CS 103: Representation Learning, Information Theory and Control

Lecture 6, Feb 15, 2019

VAEs and disentanglement

A β-VAE minimizes the loss function:

$$\mathcal{L} = H_{p,q}(x|z) + \beta\mathbb{E}_x[\mathrm{KL}(q(z|x)\|\underbrace{p(z)}_{\text{Factorized prior}})]$$

$$= H_{p,q}(x|z) + \beta\,\{\underbrace{I(z;x)}_{\text{Minimality}} + \underbrace{\mathrm{TC}(z)}_{\text{Disentanglement}}\}$$

Assuming a factorized prior for z, a β-VAE optimizes both for the IB Lagrangian and for disentanglement.

Achille and Soatto, "*Information Dropout: Learning Optimal Representations Through Noisy Computation*", PAMI 2018 (arXiv 2016)

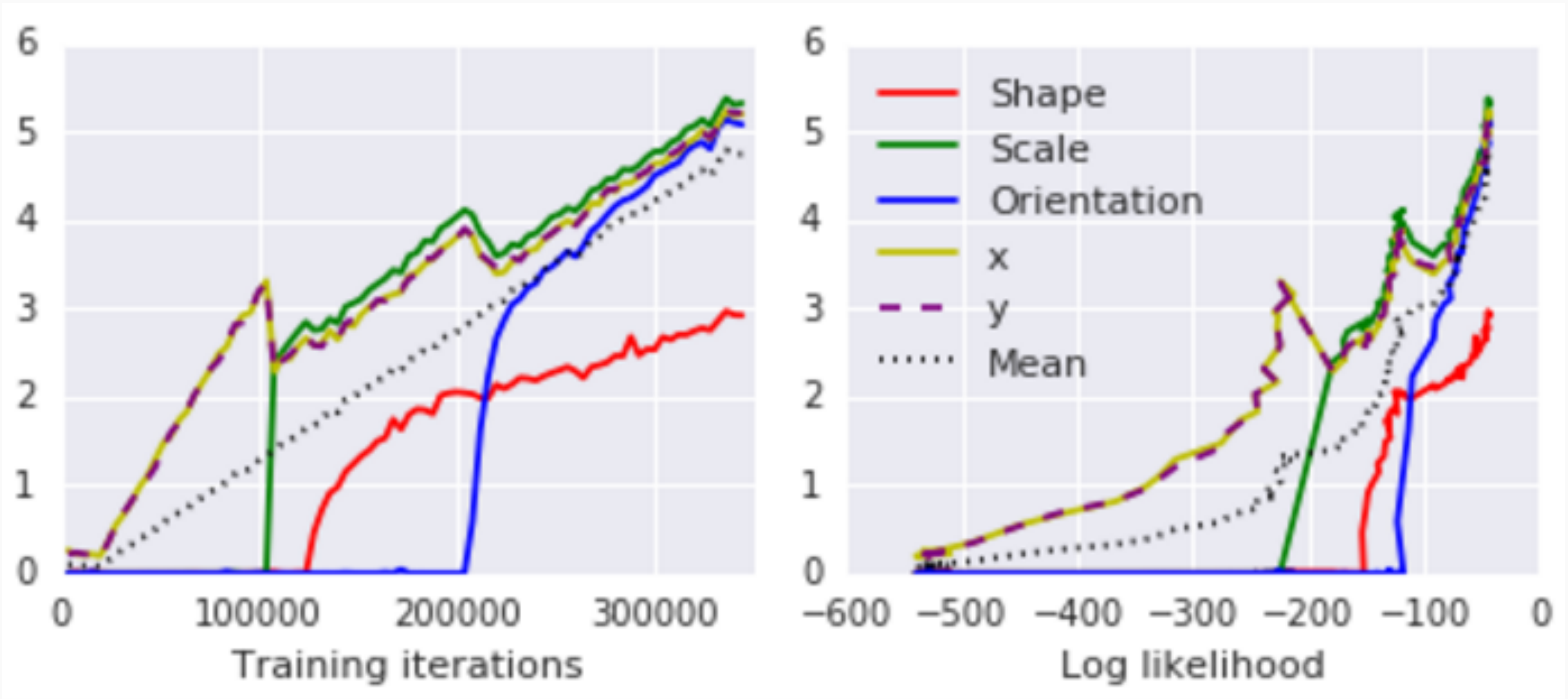
Learning disentangled representations

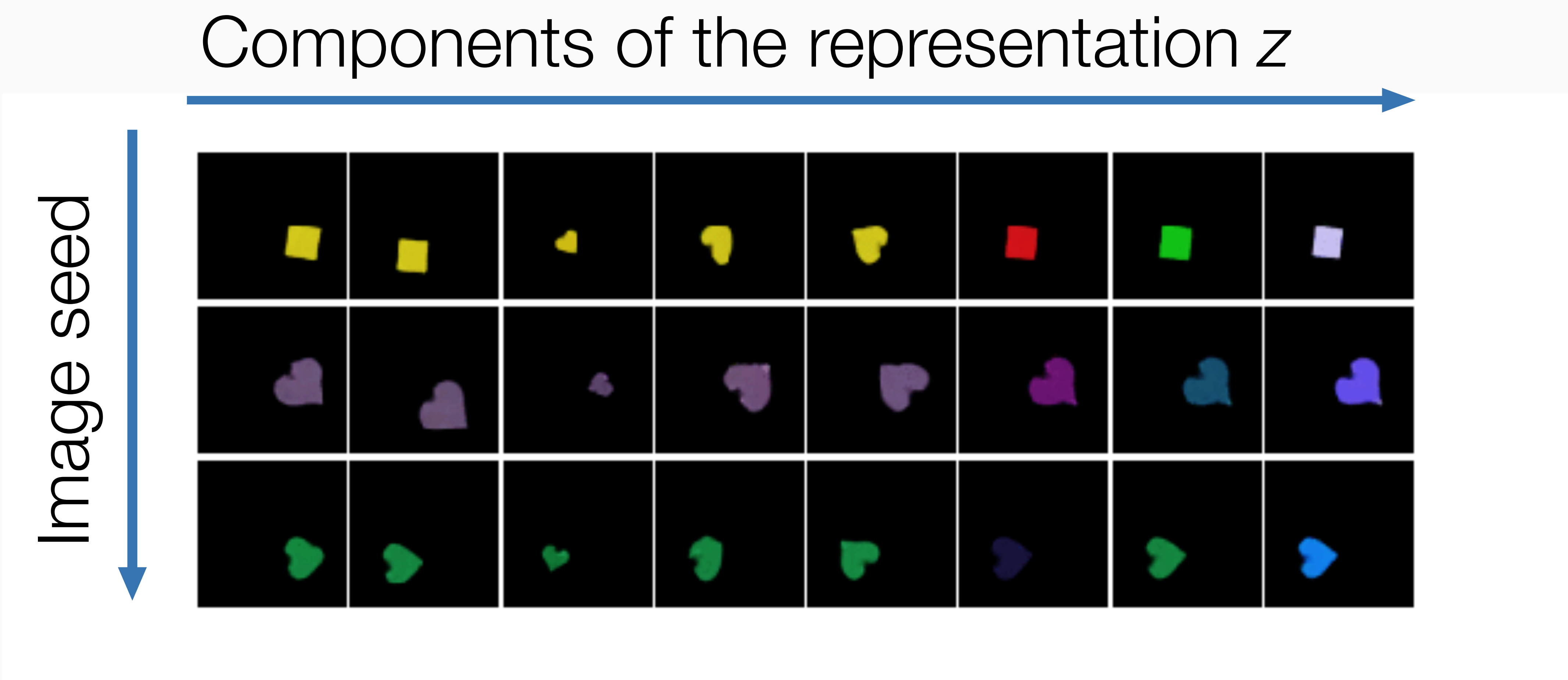
(Higgins et al., 2017, Burgess et al., 2017)

Start with very high β and slowly decrease during training.

