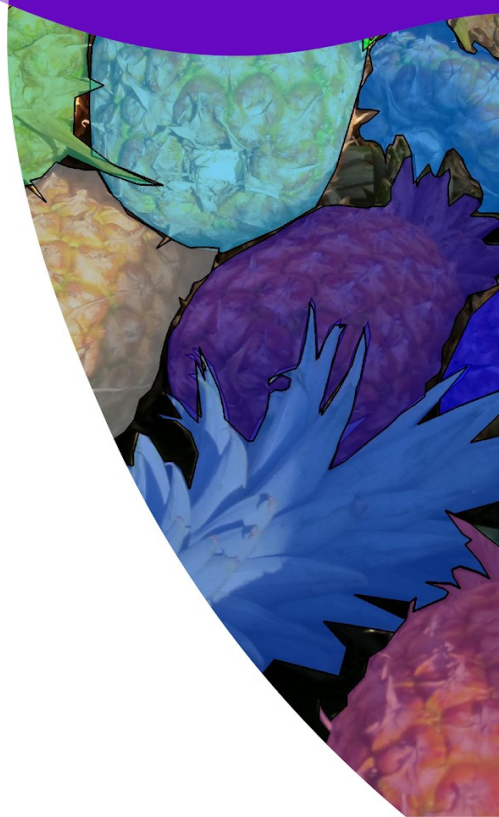
NEW YORK UNIVERSITY

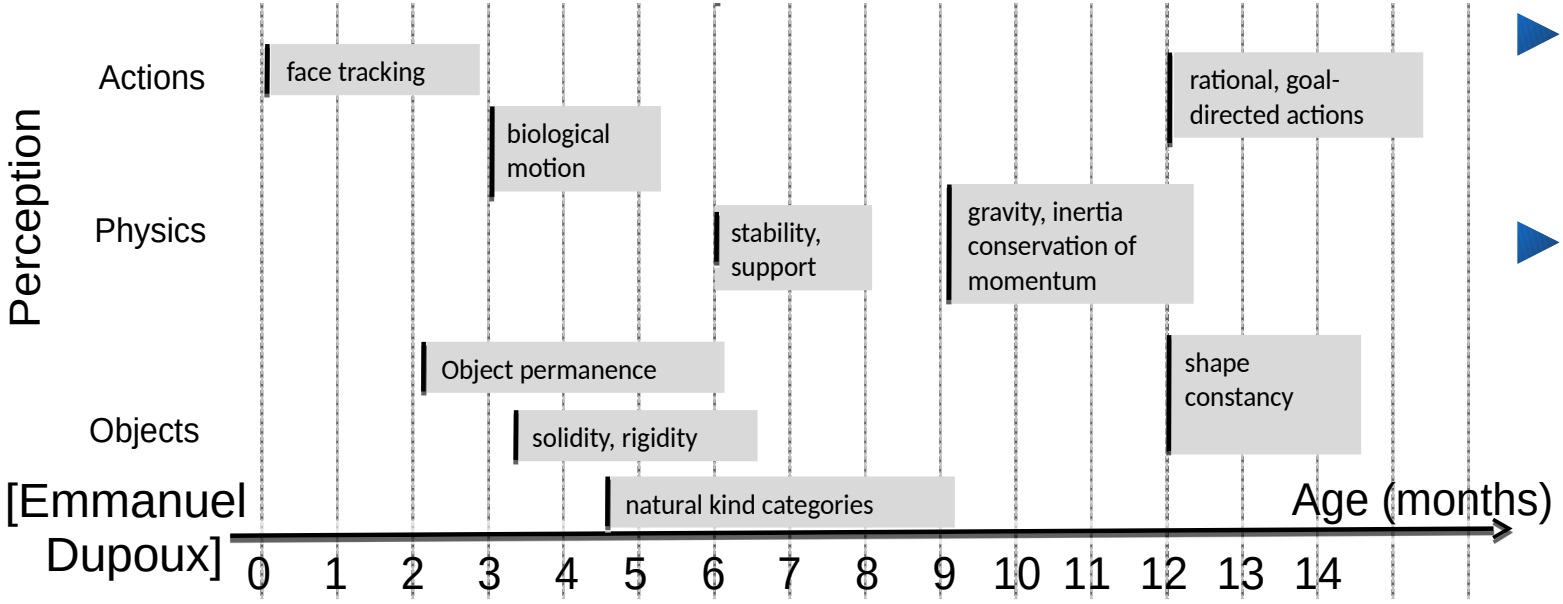
A Path Towards Human-Level AI

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



IAIFI 2022-04-01

How could machines learn like animals and humans?



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

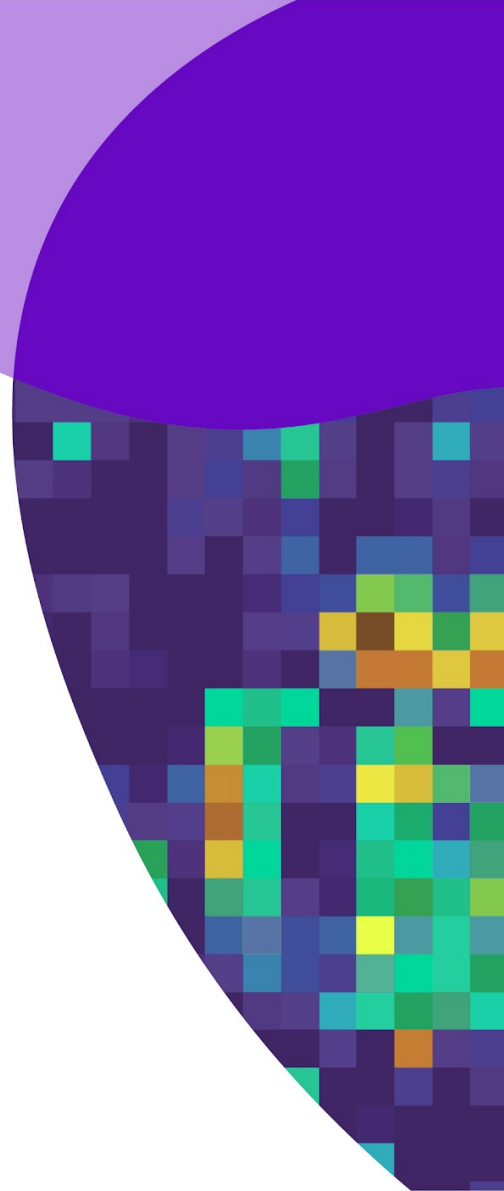
Three challenges for AI & Machine Learning

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

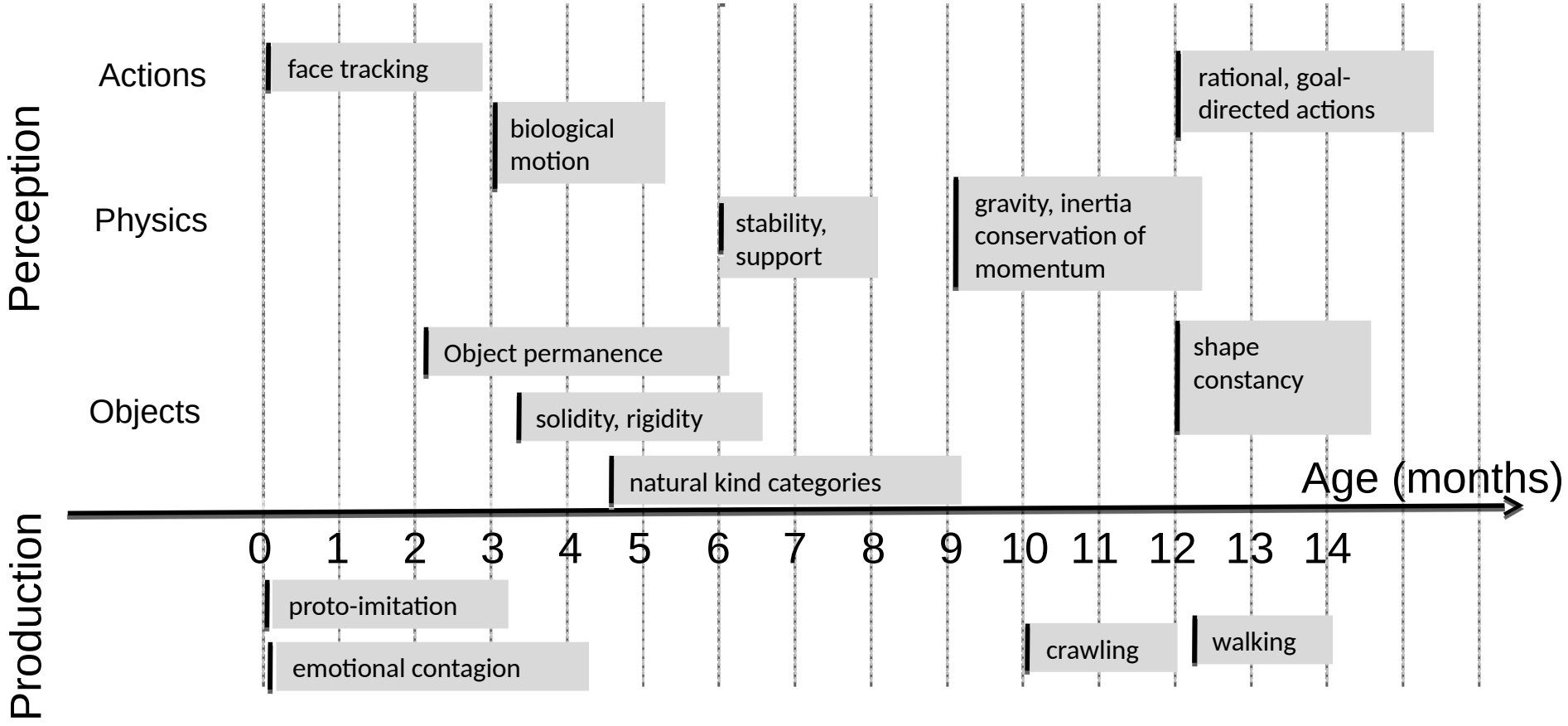
FACEBOOK AI

How do humans and animals learn so quickly?

Not supervised.
Not Reinforced.



When infants learn models of the world [after Emmanuel Dupoux]



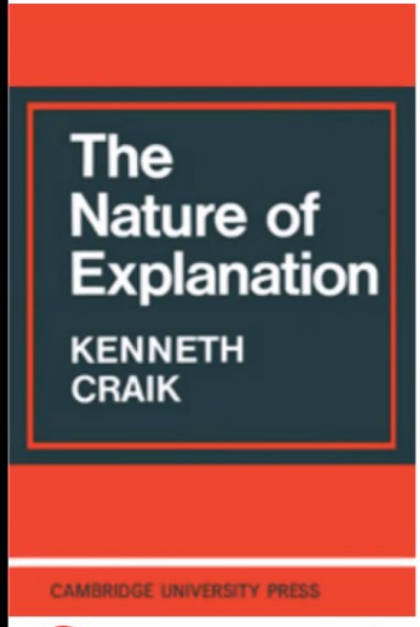
How do Human and Animal Babies Learn?

- **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

Common sense is a collection of models of the world

AI systems need to build "mental models"



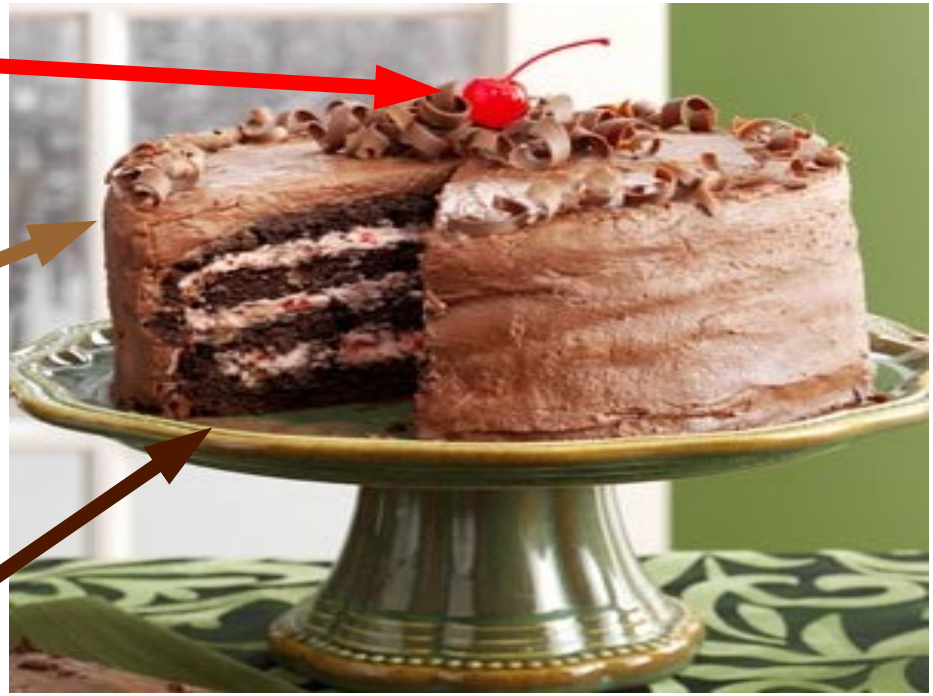
If the organism carries a `small-scale model' of external reality and of its own possible actions within its head, it is able to try out various alternatives, conclude which is the best of them, react to future situations before they arise, utilize the knowledge of past events in dealing with the present and the future, and in every way to react in a much fuller, safer, and more competent manner to the emergencies which face it (Craik, 1943,Ch. 5, p.61)

Commonsense is not just facts, it is a collection of models

Jitendra Malik

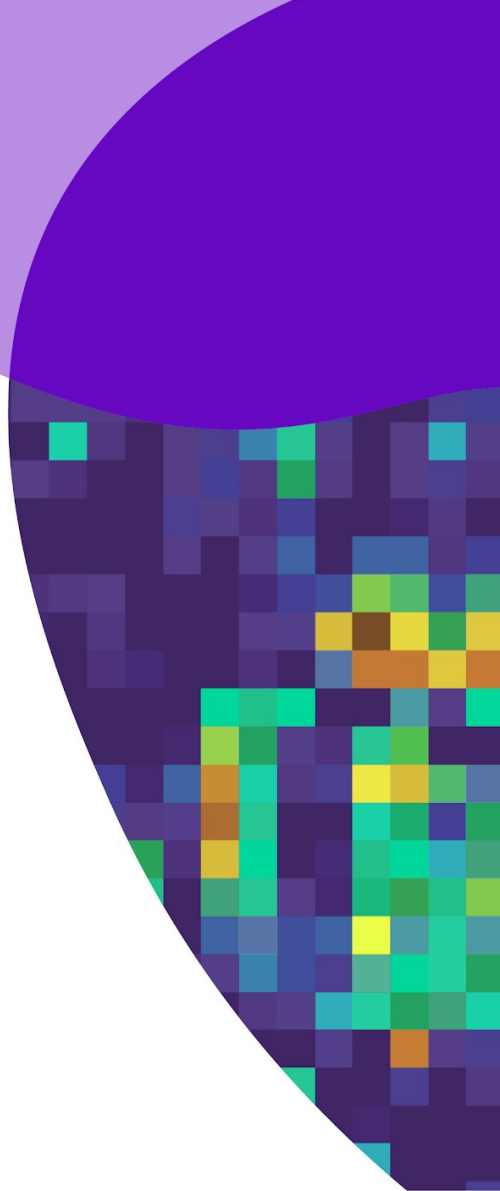
Learning Paradigms: information content per sample

