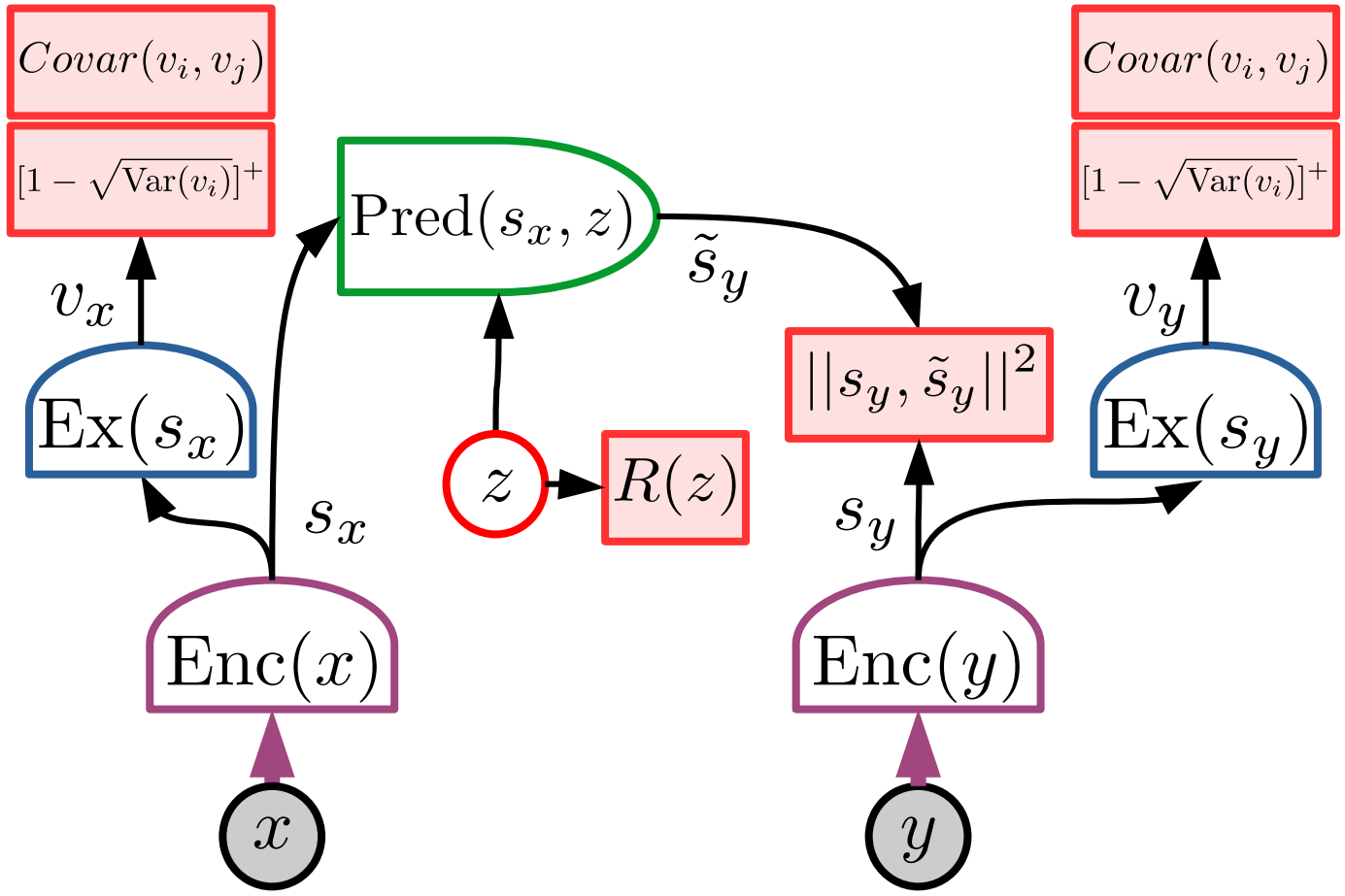
Training a JEPA non contrastively

- **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

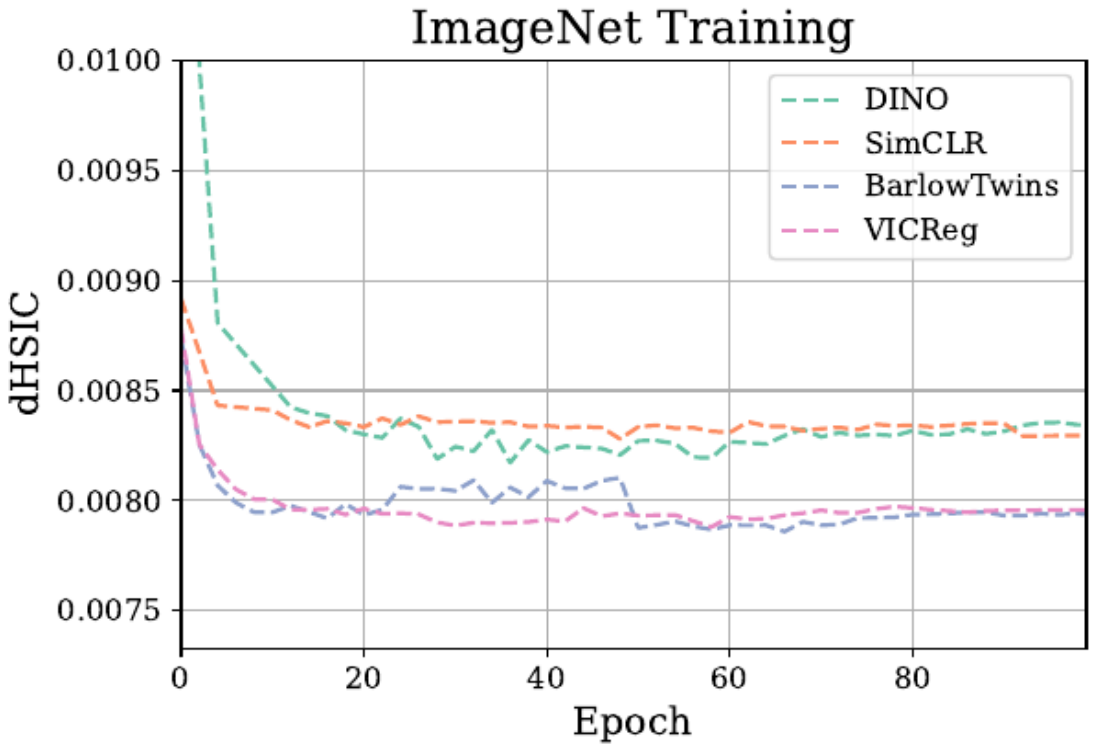
- **Variance:**
  - Maintains variance of components of representations
- **Covariance:**
  - Decorrelates components of covariance matrix of representations
- **Invariance:**
  - Minimizes prediction error.

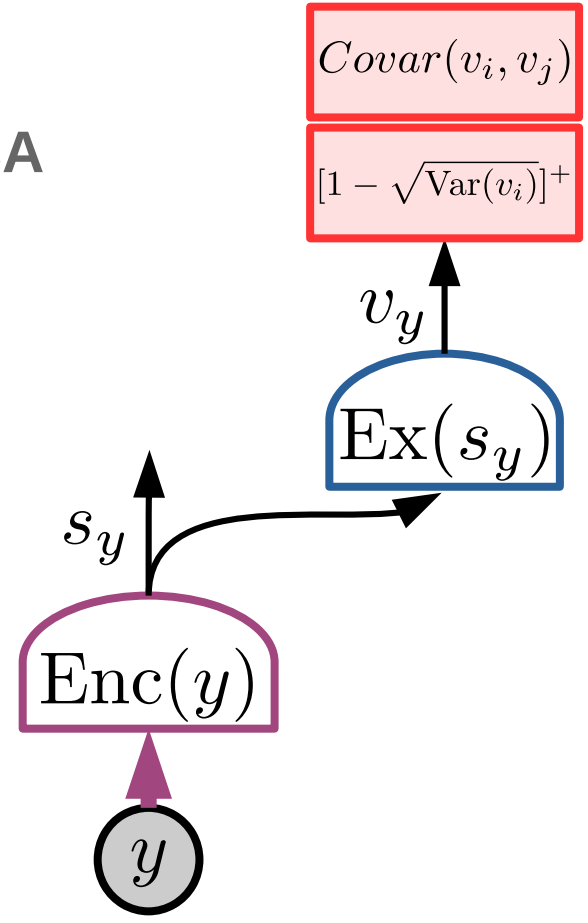


Barlow Twins [Zbontar et al. ArXiv:2103.03230], VICReg [Bardes, Ponce, LeCun arXiv:2105.04906, ICLR 2022], VICRegL [Bardes et al. NeurIPS 2022], MCR2 [Yu et al. NeurIPS 2020][Ma, Tsao, Shum, 2022]

VICReg: expander makes variables pairwise independent

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





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 He et al. (2020)	60.6	-	-	-	-	-
PIRL Misra & Maaten (2020)	63.6	-	-	-	57.2	83.8
CPC v2 Hénaff et al. (2019)	63.8	-	-	-	-	-
CMC Tian et al. (2019)	66.2	-	-	-	-	-
SimCLR Chen et al. (2020a)	69.3	89.0	48.3	65.6	75.5	87.8
MoCo v2 Chen et al. (2020c)	71.1	-	-	-	-	-
SimSiam Chen & He (2020)	71.3	-	-	-	-	-
SwAV Caron et al. (2020)	71.8	-	-	-	-	-
InfoMin Aug Tian et al. (2020)	73.0	91.1	-	-	-	-
OBoW Gidaris et al. (2021)	73.8	-	-	-	82.9	90.7
BYOL Grill et al. (2020)	74.3	91.6	53.2	68.8	78.4	89.0
SwAV (w/ multi-crop) Caron et al. (2020)	75.3	-	53.9	70.2	78.5	89.9
Barlow Twins Zbontar et al. (2021)	73.2	91.0	55.0	69.7	79.2	89.3
VICReg (ours)	73.2	91.1	54.8	69.5	79.4	89.5

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 He et al. (2020)	46.9	79.8	31.5	-	-	-
PIRL Misra & Maaten (2020)	49.8	81.1	34.1	-	-	-
SimCLR Chen et al. (2020a)	52.5	85.5	37.2	-	-	-
MoCo v2 Chen et al. (2020c)	51.8	86.4	38.6	82.5	39.8	36.1
SimSiam Chen & He (2020)	-	-	-	82.4	-	-
BYOL Grill et al. (2020)	54.0	86.6	47.6	-	40.4†	37.0†
SwAV (m-c) Caron et al. (2020)	56.7	88.9	48.6	82.6	41.6	37.8
OBoW Gidaris et al. (2021)	56.8	89.3	-	82.9	-	-
Barlow Twins Grill et al. (2020)	54.1	86.2	46.5	82.6	40.0†	36.7†
VICReg (ours)	54.3	86.6	47.0	82.4	39.4	36.4

VICReg: no need for normalization, momentum encoder, predictor...

Table 3: **Effect of incorporating variance and covariance regularization in different methods.** Top-1 ImageNet accuracy with the linear evaluation protocol after 100 pretraining epochs. For all methods, pretraining follows the architecture, the optimization and the data augmentation protocol of the original method using our reimplementation. ME: Momentum Encoder. SG: stop-gradient. PR: predictor. BN: Batch normalization layers after input and inner linear layers in the expander. No Reg: No additional regularization. Var Reg: Variance regularization. Var/Cov Reg: Variance and Covariance regularization. Unmodified original setups are marked by a †.