Beginning: Very strict bottleneck, only encode most important factor

End: Very large bottleneck, encode all remaining factors



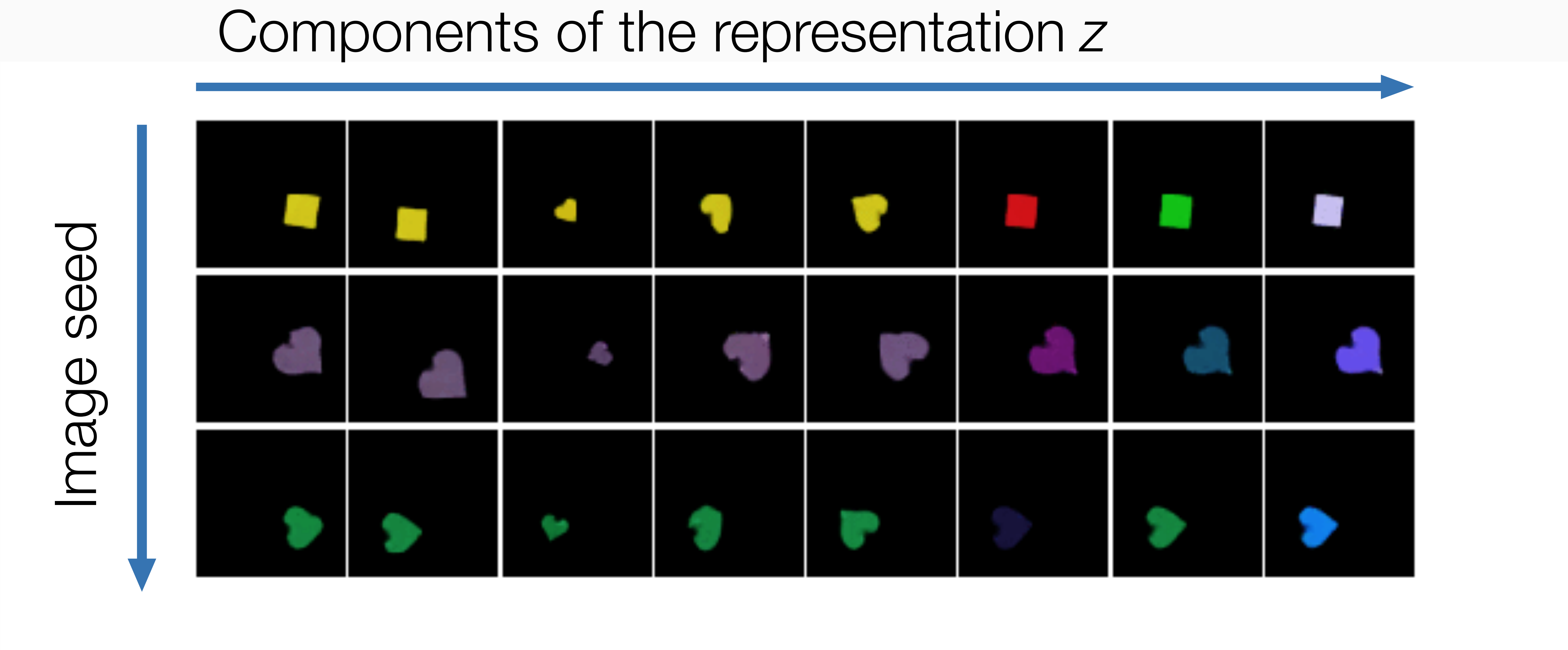


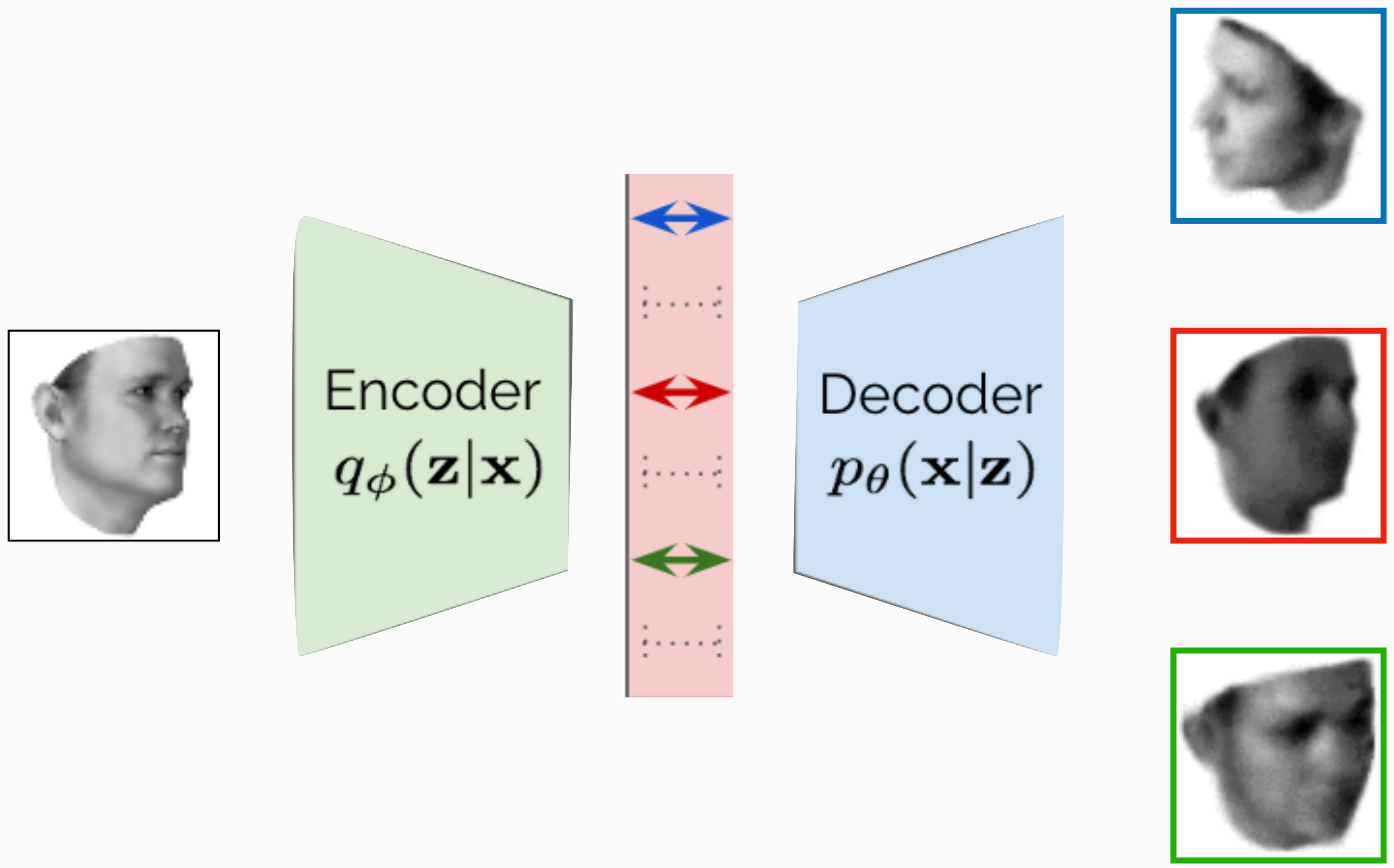
Think of it as a non-linear PCA, where *training time* disentangles the factors.

Learning disentangled representations