- **"Pure" Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**



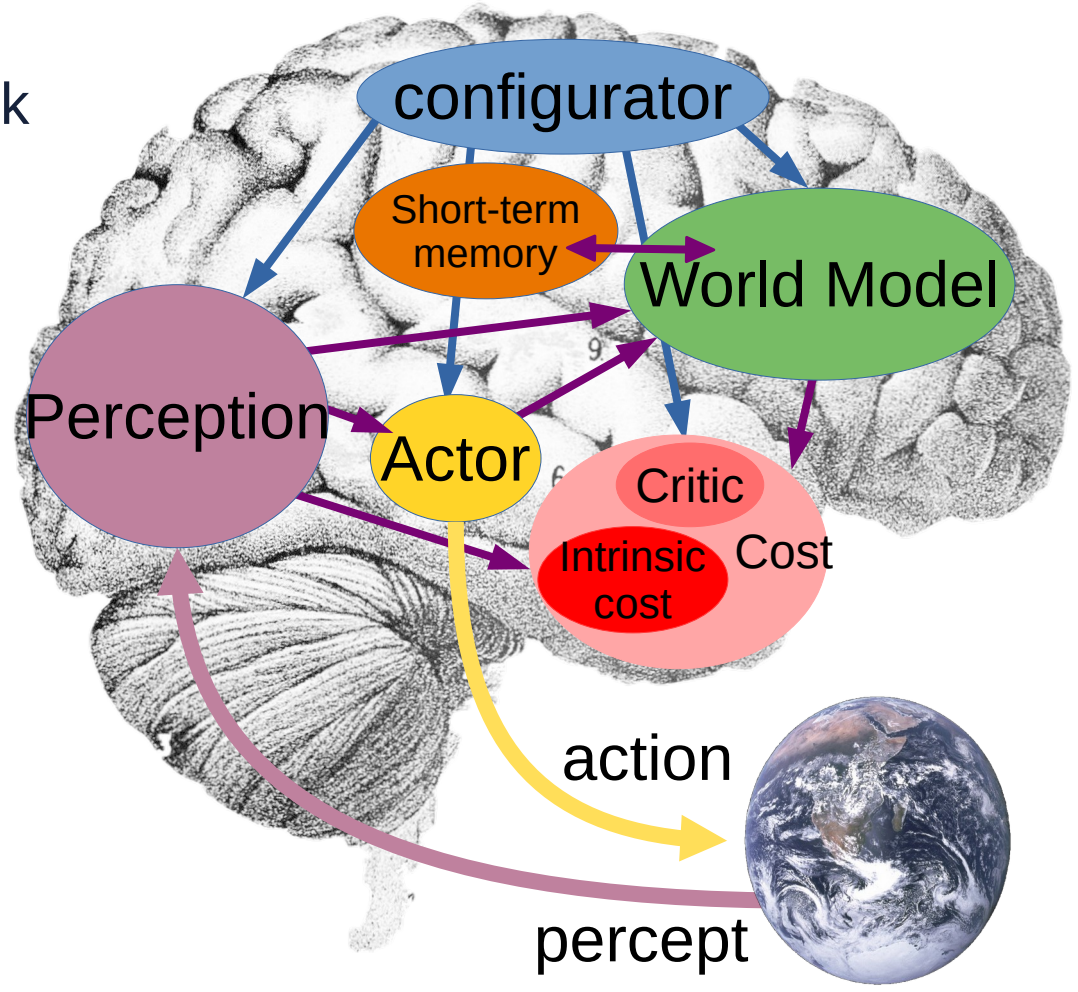
FACEBOOK AI

Architecture of Autonomous AI



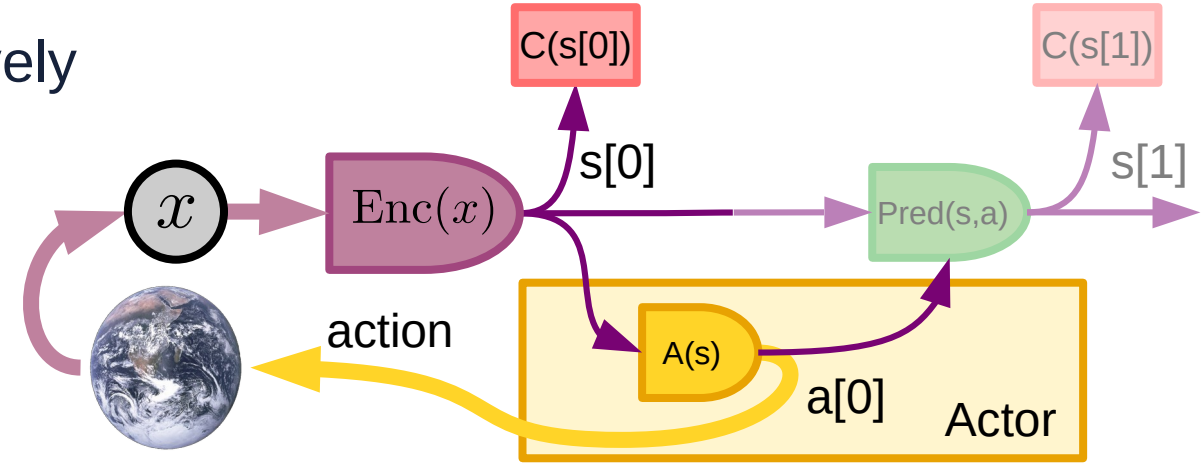
Modular Architecture for Autonomous AI

- **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



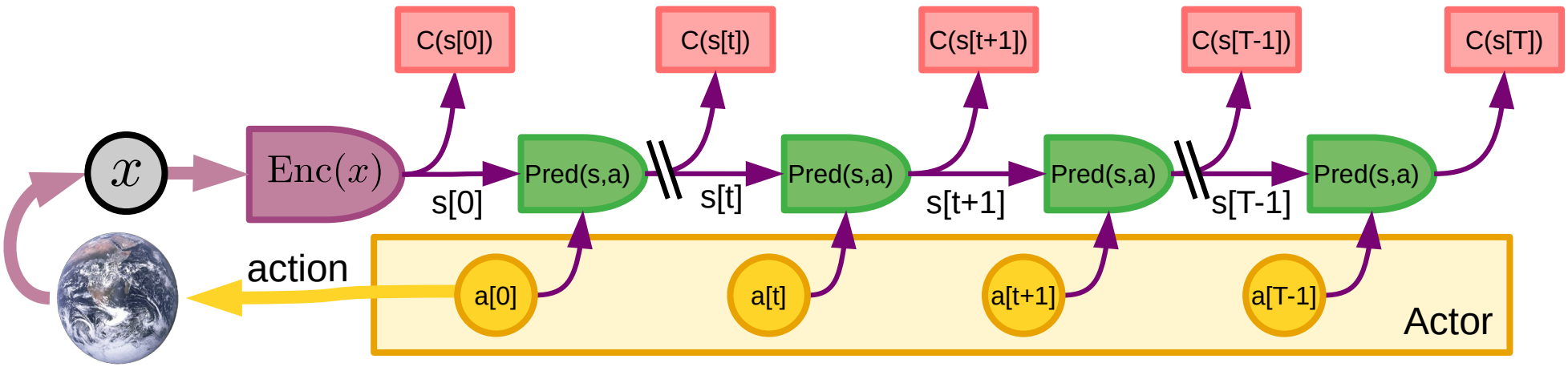
Mode-1 Perception-Action Cycle

- **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



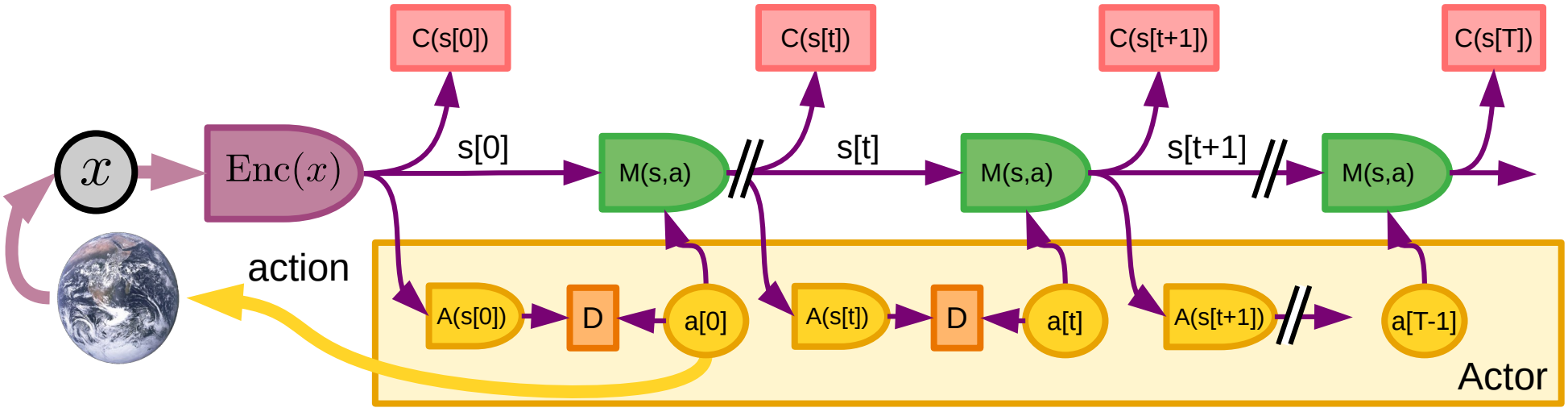
Mode-2 Perception-Planning-Action Cycle

- **Akin to 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**



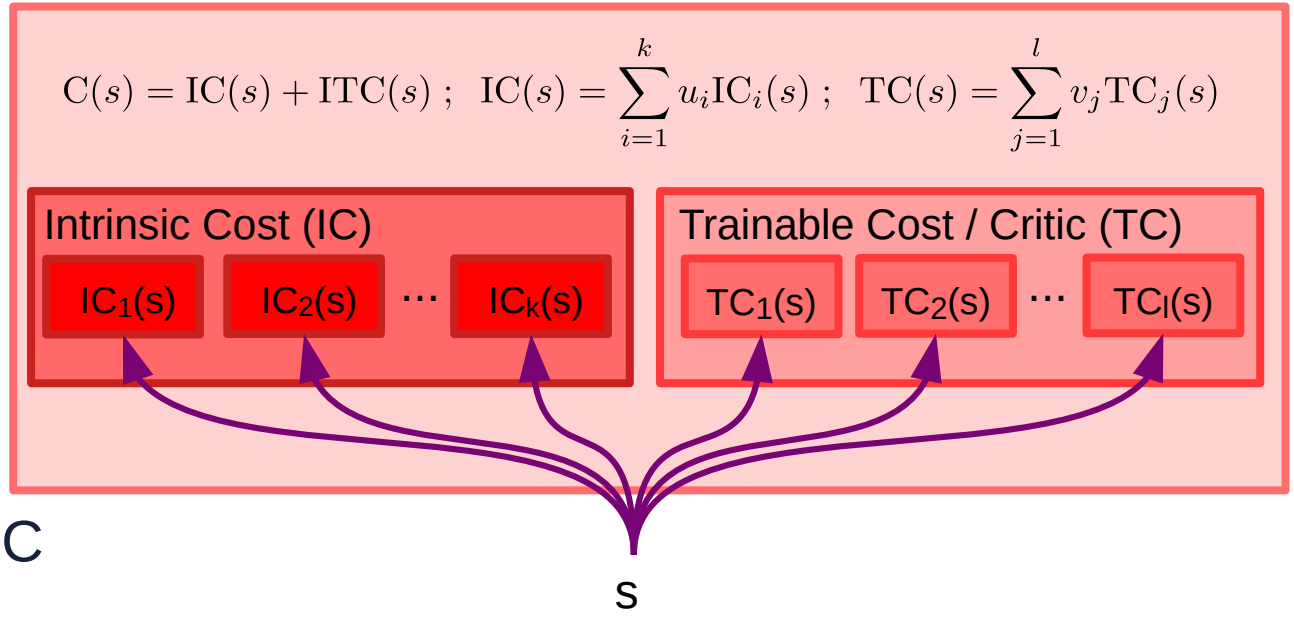
Compiling Mode-2 into Mode-1

- **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.**



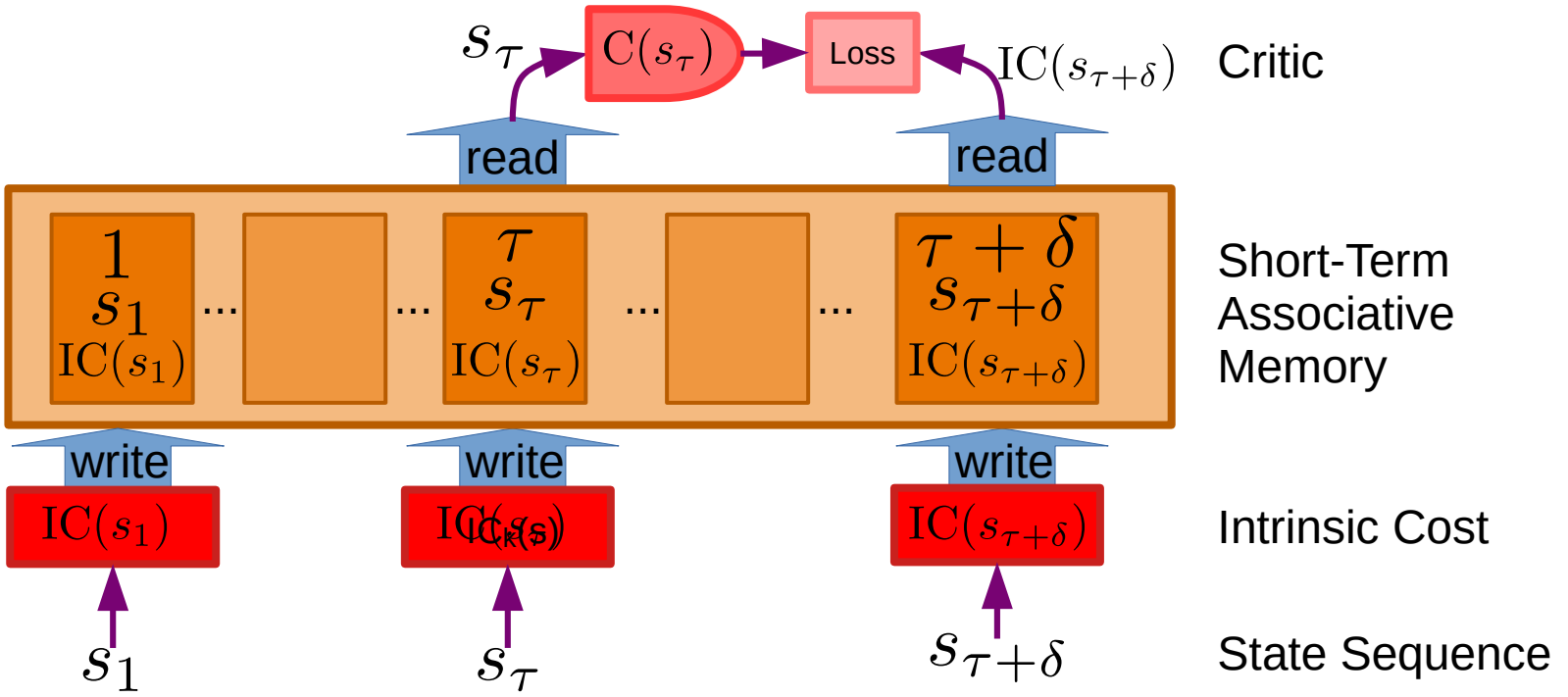
Cost Module

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



Training the Critic

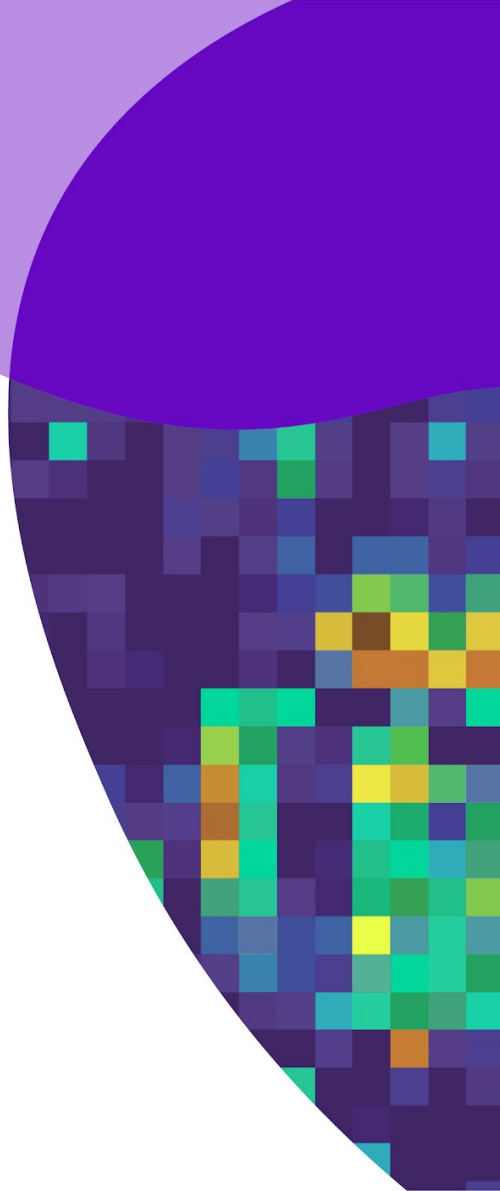
- **Critic is trained to predict future values of the intrinsic cost from the current state**
  - Uses the short term memory to produce training pairs.



FACEBOOK AI

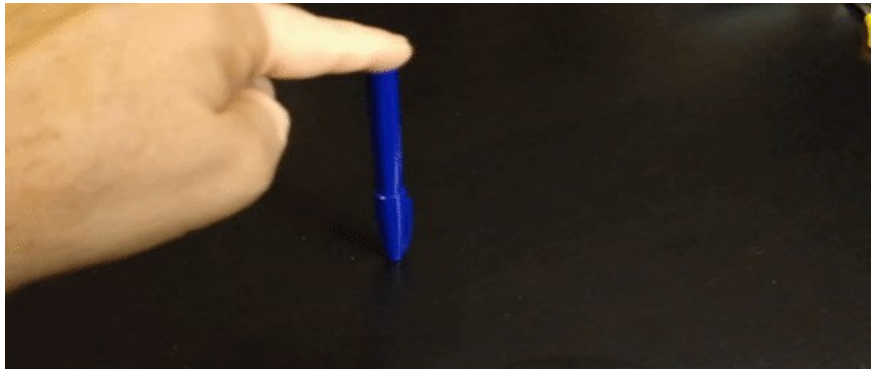
Building & Training the World Model

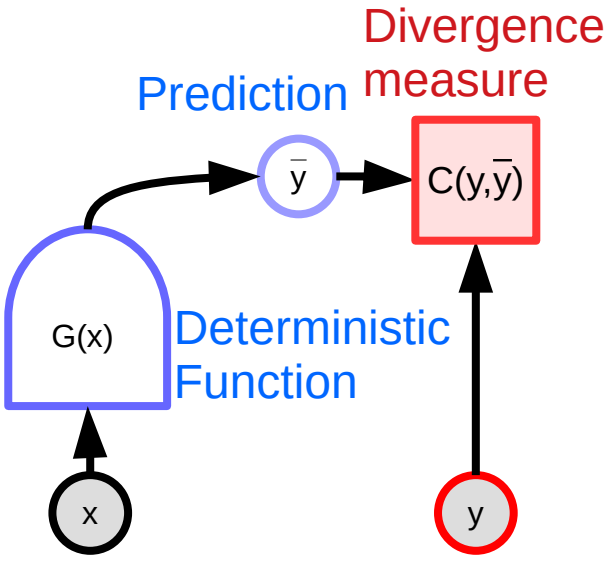
Energy-Based Models
Joint-Embedding Architecture



The world is stochastic

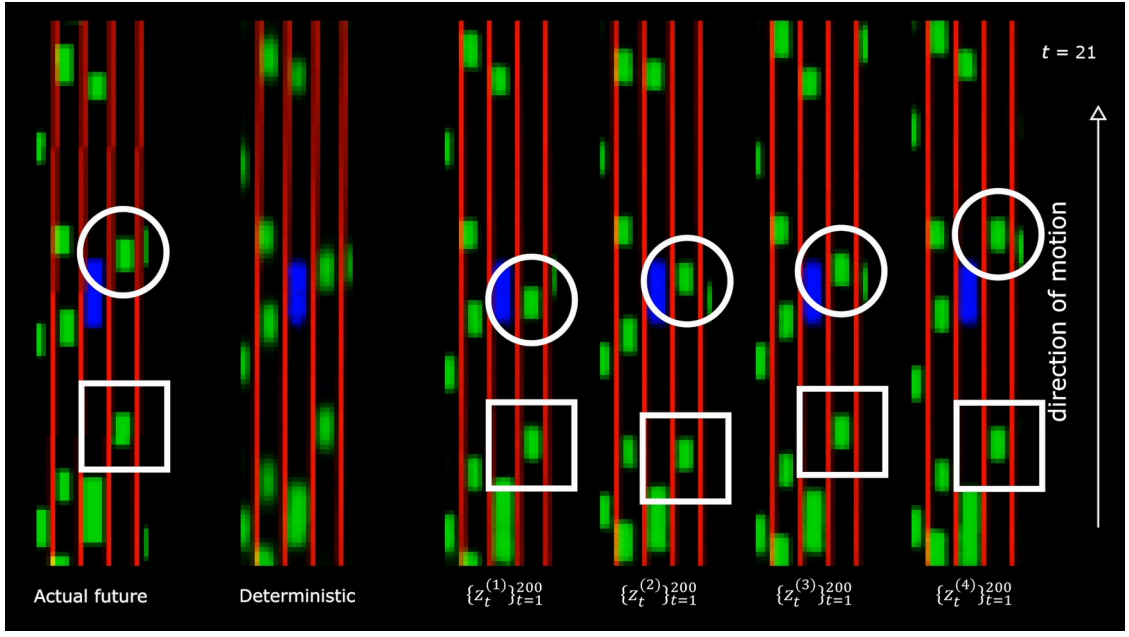
- Training a system to make a single prediction makes it predict the average of all plausible predictions
- **Blurry predictions!**





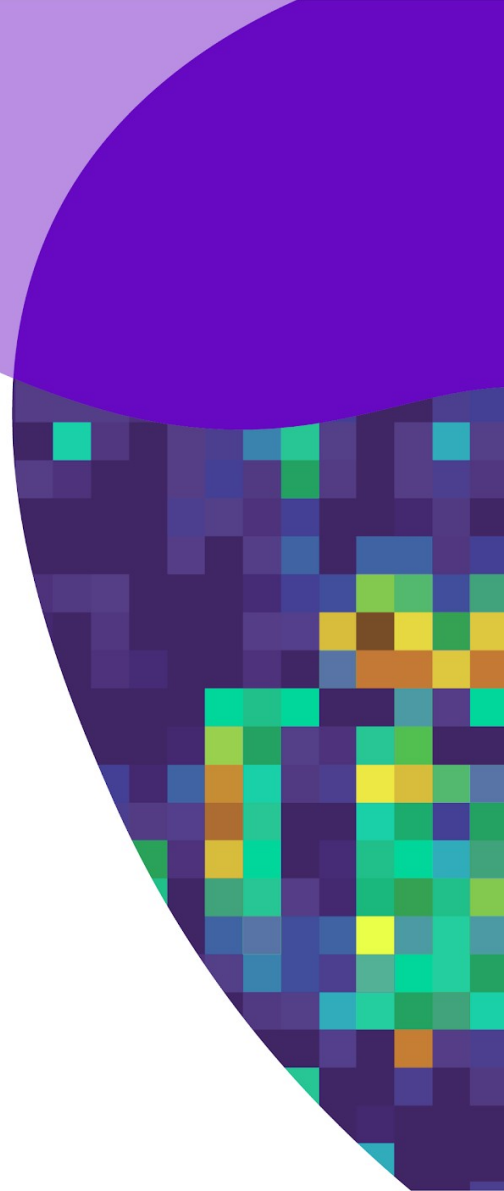
How do we represent uncertainty in the predictions?

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



FACEBOOK AI

Self-Supervised Learning

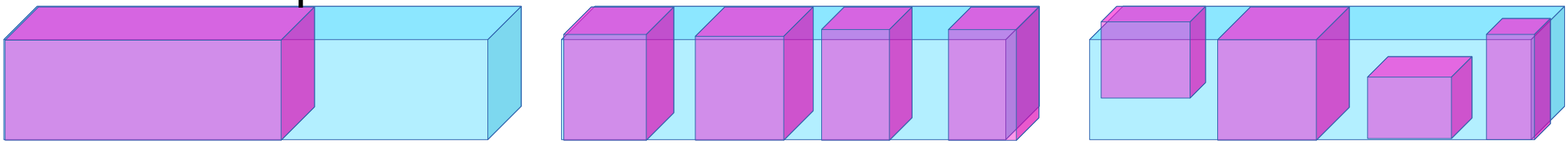


Capture dependencies between inputs.