Method	ME	SG	PR	BN	No Reg	Var Reg	Var/Cov Reg
BYOL	✓	✓	✓	✓	69.3†	70.2	69.5
SimSiam		✓	✓	✓	67.9†	68.1	67.6
SimSiam		✓	✓		35.1	67.3	67.1
SimSiam		✓			collapse	56.8	66.1
VICReg			✓		collapse	56.2	67.3
VICReg			✓	✓	collapse	57.1	68.7
VICReg				✓	collapse	57.5	68.6†
VICReg					collapse	56.5	67.4

VICReg: Variance/Covariance regularization helps other methods

Table 5: **Impact of variance-covariance regularization.** Inv: a invariance loss is used, $\lambda > 0$, Var: variance regularization, $\mu > 0$, Cov: covariance regularization, $\nu > 0$, in Eq. (6).

Method	λ	μ	ν	Top-1
Inv	1	0	0	collapse
Inv + Cov	25	0	1	collapse
Inv + Cov	0	25	1	collapse
Inv + Var	1	1	0	57.5
Inv + Var + Cov (VICReg)	25	25	1	68.6

Table 6: **Impact of normalization.** Std: variables are centered and divided by their standard deviation over the batch. This is applied or not to the embedding and the expander hidden layers. l_2: the embedding vectors are l_2-normalized.

Expander	Embedding	Top-1
Std	None	68.6
Std	Std	68.4
None	None	67.4
None	Std	67.2
Std	l_2	65.1

VICReg: No need for weight sharing between the branches!

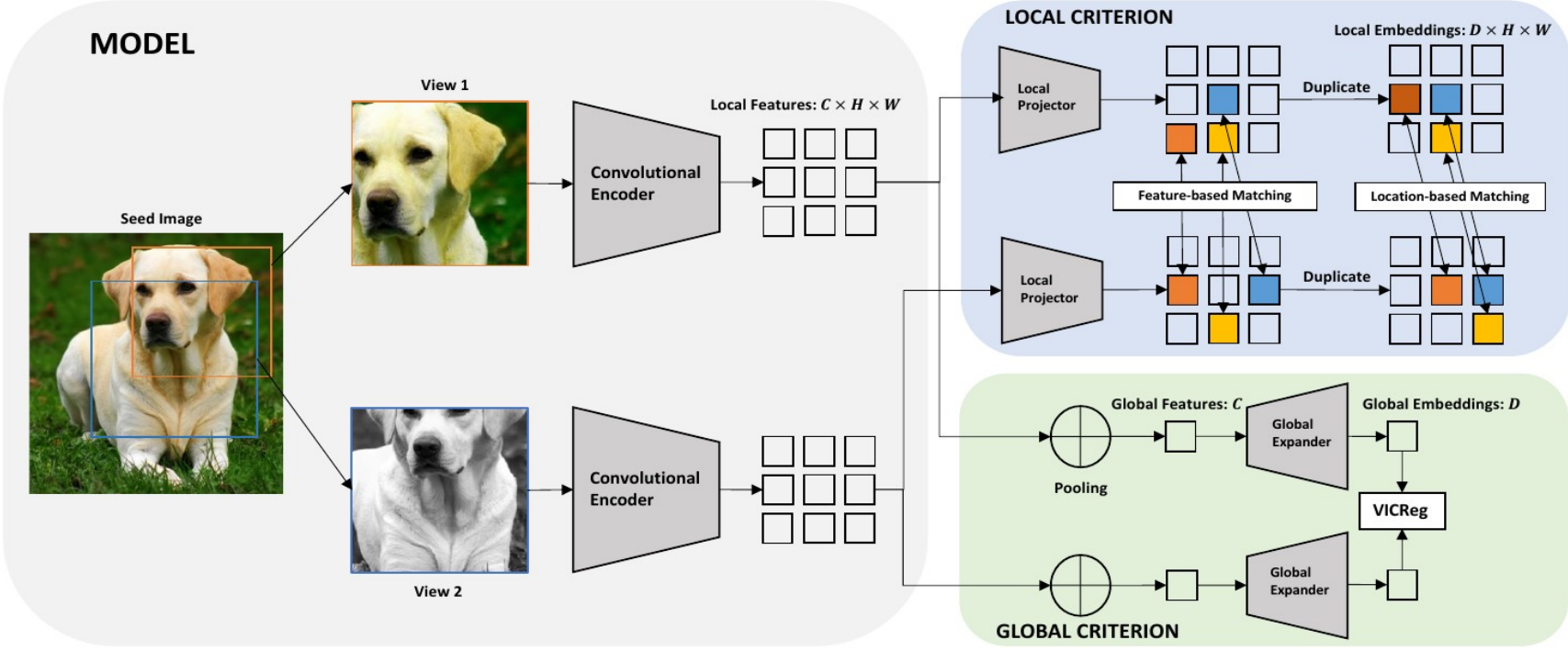
- **No need for weight sharing!**
- **The two branches can take inputs of different nature.**
- **Opens the door to many applications of non-contrastive SSL to many domains**

Table 4: **Impact of sharing weights or not between branches.** Top-1 accuracy on linear classification with 100 pretraining epochs. In all settings, the encoder and expander of both branches share the same architecture, but either share weights (✓), or have different weights in the two branches.

Encoder	Expander	Top-1
		66.5
	✓	67.3
✓		67.8
✓	✓	68.6

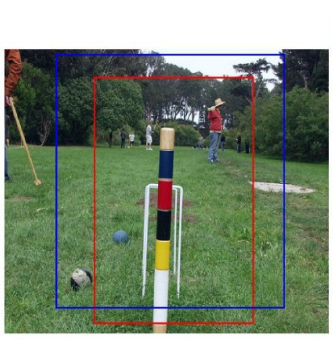
VICRegL: local matching latent variable for segmentation

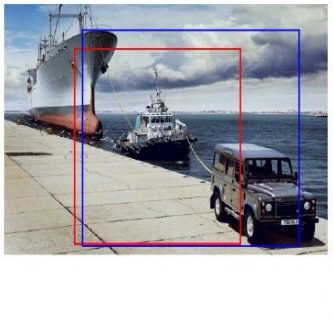
- **Latent variable optimization:**
  - Finds a pairing between local feature vectors of the two images
  - [Bardes, Ponce, LeCun NeurIPS 2022, arXiv:2210.01571]



VICRegL: local matching latent variable for segmentation

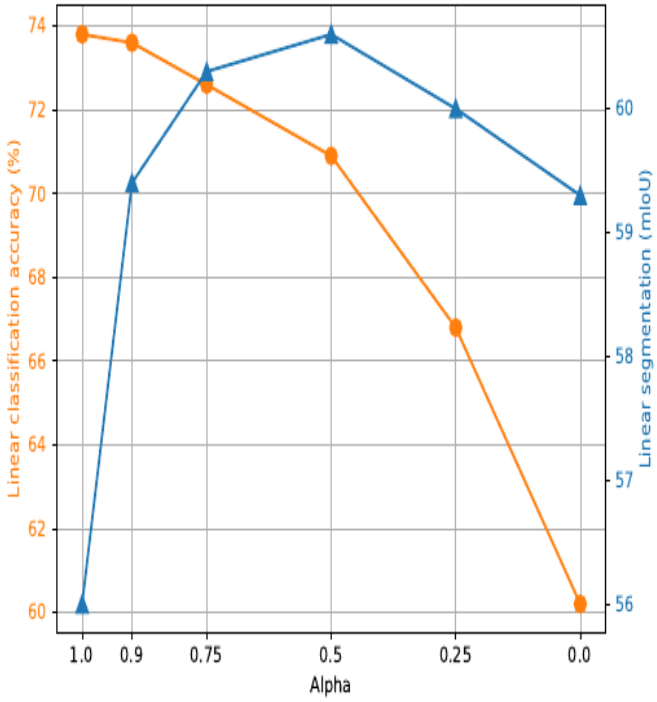
Method	Epochs	Linear Cls. (%) ImageNet Frozen	Linear Seg. (mIoU) Pascal VOC Frozen	Pascal VOC Fine-Tuned	Cityscapes Frozen
Global features					
MoCo v2 [Chen et al., 2020b]	200	67.5	35.6	64.8	14.3
SimCLR [Chen et al., 2020a]	400	68.2	45.9	65.4	17.9
BYOL [Grill et al., 2020]	300	**72.3**	47.1	65.7	22.6
VICReg [Bardes et al., 2022]	300	71.5	47.8	65.5	23.5
Local features					
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^2 [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
Ours					
VICRegL $\alpha = 0.9$	300	71.2	54.0	66.6	25.1
VICRegL $\alpha = 0.75$	300	70.4	**55.9**	**67.6**	**25.2**





VICRegL: local matching latent variable for segmentation

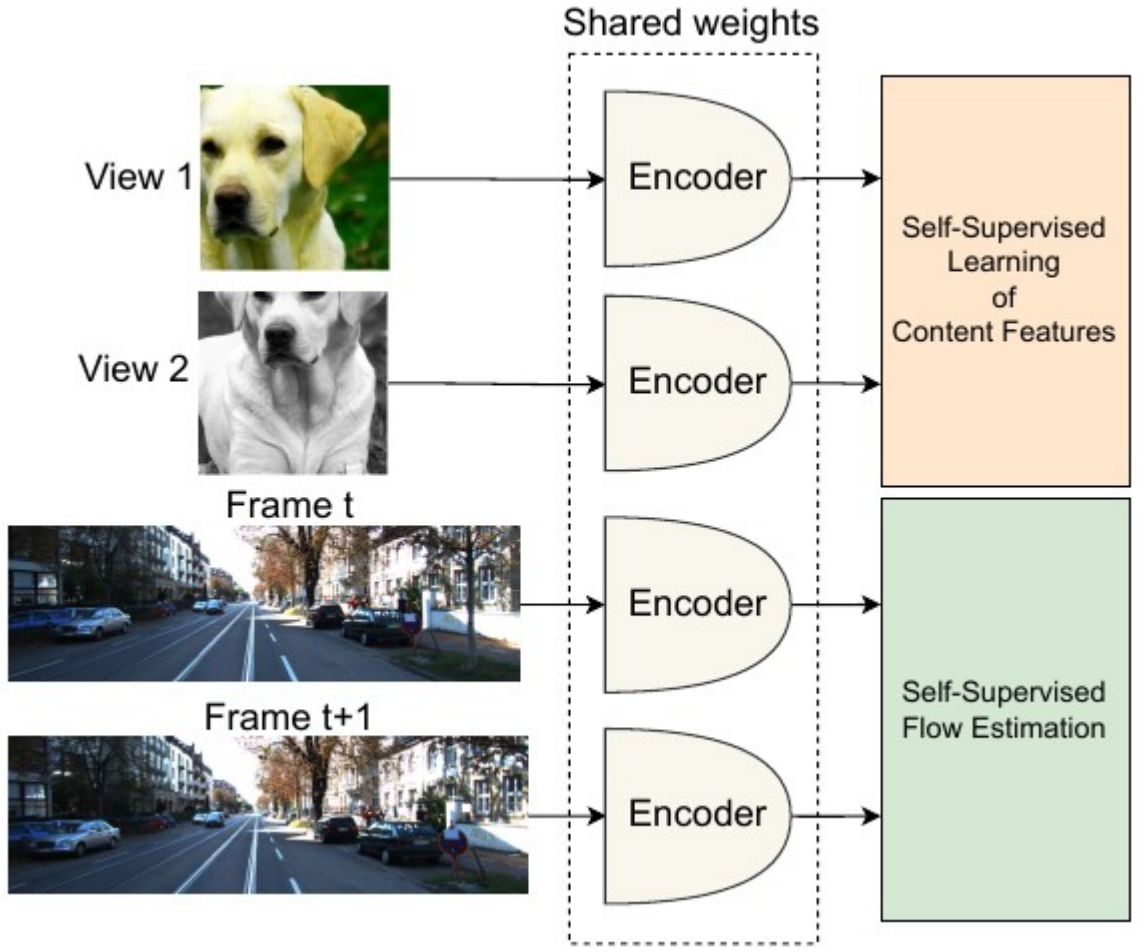
Method	Backbone	Params	Epochs	Linear Cls. (%) ImageNet Frozen	Linear Seg. (mIoU) Pascal VOC Frozen	Pascal VOC FT	ADE20k Frozen
Global features							
MoCo v3 [Chen et al., 2021]	ViT-S	21M	300	73.2	57.1	75.9	23.7
DINO [Caron et al., 2021]	ViT-S	21M	400	77.0	65.2	79.5	30.5
IBOT [Zhou et al., 2022a]	ViT-S	21M	400	77.9	68.2	79.9	33.2
VICReg [Bardes et al., 2022]	CNX-S	50M	400	76.2	60.1	77.8	28.6
MoCo v3	ViT-B	85M	300	76.7	64.8	78.9	28.7
DINO	ViT-B	85M	400	78.2	70.1	82.0	34.5
IBOT [Zhou et al., 2022a]	ViT-B	85M	400	**79.5**	73.0	82.4	38.3
MAE [He et al., 2022]	ViT-B	85M	400	68.0	59.6	82.4	27.0
VICReg	CNX-B	85M	400	77.6	67.2	81.1	32.7
Local features							
CP^2 [Wang et al., 2022]	ViT-S	21M	320	62.8	63.5	79.6	25.3
Ours							
VICRegL $\alpha = 0.9$	CNX-S	50M	400	75.9	66.7	80.0	30.8
VICRegL $\alpha = 0.75$	CNX-S	50M	400	74.6	67.5	80.6	31.2
VICRegL $\alpha = 0.9$	CNX-B	85M	400	77.1	69.3	81.2	33.5
VICRegL $\alpha = 0.75$	CNX-B	85M	400	76.3	70.4	82.5	35.3
VICRegL $\alpha = 0.75^{\dagger}$	CNX-XL	350M	150	79.4	**78.7**	**84.1**	**43.2**