(Higgins et al., 2017, Burgess et al., 2017)

Each component of the learned representation corresponds to a different semantic factor.





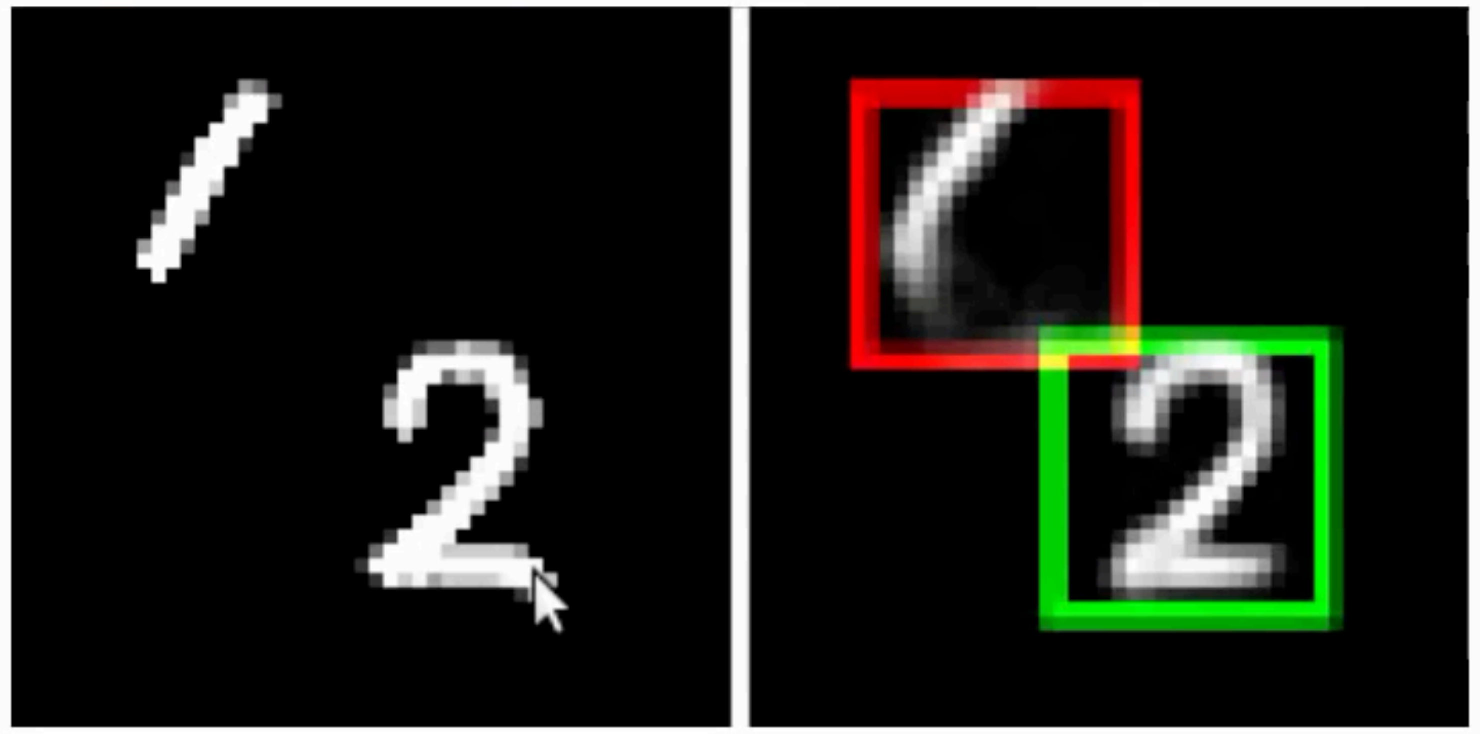
Higgins et al., β-*VAE: Learning Basic Visual Concepts with a Constrained Variational Framework*, 2017
Burgess et al., *Understanding Disentangling in beta-VAE"* 2017