Self-Supervised Learning = Learning to Fill in the Blanks

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

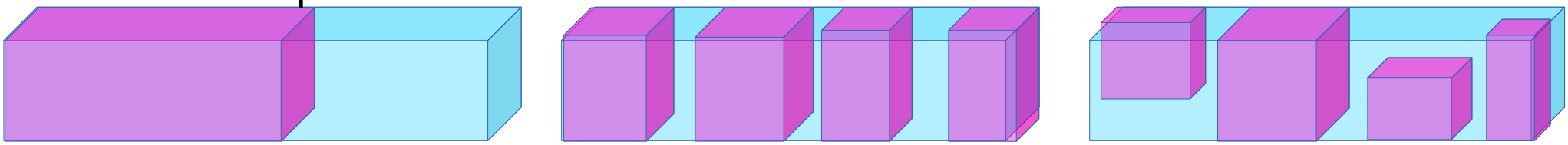
time or space →



Self-Supervised Learning = Learning to Fill in the Blanks

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

time or space →



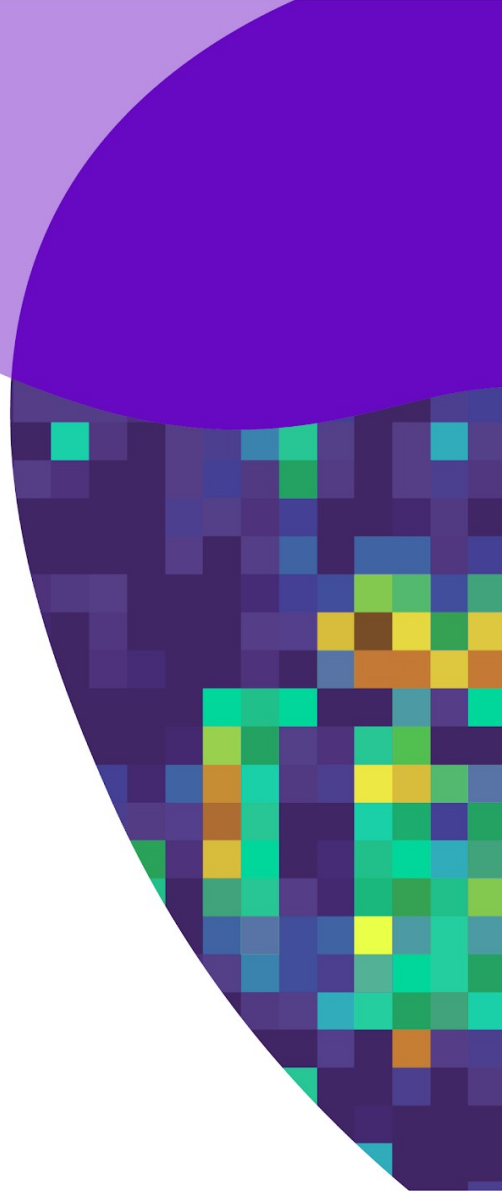
Two Uses for Self-Supervised Learning

- **1. Learning hierarchical representations of the world**
  - SSL pre-training precedes a supervised or RL phase
- **2. Learning predictive (forward) models of the world**
  - Learning models for Model-Predictive Control, policy learning for control, or model-based RL.
- **Question: how to represent uncertainty & multi-modality in the prediction?**

FACEBOOK AI

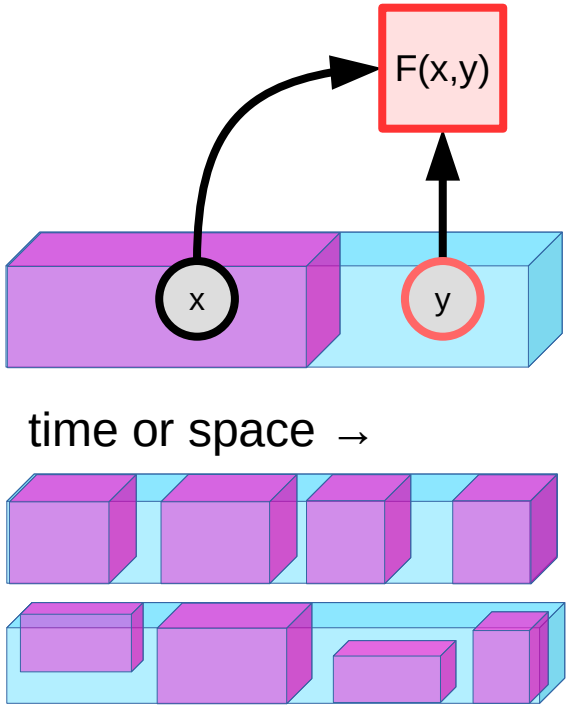
Energy-Based Models

Capture dependencies through an energy function.

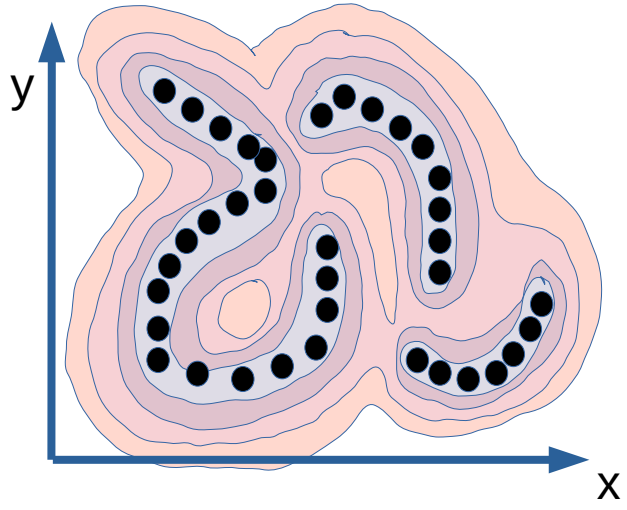


Energy-Based Models: Implicit function

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

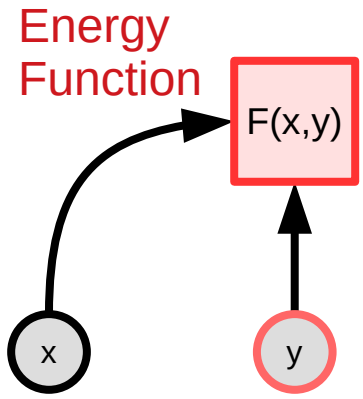


Energy
Function



EBM Inference by Minimization

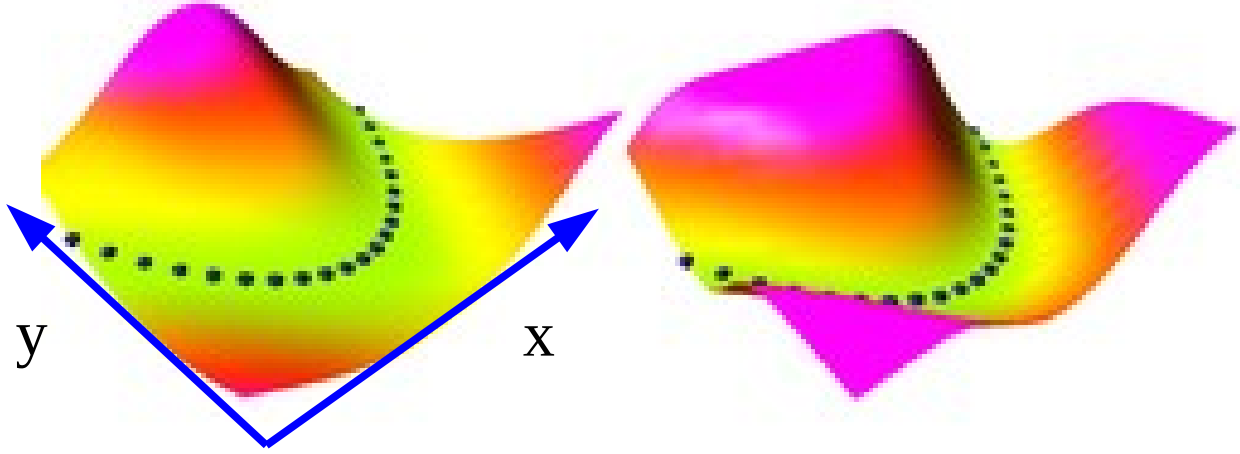
- **Energy function F(x,y) scalar-valued.**
  - Takes low values when y is compatible with x and higher values when y is less compatible with x
- **Inference: find values of y that make F(x,y) small.**
  - There may be multiple solutions



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

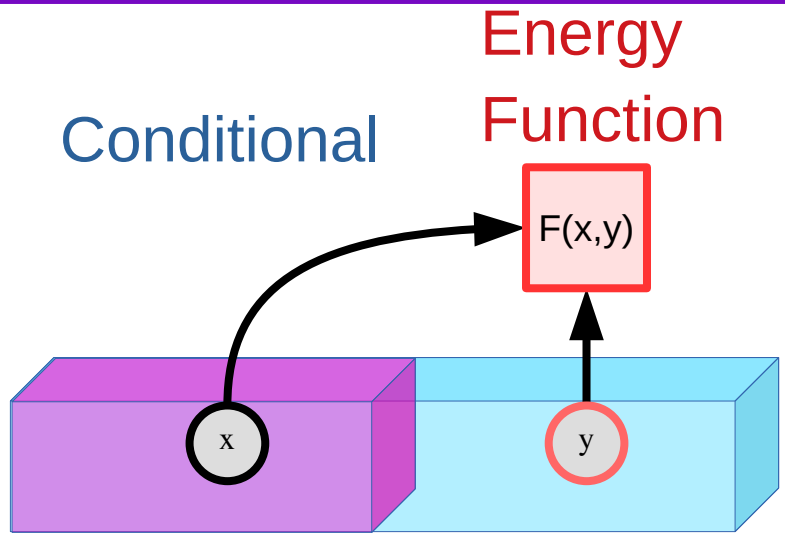
- **Note: the energy is used for inference, not for learning**

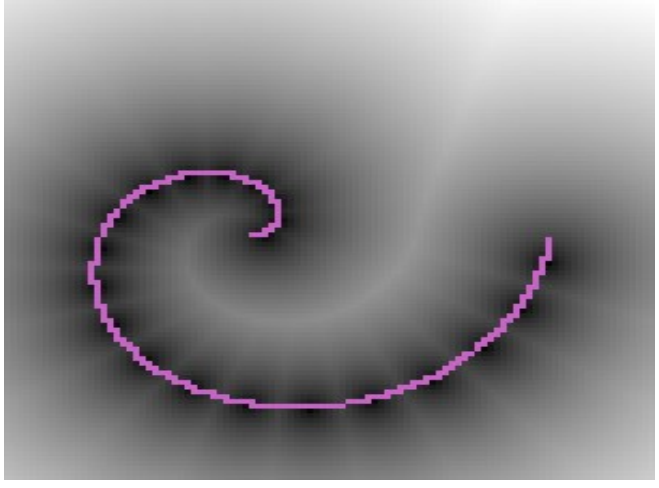
- **Example**
  - Blue dots are data points



Conditional and Unconditional Energy-Based Models

- **Conditional EBM: F(x,y)**
- **Unconditional EBM: F(y)**
  - measures the compatibility between the components of y
  - If we don't know in advance which part of y is known and which part is unknown

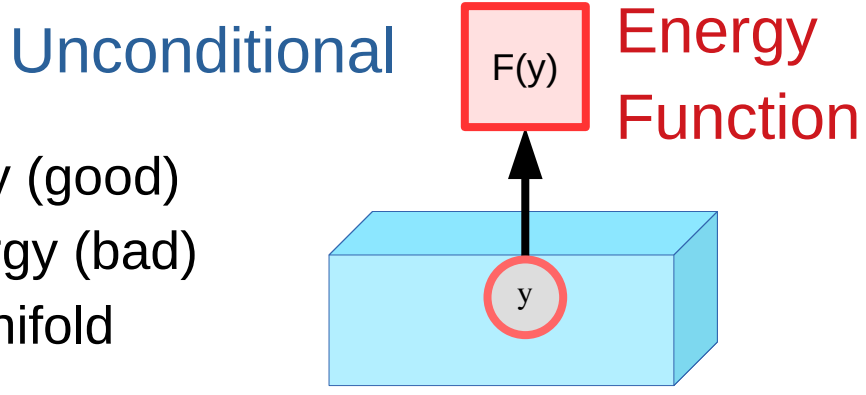




Dark = low energy (good)

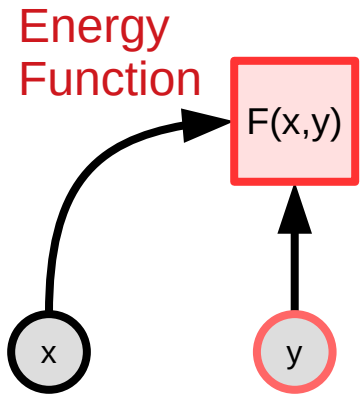
Bright = high energy (bad)

Purple = data manifold



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

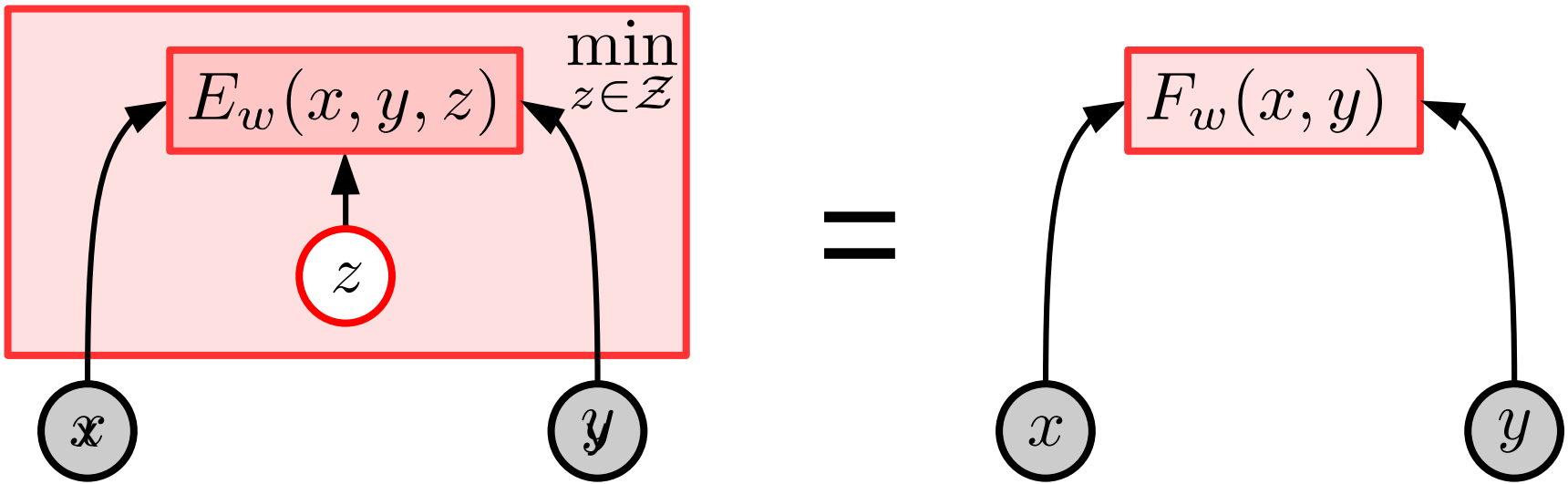


$$P(y|x) = \frac{e^{-\beta F(x,y)}}{\int_{y'} e^{-\beta F(x,y')}}$$

Latent-Variable EBM

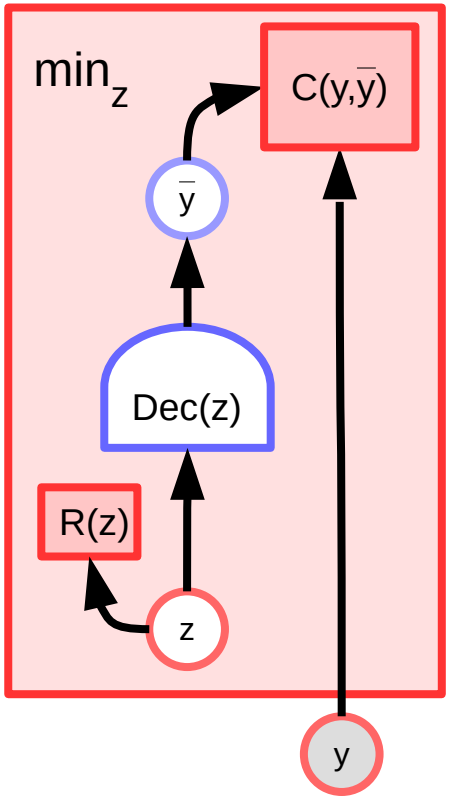
- **Latent variable z:**
  - Captures the information in y that is not available in x
  - Computed by minimization

$$\check{z} = \text{argmin}_{z \in \mathcal{Z}} E_w(x, y, z) \qquad F_w(x, y) = E_w(x, y, \check{z})$$



Example: Unconditional Latent-Variable EBM

- **Basic idea: limiting the information capacity of the representation**
- **Examples: K-Means, sparse coding**



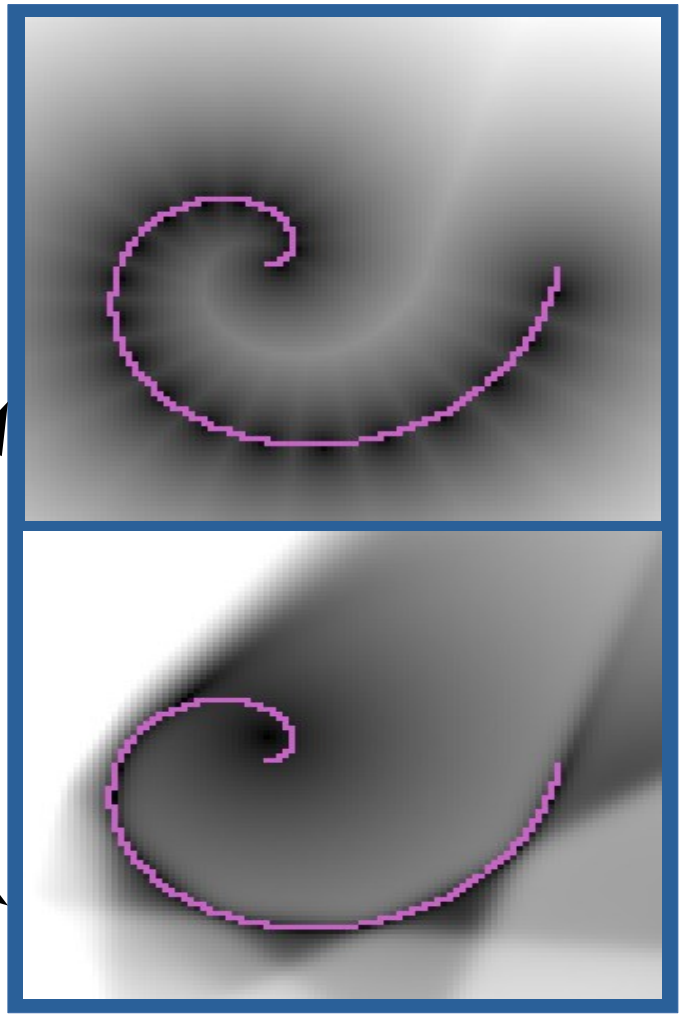
$$E(y,z) = C(y, \mathrm{Dec}(z)) + R(z)$$

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

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

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

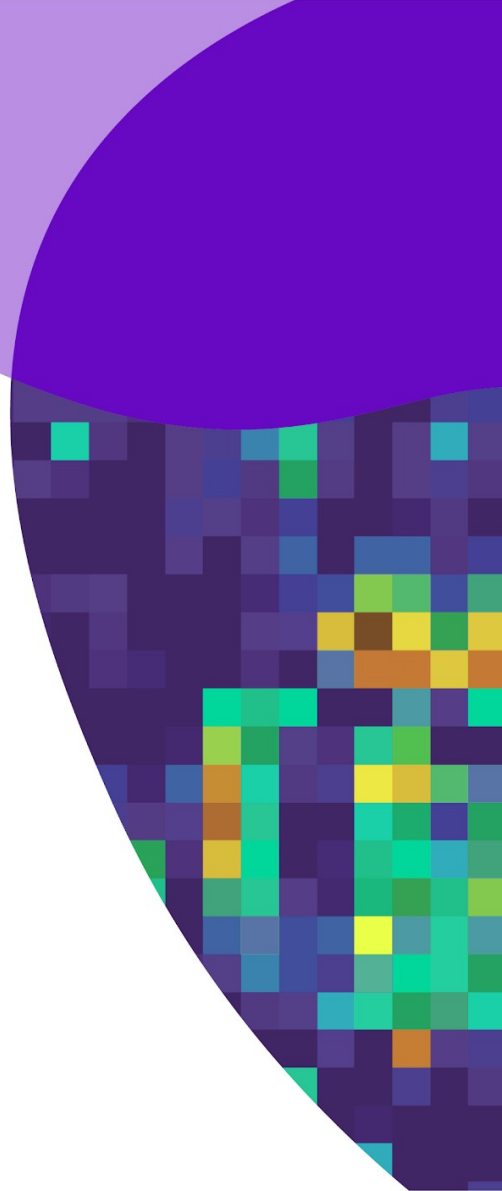
$$E(y,z) = ||y - wz||^2 + \lambda |z|_{L1|}$$



FACEBOOK AI

EBM Training

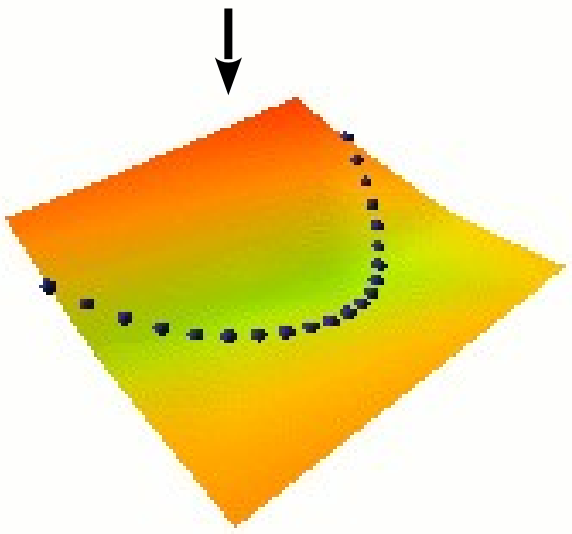
How to avoid a collapse?
1. Contrastive methods
2. Regularized & Architectural methods



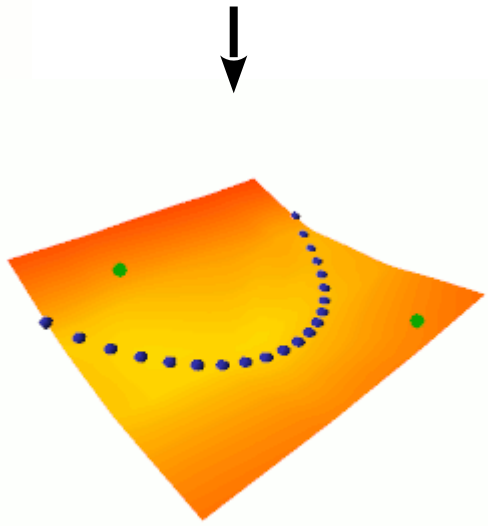
Shaping the energy surface / preventing collapse

- **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.