Segmentation
Classification

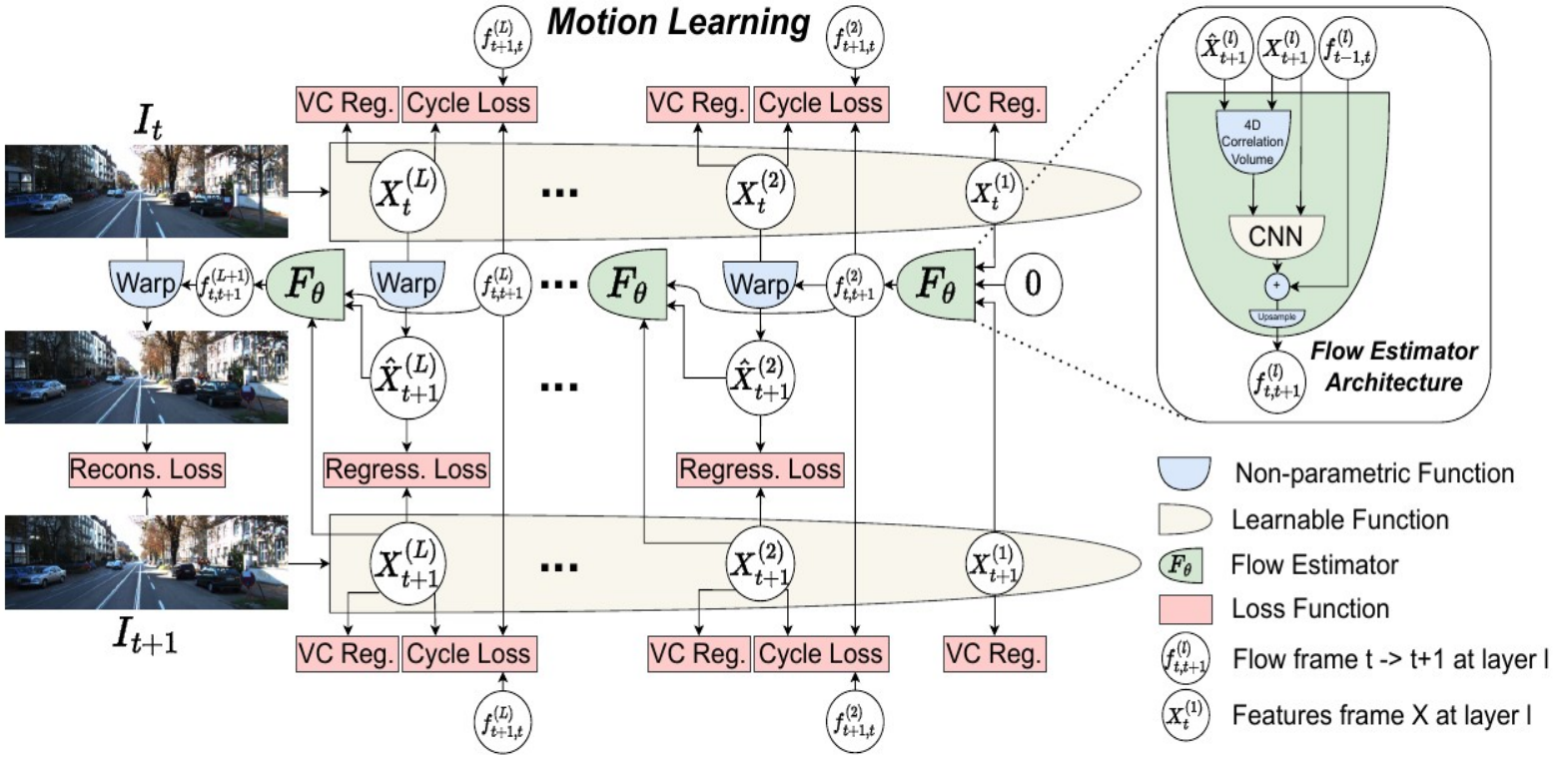
MC-JEPA: Motion & Content JEPA

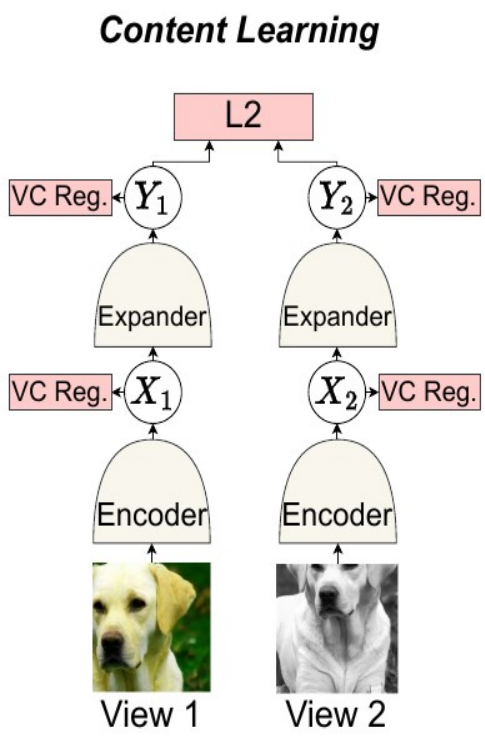
- **Simultaneous SSL for**
  - Image recognition
  - Motion estimation
- **Trained on**
  - ImageNet 1k
  - Various video datasets
- **Uses VCReg to prevent collapse**
  - ConvNext-T backbone



MC-JEPA: Motion & Content JEPA

- **Motion estimation architecture uses a top-down hierarchical predictor that "warp" feature maps.**

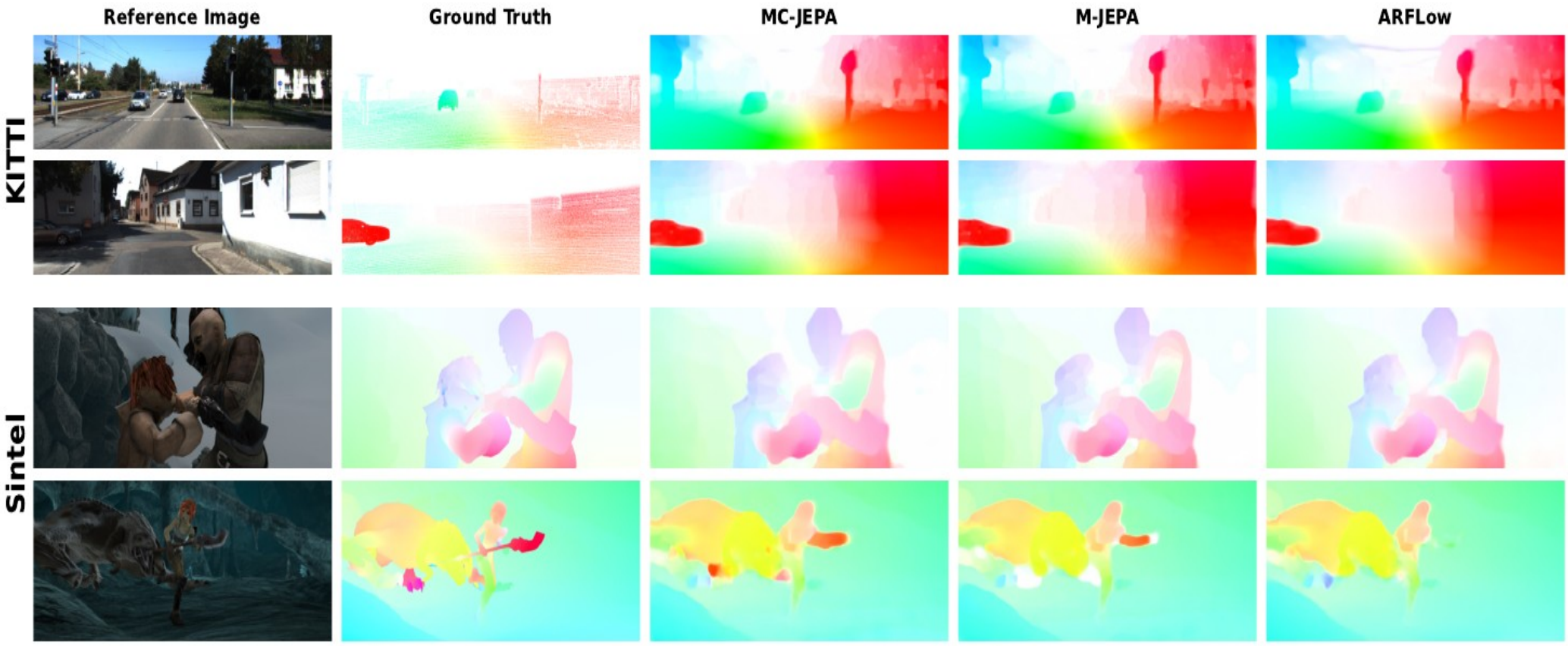




MC-JEPA: Results

Method	Backbone	Optical Flow Estimation						Image Segmentation						Video Seg.
		Sintel Clean		Sintel Final		KITTI 2015		Pascal VOC		CityScapes		ADE20k		Davis 2017
		train EPE	test EPE	train EPE	test EPE	train EPE	test F1	Frozen mIoU	FT mIoU	Frozen mIoU	FT mIoU	Frozen mIoU	FT mIoU	$(\mathcal{J}\&\mathcal{F})_m$
Rand. weights	CNX-T	23.71	-	24.02	-	24.88	-	0.5	-	-	-	-	-	-
flow methods														
UFlow [45]	PWC	2.50	5.21	3.39	6.50	2.71	11.13	7.8	-	-	-	-	-	42.0
ARFLow [51]	PWC	2.79	4.78	3.73	5.89	2.85	11.80	7.9	-	-	-	-	-	-
UPFlow [56]	PWC	2.33	4.68	2.67	5.32	2.45	9.38	8.8	-	-	-	-	-	-
SMURF [66]	RAFT	**1.71**	**3.15**	**2.58**	**4.18**	**2.00**	**6.83**	10.4	-	-	-	-	-	-
correspondence methods														
VFS [77]	R50	-	-	-	-	-	-	51.2	-	-	-	-	-	68.9
MCRW [8]	PWC	2.84	5.68	3.82	6.72	2.81	11.67	39.8	-	-	-	-	-	57.9
content methods														
VICReg [6]	CNX-T	-	-	-	-	13.5	-	60.1	77.8	59.8	76.3	28.6	41.1	58.1
VICRegL [7]	CNX-T	-	-	-	-	11.4	-	66.8	79.7	64.9	78.3	30.6	44.1	66.7
MoCo v3	ViT-S	-	-	-	-	12.9	-	57.1	75.9	56.5	74.0	23.7	39.8	-
DINO [14]	ViT-S	-	-	-	-	11.8	-	65.2	79.5	64.8	78.1	30.5	43.5	69.9
ours														
M-JEPA	CNX-T	2.98	-	3.82	-	3.01	-	9.4	-	-	-	-	-	-
MC-JEPA	CNX-T	2.81	5.01	3.51	6.12	2.67	11.33	**67.1**	**79.9**	**65.5**	**78.4**	**30.8**	**44.2**	**70.5**