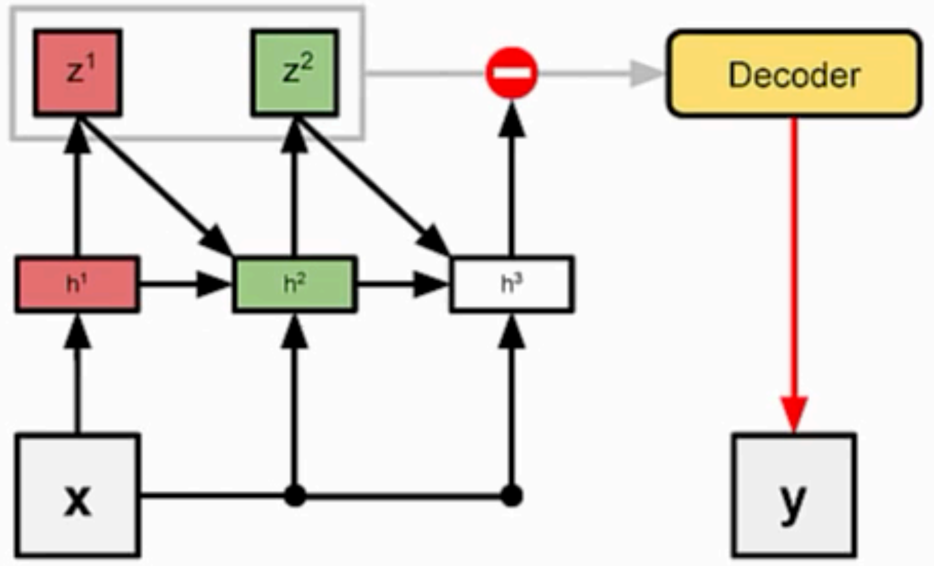
Pictures courtesy of Higgins et al., Burgess et al.

Multiple Objects

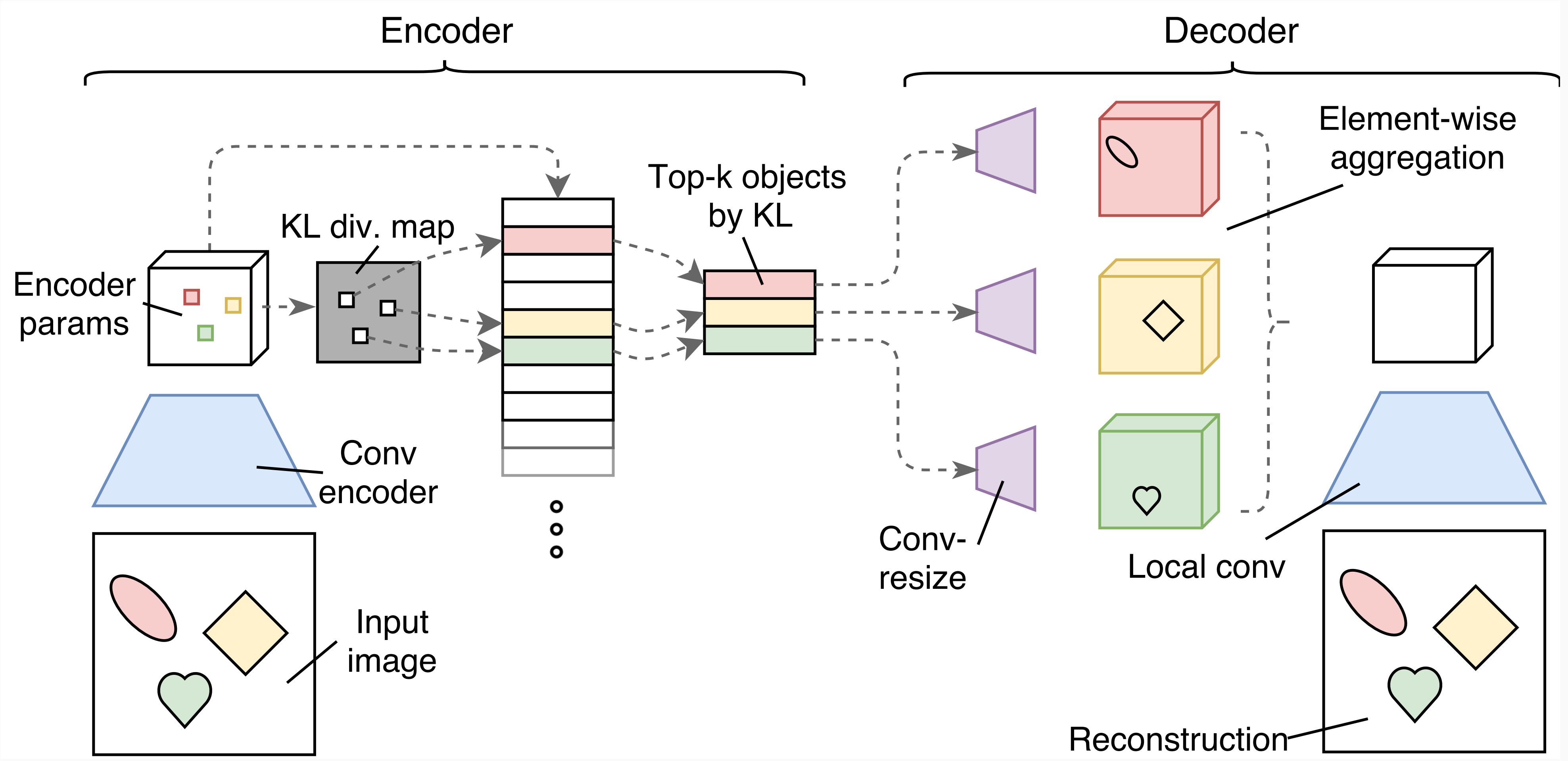
Attend, Infer, Repeat (Eslami et al.)