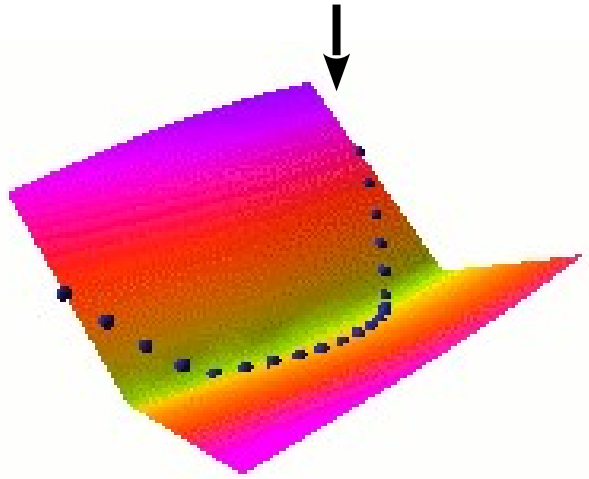
Collapse!



Contrastive Method

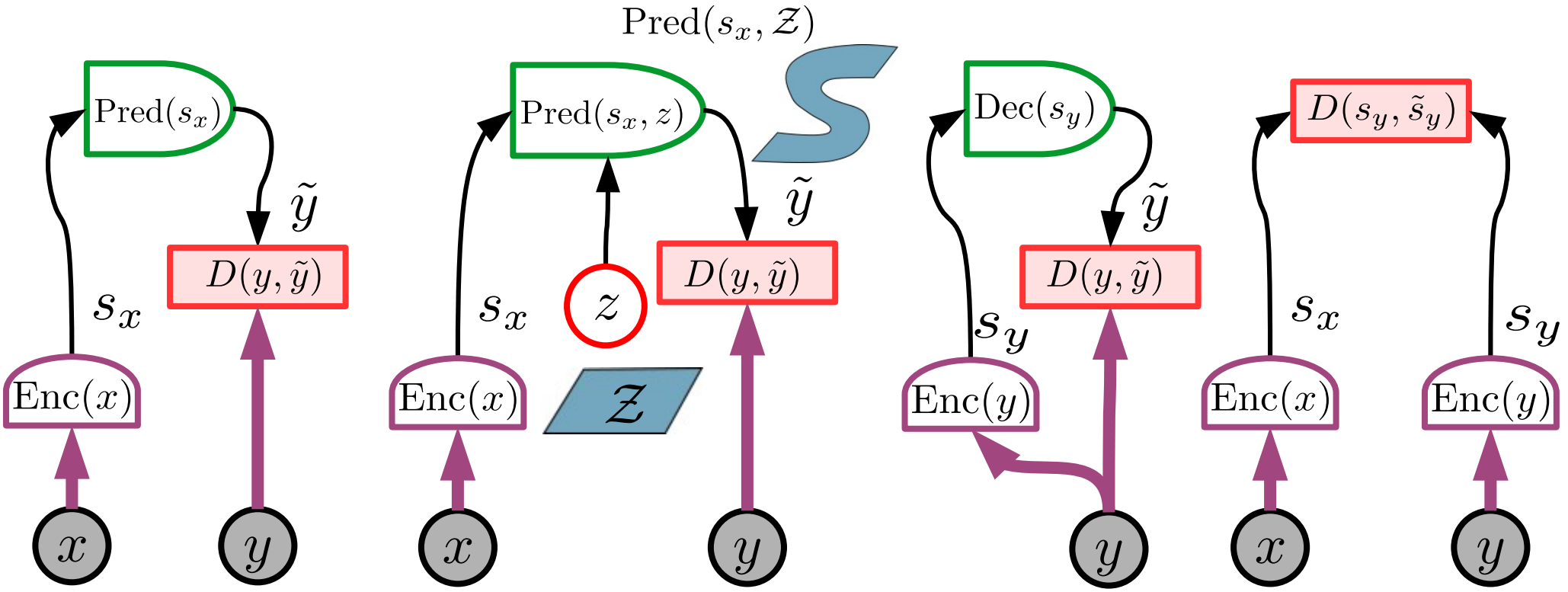


Regularized Methods



EBM Architectures

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



a) Prediction / regression
NO COLLAPSE

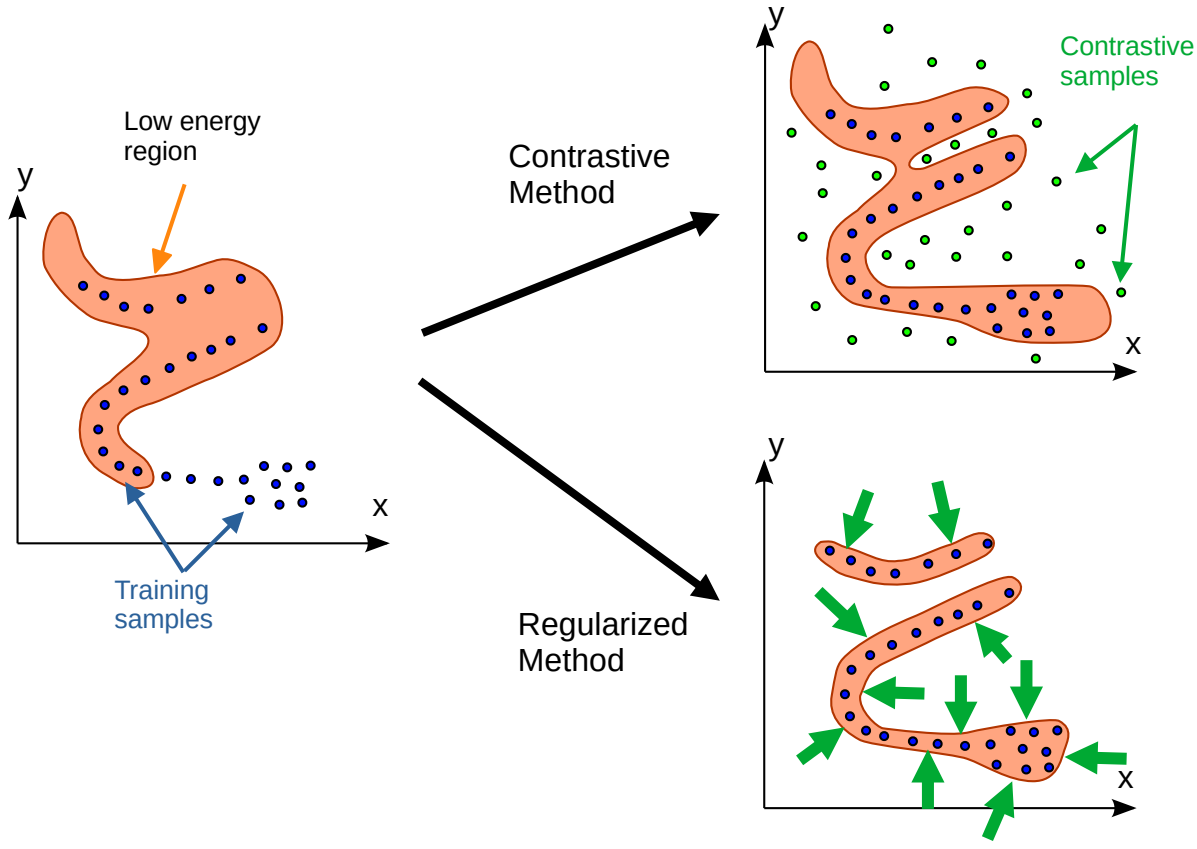
b) Generative latent-variable Architecture
CAN COLLAPSE

c) Auto-Encoder
CAN COLLAPSE

d) Joint Embedding Architecture
CAN COLLAPSE

EBM Training: two categories of methods

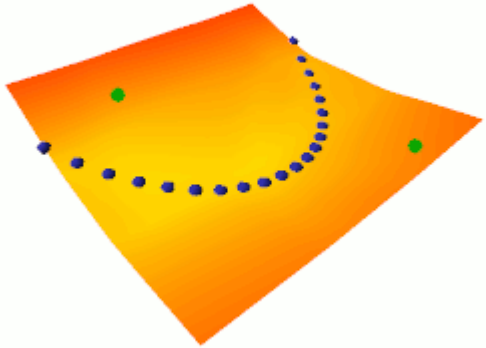
- **Contrastive methods**
  - Push down on energy of training samples
  - Pull up on energy of suitably-generated contrastive samples
  - Scales very badly with dimension
- **Regularized Methods**
  - Regularizer minimizes the volume of space that can take low energy



Contrastive methods vs Regularized Methods

- **Contrastive methods: works with any architecture**
  - Expensive in high dimension
  - Example of contrastive loss: pick a ŷ to push up.

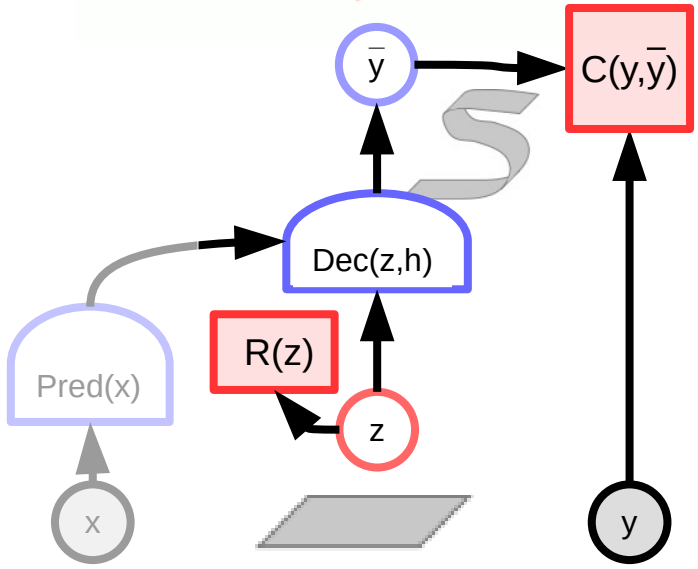
$$\mathcal{L}(x, y, \hat{y}, w) = [F_w(x, y) - F_w(x, \hat{y}) + m(y, \hat{y})]^+$$



- **Regularized methods: minimizing the volume of low-energy space**
  - E.g. by limiting the capacity of the latent

$$\mathcal{L}(x, y, w) = F_w(x, y)$$

$$F_w(x, y) = \min_z [C(\mathrm{Dec}(\mathrm{Pred}(x), z), y) + R(z)]$$



Contrastive Methods vs Regularized/Architectural Methods

- **Contrastive:** **[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:** **[Different ways tocontrol the information capacity of the latent representation]**
  - A1: build the machine so that the volume of low energy space is upper-bounded: PCA, K-means, Gaussian Mixture Model, Square ICA, normalizing flows…
  - A2: use a regularization term that minimizes the volume of space that has low energy: Sparse coding, sparse auto-encoder, LISTA, Discretized AE/VQVAE, Contracting AE, Saturating AE, Noisy AE, Variational AE, SwAV, BYOL, Barlow Twins, VICReg.
  - A4: minimize the gradient and maximize the curvature around data points: score matching

Loss function zoo for contrastive EBM training

	Method	Energy	$\hat{y}$ Generation	Loss
1	Max Likelihood	discrete y	exhaustive	$F_w(x,y) + \log \sum_{y' \in \mathcal{Y}} \exp(-F_w(x,y'))$
2	Max Likelihood	tractable	exhaustive	$F_w(x,y) + \log \int_{y' \in \mathcal{Y}} \exp(-F_w(x,y'))$
3	Max likelihood	any	MC or MCMC	$F_w(x,y) - F_w(x,\hat{y})$
4	Contr. Divergence	any	trunc'd MCMC	$F_w(x,y) - F_w(x,\hat{y})$
5	Pairwise Hinge	any	most offending	$[F_w(x,y) - F_w(x,\hat{y}) + m(y,\hat{y})]^+$
6	Min-Hinge	positive	most offending	$F_w(x,y) + [m(y,\hat{y}) - F_w(x,\hat{y})]^+$
6	Square-Hinge	divergence	most offending	$F_w(x,y)^2 + \left([m(y,\hat{y}) - F_w(x,\hat{y})]^+\right)^2$
7	Square-Exp	any	most offending	$F_w(x,y)^2 + \exp(-\beta F_w(x,\hat{y}))$
8	Logistic	any	most offending	$\log(1 + \exp(F_w(x,y) - F_w(x,\hat{y}))$
9	GAN	any	$\hat{y} = g_u(z)$	$H(F_w(x,y), F_w(x,\hat{y}), m(y,\hat{y}))$
10	Denoising AE	$D(y, g_w(y))$	$\hat{y} = N(y)$	$D(y, g_w(\hat{y})$

Contrastive Methods: group losses

- **Push down on a group of data points, push up on a group of contrastive points**
  - General group loss on p^+ data points and p^- contrastive points:

$$\mathcal{L}(x_1 \dots x_{p^+}, y_1 \dots y_{p^+}, \hat{y}_1 \dots \hat{y}_{p^-}, w) = H\left(F(x_1, y_1), \dots F(x_{p^+}, y_{p^+}), F(x_1, \hat{y_1}), \dots F(x_{p^+}, \hat{y_{p^+}}), M(Y_{1\dots p^+}, \hat{Y}_{1\dots p^-})\right)$$

    - Where H must be an increasing fn of the data energies and decreasing fn of the contrastive point energies within the margin.
    - M is a margin matrix for all pairs of y and ŷ in the group.
  - Example: Neighborhood Component Analysis, Noise Contrastive Estimation, InfoNCE (implicit infinite margin) [Goldberger 2005] [Gutmann 2010]...[Misra 2019] [Chen 2020]

$$\mathcal{L}(x, y, \hat{y}_1, \dots \hat{y}_{p^-}, w) = -\log \frac{e^{-F_w(x,y)}}{e^{-F_w(x,y)} + \sum_{i=1}^{p^-} e^{-F_w(x,\hat{y}_i,w)}}$$

Max Likelihood is (generally) a (bad) Contrastive Method

- **Push down on data points,**
- **Push up on all points**
- **Max likelihood / probabilistic models**

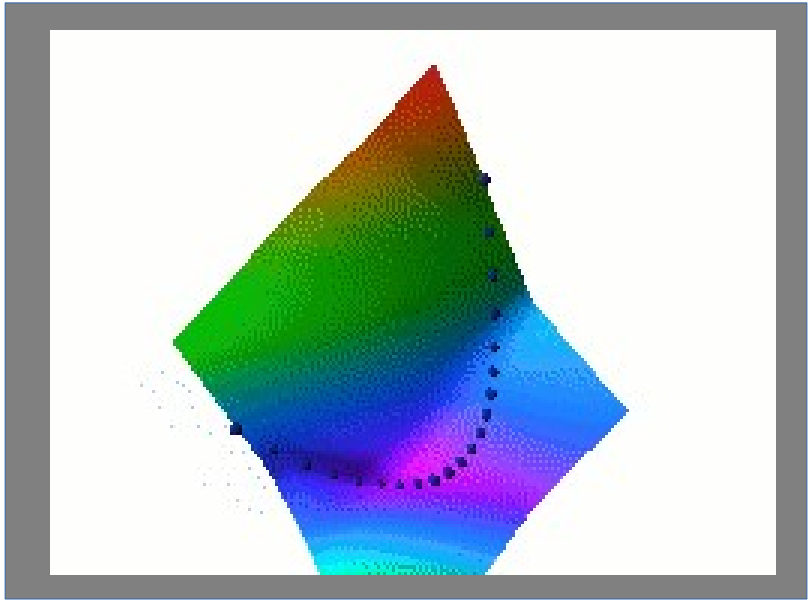
$$P_w(y|x) = \frac{e^{-\beta F_w(x,y)}}{\int_{y'} e^{-\beta F_w(x,y')}}$$

- Loss: $$\mathcal{L}(x,y,w) = F_w(x,y) + \frac{1}{\beta}\log\int_{y'} e^{-\beta F_w(x,y')}$$
- Gradient: $$\frac{\partial \mathcal{L}(x,y,w)}{\partial w} = \frac{\partial F_w(x,y)}{\partial w} - \int_{y'} P_w(y'|x)\frac{\partial F_w(x,y')}{\partial w}$$
- 2nd term is intractable: MC/MCMC/HMC/CD: $\hat{y}$ sampled from $P_w(y|x)$

$$\frac{\partial \mathcal{L}(x,y,w)}{\partial w} = \frac{\partial F_w(x,y)}{\partial w} - \frac{\partial F_w(x,\hat{y})}{\partial w}$$

Problem with Max Likelihood / Probabilistic Methods

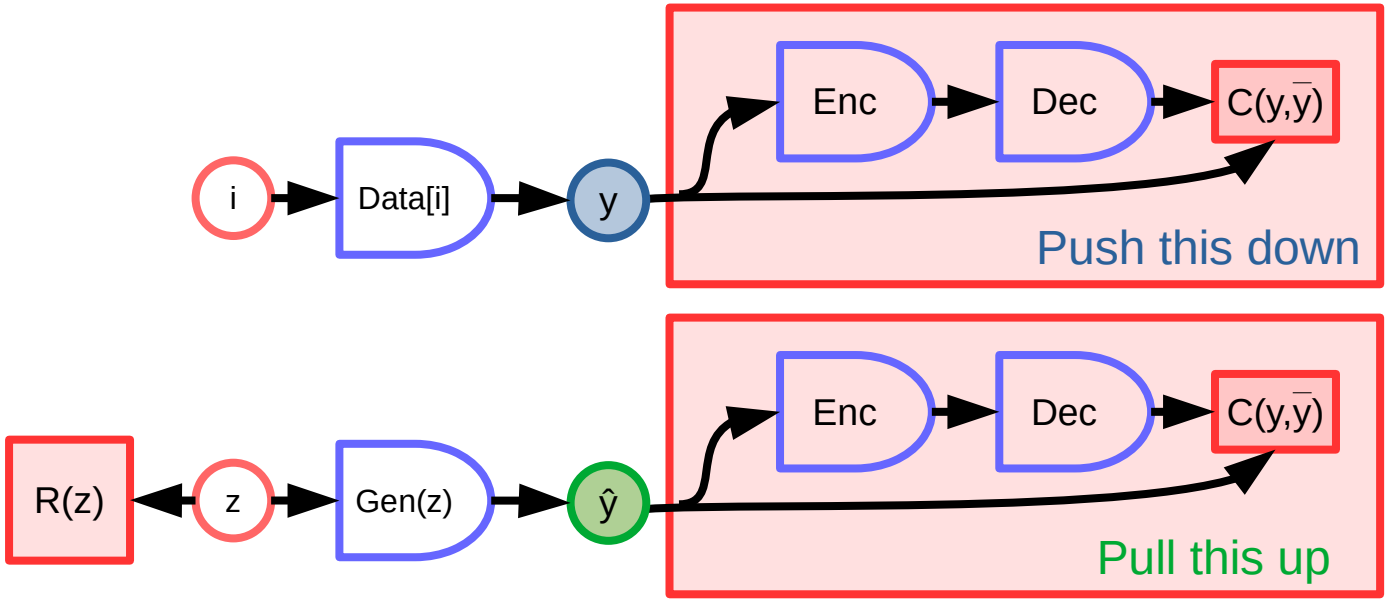
- **It wants to make the difference between the energy on the data manifold and the energy just outside of it infinitely large!**
- **It wants to make the data manifold an infinitely deep and infinitely narrow canyon.**
- **The loss must be regularized to keep the energy smooth**
  - e.g. à la Wassertstein GAN.
  - So that gradient-based inference works
  - Equivalent to a prior
  - But then, why use a probabilistic model?

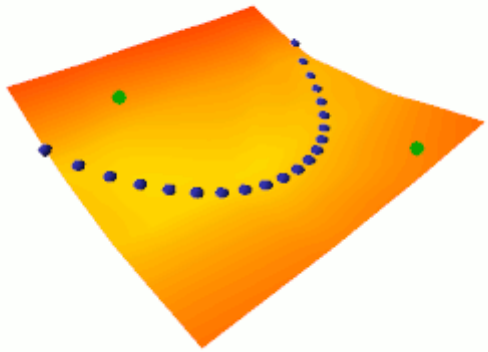


GAN is secretly a contrastive method for EBM

- **Energy-Based GAN** [Zhao 2016]**, Wasserstein GAN** [Arjovsky 2017]**,...**
  - GANs generate nice images
  - But GANs have not been successful for learning representations of images

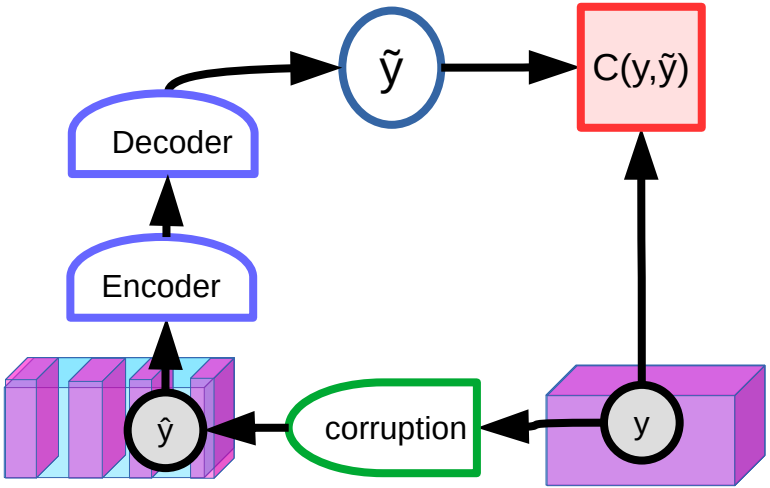
$$\mathcal{L}(x, y, \hat{y}, w) = H(F_w(x, y), F_w(x, \hat{y}), m(y, \hat{y}))$$





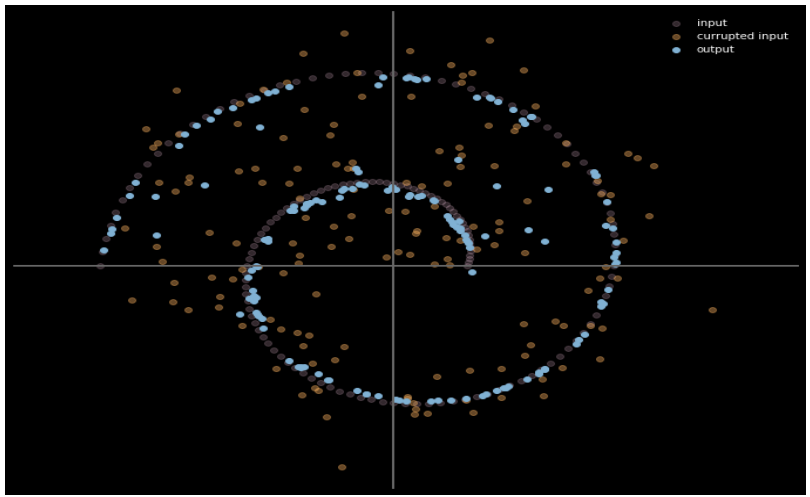
Denoising AE / Masked AE are contrastive methods

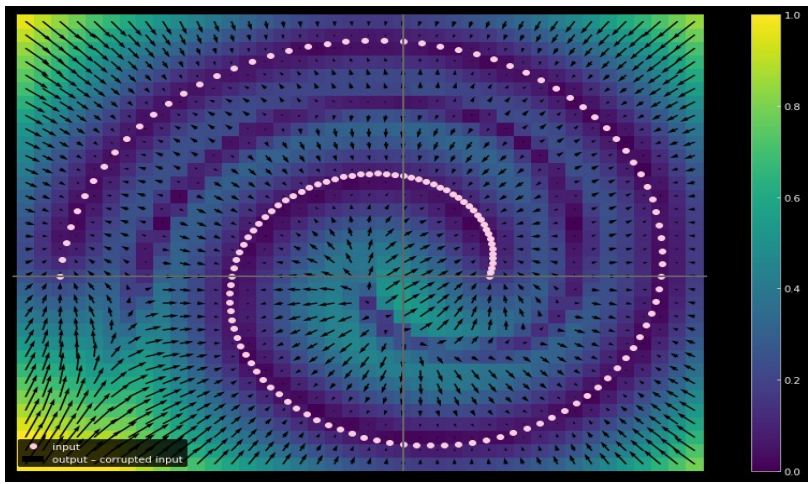
- **Denoising AE** [Vincent 2008]
- **Contrastive method for NLP**
  - [Collobert-Weston 2011]
- **Masked AE: Learning text representations**
  - BERT [Devlin 2018], RoBERTa [Ott 2019]



This is a [...] of text extracted [...] a large set of [...] articles

This is a piece of text extracted from a large set of news articles





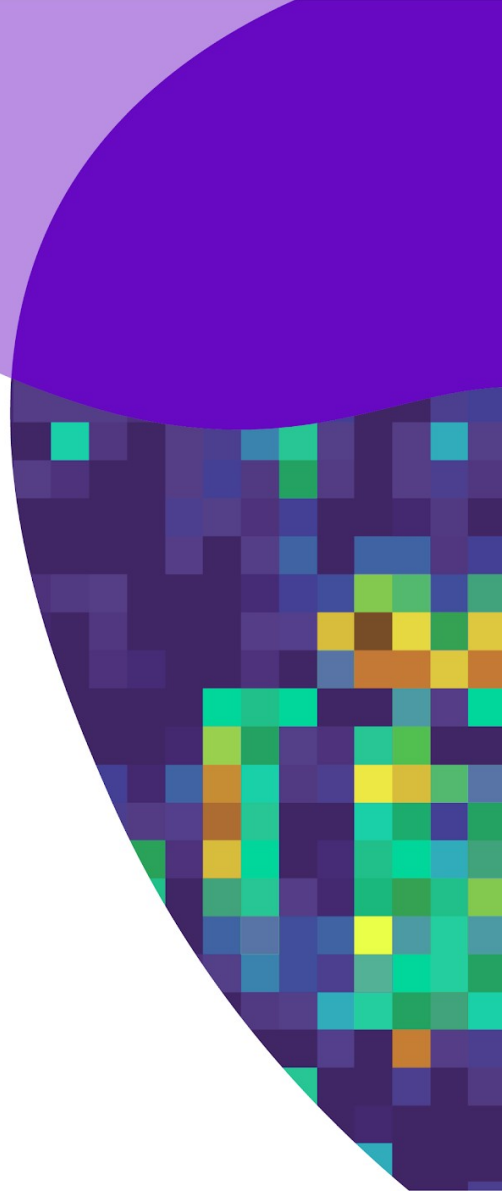
Figures: Alfredo Canziani

FACEBOOK AI

EBM Architectures

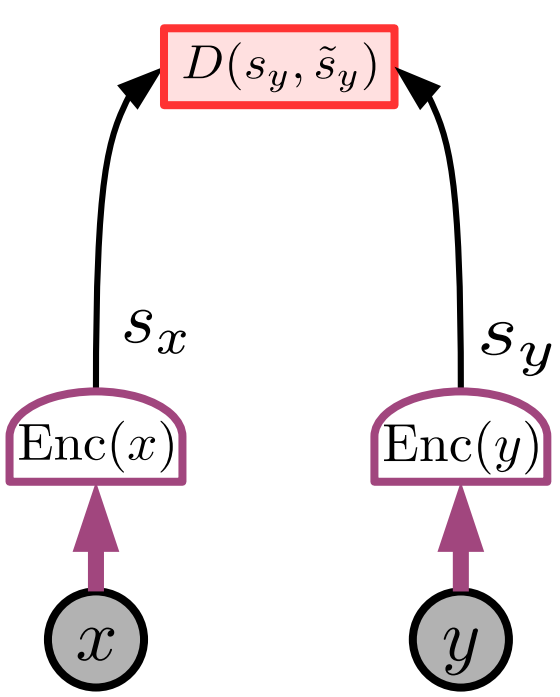
that can handle multimodality

1. Latent variable models
2. Joint embedding architectures

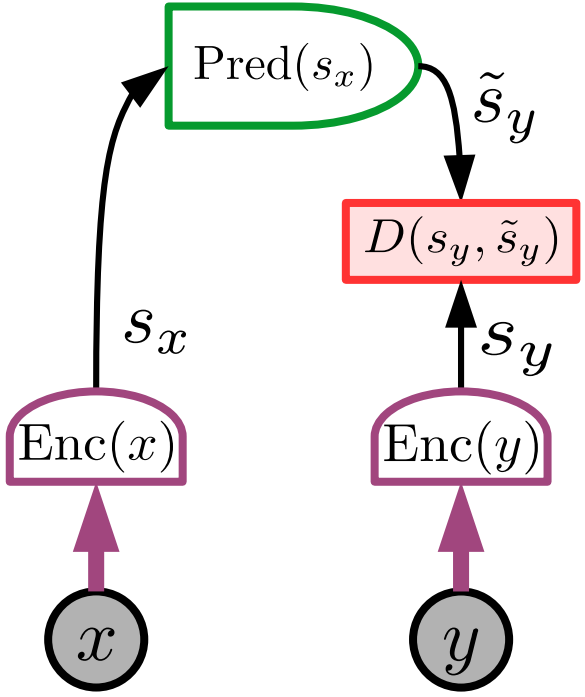


Joint Embedding Architectures

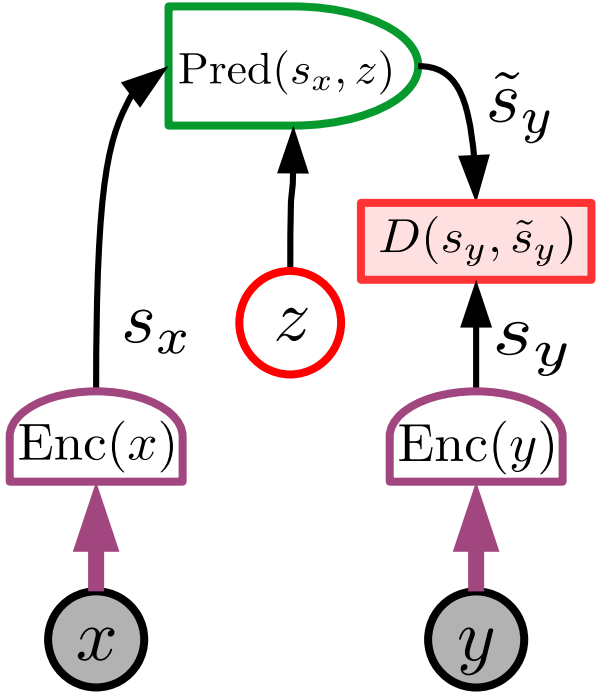
- **Computes abstract representations for x and y**
- **Tries to make them equal or predictable from each other.**



a) Joint Embedding Architecture (JEA)



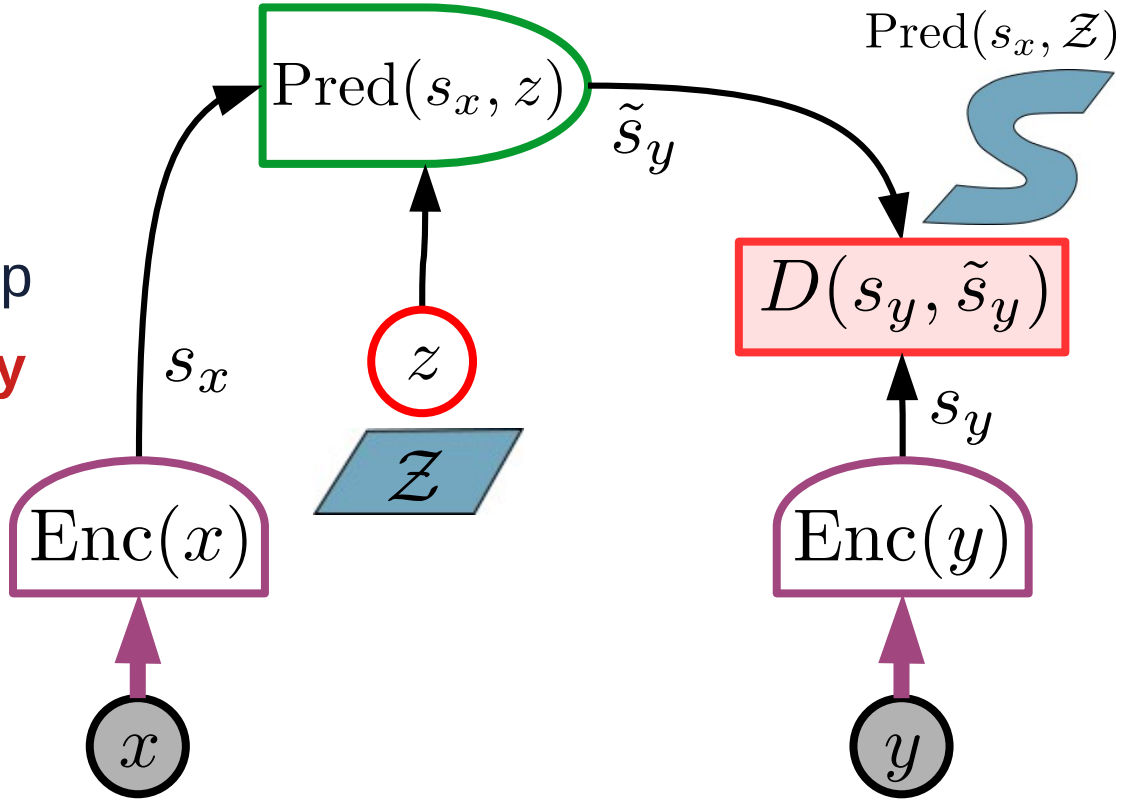
b) Deterministic Joint Embedding Predictive Architecture (DJEPA)



c) Joint Embedding Predictive Architecture (JEPA)

Joint Embedding Predictive Architecture (JEPA)

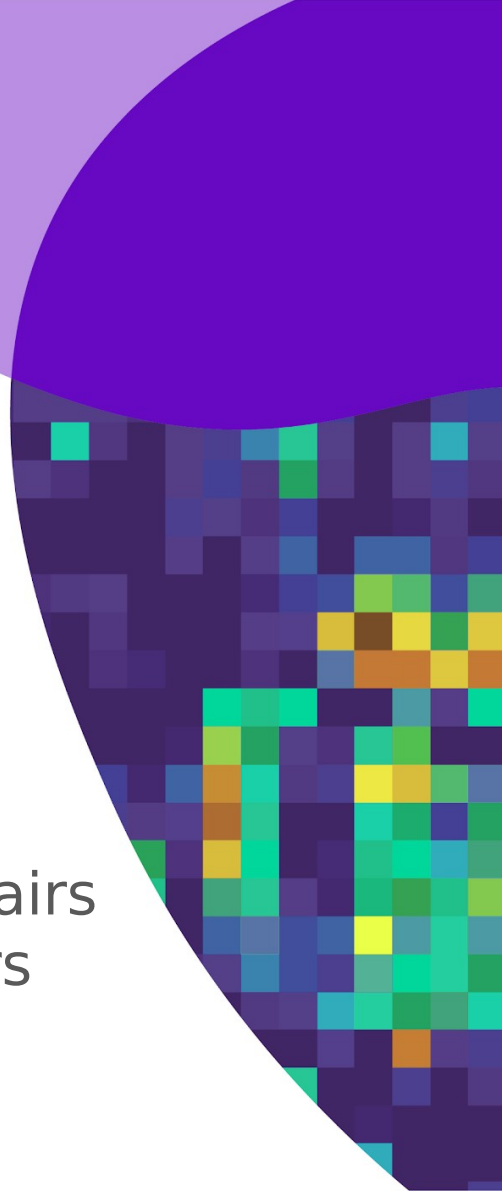
- **Computes abstract representations for x and y**
- **Makes predictions in representation space**
  - Can use a latent variable to help
- **Does not need to predict every details of y**
  - Enc(y) can eliminate irrelevant details through invariances
- **Tries to make the representations predictable from each other.**



FACEBOOK AI

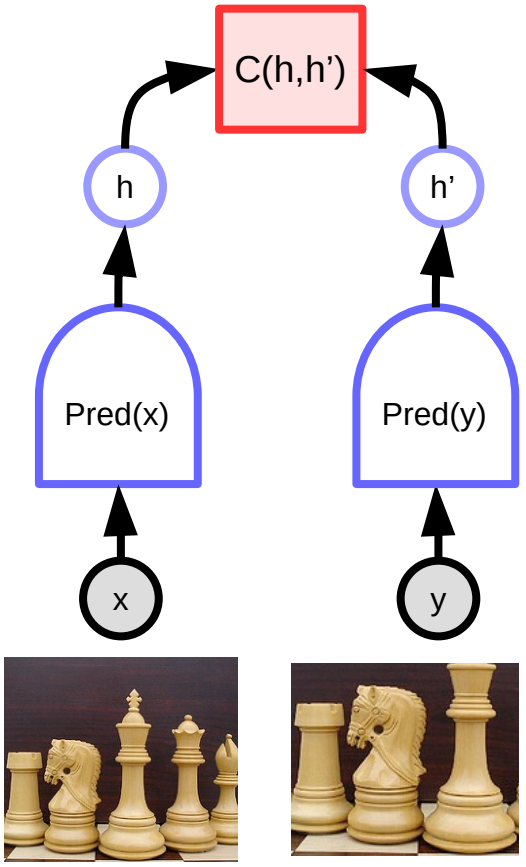
Contrastive Methods for joint embedding architectures

Push down on the energy of compatible sample pairs
Pull up on the energy of incompatible sample pairs

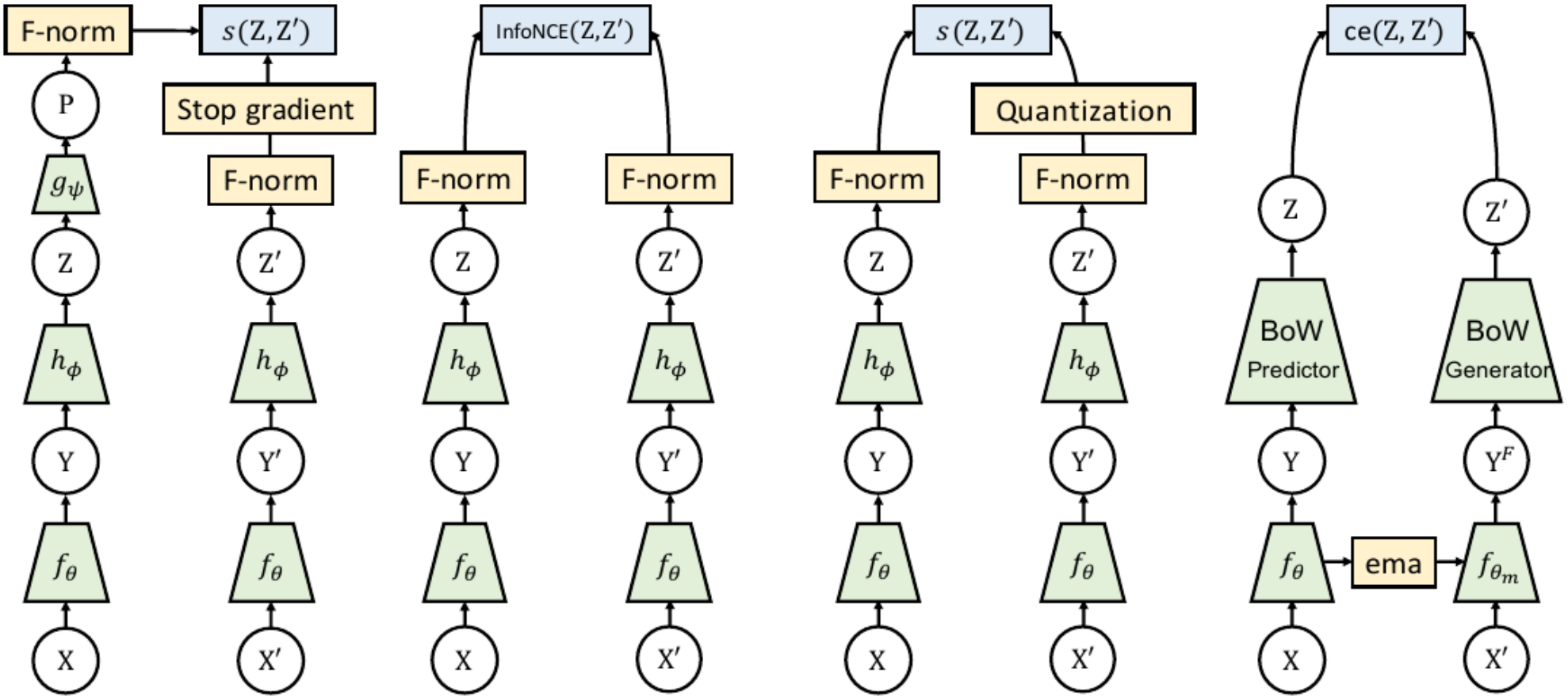


Joint Embedding Architectures

- **Distance measured in feature space**
- **Multiple “predictions” through feature invariance**
- **Siamese nets, metric learning**
  - [Bromley NIPS’93] [Chopra CVPR’05] [Hadsell CVPR’06]
- **Advantage: no pixel-level reconstruction**
- **Difficulty: <in a few slides>**
- **Many successful examples for image recognition:**
  - DeepFace [Taigman et al. CVPR 2014]
  - PIRL [Misra et al. Arxiv:1912.01991]
  - MoCo [He et al. Arxiv:1911.05722]
  - SimCLR [Chen et al. Arxiv:2002.05709]
  - …..



Contrastive Joint Embedding Methods



(e) SimSiam (f) SimCLR (g) SwAV (h) OBoW

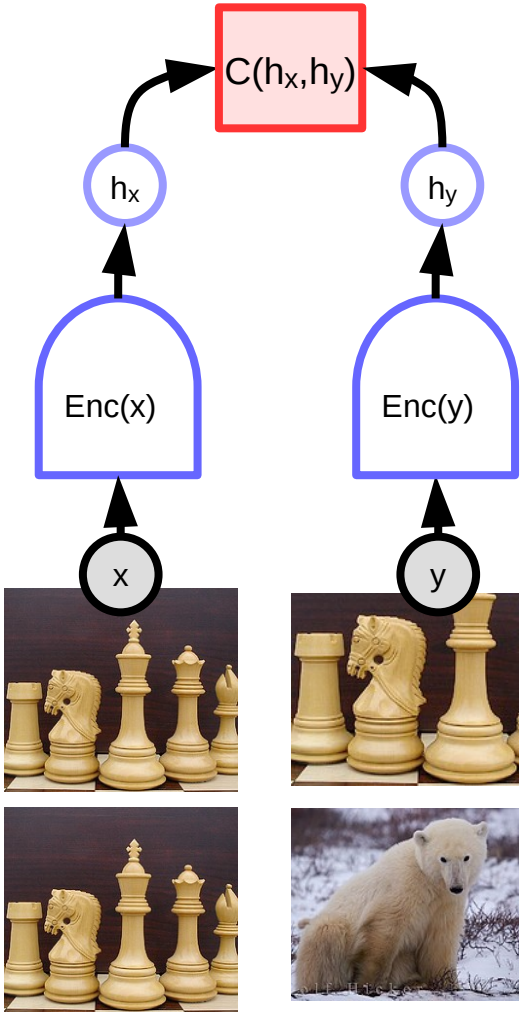
Contrastive Joint Embedding

$$F(x, y) = C(\text{Enc}(x), \text{Enc}(y))$$

- **Siamese nets, metric learning**
- **Two identical networks with shared weights**
  - Signature verification: [Bromley NIPS'93],
  - Face verification [Chopra CVPR'05]
  - Face reco, DeepFace [Taigman et al. CVPR 2014]
  - Video feature learning [Taylor CVPR 2011]
  - Use square-square or square-exp loss

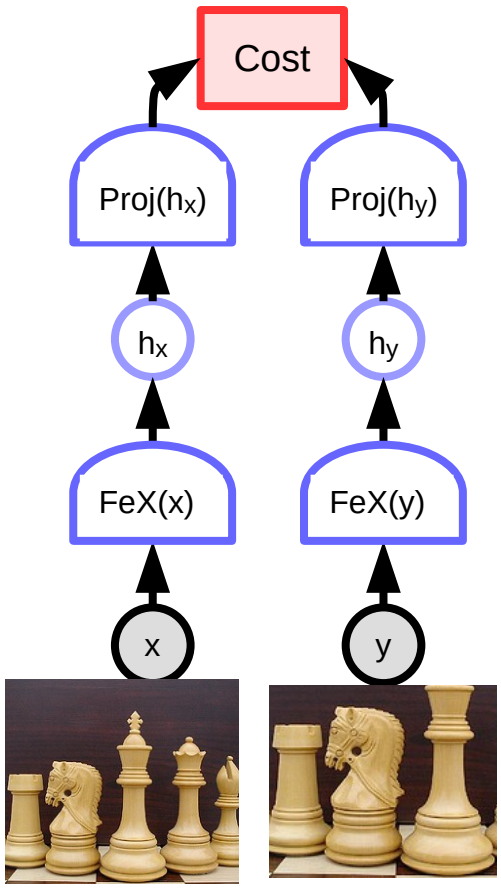
$$\mathcal{L}(x, y, \hat{y}, w) = \left([F_w(x, y)]^+\right)^2 + \left([m - F_w(x, \hat{y})]^+\right)^2$$

- **Advantages:**
  - no pixel-level reconstruction
  - Learns a similarity metric
  - Multimodality through encoder invariance

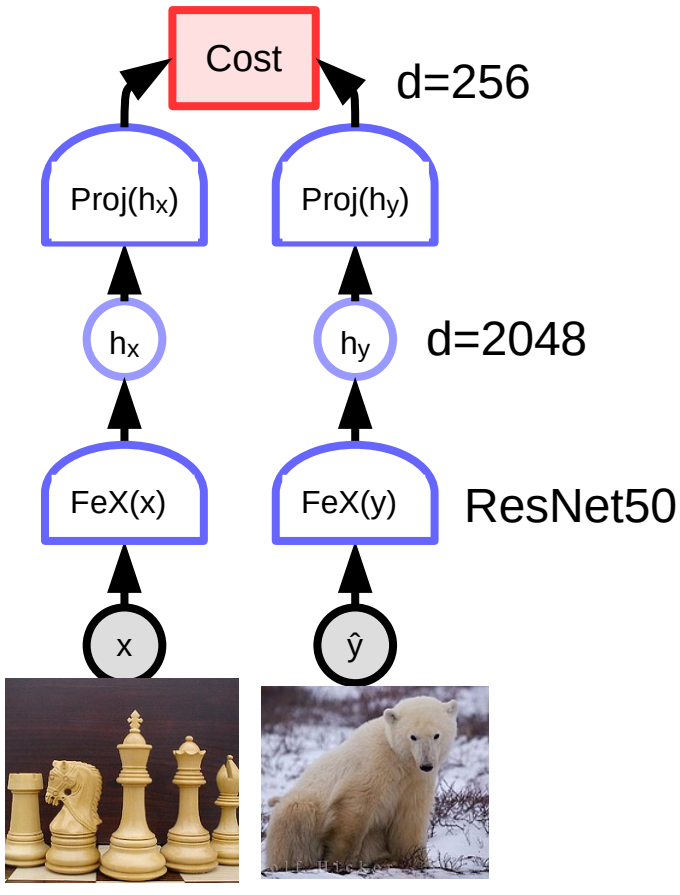


Contrastive Joint Embedding

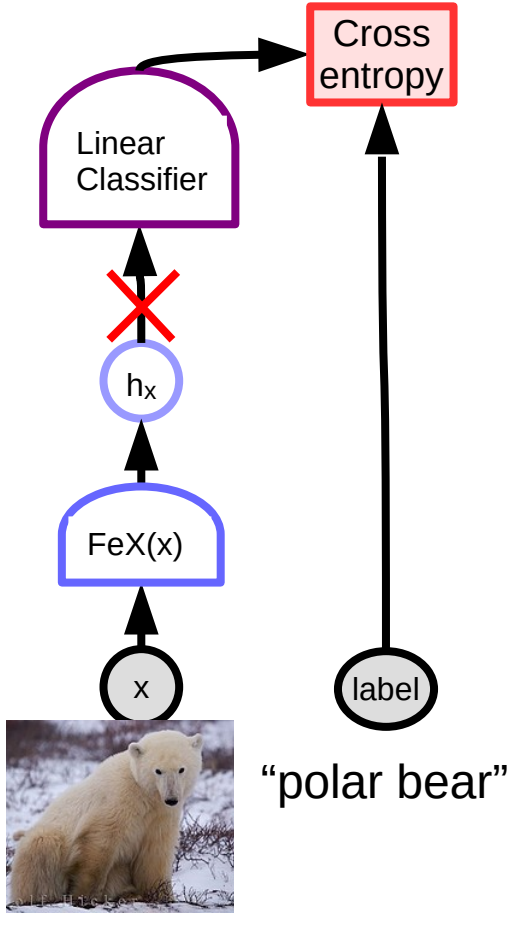
Make F(x,y) small



Make F(x,ŷ) large

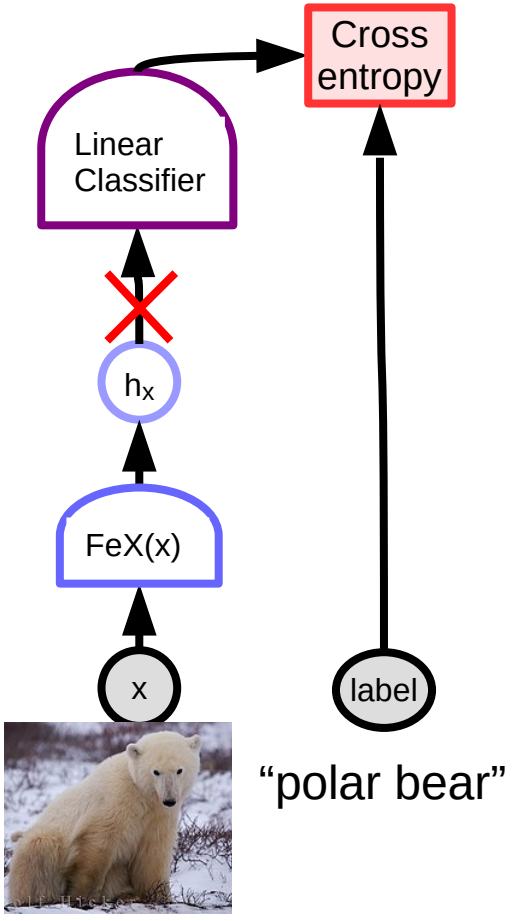


Training a supervised linear head



Contrastive Joint Embedding

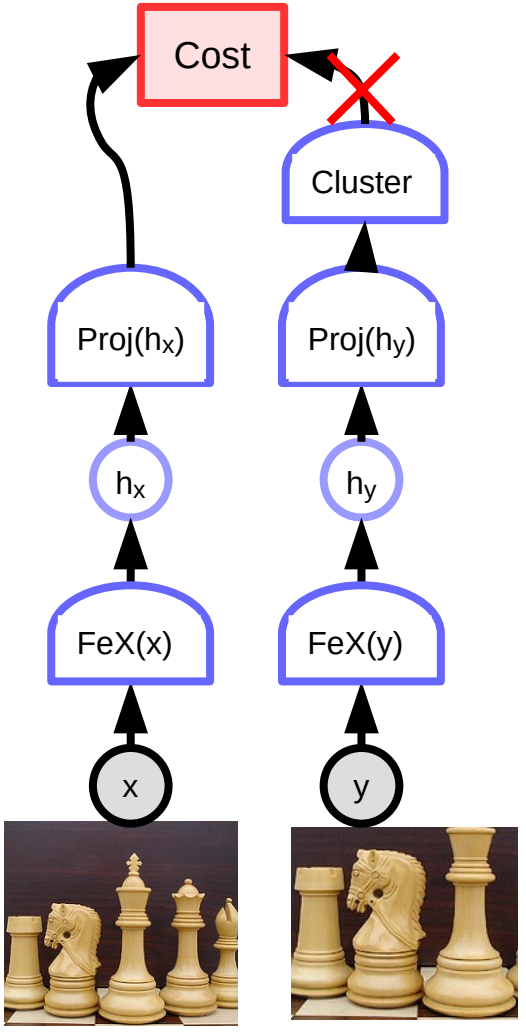
- **Issues:**
  - Hard negative mining
  - Expensive computationally
  - Only works for small dimension of embeddings (256)
- **Successful examples for image recognition:**
  - PIRL [Misra et al. Arxiv:1912.01991]
  - MoCo [He et al. Arxiv:1911.05722]
  - SimCLR [Chen et al. Arxiv:2002.05709]
- **Use InfoNCE group loss**



$$\mathcal{L}(x, y, \hat{y_1}, \ldots \hat{y}_q, w) = F_w(x, y) + \log \left[e^{-F_w(x,y)} + \sum_{i=1}^{q} e^{-F_w(x,\hat{y}_i,w)} \right]$$

Quantization Methods

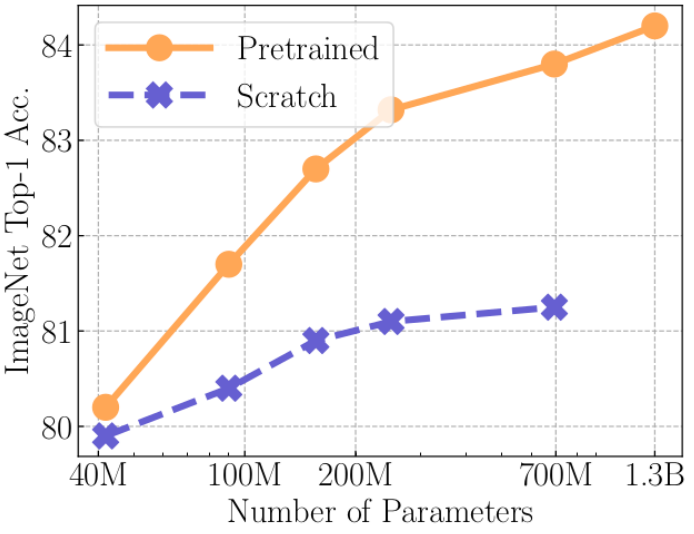
- **K-means clustering on embedding vectors**
  - Ensuring that all clusters are populated
    - Sinkhorn-Knapp procedure (information maximization)
  - Cluster centers used as targets for student branch
- **Examples**
  - DeepCluster [Caron arXiv:1807.05520]
  - SwAV [Caron arXiv:2006.09882]
- **Advantage:**
  - Works really well!
  - Uses large distortions (multicrop)
  - Scales to very large datasets

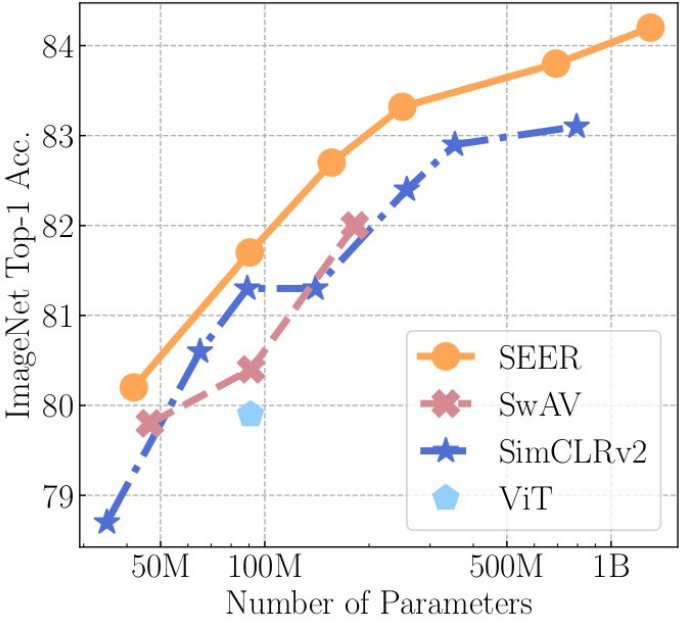


SEER [Goyal et al. ArXiv:2103.01988]

- **SwAV training on 1 billion random IG images**
- **RegNet architecture**
- **Fine-tuned on various datasets**
  - ImgNet full: 84.% top-1 correct
  - ImgNet 10%: 77.9%, ImgNet 1%: 60.5%
  - Inaturalist: 50.8%, Places205: 62.7%
  - Pascal VOC2007: 92.6%
- **Code: https://vissl.ai/**

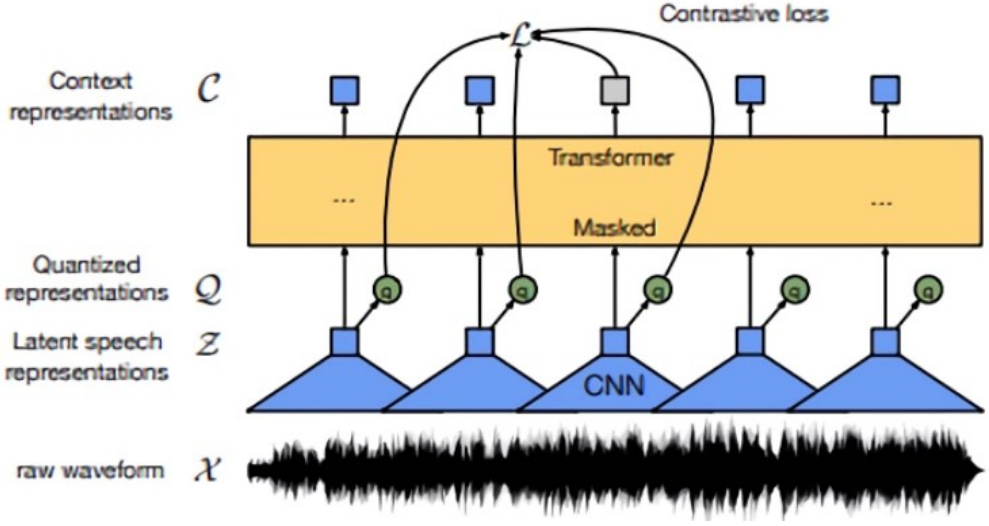
Method	Data	#images	Arch.	#param.	Top-1
DeeperCluster [6]	YFCC100M	96M	VGG16	138M	74.9
ViT [14]	JFT	300M	ViT-B/16	91M	79.9
SwAV [7]	IG	1B	RX101-32x16d	182M	82.0
SimCLRv2 [9]	ImageNet	1.2M	RN152w3+SK	795M	83.1
SEER	IG	1B	RG128	693M	83.8
SEER	IG	1B	RG256	1.3B	**84.2**

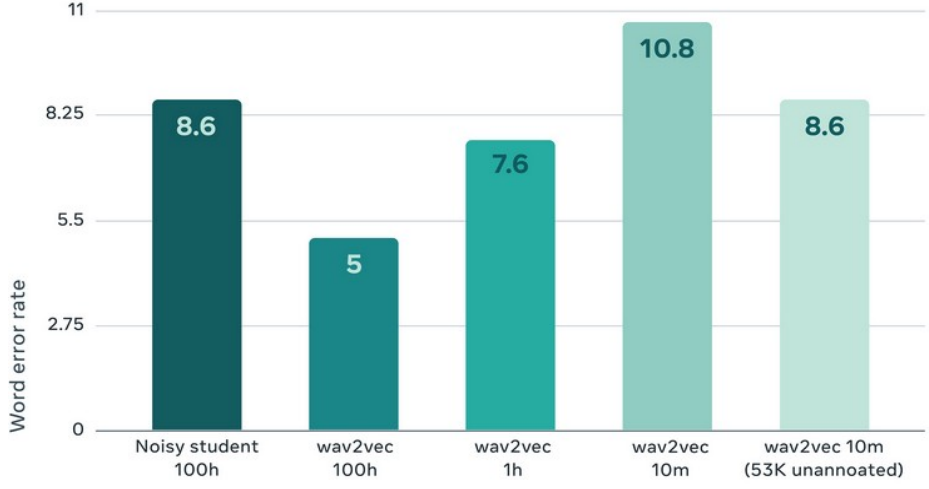




Wav2Vec 2.0: SSL for speech recognition

- **Pre-train on 960h of unlabeled speech,**
- **then train with 10 minutes, 1h or 100h of labeled speech**
- **Results on LibriSpeech**
  - Wav2vec on 10 minutes = Same WER as previous SOTA on 100h
  - Papers: [Baevski et al. NeurIPS 2020] [Xu et al. ArXiv:2010.11430]
  - Code: Github: PyTorch/fairseq

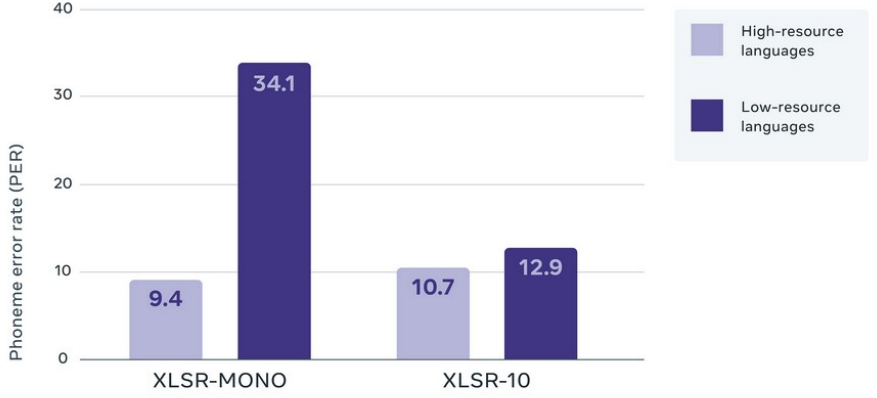




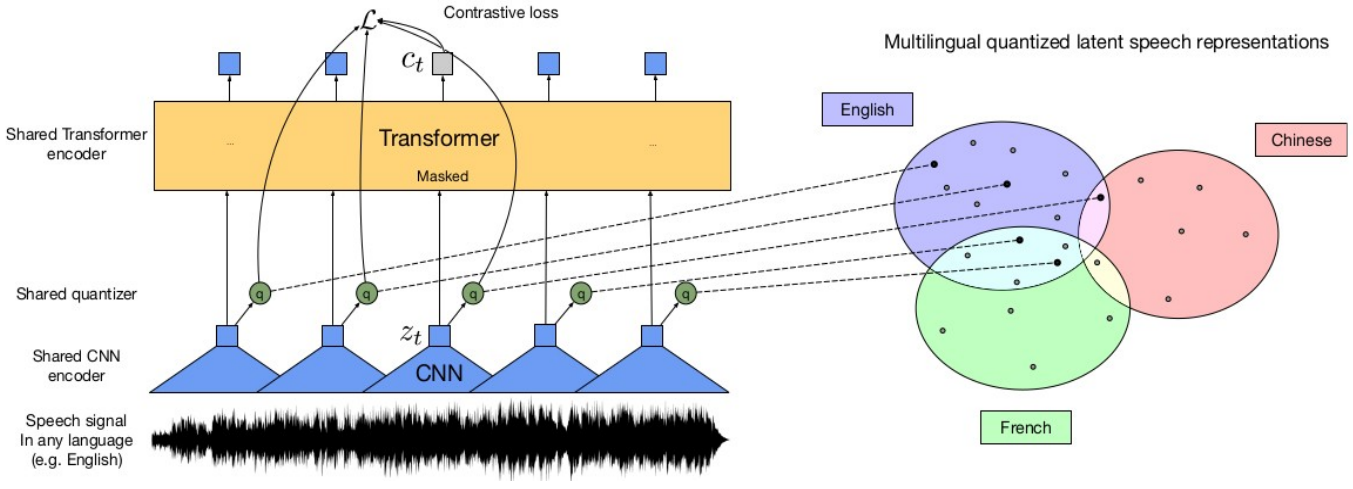
WER for Noisy Student self-training with 100 hours of labeled data. Wav2vec 2.0 with 100 hours, 1 hour, and only 10 minutes of labeled data. All models use the remainder of the LibriSpeech corpus (total 960 hours) as unannotated data, except for the last result, which uses 53K hours from LibriVox.

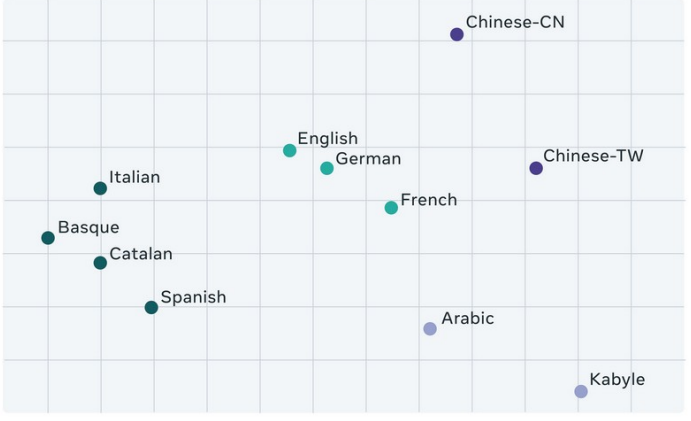
XLSR: multilingual speech recognition

- **Multilingual self-supervised ASR**
  - [Conneau arXiv:2006.13979]
  - Raw audio → ConvNet → Transformer
  - CommonVoice: 72% reduction of PER
  - BABEL: 16% reduction of WER



Results on the Common Voice benchmark in terms of phoneme error rate (PER), comparing training on each language individually (XLSR-Mono) with training on all 10 languages simultaneously (XLSR-10).





Visualization of how the learned units are used across languages. Graph shows a 2D PCA plot of how units are used in each language. Languages closer to each other, like English and German or Basque and Catalan, tend to use similar units.

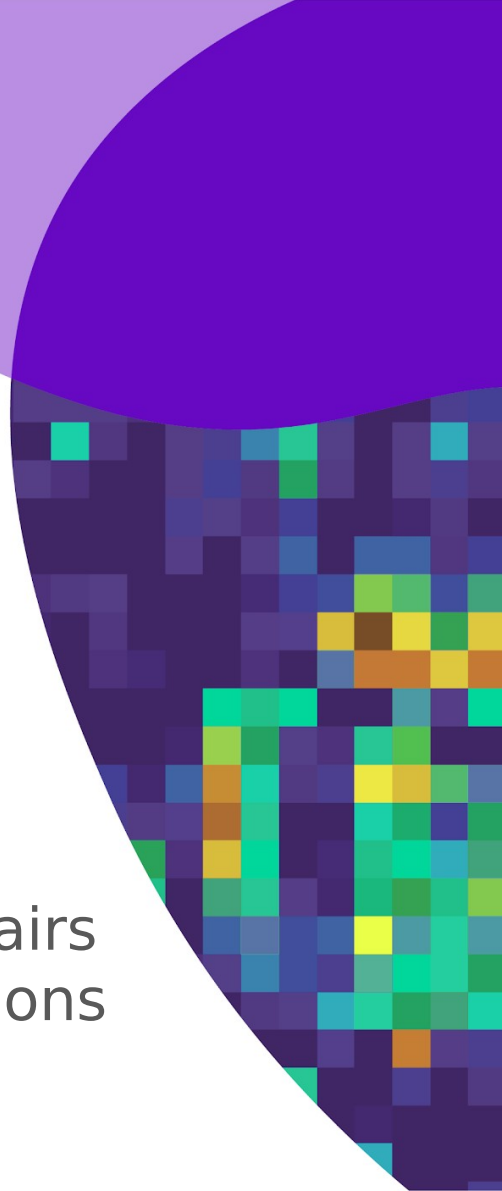
FACEBOOK AI

Regularized Methods
for joint embedding architectures

This is the cool stuff!

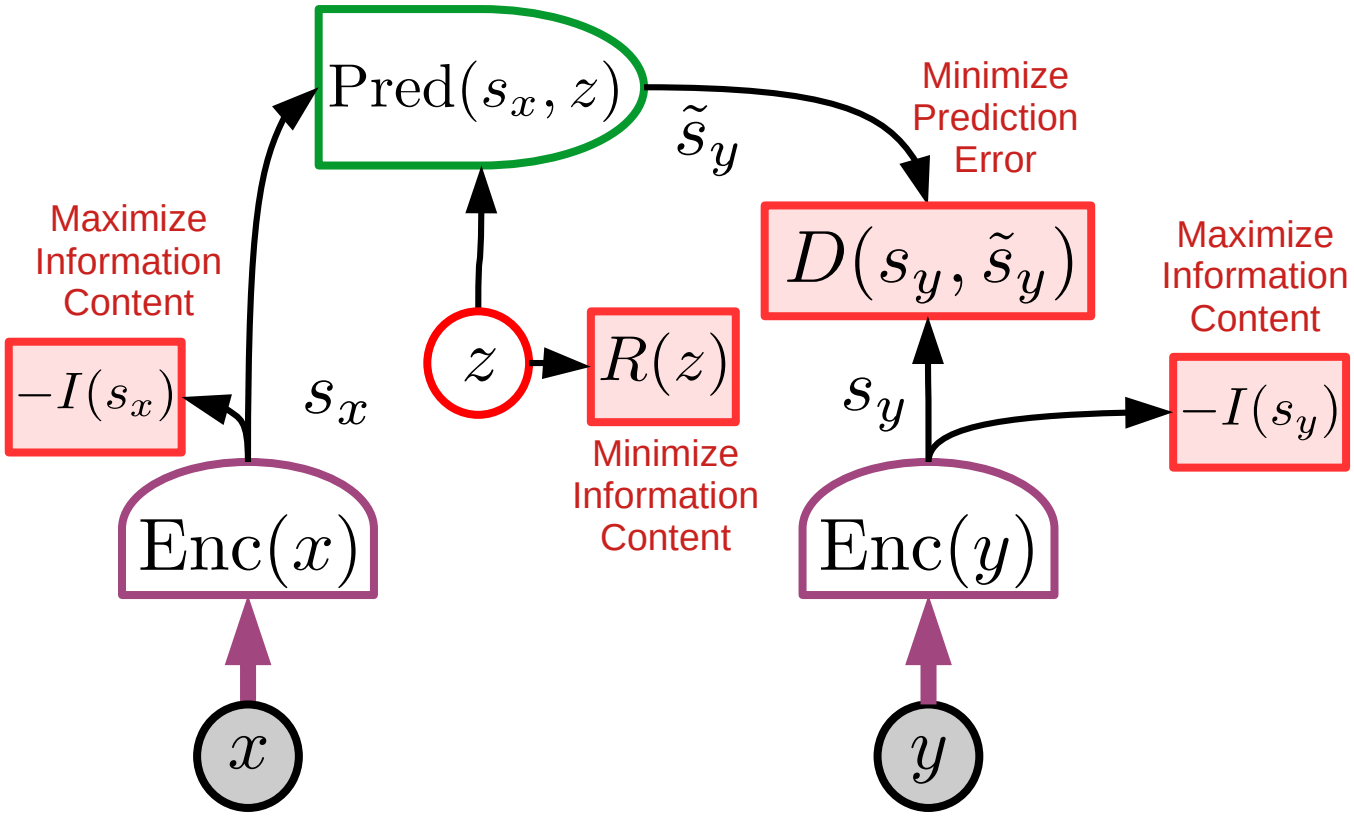
Push down on the energy of compatible sample pairs

Maximize the information capacity of representations

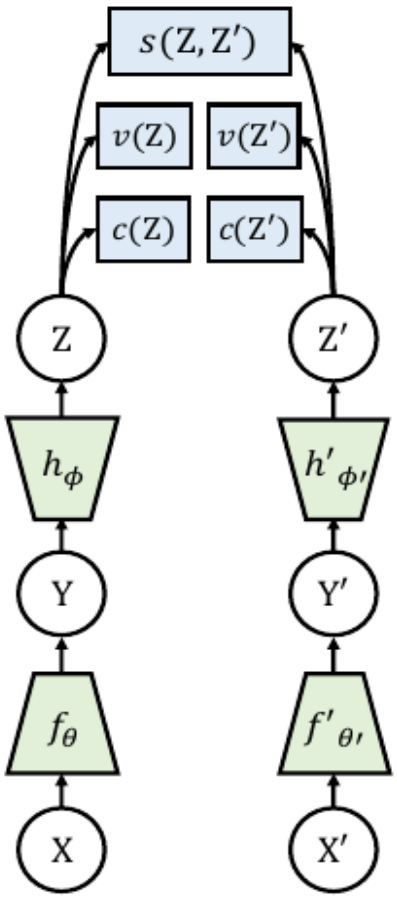


Training a JEPA (non contrastively)

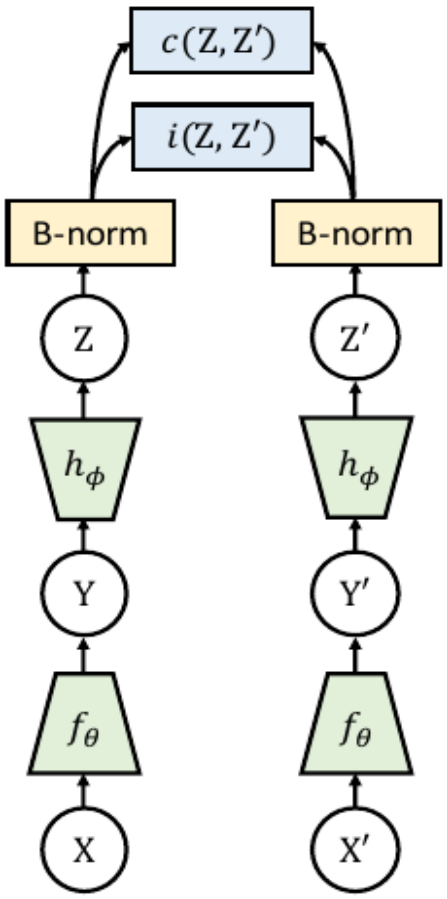
- **Four terms in the cost**
  - Maximize information content in representation of x
  - Maximize information content in representation of y
  - Minimize Prediction error
  - Minimize information content of latent variable z



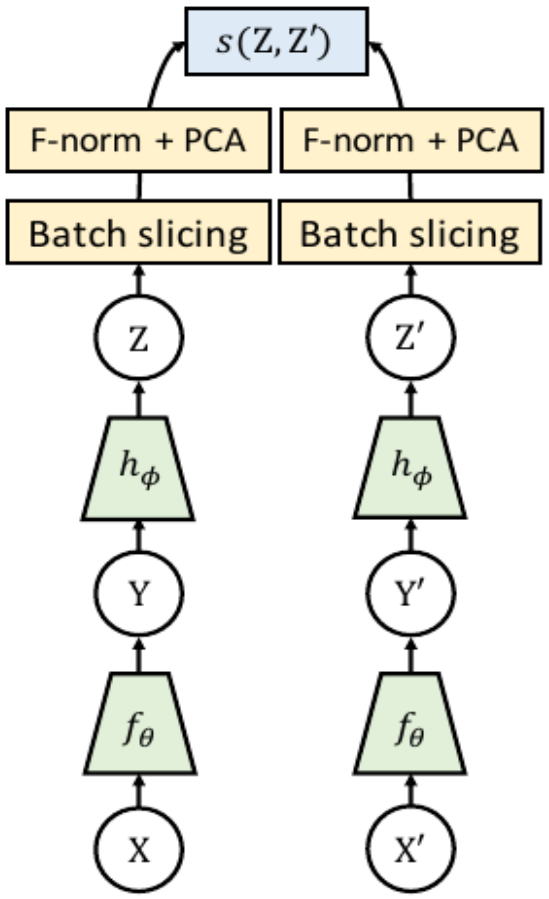
Regularized (Non-Contrastive) Joint Embedding Methods



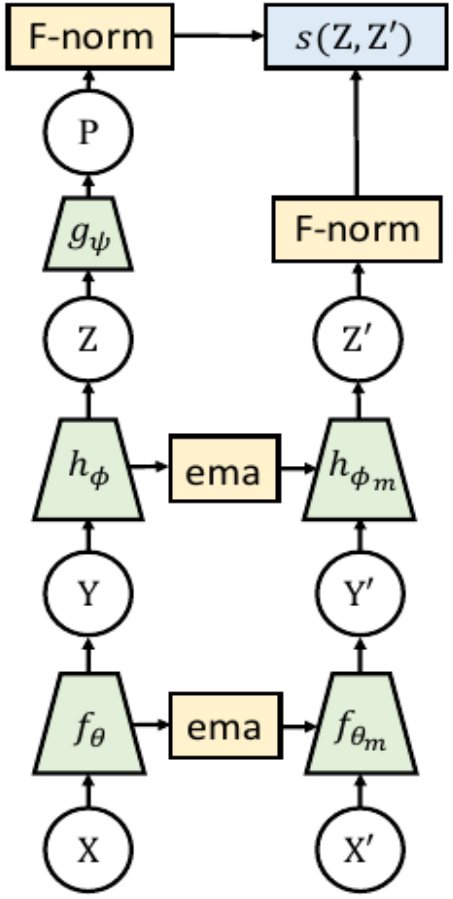
(a) VICReg



(b) Barlow Twins



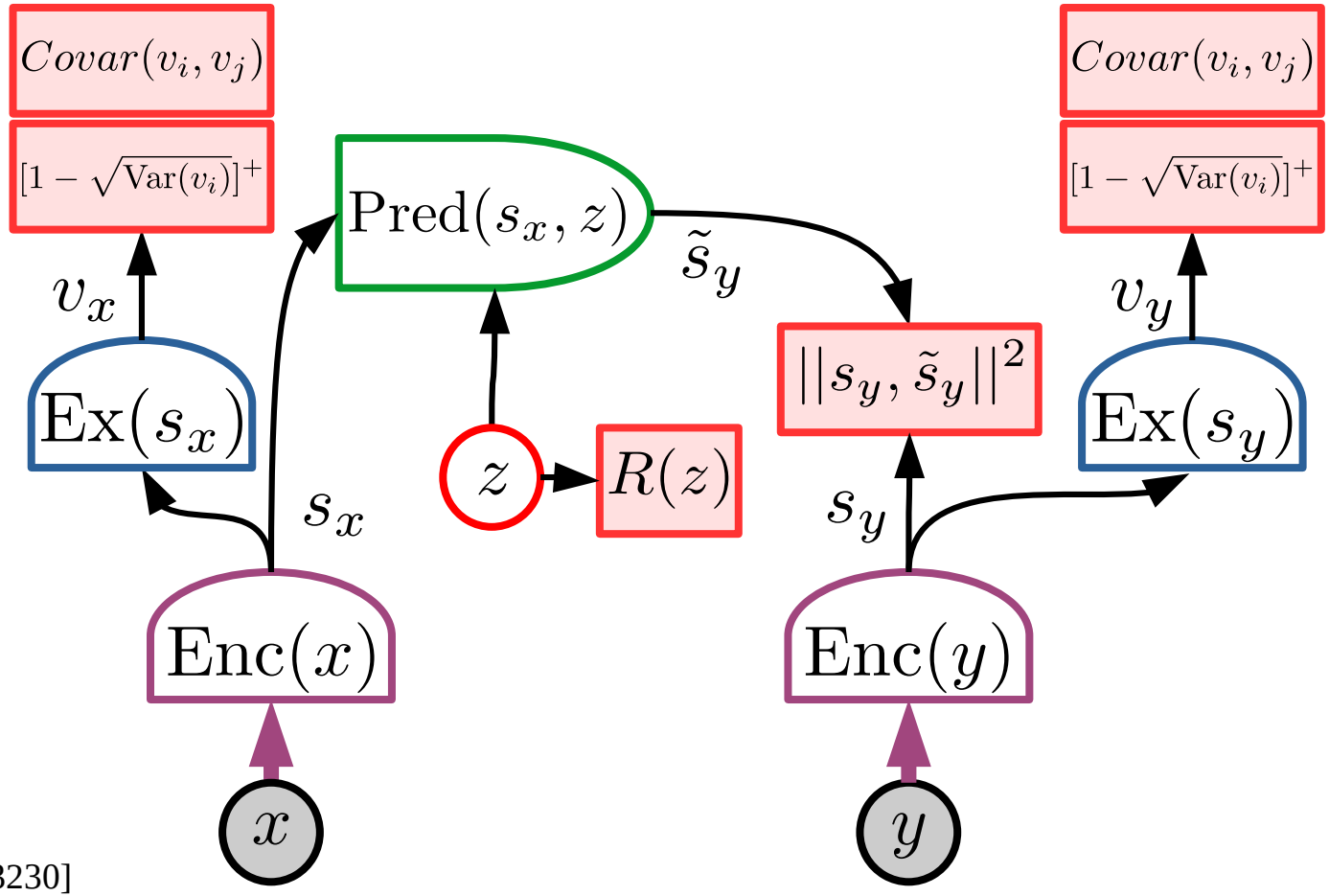
(c) W-MSE



(d) BYOL

VICReg: Variance, Invariance, Covariance Regularization

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

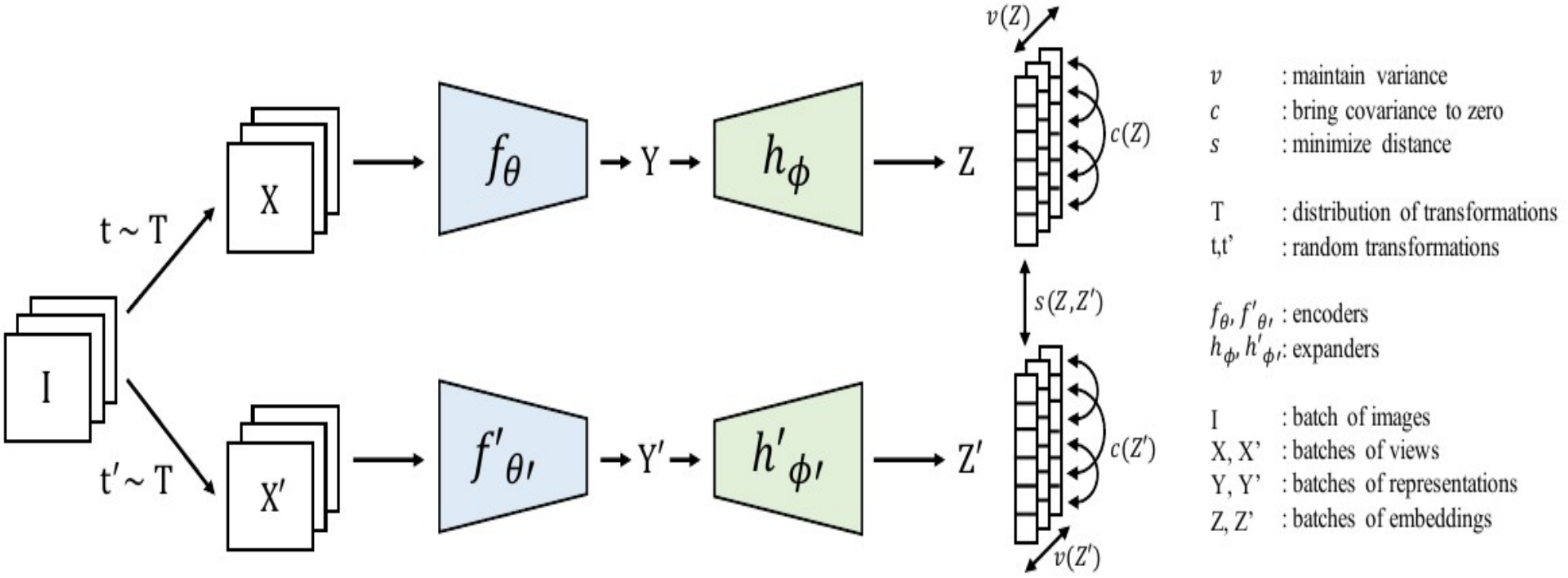


Barlow Twins [Zbontar et al. ArXiv:2103.03230]

VICReg [Bardes, Ponce, LeCun arXiv:2105.04906, ICLR 2022]

VICReg: Variance-Invariance-Covariance Regularization

- VICReg: [Bardes, Ponce, LeCun arXiv:2105.04906, ICLR 2022]
- Improvement on Barlow Twins [Zbontar et al. ArXiv:2103.03230]
- Maximizes a measure of mutual information between the two outputs



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**

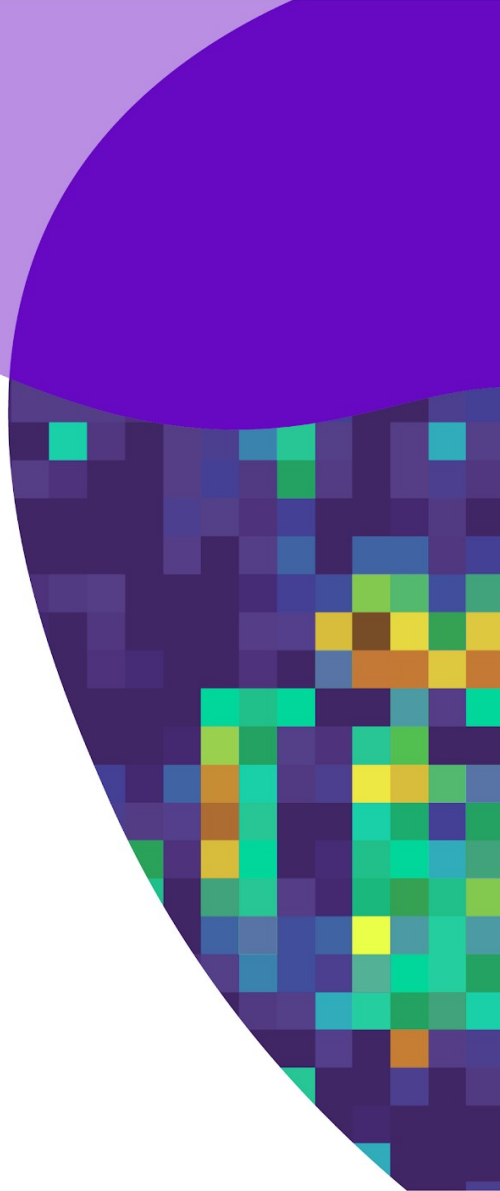
Table 4: **Impact of sharing weights or not between branches.** Top-1 accuracy on linear classification with 100 pretraining epochs. In all settings, the encoder and expander of both branches share the same architecture, but either share weights (✓), or have different weights in the two branches.

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

FACEBOOK AI

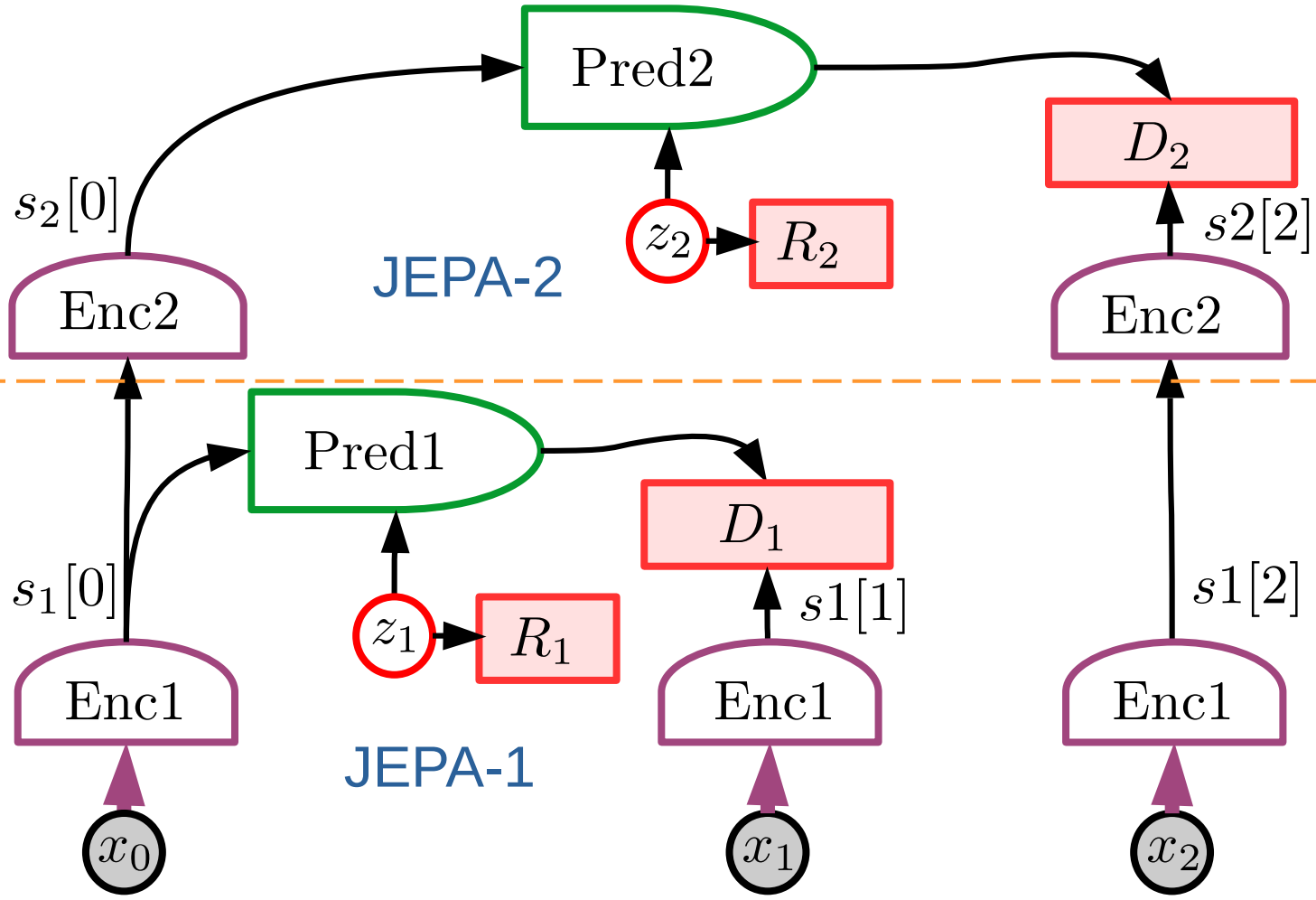
Hierarchical JEPA

For control, planning, and policy learning.



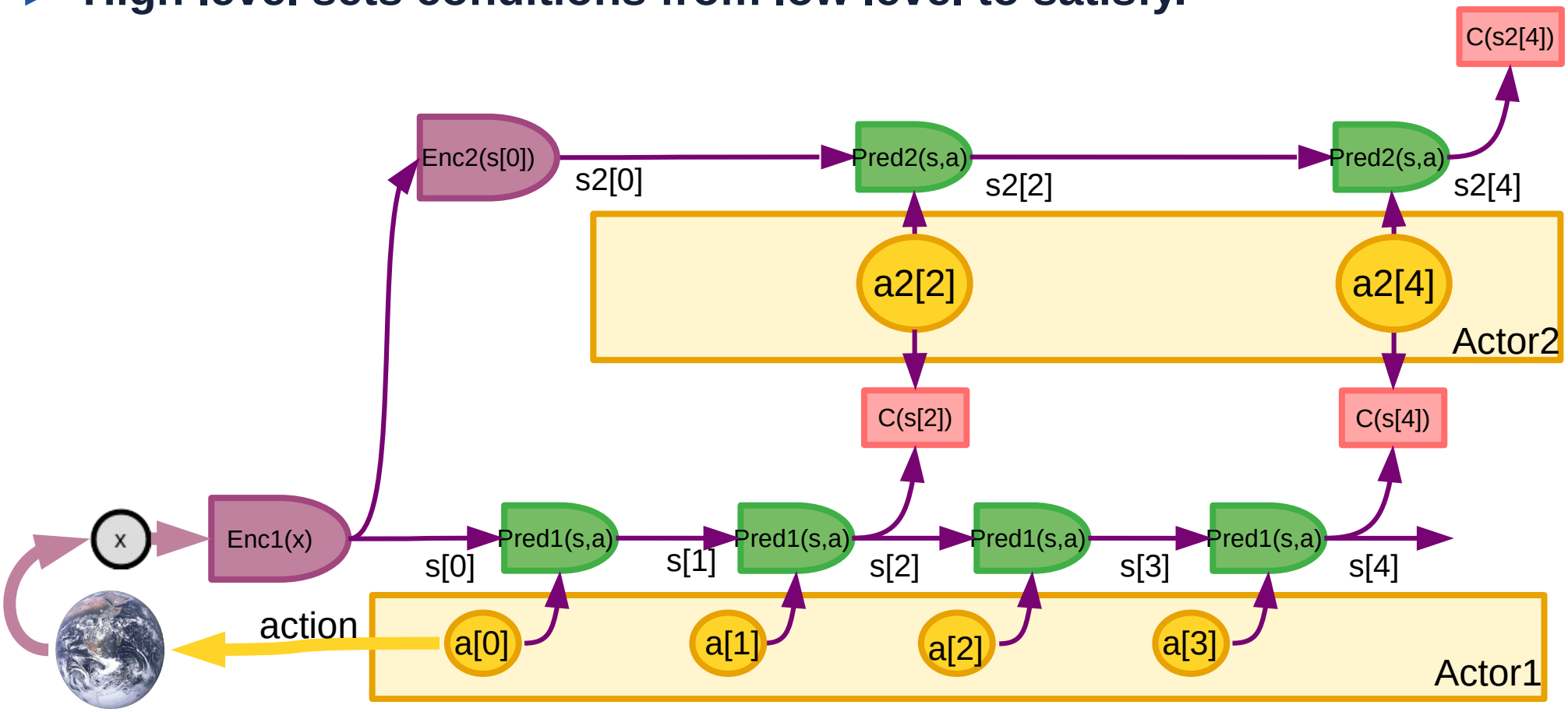
Multi time-scale Predictions

- **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



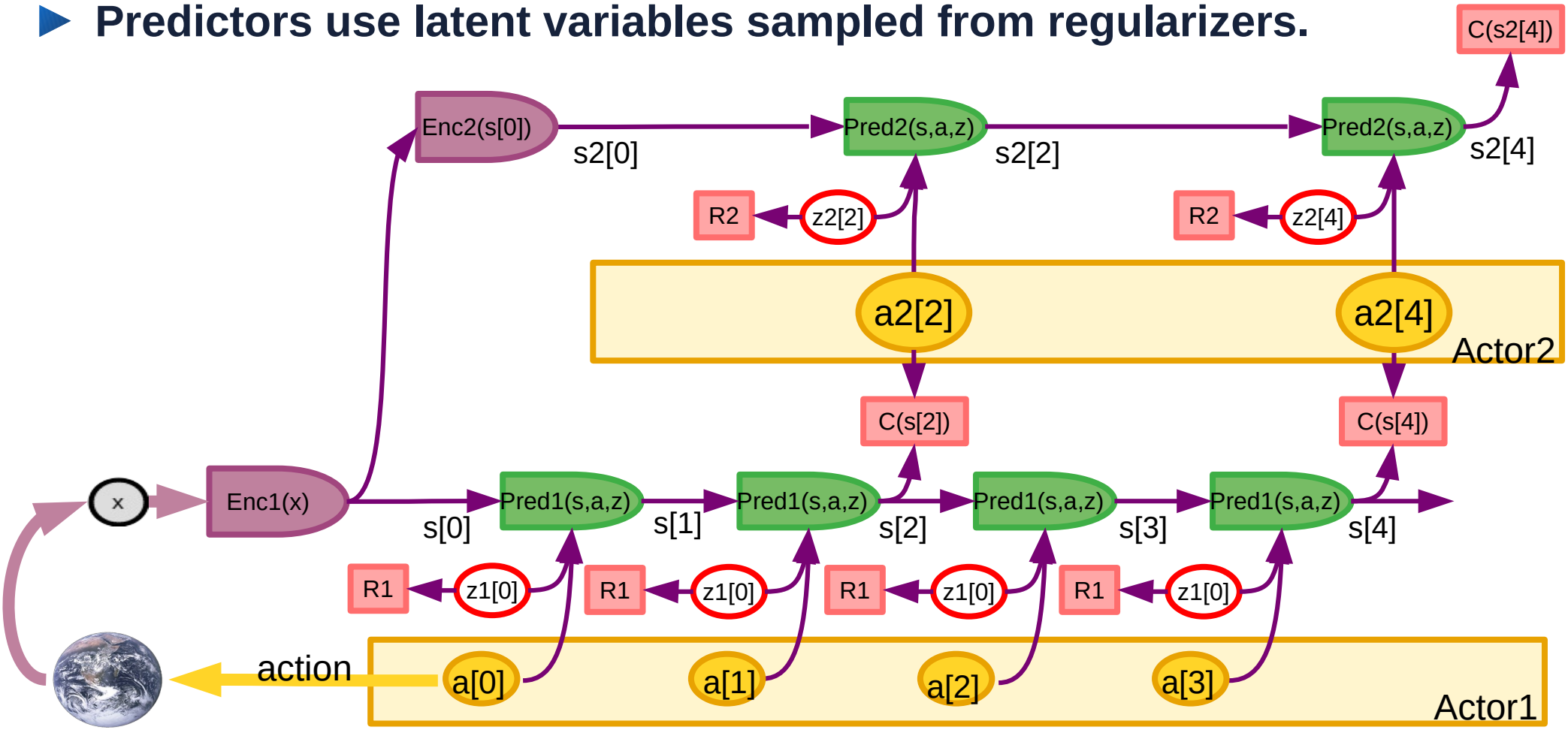
Mode-2 Planning with H-JEPA-based World Model

- **High level sets conditions from low level to satisfy.**



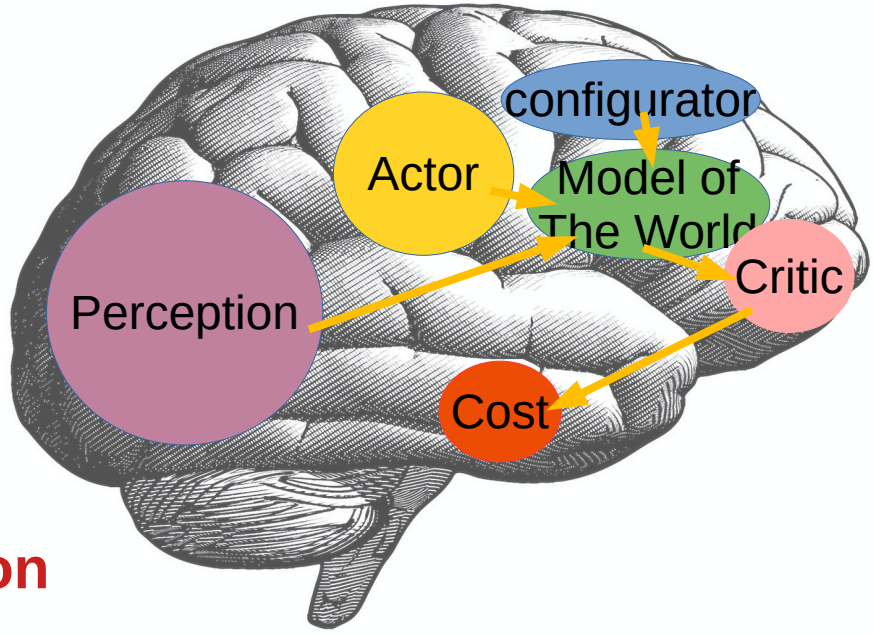
Hierarchical Planning with Uncertainty

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



Obstacles on the Way to Autonomous AI Systems

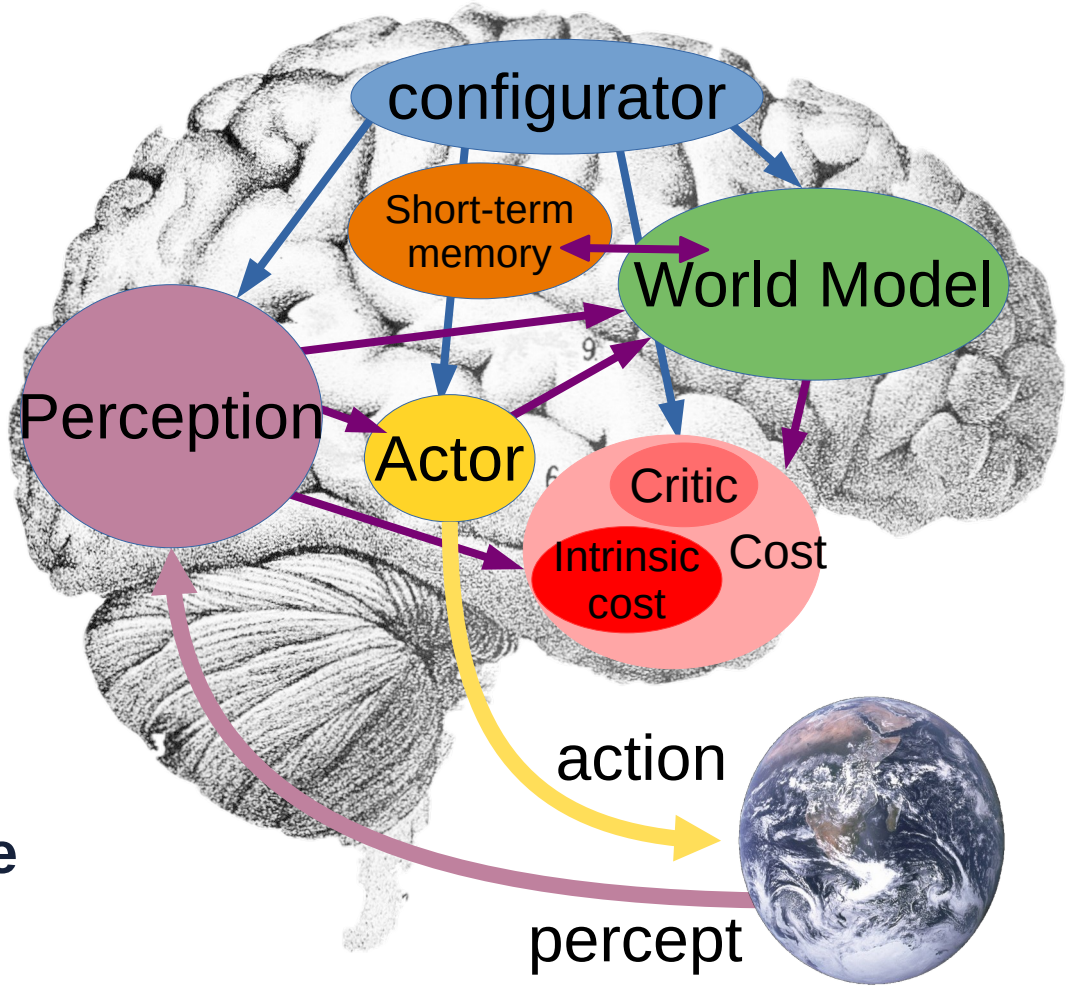
- **Self-Supervised Learning**
  - To learn representations of the world
  - To learn predictive models of the world
- **Handling uncertainty in predictions**
  - Joint-embedding predictive architectures
  - Energy-Based Model framework
- **Learning world models from observation**
  - Like animals and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



A Single World Model Engine

- **Configurator?**

- **Configures other modules for a task**
- **Provides executive control**
- **Sets subgoals**
- **Primes the perception module for the task at hand**
- **Configures the world model for the task at hand.**
  - There is a single world model.
- **Focuses the agent on deliberate (“conscious”) tasks.**



FACEBOOK AI

Thank You!