MC-JEPA: Optical Flow Estimation Results



MC-JEPA: Video Segmentation Results

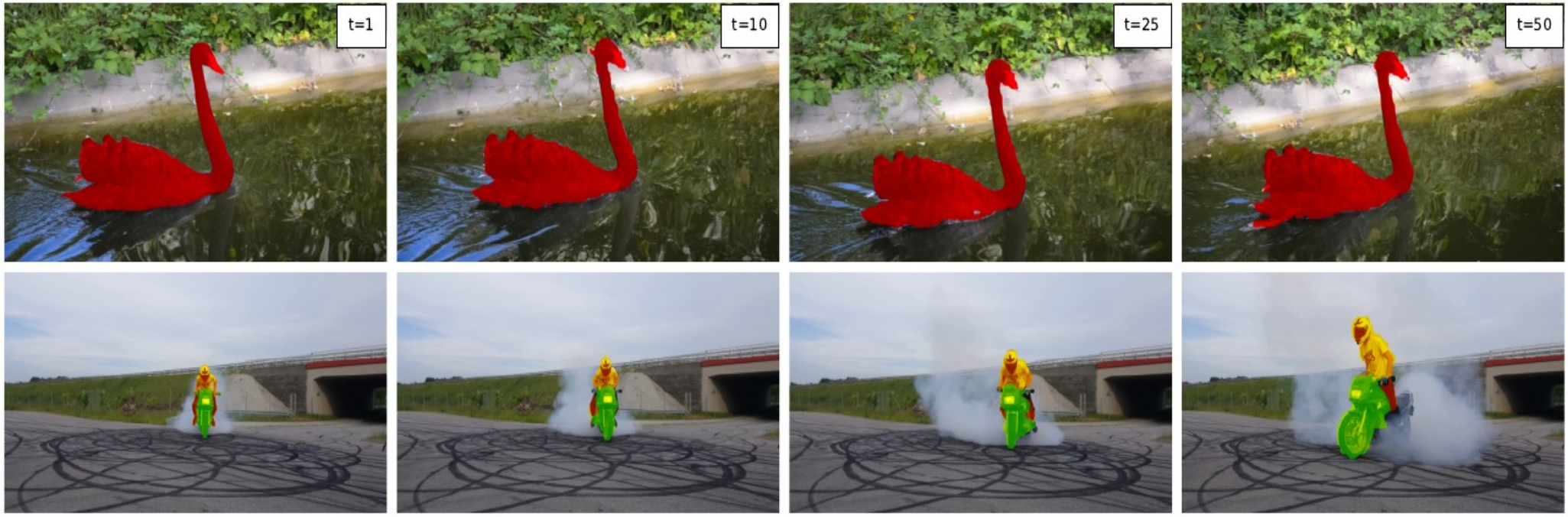
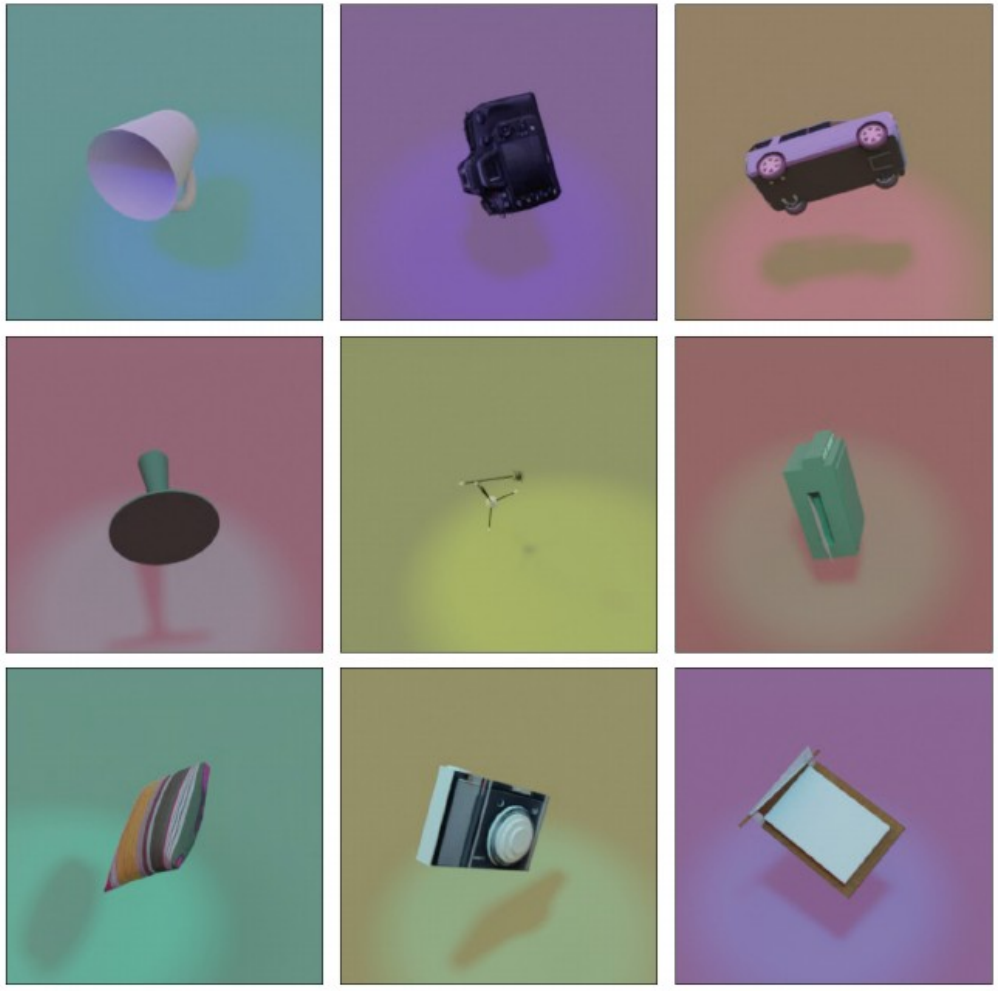


Figure 4. **Qualitative visualization: video segmentation.** We visualize the segmentation maps obtained by the frozen features learnt with MC-JEPA on the video instance tracking task on DAVIS 2017, for several video sequences, at frames t=1,10,25,50. Frame 1 is given as ground truth, and the others are predicted by our model.

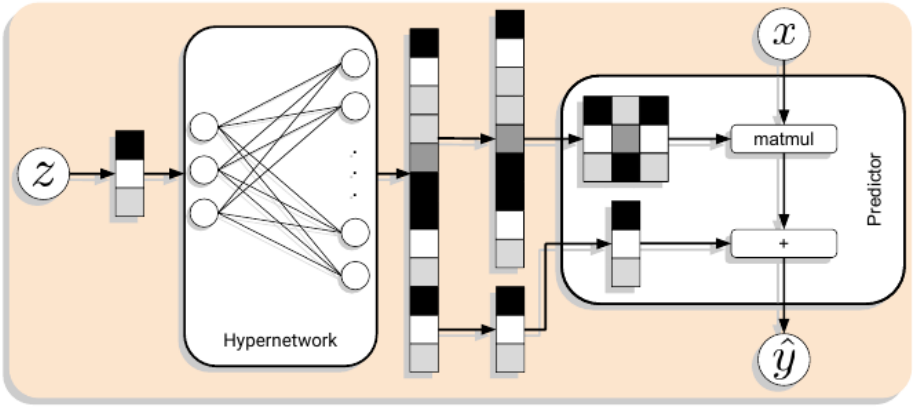
Split Invariant-Equivariant Representation Learning

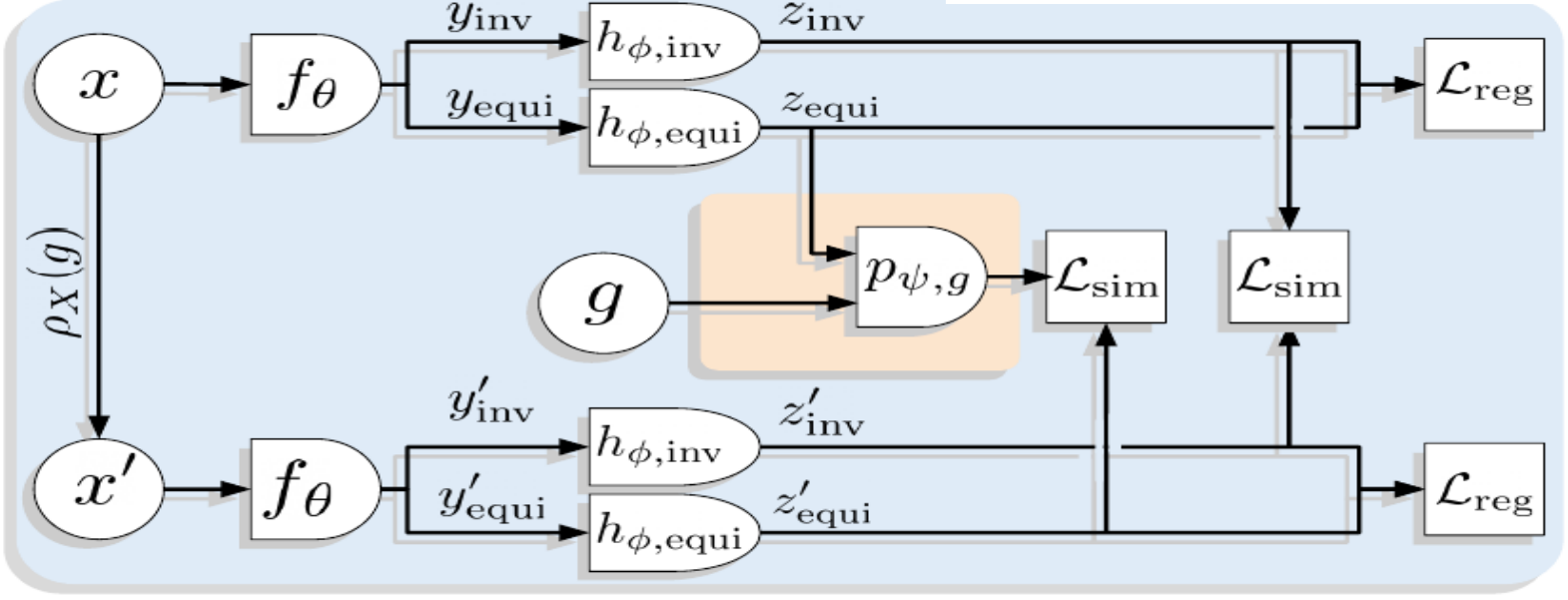
- **Training on multiple rendered views of 3D objects**
  - 3DIEBench dataset
- **Split representation**
  - Invariant part:
    - encodes shape identity
  - Equivariant part:
    - Encodes pose
- **[Garrido ArXiv:2302.10283]**



Split Invariant-Equivariant Representation Learning

- ConvNext backbone
- 2 heads for invariant and equivariant
- Predictor for equivariant part (JEPA)
- Predictor is a hypernetwork
- VC regularization