Good reconstruction, correct count

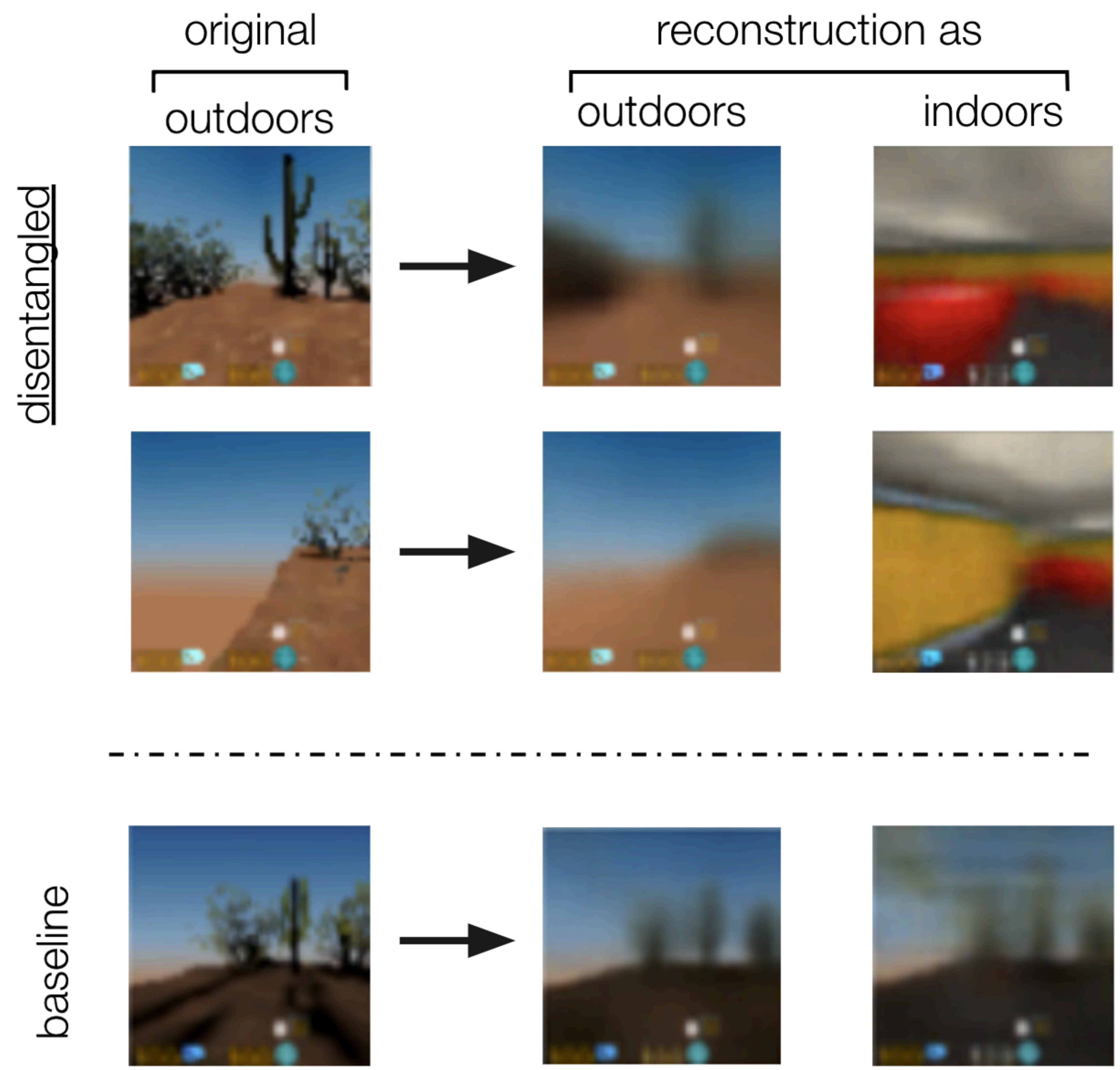


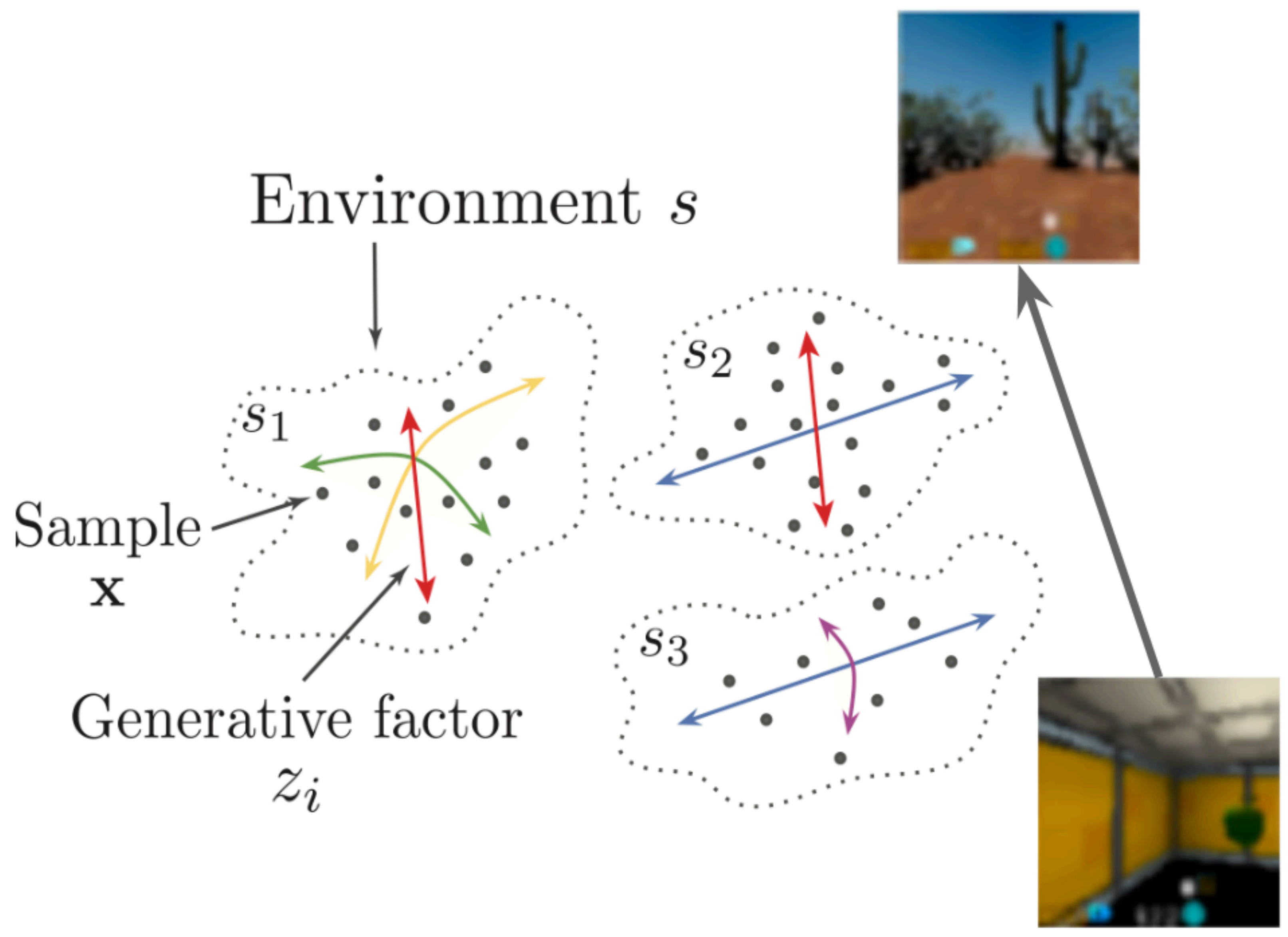


Multi-Entity VAE (Nash et al.)



Is the representation "semantic" and domain invariant?





Achille et al., *Life-Long Disentangled Representation Learning with Cross-Domain Latent Homologies*, 2018