Split Invariant-Equivariant Representation Learning

Method	Classification (top-1)			Rotation prediction (R^2)			Color prediction (R^2)		
	All	Inv.	Equi.	All	Inv.	Equi.	All	Inv.	Equi.
Baselines									
Supervised	87.47			0.76					
Random				0.23					
Invariant and parameter prediction methods									
VICReg	84.74			0.41			0.06		
VICReg, g kept identical	72.81			0.56			0.25		
SimCLR	**86.73**			0.50			0.30		
SimCLR, g kept identical	71.21			0.54			0.83		
Parameter prediction	85.11			**0.75**			0.12		
Equivariant methods									
Only equivariant (Original predictor)	86.93			0.51			0.23		
Only equivariant (Our predictor)	86.10			0.60			0.24		
EquiMod (Original predictor)	**87.19**			0.47			0.21		
EquiMod (Our predictor)	**87.19**			0.60			0.13		
SIE (Ours)	82.94	82.08	80.32	**0.73**	0.23	0.73	0.07	0.05	0.02

Split Invariant-Equivariant Representation Learning

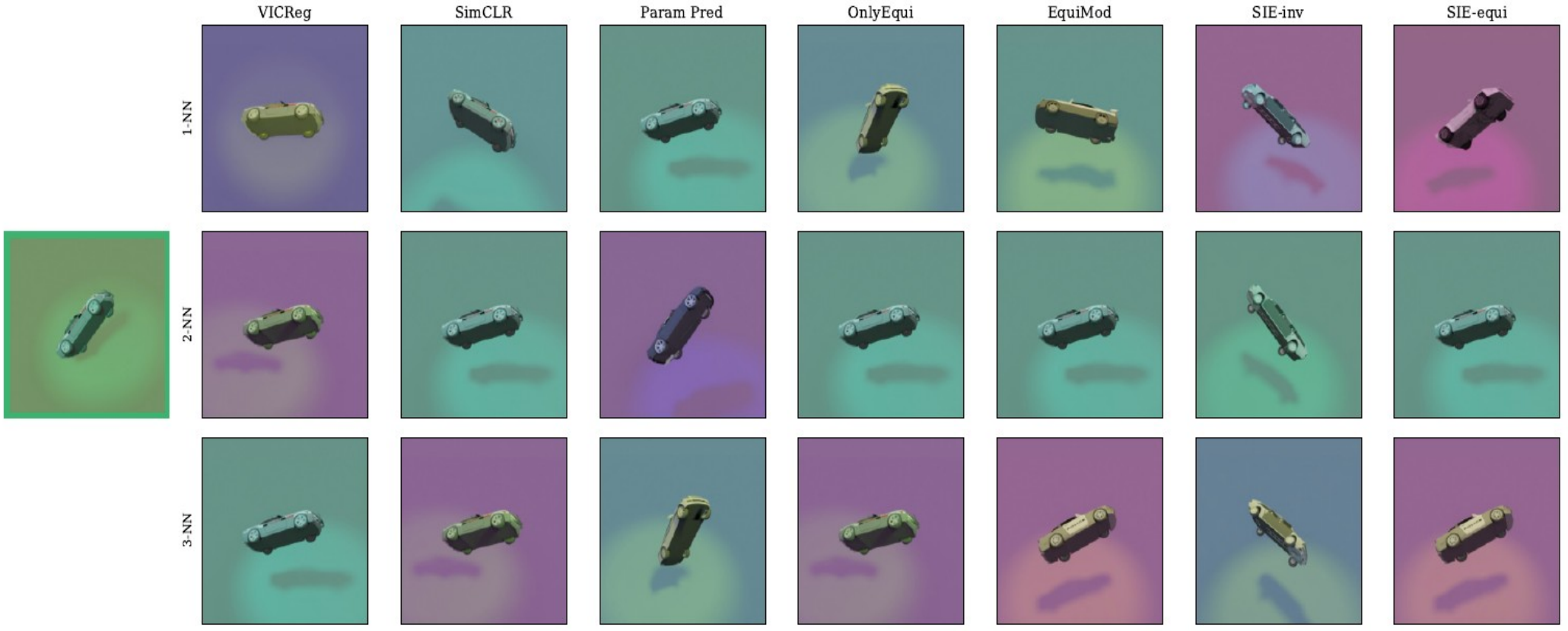


Figure 3: Retrieval of nearest representations. Starting from the representation associate to the object in the green frame on the left, we compute its nearest neighbours for all considered methods and show the 3 closest.

Split Invariant-Equivariant Representation Learning

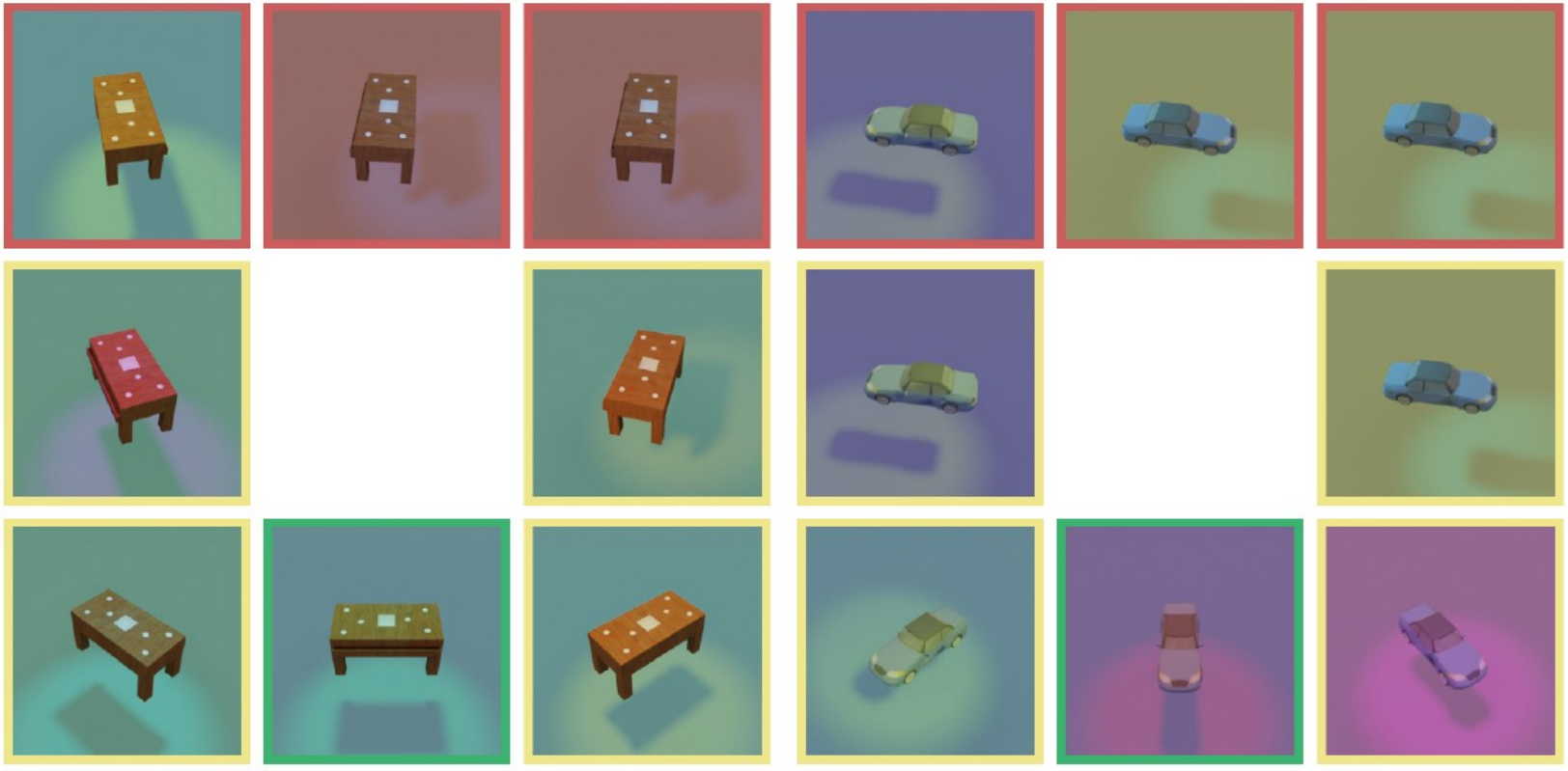


Figure 7: Generalization to unseen rotations during training. Starting from the canonical view (green frame), we apply rotations through the predictor of a trained SIE. Rotation were either possibly seen during training (orange frame) or could not have been seen (red frame).

Distillation Methods

- **Modified Siamese nets**
  - Predictor head eliminates variation of representations due to distortions
  - BYOL: Teacher branch uses a moving-average of the parameters of the student branch
- **Examples:**
  - Bootstrap Your Own Latents [Grill arXiv:2006.07733]
  - SimSiam [Chen & He arXiv:2011.10566]
- **Advantages**
  - No negative samples
- **Issues**
  - Not clear why they don't collapse (normalization?)

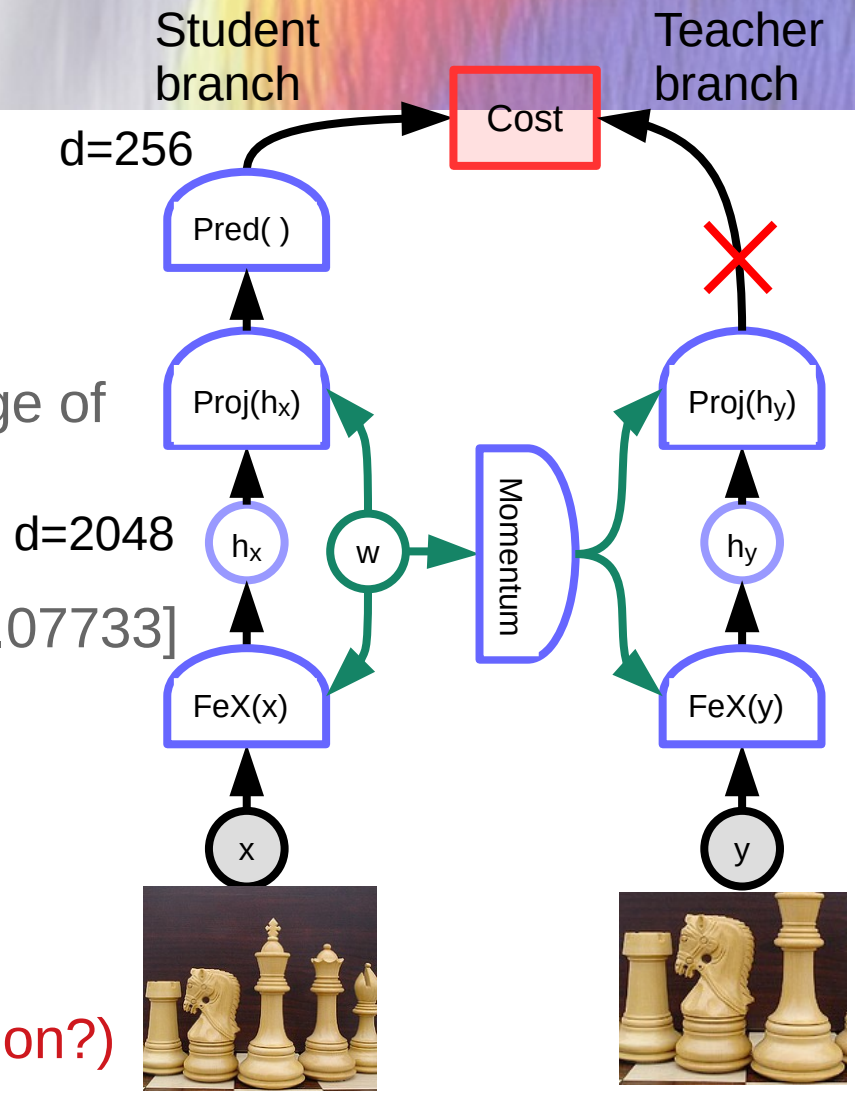
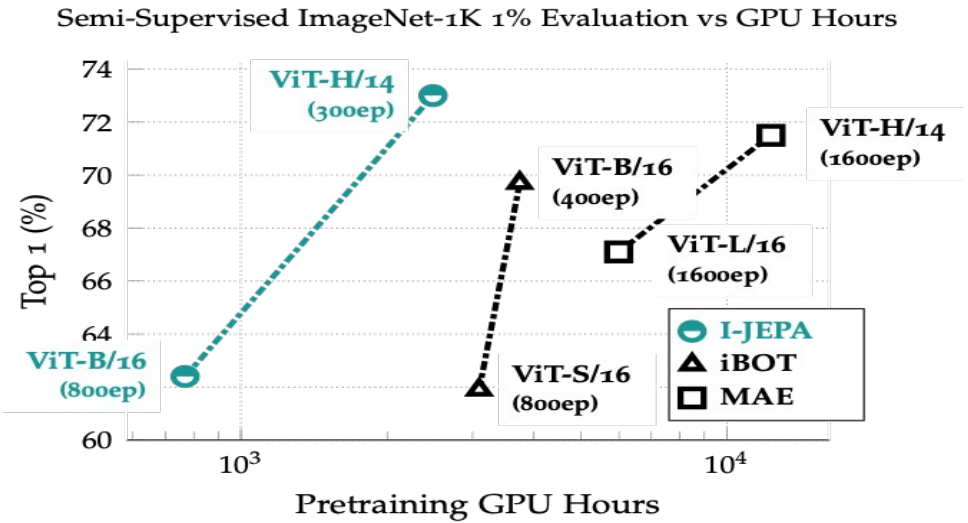
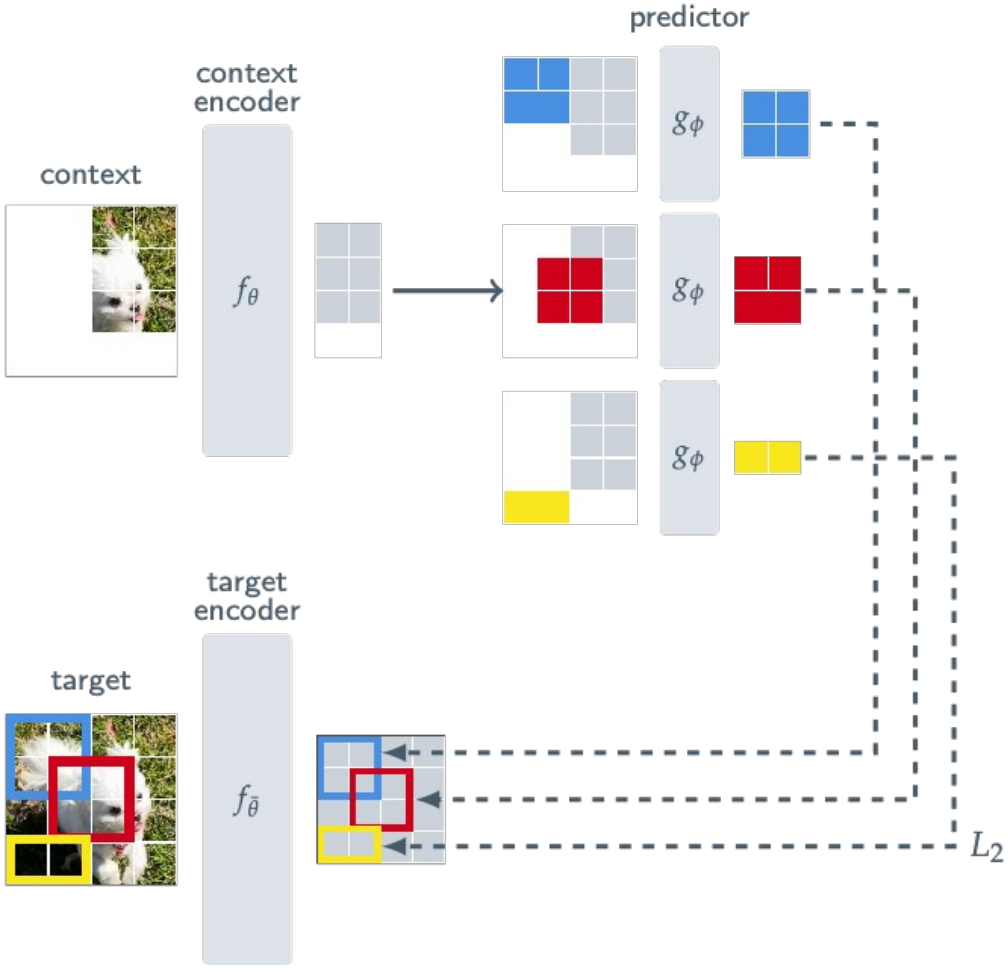


Image-JEPA: uses masking, transformer, EMA weights

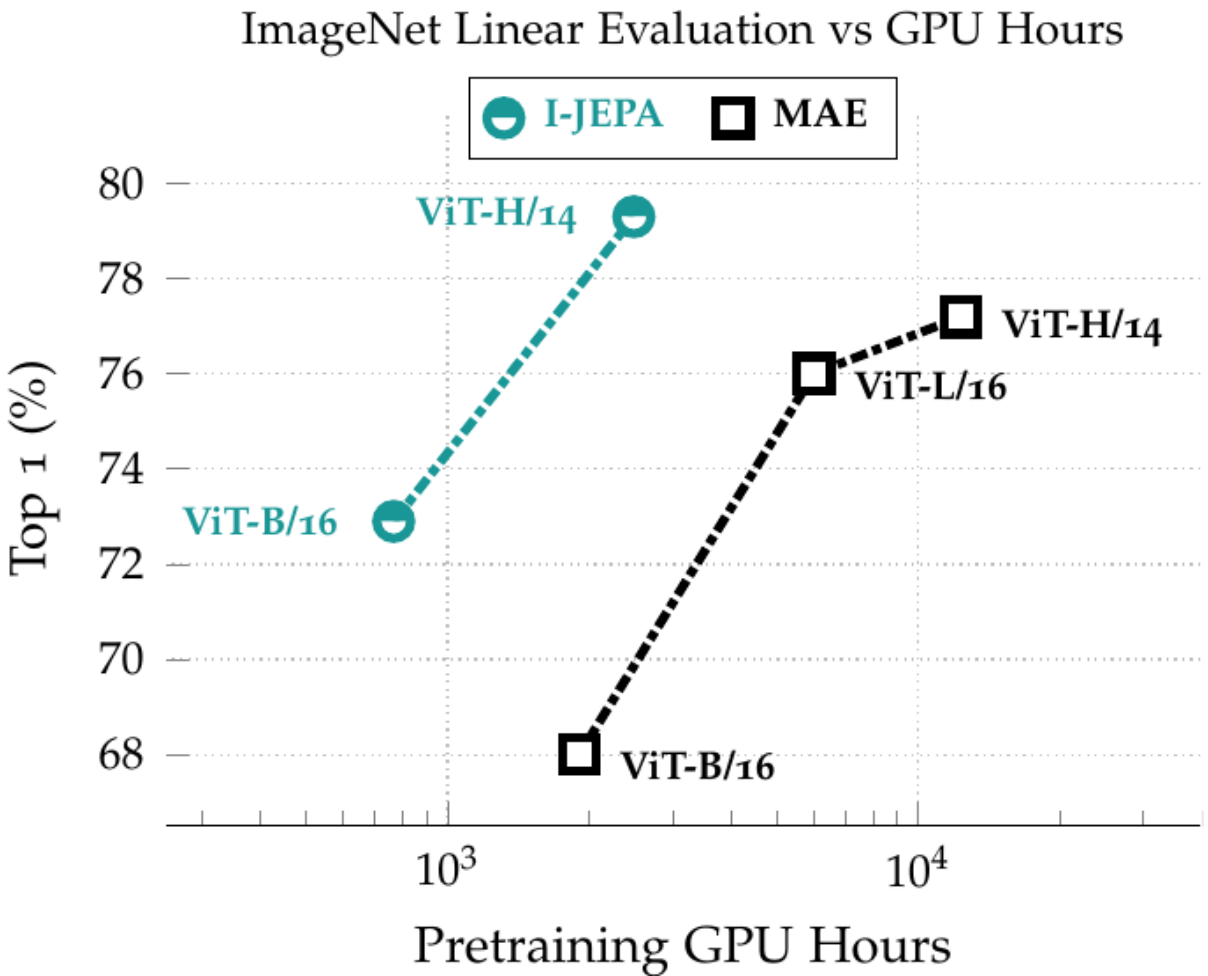
- **"SSL from images with a JEPA"**
  - M. Assran et al arxiv:2301.08243
- **Jointly embeds a context and a number of neighboring patches.**
  - Uses predictors
  - Uses only masking





I-JEPA Results

- Training is fast
- Non-generative method seems to beat reconstruction-based methods (MAE)



I-JEPA Results on ImageNet

- **JEPA better than generative architecture on pixels.**

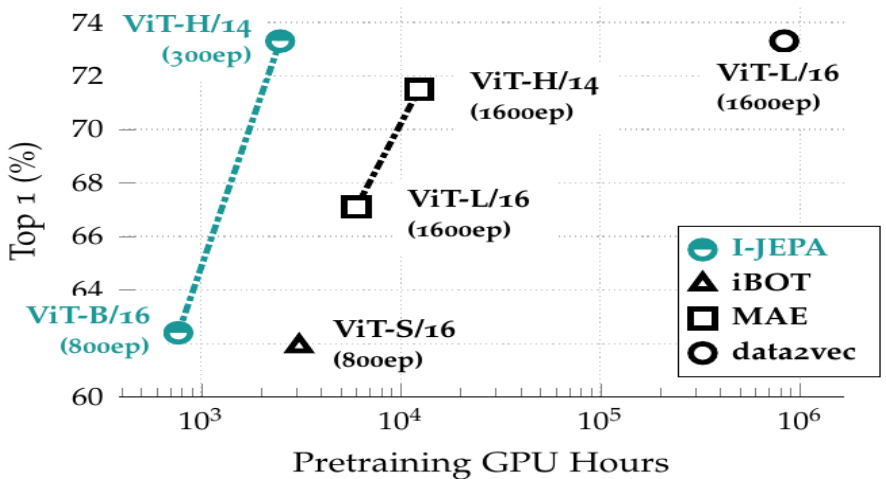
Targets	Arch.	Epochs	Top-1
Target-Encoder Output	ViT-L/16	500	**66.9**
Pixels	ViT-L/16	800	40.7

- **Closing the gap with methods that use data augments**
- **Methods with only masking**
  - No data augmentation
- **Methods with data augmentation**
  - Similar to SimCLR

Method	Arch.	Epochs	Top-1
Methods without view data augmentations			
data2vec [7]	ViT-L/16	1600	53.5
MAE [34]	ViT-B/16	1600	68.0
	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_{448}	300	**81.1**
Methods using extra view data augmentations			
SimCLR v2 [20]	RN152 (2×)	800	79.1
DINO [17]	ViT-B/8	300	80.1
iBOT [74]	ViT-L/16	250	**81.0**

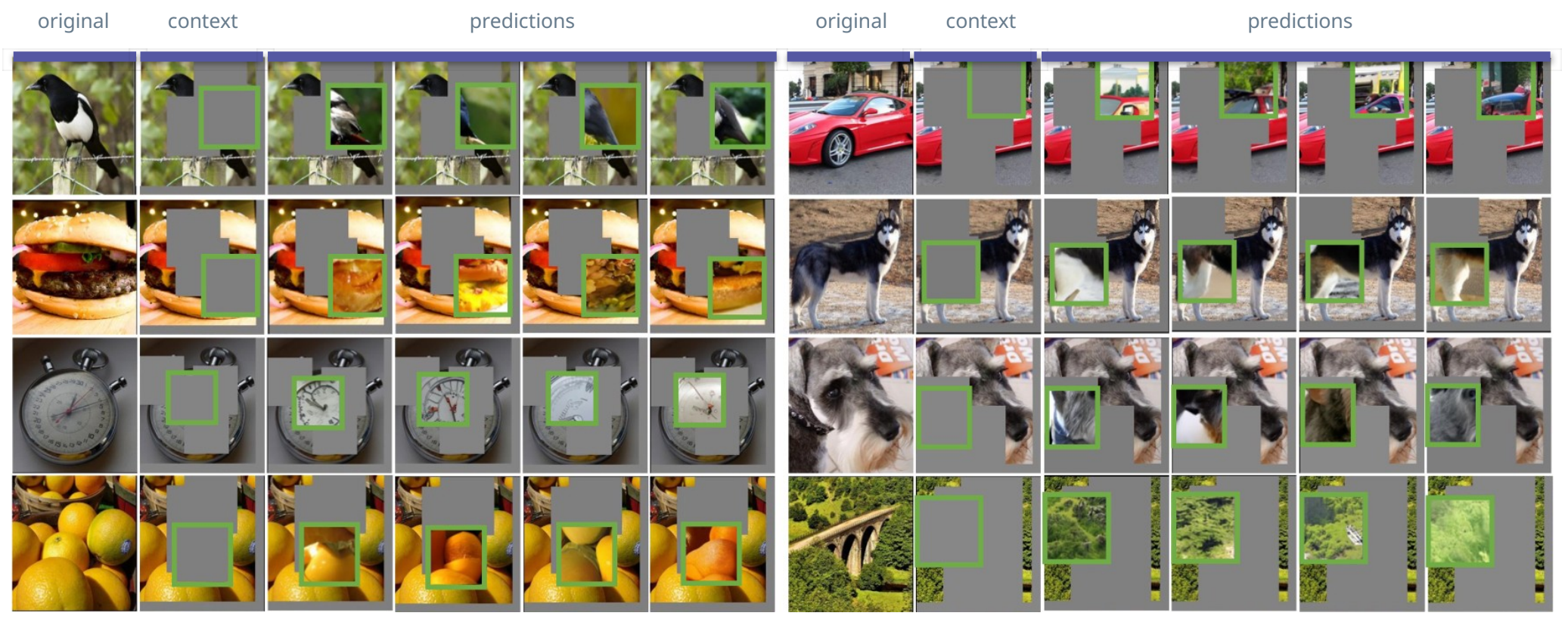
I-JEPA Results on ImageNet with 1% training

- 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
Methods without view data augmentations			
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_{448}	300	**77.3**
Methods using extra view data augmentations			
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	**75.7**

I-JEPA: Visualizing Predicted Representations



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"

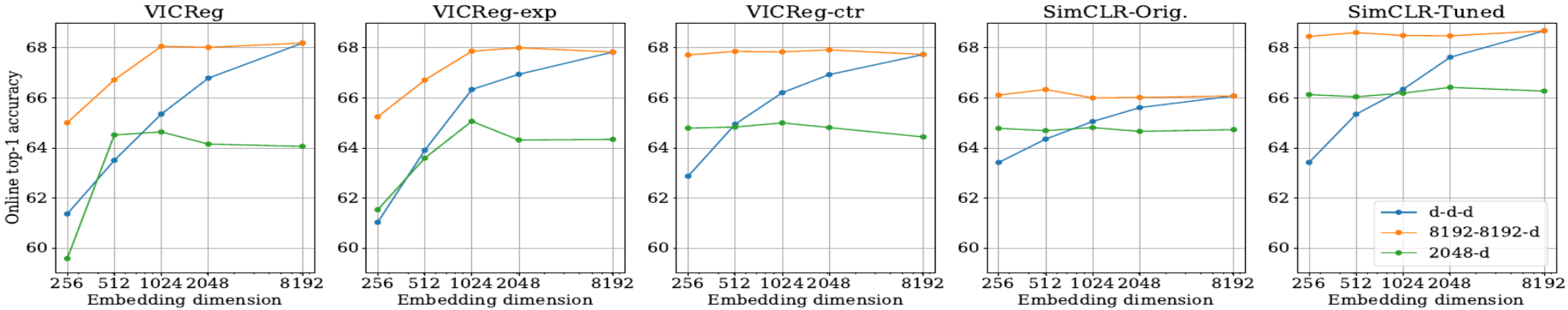


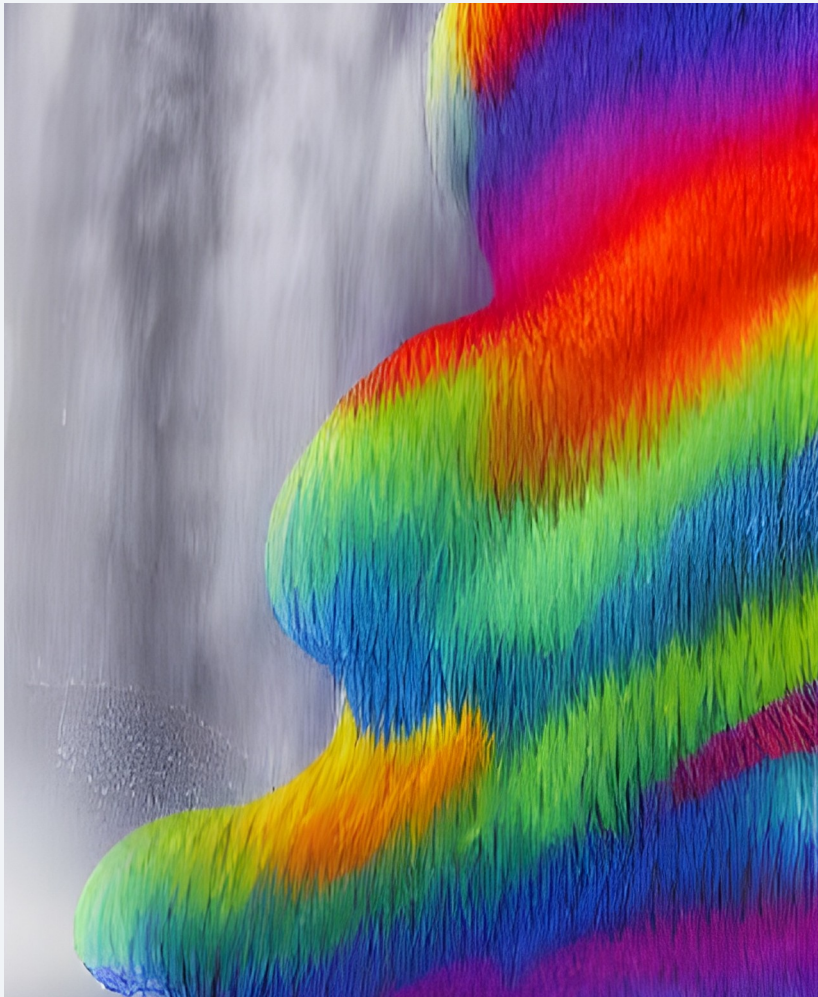
Figure 1: VICReg, VICReg-exp and VICReg-ctr perform similarly in 100 epochs training, validating empirically our theoretical result. While the original implementation of SimCLR performs significantly worse – which is unexpected per our theory – we are able to improve its performance to VICReg's level. 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.

NEW YORK UNIVERSITY

Meta AI

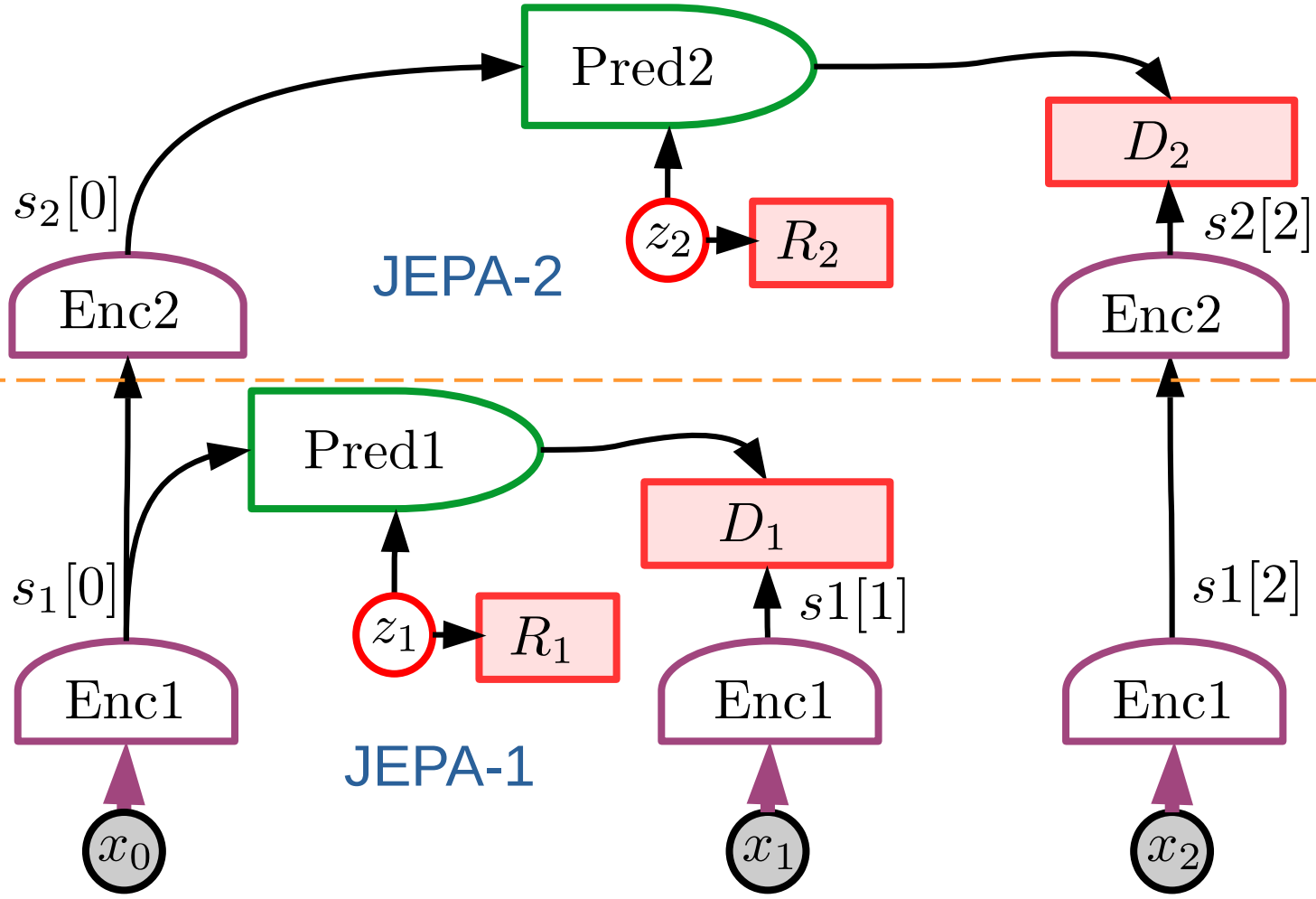
Hierarchical JEPA for Hierarchical Planning

Control, planning, and policy learning.