Corollary: Ways of enforcing invariance

The standard architecture alone already promotes invariant representations

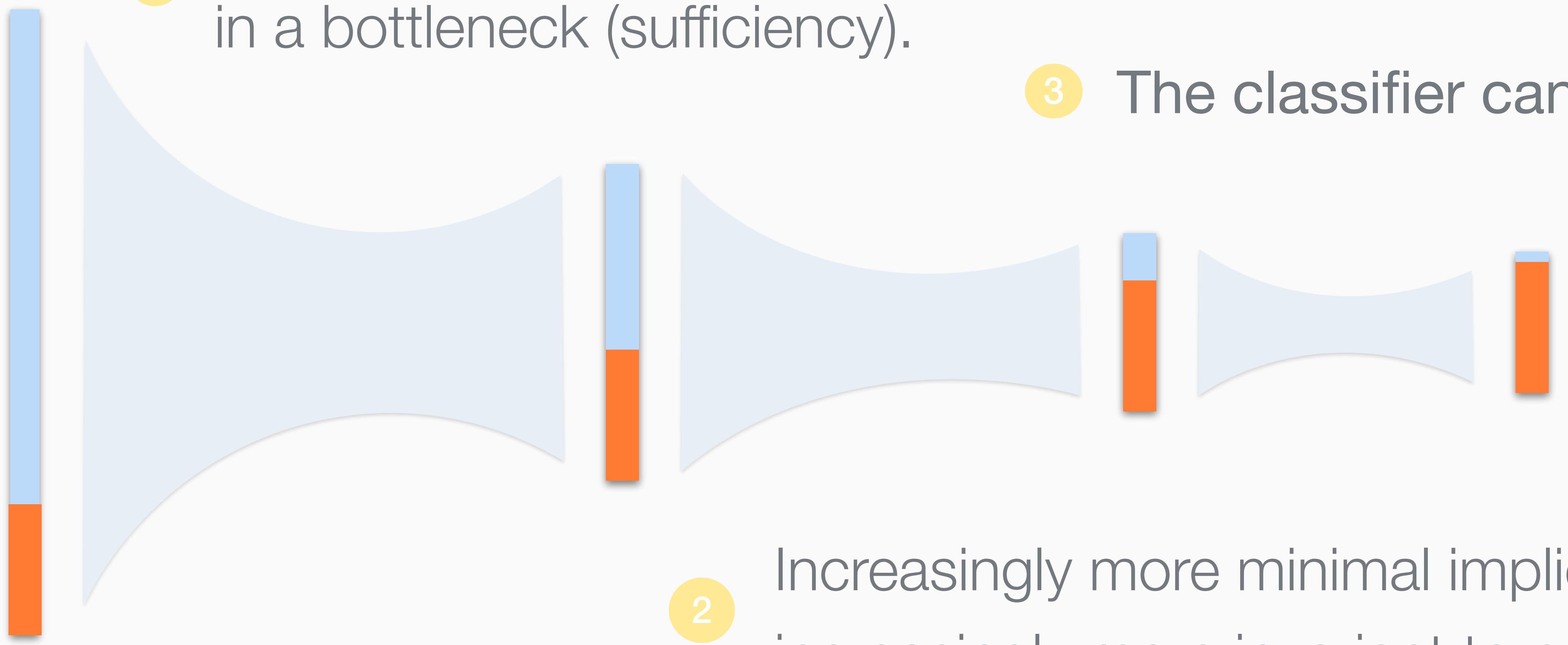
Regularization by architecture

Reducing dimension (max-pooling) or adding noise (dropout) increases minimality and invariance.