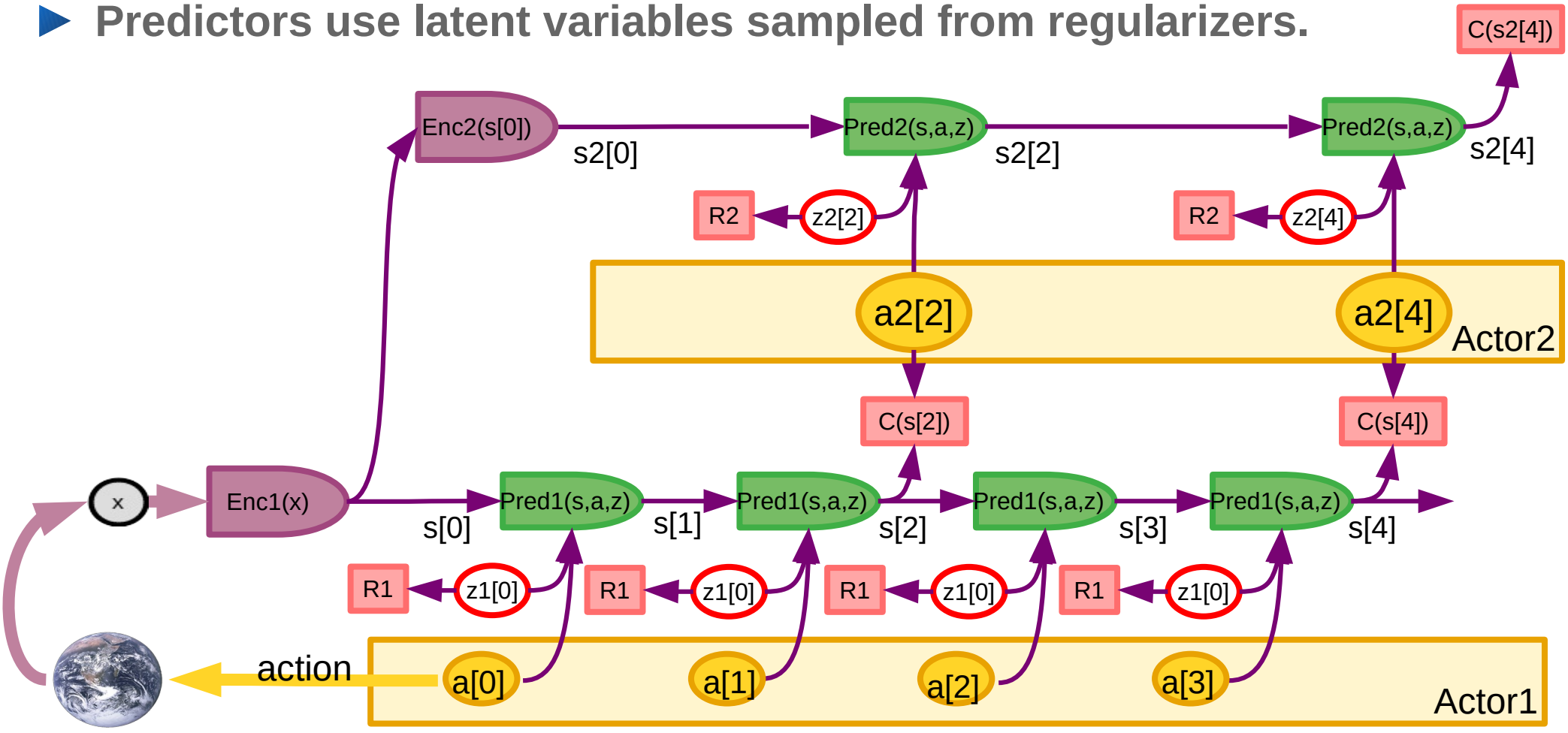
Multi time-scale Predictions

- **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

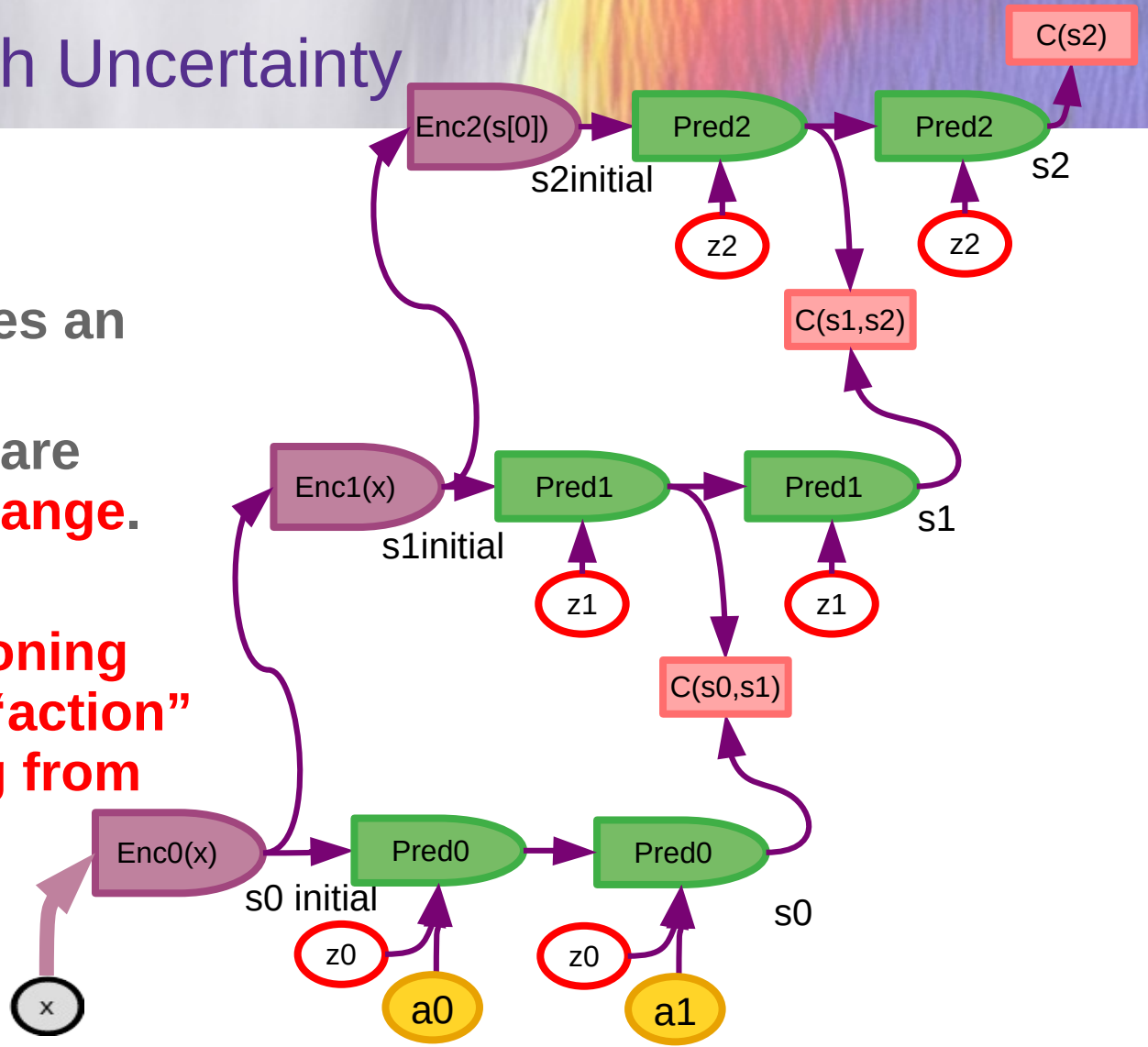
- **Predictors use latent variables sampled from regularizers.**



Hierarchical Planning with Uncertainty

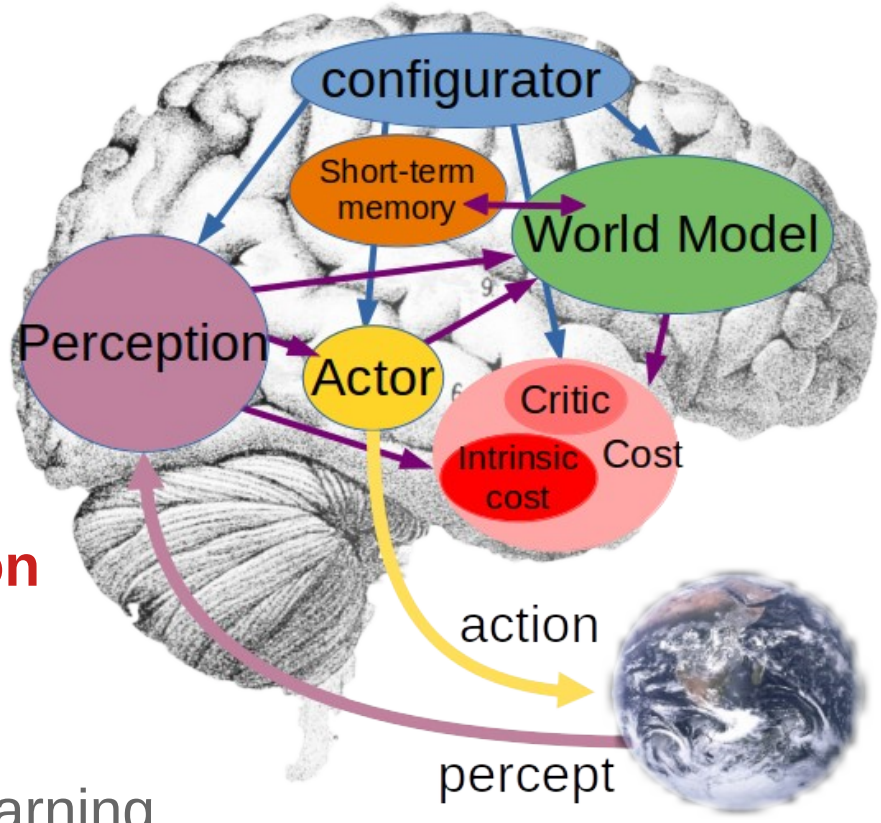
- Hierarchical world model
- Hierarchical planning
- An **action** at level k specifies an **objective** for level k-1
- Prediction in higher levels are more **abstract** and **longer-range**.

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



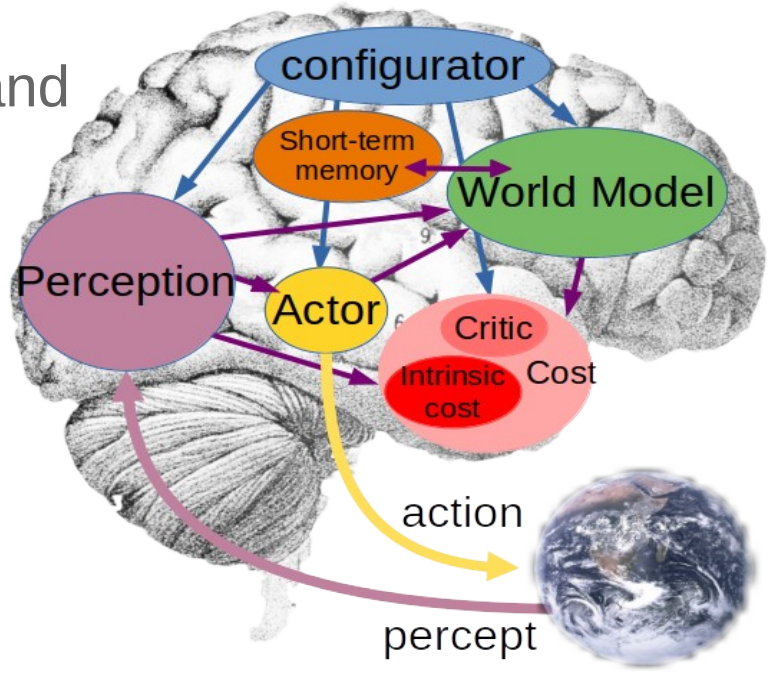
Steps towards Autonomous AI Systems

- **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



A Single, Configurable World Model Engine

- **What is the Configurator?**
- **The configurator configures the agent for a deliberate ("conscious") tasks.**
  - Configures all other modules for the task at hand
  - Primes the perception module
  - Provides executive control
  - Sets subgoals
  - Configures the world model for the task.



- **There is a single world model engine**
  - The system can only perform one "conscious" task at a time
  - Consciousness is a consequence of the single-world-model limitation

Challenges for SSL Research

- Finding a **general recipe** for training Hierarchical Joint Embedding Architectures 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 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**

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.

NEW YORK UNIVERSITY

Meta AI

Thank You!