1 Only nuisance information dropped in a bottleneck (sufficiency).

3 The classifier cannot overfit to nuisances.

Nuisance information
I(x; n)



Task information
I(x; y)

2 Increasingly more minimal implies increasingly more invariant to nuisances.

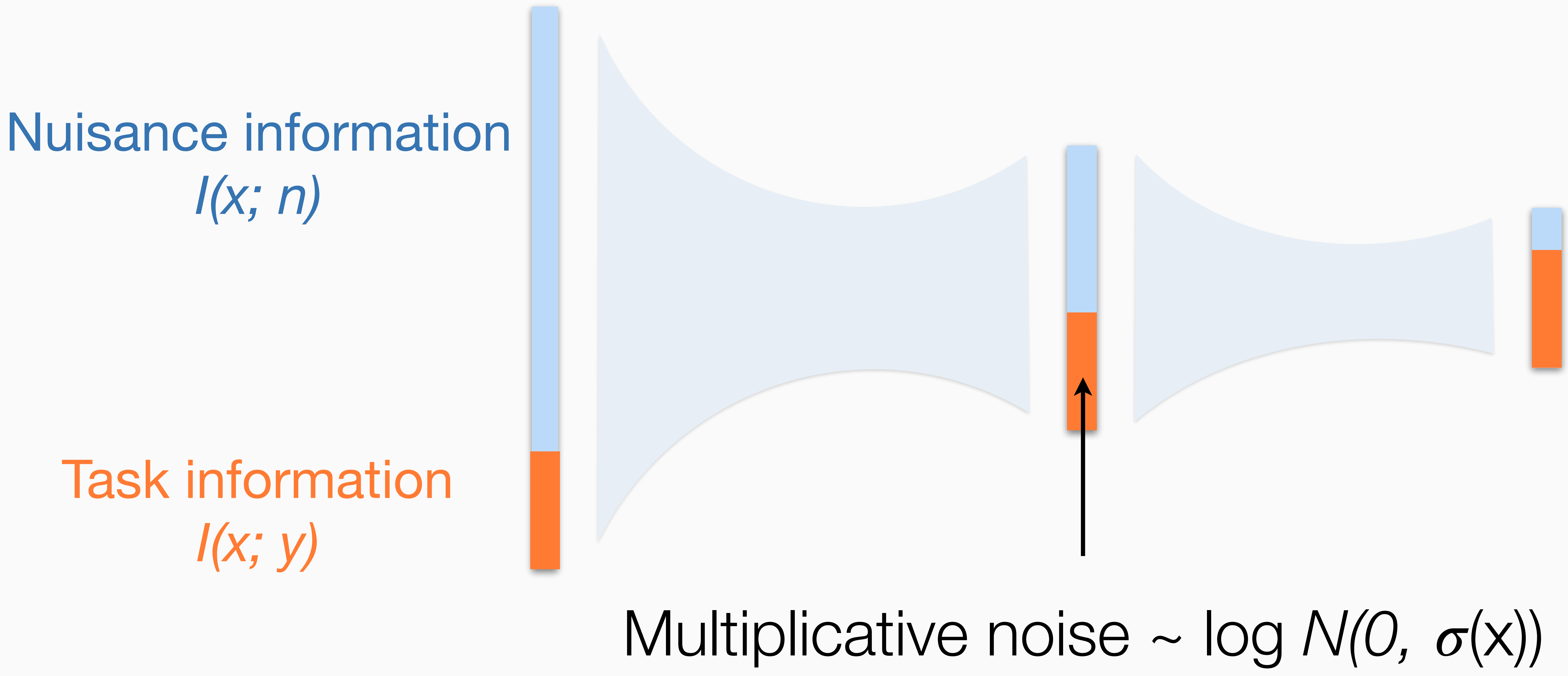
Stacking layers

Stacking multiple layers makes the representation increasingly minimal.

Information Dropout: a Variational Bottleneck

Creating a soft bottleneck with controlled noise

$$\mathscr{L} = H_{p,q}(y\,|\,x) + \underbrace{\mathbb{E}_x\,\mathrm{KL}(p(z\,|\,x)\|q(z))}_{\text{bottleneck}} = \boxed{H_{p,q}(y\,|\,x) + \underbrace{\mathbb{E}_x[-\log|\Sigma(x)|]}_{\text{Average log-variance of noise}}}$$

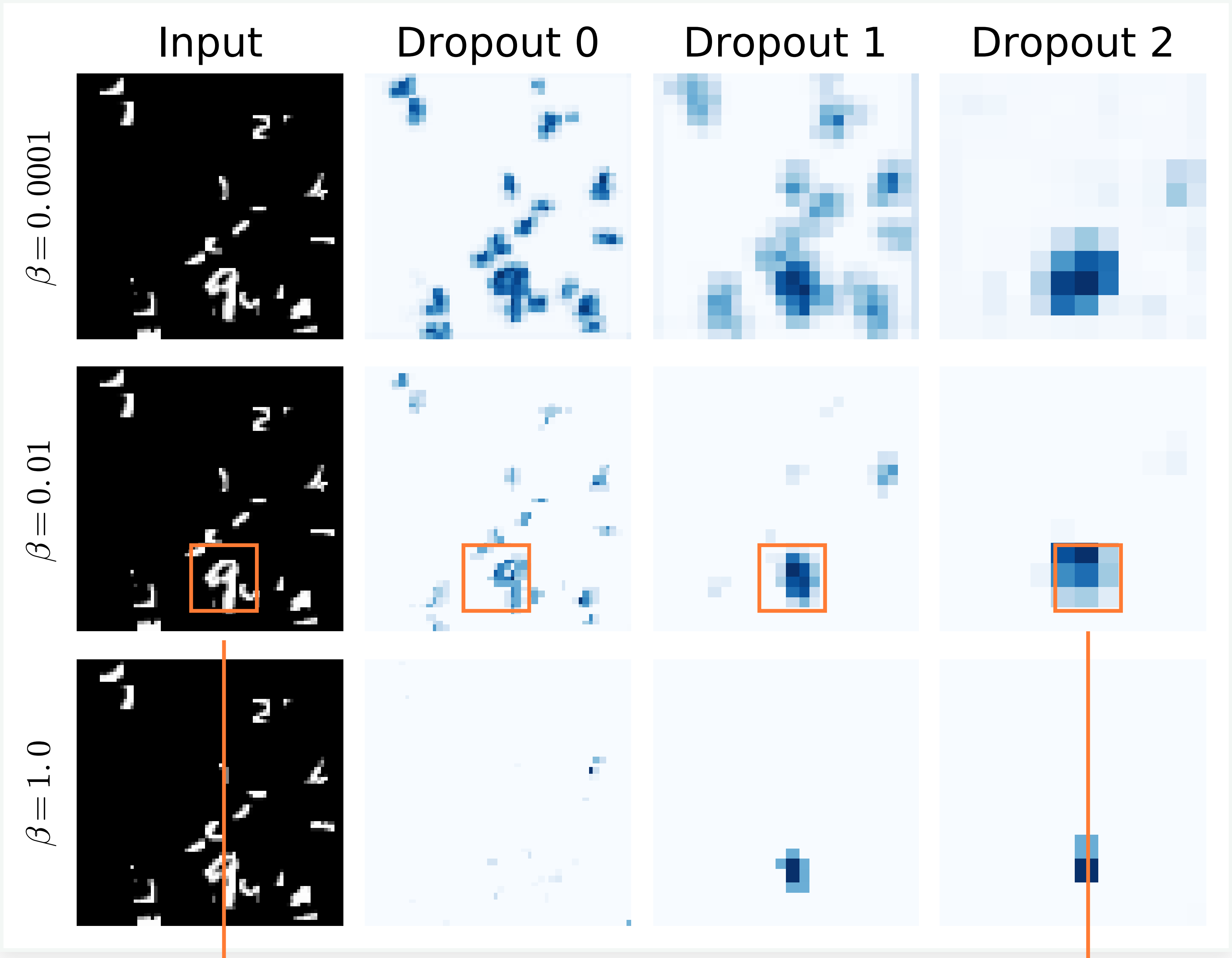


Achille and Soatto, "*Information Dropout: Learning Optimal Representations Through Noisy Computation*", PAMI 2018 (arXiv 2016)

Learning invariant representations

(Achille and Soatto, 2017)

Deeper layers filter increasingly more nuisances



Stronger bottleneck = more filtering

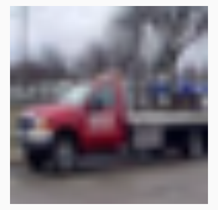
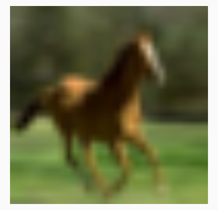
Only informative part of the image

Other information is discarded

Achille and Soatto, "*Information Dropout: Learning Optimal Representations Through Noisy Computation*", PAMI 2018 (arXiv 2016)

The catch

What if we just represent an image by its index in the training set (or by a unique hash)?

x		z		y
24,576 bits		*16 bits*		*4 bits*
	→	0000000000000000	→	0100
	→	0000000000000001	→	0001
	→	0000000000000010	→	0010
	→	0000000000000011	→	0101

It is a sufficient representation and it is close to minimal.

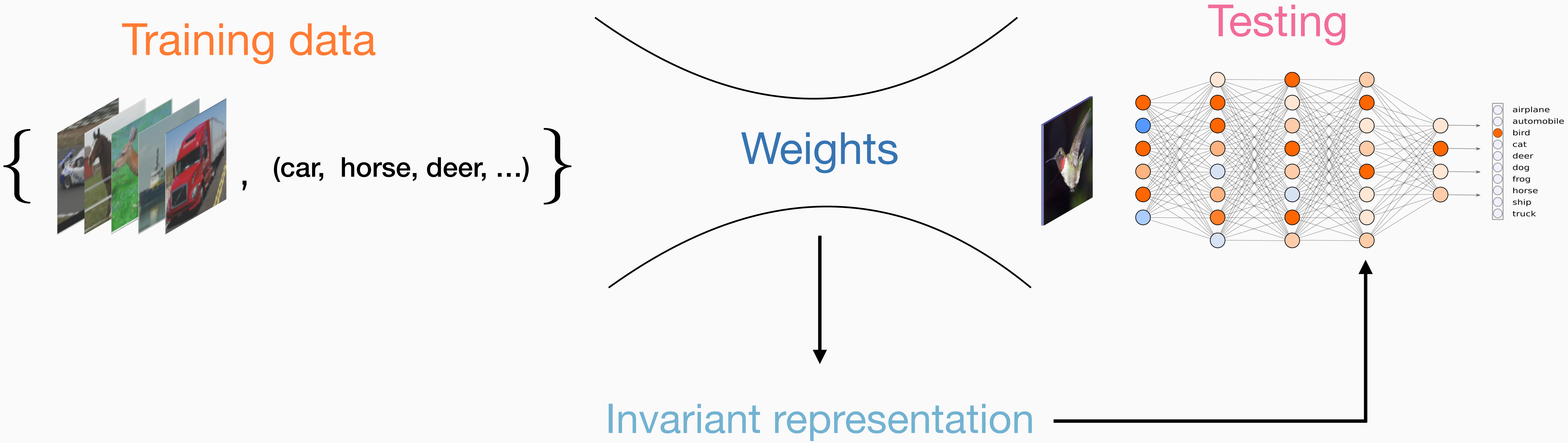
This Information Bottleneck is wishful thinking

The IB is a statement of desire for future data we do not have:

$$\min_{q(z|x)} \mathcal{L} = H_{p,q}(y|z) + \beta\, I(z;x)$$

What we have is the data collected in the past.

What is the best way to use the past data in view of future tasks?


Training data
(car, horse, deer, ...)
Weights
Invariant representation
Testing
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck