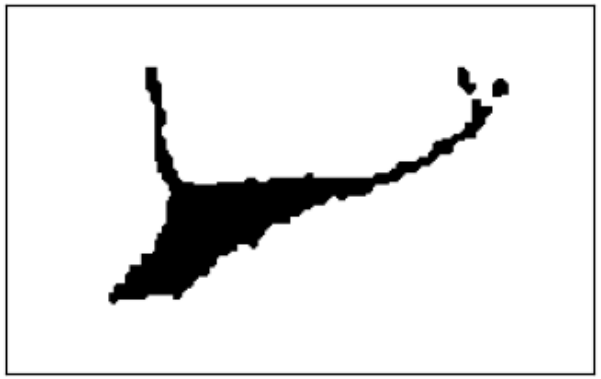
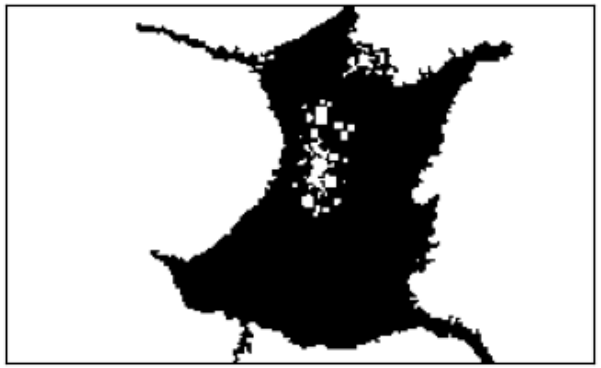
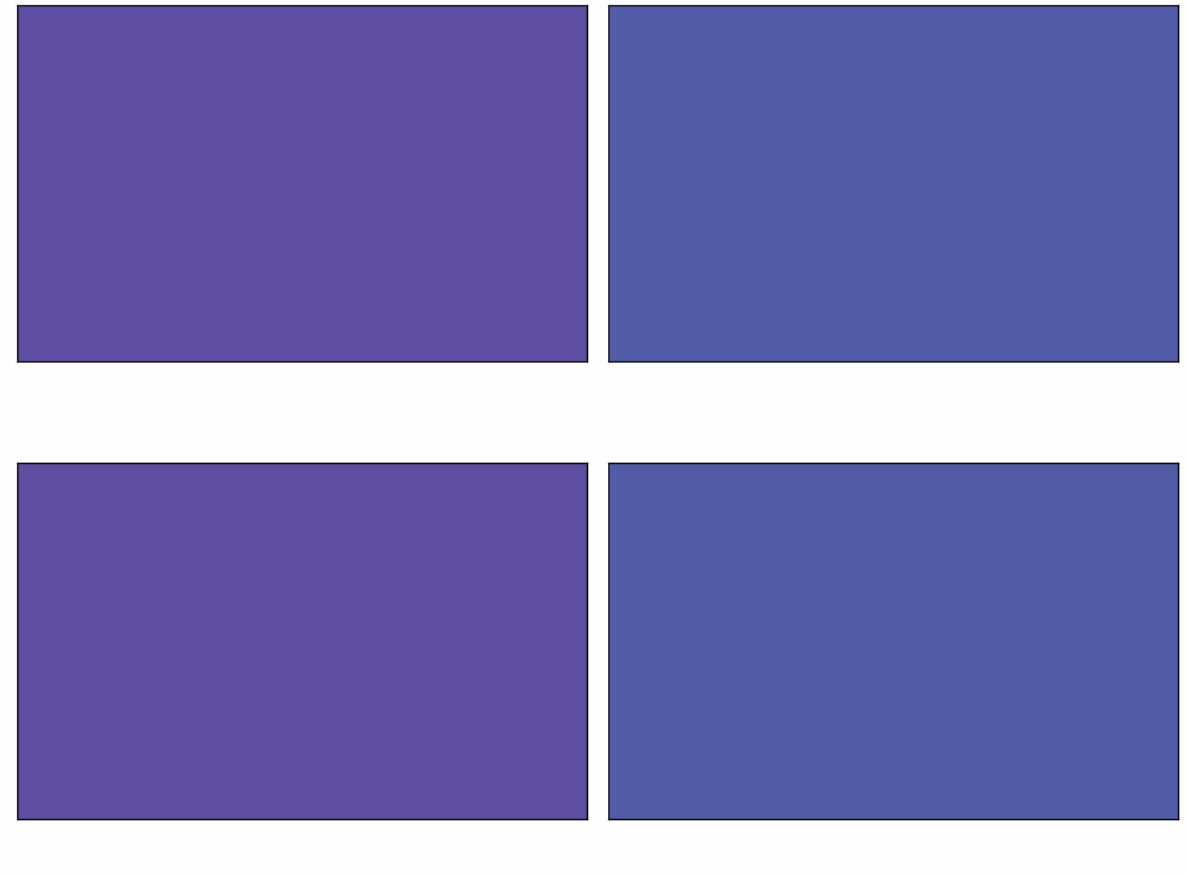
Deep Reduced Order Modeling

Stephen Baek

Motivation: Complex Nonlinear Dynamics







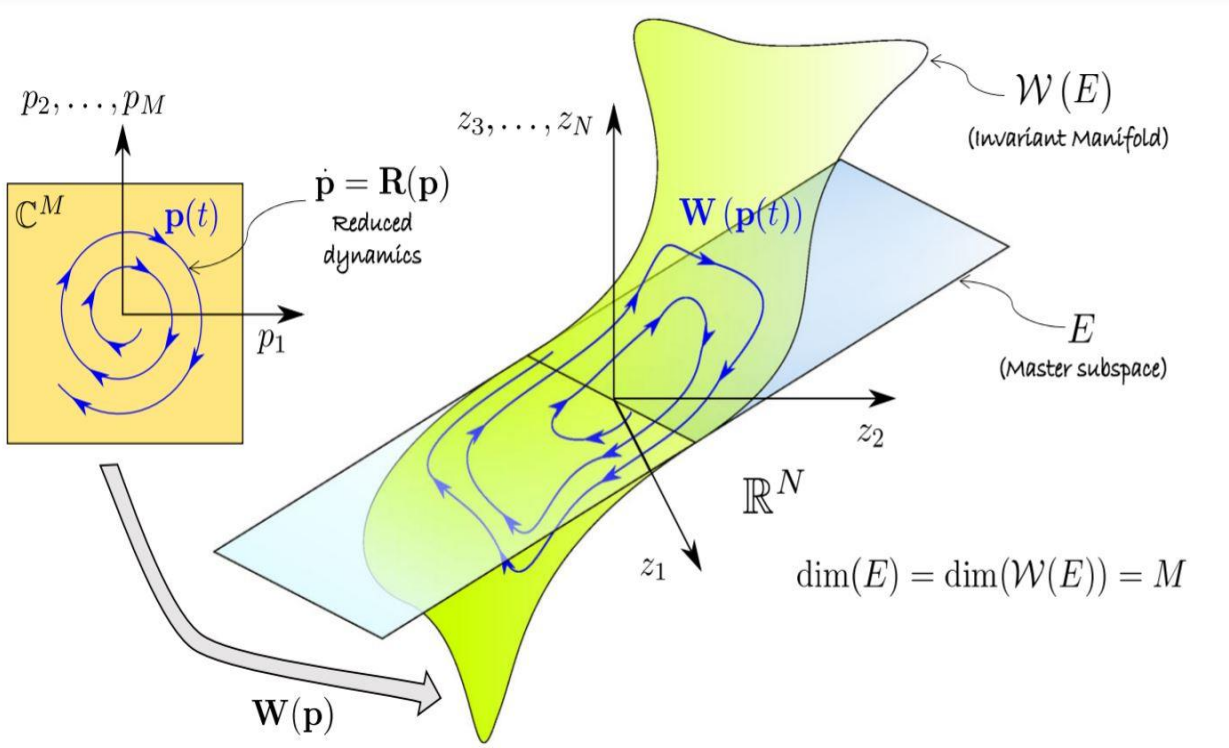
Pore geometry

Temperature evolution

Pressure evolution

Motivation: Encoding Dynamics into Latent Space

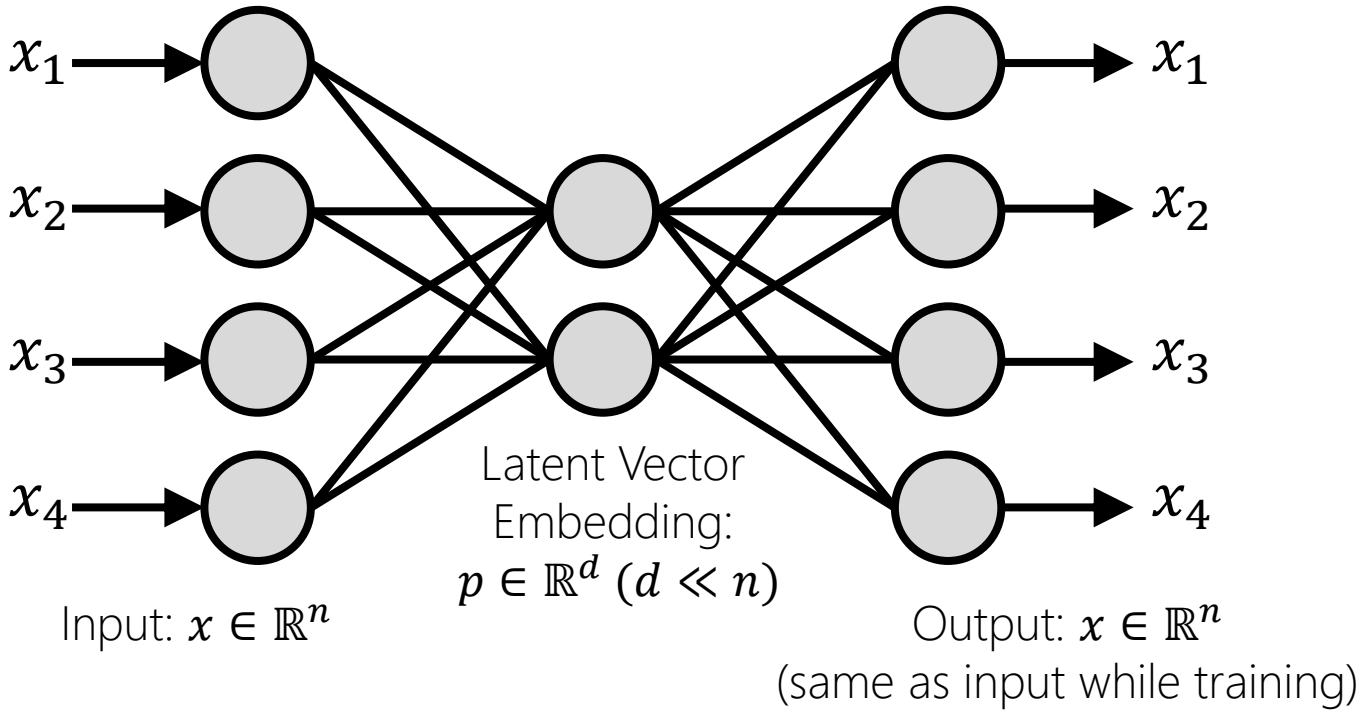
- The effective dynamics of the system happens over a lower dimensional space (dynamics manifold / invariant manifold) embedded into the high dimensional space of raw data.
- Look for an embedding of the dynamics manifold into a latent space which preserve the physical/geometrical properties.
- Learn the reduced dynamics over the latent space.



Jain et al., How to compute invariant manifolds and their reduced dynamics in high dimensional finite element models, 2021

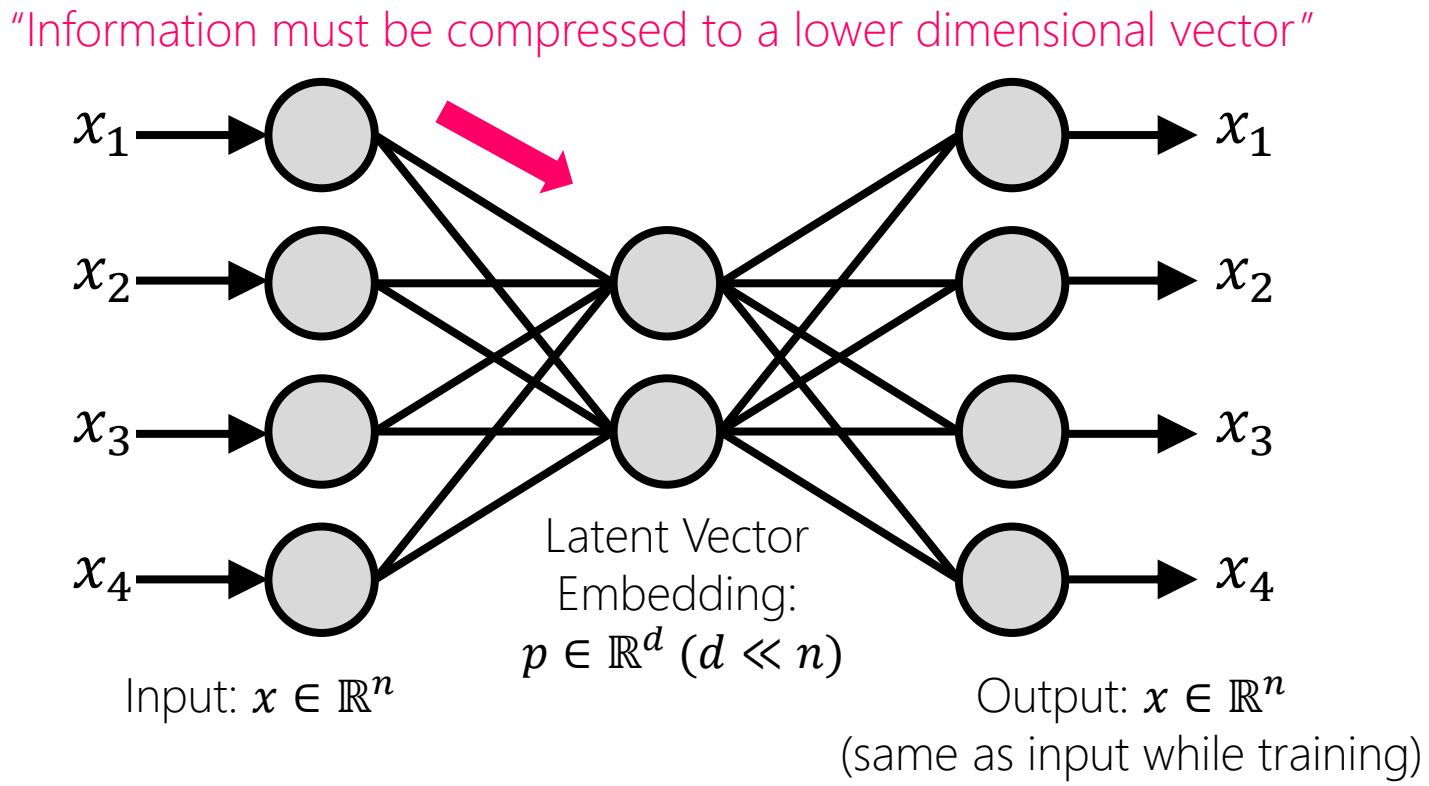
Autoencoder

- MLP with a "Dimensional Bottleneck"



Autoencoder

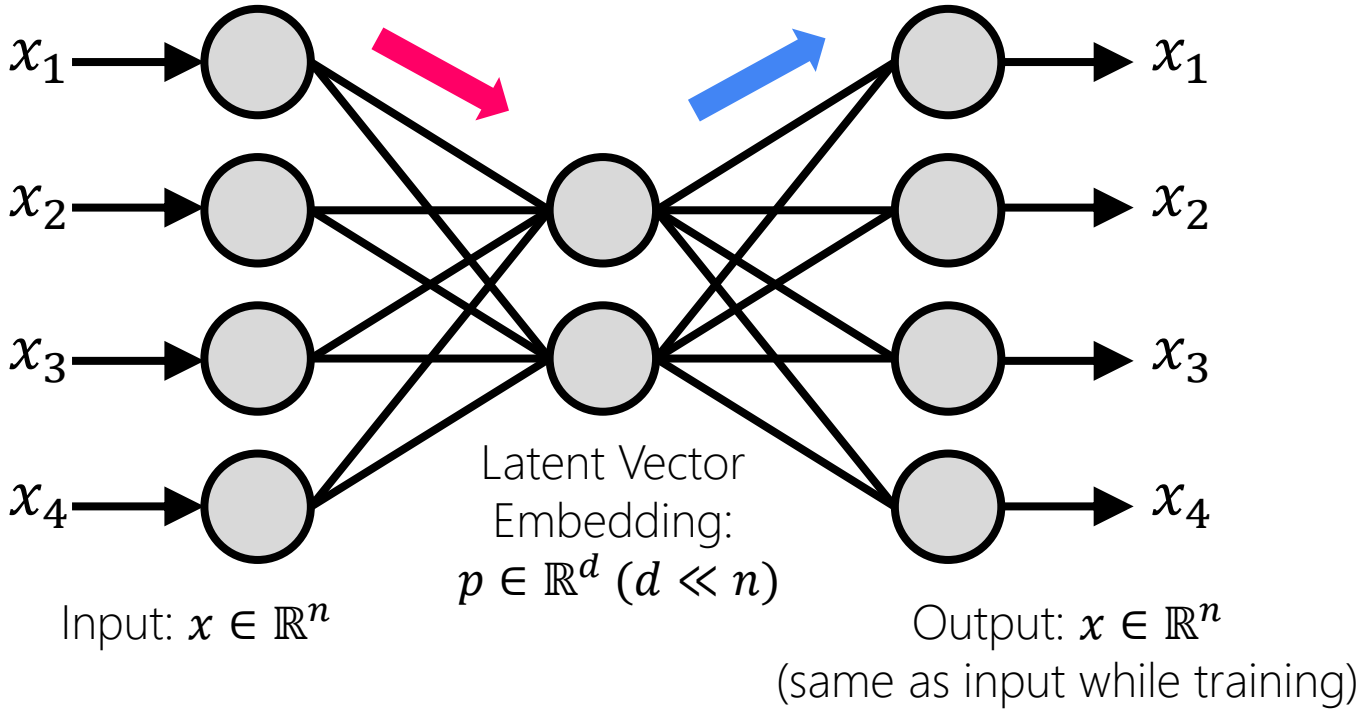
- MLP with a "Dimensional Bottleneck"



Autoencoder

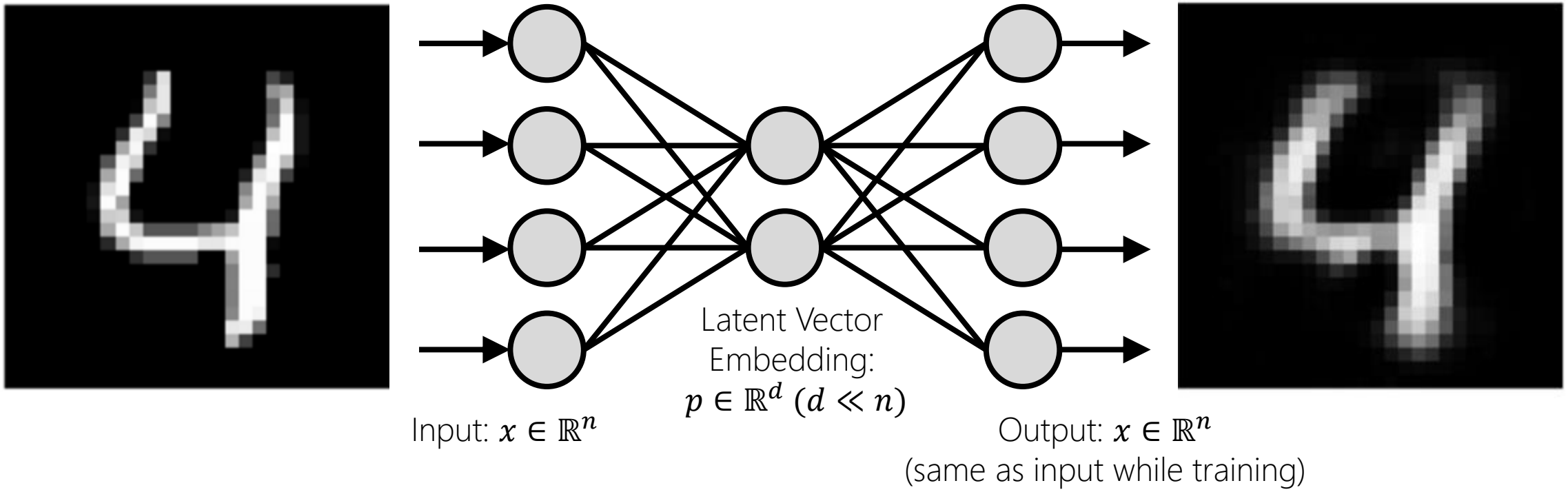
- MLP with a "Dimensional Bottleneck"

"The original information must be reconstructed from the compressed information"



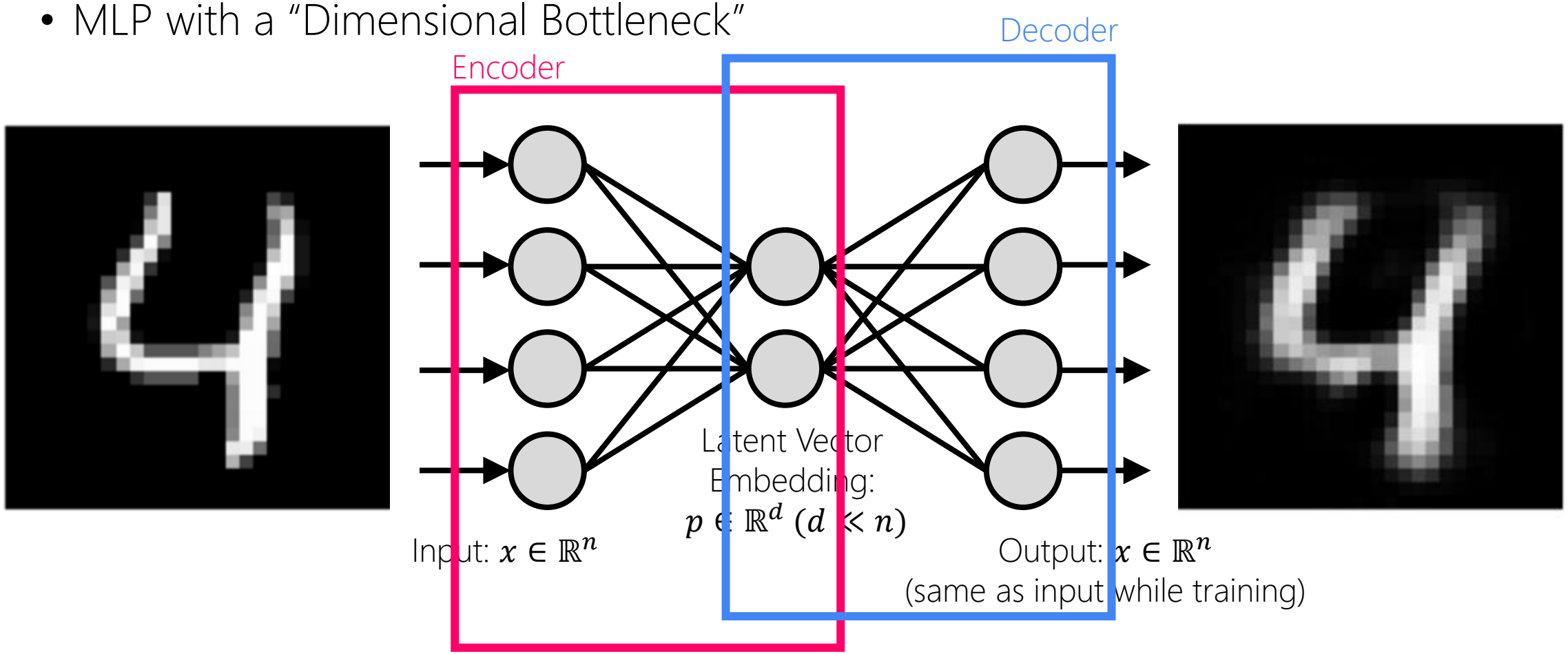
Autoencoder

- MLP with a "Dimensional Bottleneck"



Autoencoder

- MLP with a "Dimensional Bottleneck"



Training Objectives

- Autoencoder:

$$P = f(X) = \sigma_e(XW_e + b_e)$$
$$X' = g(P) = \sigma_p(PW_d + b_d)$$

- Lp norm as reconstruction error

$$\mathcal{L}(W, b \mid x, x') = \|x - x'\|_p$$

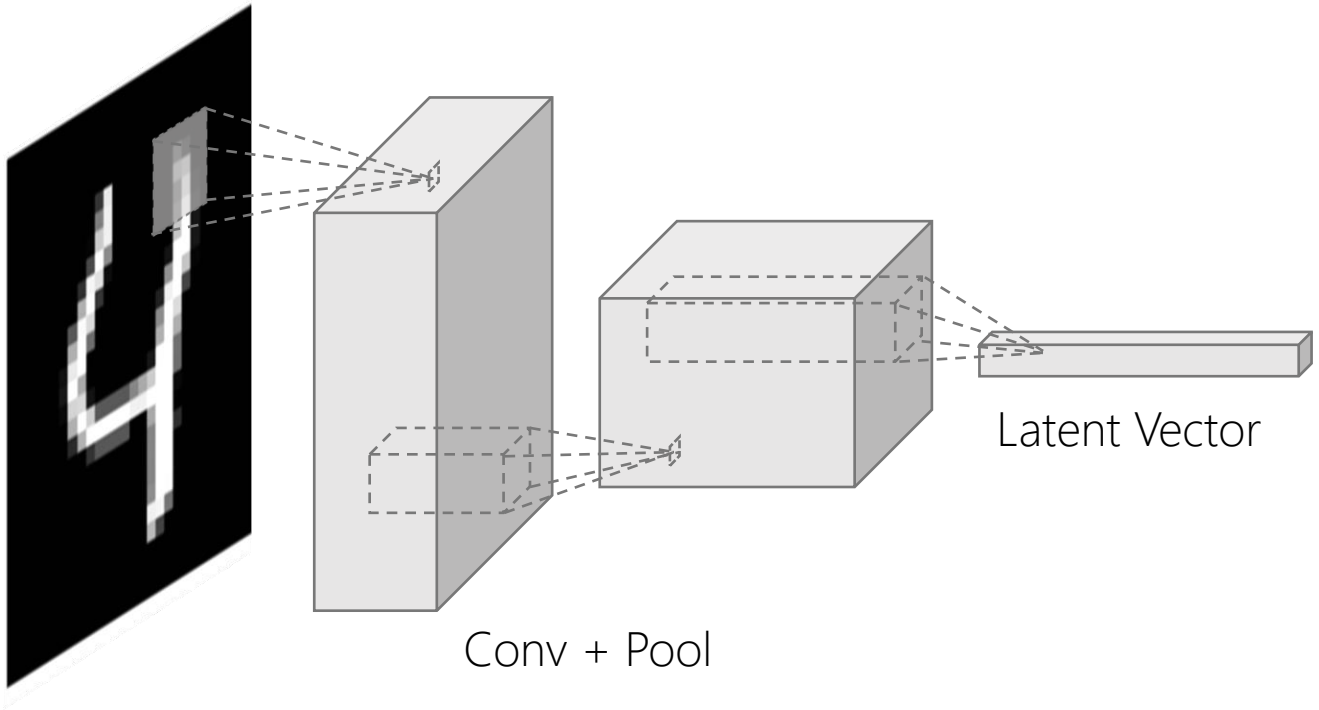
- Cross-entropy as reconstruction error (common for images)

$$\mathcal{L}(W, b \mid x, x') = -\sum_{i=1}^{M} x_i' \log x_i$$

- Sparsity constraints: L1 and/or L2 regularizer

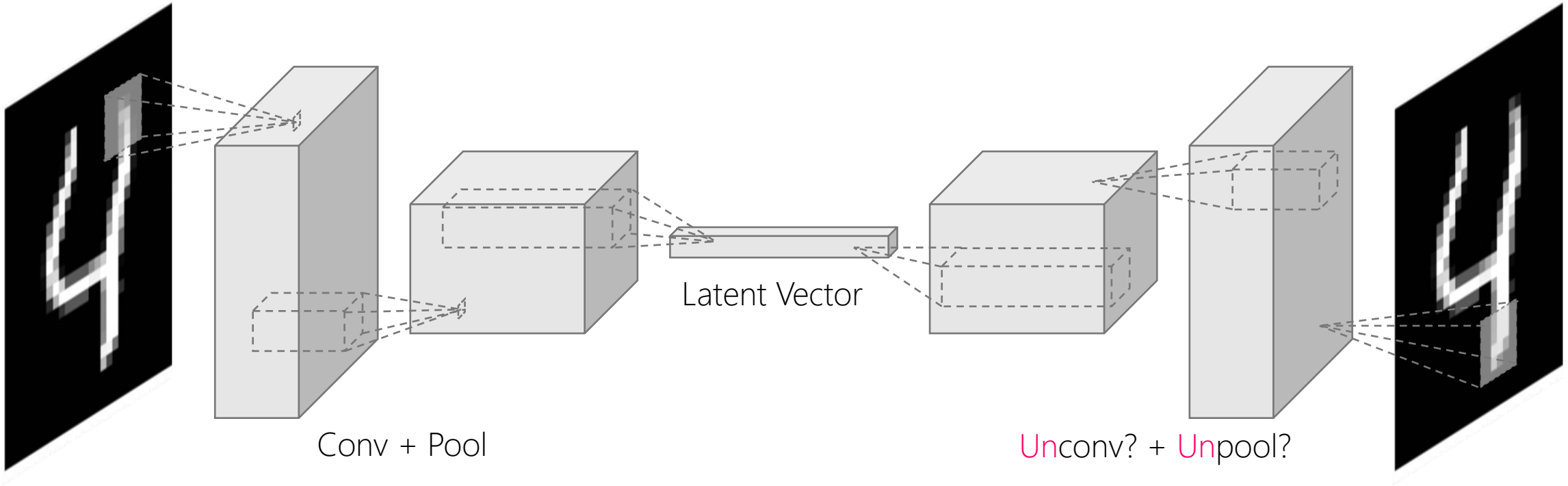
Convolutional Autoencoder

- Convolution layers instead of dense layers.



Convolutional Autoencoder

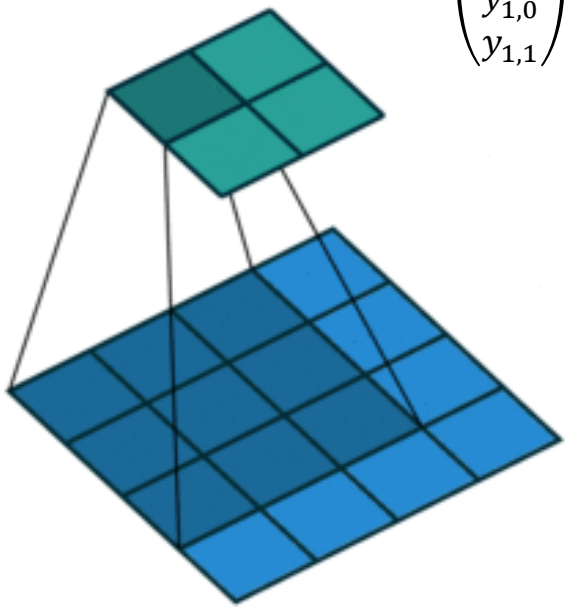
- Convolution layers instead of dense layers.



So, you know what convolution is?

- Convolution as a matrix operation

output



input

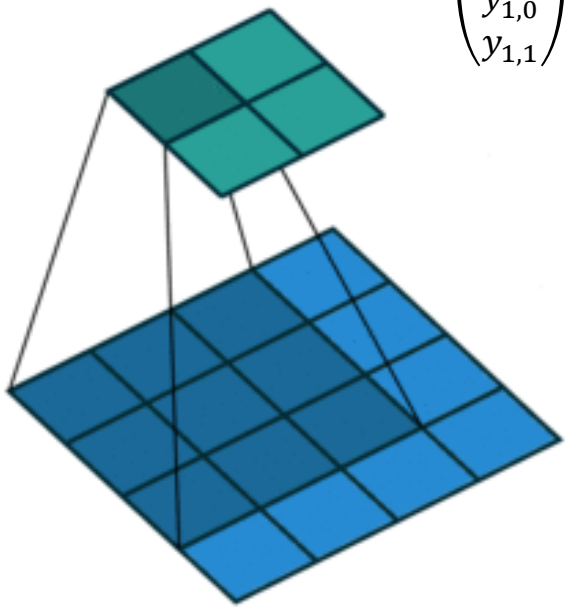
$$\begin{pmatrix} y_{0,0} \\ y_{0,1} \\ y_{1,0} \\ y_{1,1} \end{pmatrix} = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix} \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{1,0} \\ x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{2,0} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \end{pmatrix}$$

Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

So, you know what convolution is?

- Convolution as a matrix operation

output



input

$$\begin{pmatrix} y_{0,0} \\ y_{0,1} \\ y_{1,0} \\ y_{1,1} \end{pmatrix} = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix} \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{1,0} \\ x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{2,0} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \end{pmatrix}$$

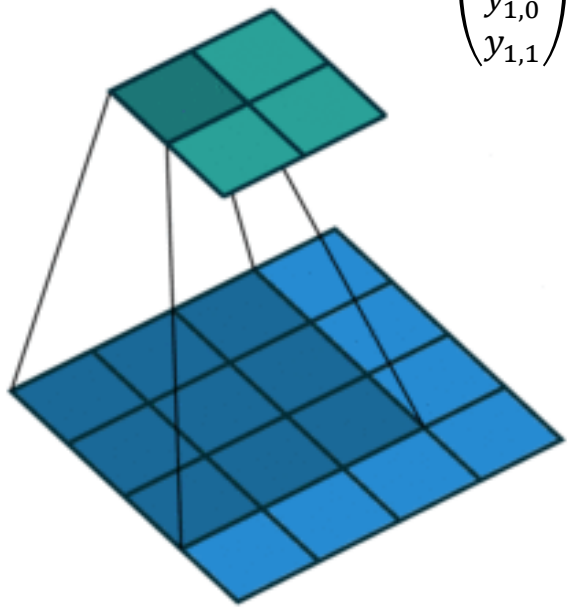
Backprop?

Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

So, you know what convolution is?

- Convolution as a matrix operation

output



input

$$\begin{pmatrix} y_{0,0} \\ y_{0,1} \\ y_{1,0} \\ y_{1,1} \end{pmatrix} = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix} \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{1,0} \\ x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{2,0} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \end{pmatrix}$$

Backprop?

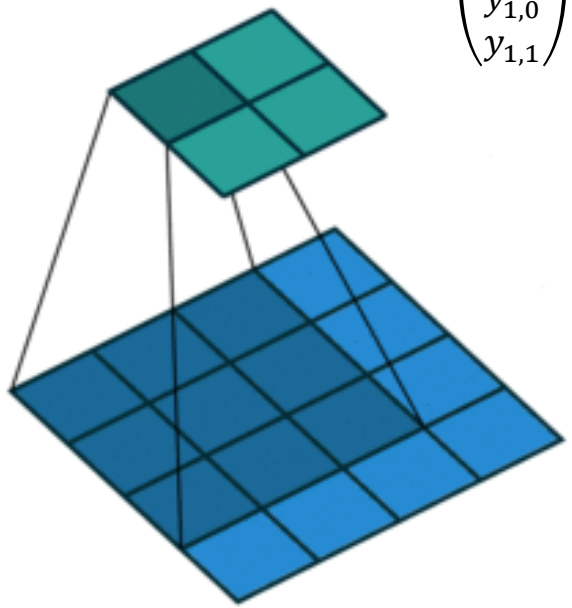
$$\begin{pmatrix} \partial\mathcal{L}/\partial x_{0,0} \\ \partial\mathcal{L}/\partial x_{0,1} \\ \partial\mathcal{L}/\partial x_{0,2} \\ \partial\mathcal{L}/\partial x_{0,3} \\ \partial\mathcal{L}/\partial x_{1,0} \\ \partial\mathcal{L}/\partial x_{1,1} \\ \partial\mathcal{L}/\partial x_{1,2} \\ \partial\mathcal{L}/\partial x_{1,3} \\ \partial\mathcal{L}/\partial x_{2,0} \\ \partial\mathcal{L}/\partial x_{2,1} \\ \partial\mathcal{L}/\partial x_{2,2} \\ \partial\mathcal{L}/\partial x_{2,3} \\ \partial\mathcal{L}/\partial x_{3,0} \\ \partial\mathcal{L}/\partial x_{3,1} \\ \partial\mathcal{L}/\partial x_{3,2} \\ \partial\mathcal{L}/\partial x_{3,3} \end{pmatrix} =$$

Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

So, you know what convolution is?

- Convolution as a matrix operation

output



input

$$\begin{pmatrix} y_{0,0} \\ y_{0,1} \\ y_{1,0} \\ y_{1,1} \end{pmatrix} = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix} \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{1,0} \\ x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{2,0} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \end{pmatrix}$$

Backprop?

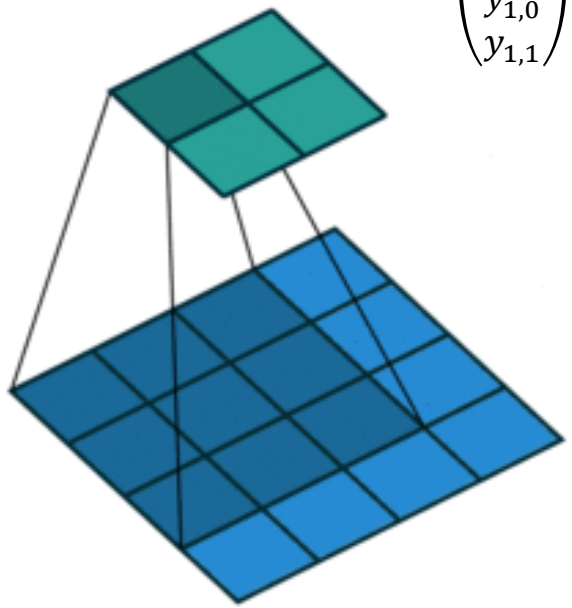
$$\begin{pmatrix} \partial\mathcal{L}/\partial x_{0,0} \\ \partial\mathcal{L}/\partial x_{0,1} \\ \partial\mathcal{L}/\partial x_{0,2} \\ \partial\mathcal{L}/\partial x_{0,3} \\ \partial\mathcal{L}/\partial x_{1,0} \\ \partial\mathcal{L}/\partial x_{1,1} \\ \partial\mathcal{L}/\partial x_{1,2} \\ \partial\mathcal{L}/\partial x_{1,3} \\ \partial\mathcal{L}/\partial x_{2,0} \\ \partial\mathcal{L}/\partial x_{2,1} \\ \partial\mathcal{L}/\partial x_{2,2} \\ \partial\mathcal{L}/\partial x_{2,3} \\ \partial\mathcal{L}/\partial x_{3,0} \\ \partial\mathcal{L}/\partial x_{3,1} \\ \partial\mathcal{L}/\partial x_{3,2} \\ \partial\mathcal{L}/\partial x_{3,3} \end{pmatrix} = \left(\frac{\partial y}{\partial x}\right)\left(\frac{\partial \mathcal{L}}{\partial y}\right)$$

Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

So, you know what convolution is?

- Convolution as a matrix operation

output



input

$$\begin{pmatrix} y_{0,0} \\ y_{0,1} \\ y_{1,0} \\ y_{1,1} \end{pmatrix} = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix} \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{1,0} \\ x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{2,0} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \end{pmatrix}$$

Backprop?

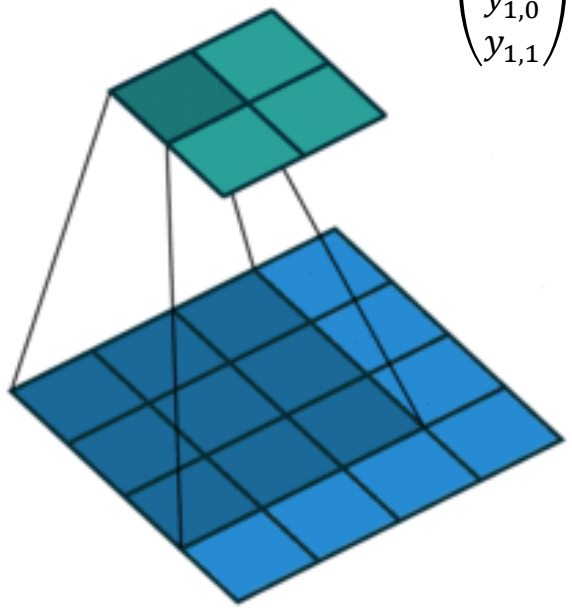
$$\begin{pmatrix} \partial\mathcal{L}/\partial x_{0,0} \\ \partial\mathcal{L}/\partial x_{0,1} \\ \partial\mathcal{L}/\partial x_{0,2} \\ \partial\mathcal{L}/\partial x_{0,3} \\ \partial\mathcal{L}/\partial x_{1,0} \\ \partial\mathcal{L}/\partial x_{1,1} \\ \partial\mathcal{L}/\partial x_{1,2} \\ \partial\mathcal{L}/\partial x_{1,3} \\ \partial\mathcal{L}/\partial x_{2,0} \\ \partial\mathcal{L}/\partial x_{2,1} \\ \partial\mathcal{L}/\partial x_{2,2} \\ \partial\mathcal{L}/\partial x_{2,3} \\ \partial\mathcal{L}/\partial x_{3,0} \\ \partial\mathcal{L}/\partial x_{3,1} \\ \partial\mathcal{L}/\partial x_{3,2} \\ \partial\mathcal{L}/\partial x_{3,3} \end{pmatrix} = \begin{pmatrix} w_{0,0} & 0 & 0 & 0 \\ w_{0,1} & w_{0,0} & 0 & 0 \\ w_{0,2} & w_{0,1} & 0 & 0 \\ 0 & w_{0,2} & 0 & 0 \\ w_{1,0} & 0 & w_{0,0} & 0 \\ w_{1,1} & w_{1,0} & w_{0,1} & w_{0,0} \\ w_{1,2} & w_{1,1} & w_{0,2} & w_{0,1} \\ 0 & w_{1,2} & 0 & w_{0,2} \\ w_{2,0} & 0 & w_{1,0} & 0 \\ w_{2,1} & w_{2,0} & w_{1,1} & w_{1,0} \\ w_{2,2} & w_{2,1} & w_{1,2} & w_{1,1} \\ 0 & w_{2,2} & 0 & w_{1,2} \\ 0 & 0 & w_{2,0} & 0 \\ 0 & 0 & w_{2,1} & w_{2,0} \\ 0 & 0 & w_{2,2} & w_{2,1} \\ 0 & 0 & 0 & w_{2,2} \end{pmatrix} \begin{pmatrix} \partial\mathcal{L}/\partial y_{0,0} \\ \partial\mathcal{L}/\partial y_{0,1} \\ \partial\mathcal{L}/\partial y_{1,0} \\ \partial\mathcal{L}/\partial y_{1,1} \end{pmatrix}$$

Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

So, you know what convolution is?

- Convolution as a matrix operation

output



input

$$\begin{pmatrix} y_{0,0} \\ y_{0,1} \\ y_{1,0} \\ y_{1,1} \end{pmatrix} = \underbrace{\begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix}}_{C} \begin{pmatrix} x_{0,0} \\ x_{0,1} \\ x_{0,2} \\ x_{0,3} \\ x_{1,0} \\ x_{1,1} \\ x_{1,2} \\ x_{1,3} \\ x_{2,0} \\ x_{2,1} \\ x_{2,2} \\ x_{2,3} \\ x_{3,0} \\ x_{3,1} \\ x_{3,2} \\ x_{3,3} \end{pmatrix}$$

Backprop?

$$\begin{pmatrix} \partial\mathcal{L}/\partial x_{0,0} \\ \partial\mathcal{L}/\partial x_{0,1} \\ \partial\mathcal{L}/\partial x_{0,2} \\ \partial\mathcal{L}/\partial x_{0,3} \\ \partial\mathcal{L}/\partial x_{1,0} \\ \partial\mathcal{L}/\partial x_{1,1} \\ \partial\mathcal{L}/\partial x_{1,2} \\ \partial\mathcal{L}/\partial x_{1,3} \\ \partial\mathcal{L}/\partial x_{2,0} \\ \partial\mathcal{L}/\partial x_{2,1} \\ \partial\mathcal{L}/\partial x_{2,2} \\ \partial\mathcal{L}/\partial x_{2,3} \\ \partial\mathcal{L}/\partial x_{3,0} \\ \partial\mathcal{L}/\partial x_{3,1} \\ \partial\mathcal{L}/\partial x_{3,2} \\ \partial\mathcal{L}/\partial x_{3,3} \end{pmatrix} = \underbrace{\begin{pmatrix} w_{0,0} & 0 & 0 & 0 \\ w_{0,1} & w_{0,0} & 0 & 0 \\ w_{0,2} & w_{0,1} & 0 & 0 \\ 0 & w_{0,2} & 0 & 0 \\ w_{1,0} & 0 & w_{0,0} & 0 \\ w_{1,1} & w_{1,0} & w_{0,1} & w_{0,0} \\ w_{1,2} & w_{1,1} & w_{0,2} & w_{0,1} \\ 0 & w_{1,2} & 0 & w_{0,2} \\ w_{2,0} & 0 & w_{1,0} & 0 \\ w_{2,1} & w_{2,0} & w_{1,1} & w_{1,0} \\ w_{2,2} & w_{2,1} & w_{1,2} & w_{1,1} \\ 0 & w_{2,2} & 0 & w_{1,2} \\ 0 & 0 & w_{2,0} & 0 \\ 0 & 0 & w_{2,1} & w_{2,0} \\ 0 & 0 & w_{2,2} & w_{2,1} \\ 0 & 0 & 0 & w_{2,2} \end{pmatrix}}_{C'} \begin{pmatrix} \partial\mathcal{L}/\partial y_{0,0} \\ \partial\mathcal{L}/\partial y_{0,1} \\ \partial\mathcal{L}/\partial y_{1,0} \\ \partial\mathcal{L}/\partial y_{1,1} \end{pmatrix}$$

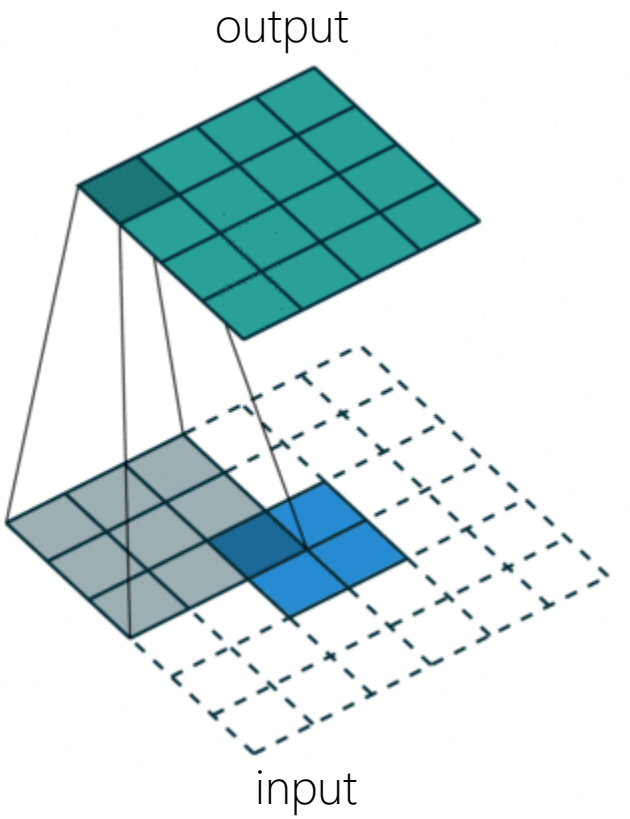
Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

Transposed Convolution

- Other names:
  - Deconvolution (wrong)
  - Upconvolution
  - Fractionally strided convolutions
  - Backward strided convolutions

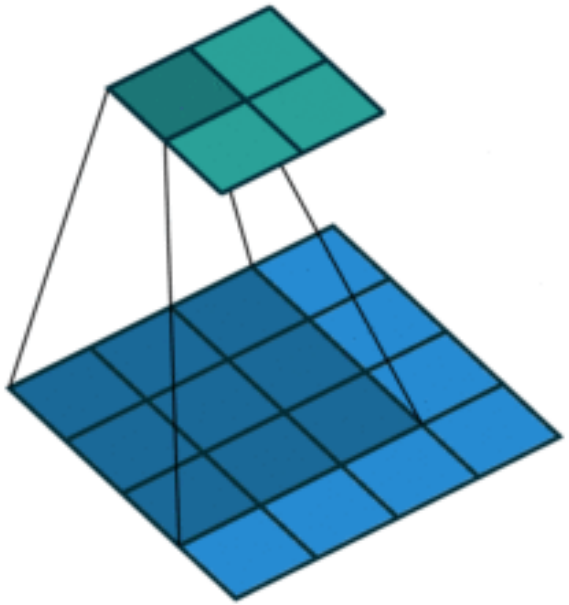
- Basically the same as (regular) conv!

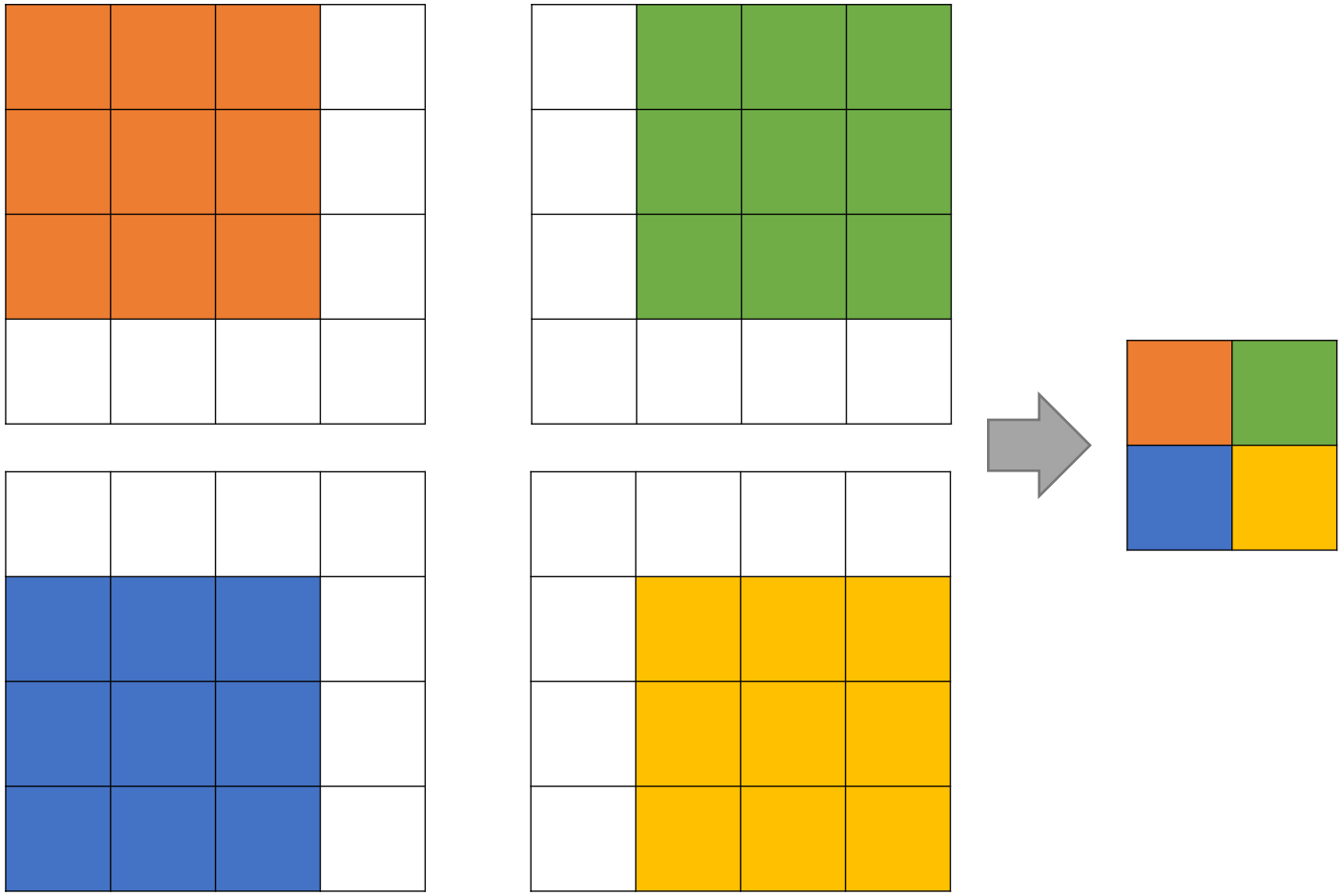
$$\begin{pmatrix} w_{0,0} & 0 & 0 & 0 \\ w_{0,1} & w_{0,0} & 0 & 0 \\ w_{0,2} & w_{0,1} & 0 & 0 \\ 0 & w_{0,2} & 0 & 0 \\ w_{1,0} & 0 & w_{0,0} & 0 \\ w_{1,1} & w_{1,0} & w_{0,1} & w_{0,0} \\ w_{1,2} & w_{1,1} & w_{0,2} & w_{0,1} \\ 0 & w_{1,2} & 0 & w_{0,2} \\ w_{2,0} & 0 & w_{1,0} & 0 \\ w_{2,1} & w_{2,0} & w_{1,1} & w_{1,0} \\ w_{2,2} & w_{2,1} & w_{1,2} & w_{1,1} \\ 0 & w_{2,2} & 0 & w_{1,2} \\ 0 & 0 & w_{2,0} & 0 \\ 0 & 0 & w_{2,1} & w_{2,0} \\ 0 & 0 & w_{2,2} & w_{2,1} \\ 0 & 0 & 0 & w_{2,2} \end{pmatrix}$$



Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

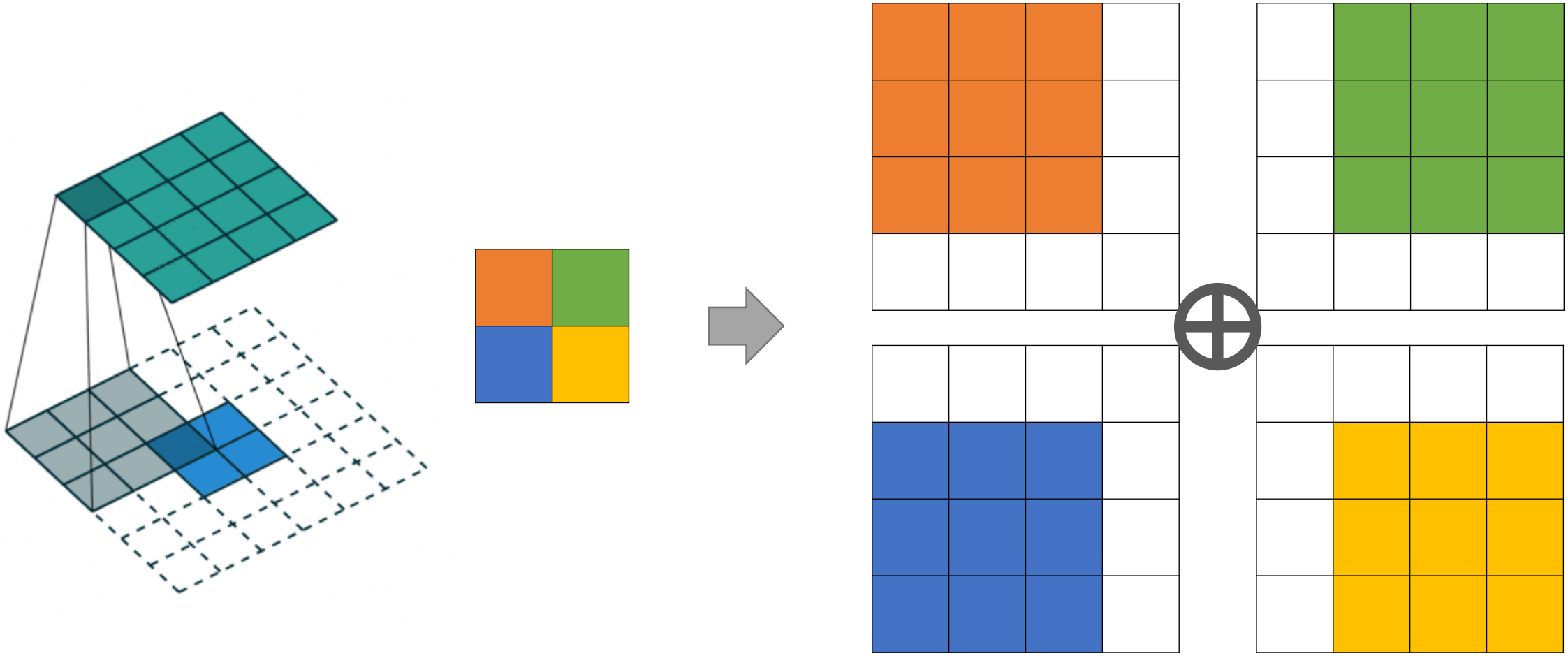
“Unconvolving” Feature Maps





Dumoulin and Visin, “A guide to convolution arithmetic for deep learning”, 2018

“Unconvolving” Feature Maps

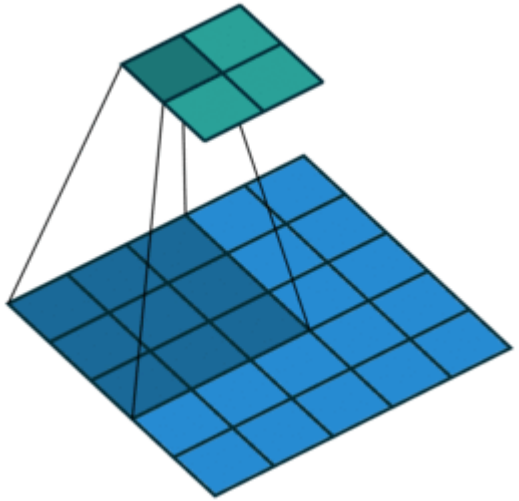


Dumoulin and Visin, “A guide to convolution arithmetic for deep learning”, 2018

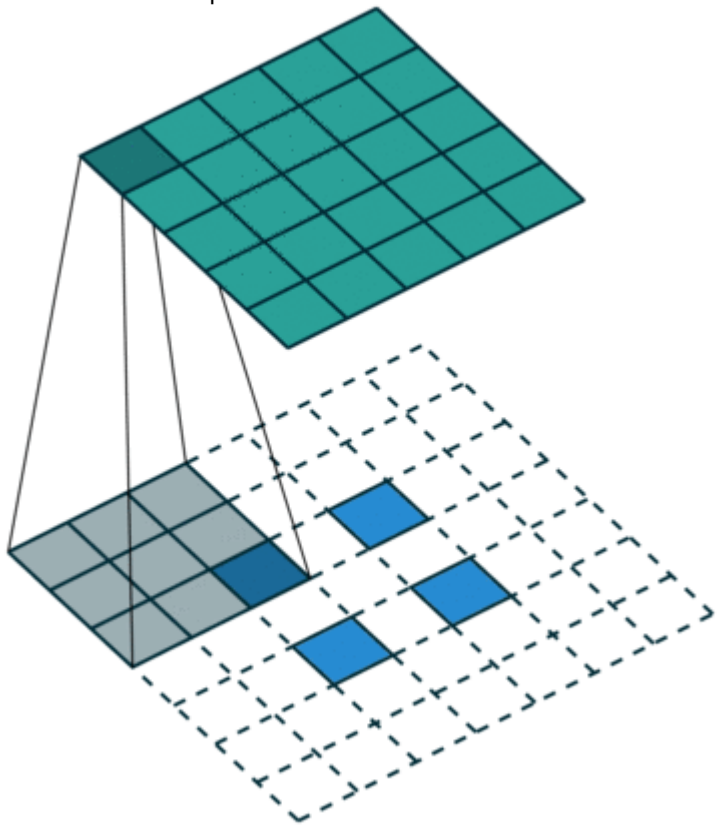
Transposed Convolution

- This holds true for arbitrary padding and strides

Conv



Transposed Conv

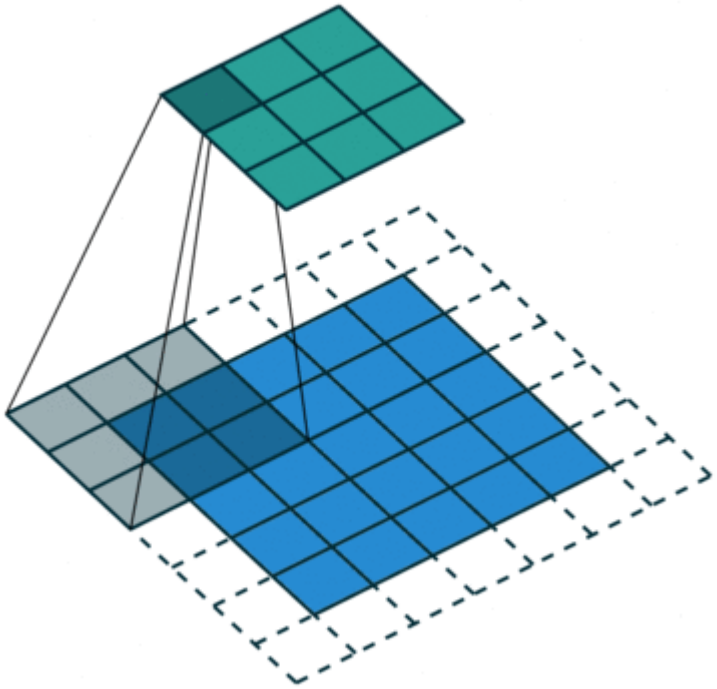


Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

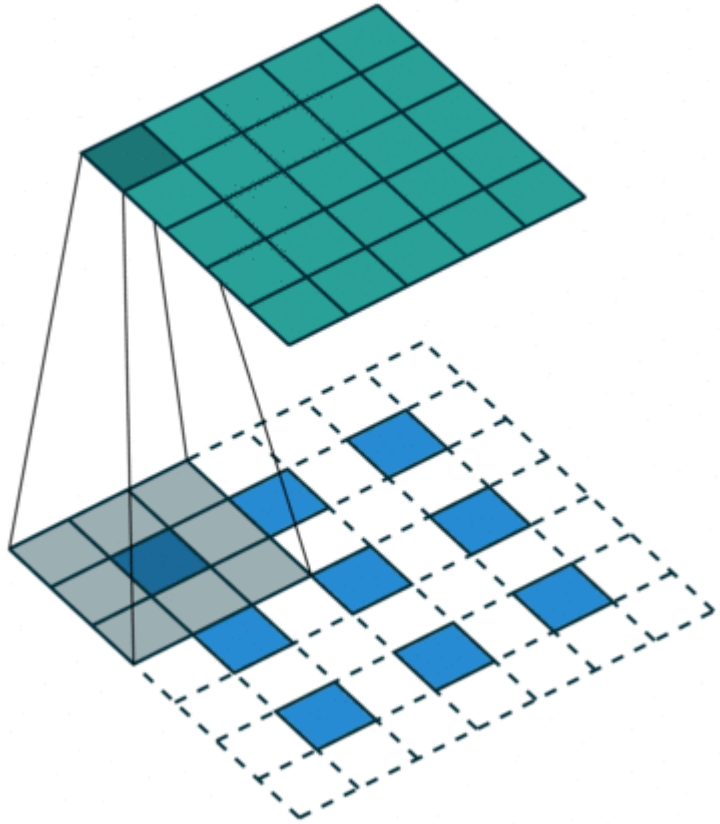
Transposed Convolution

- This holds true for arbitrary padding and strides

Conv



Transposed Conv

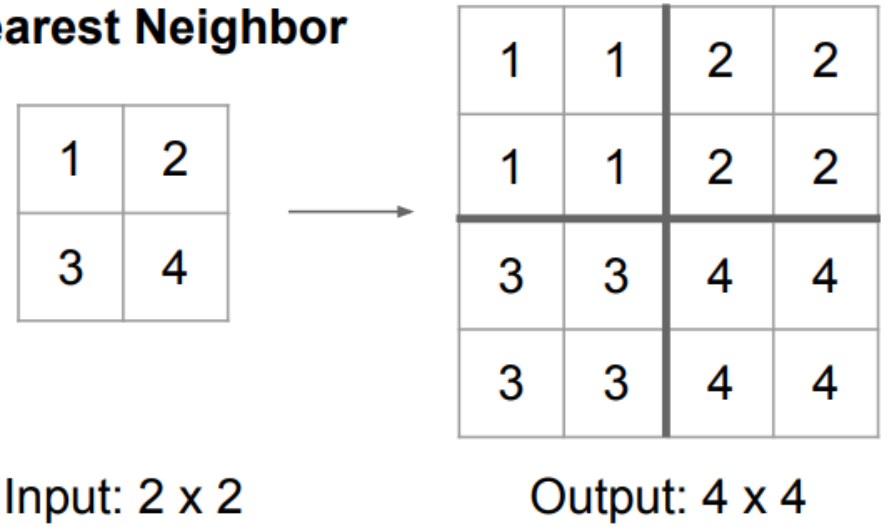


Dumoulin and Visin, "A guide to convolution arithmetic for deep learning", 2018

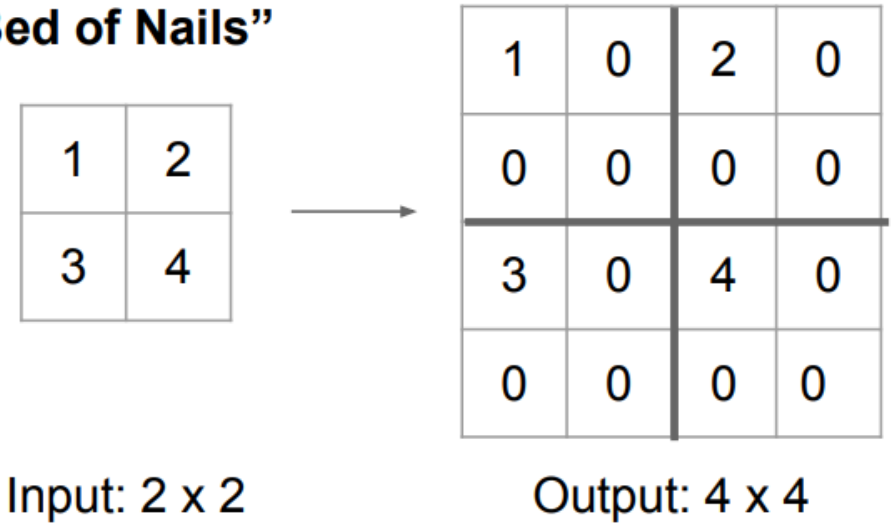
Unpooling

- Naïve strategies

Nearest Neighbor

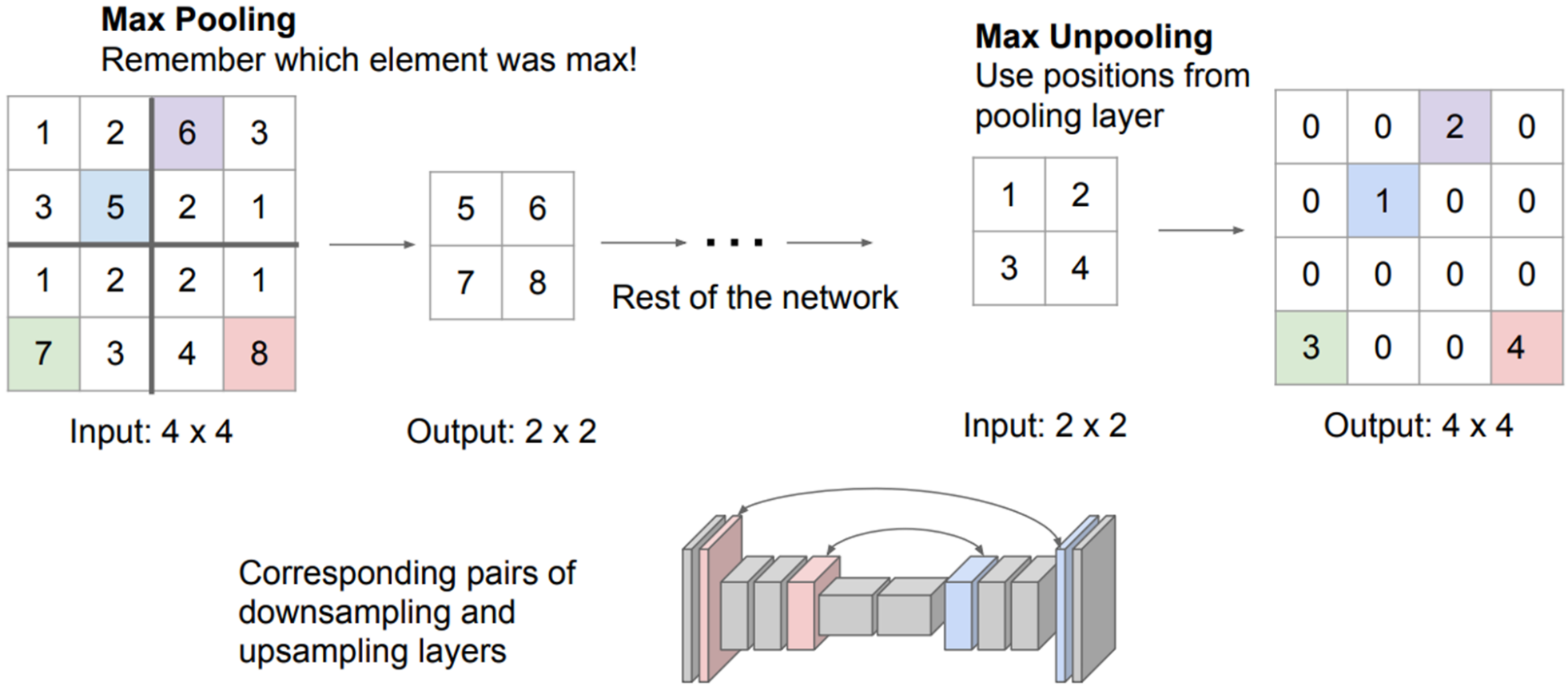


“Bed of Nails”

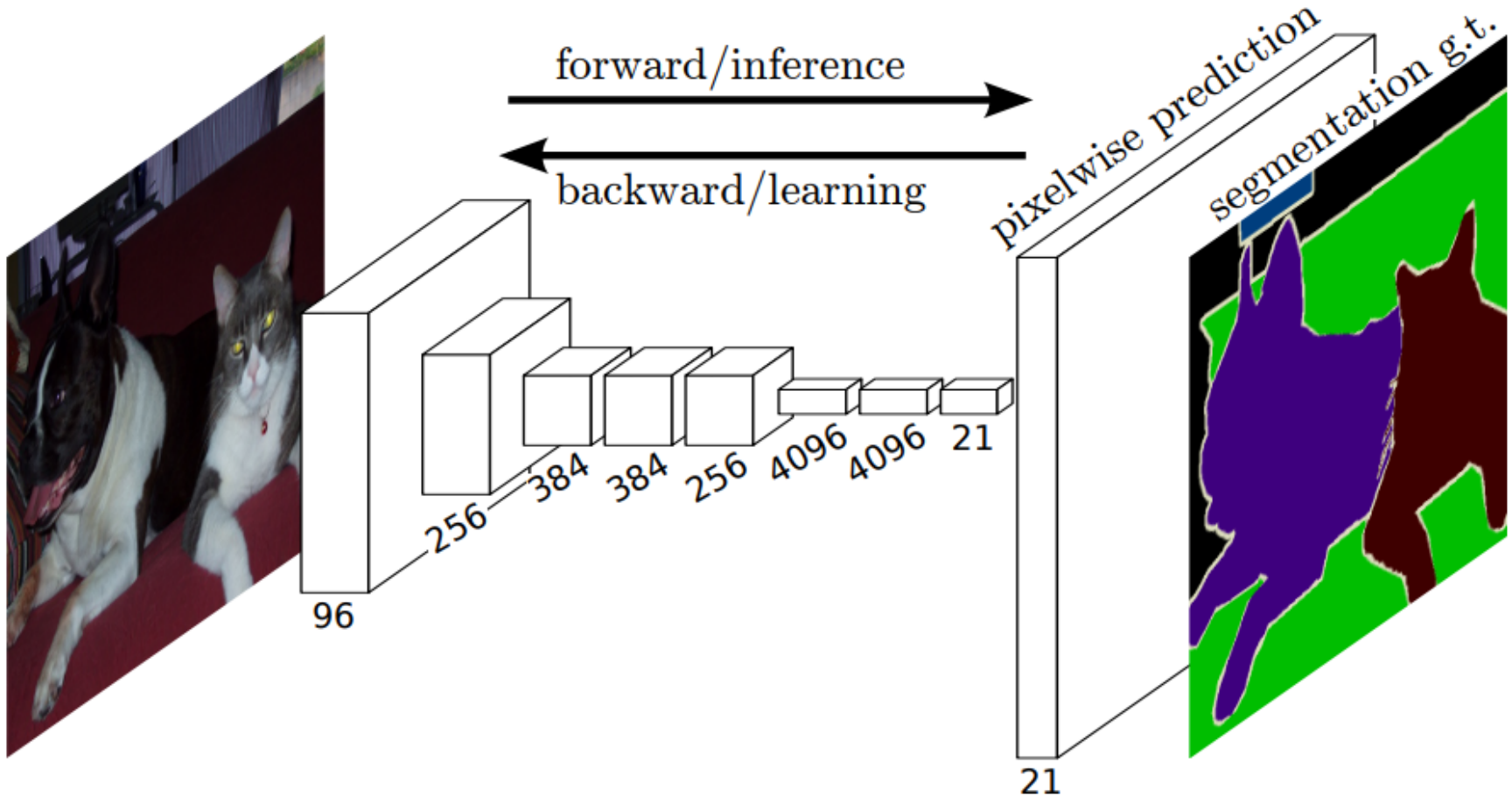


Unpooling

- Max Unpooling

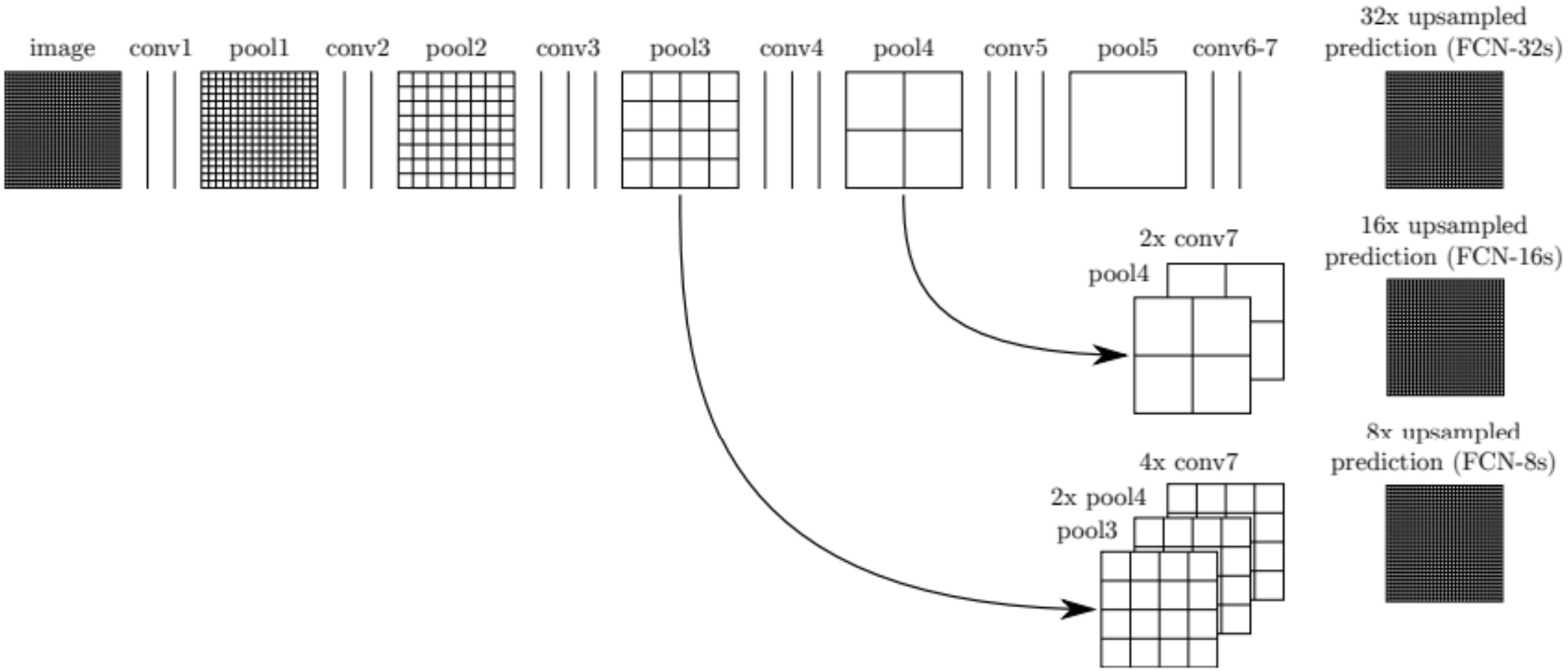


Fully Convolutional Networks



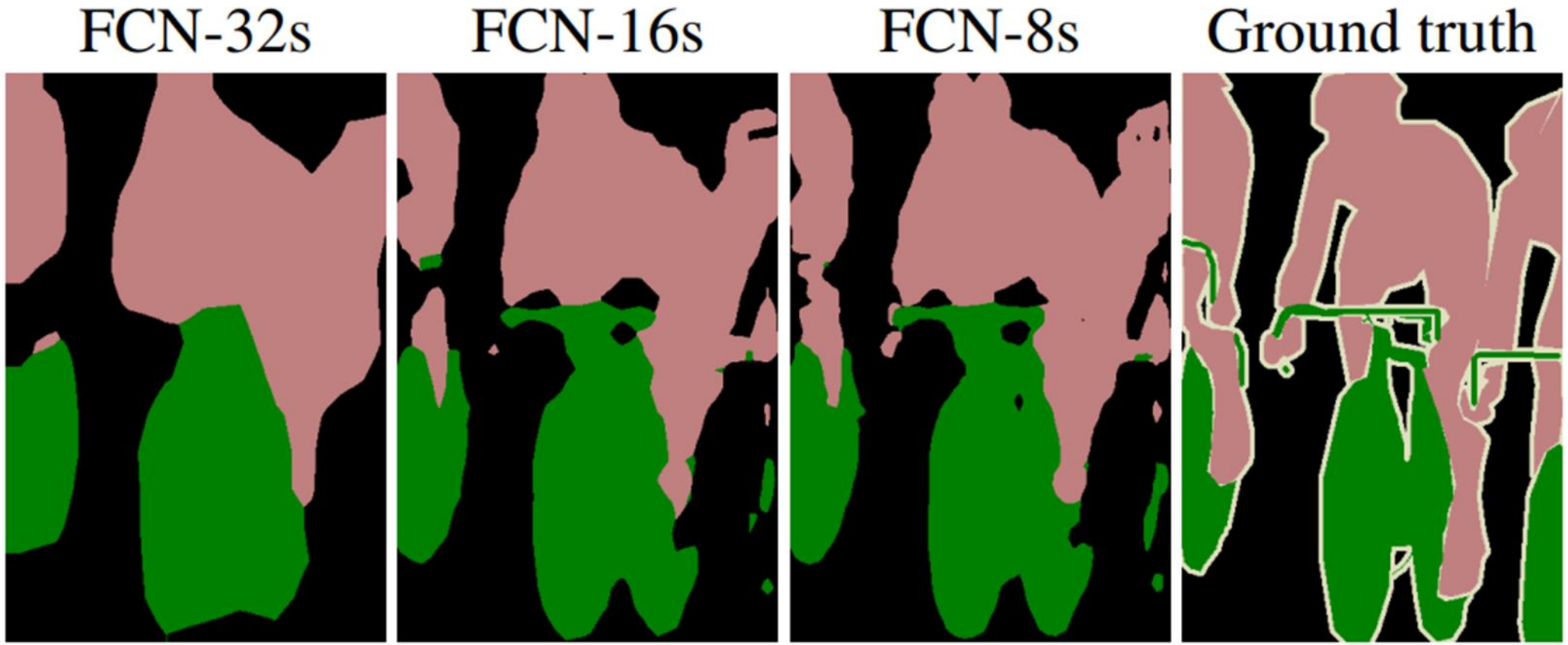
Long, Shelhamer, & Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR2015

Fully Convolutional Networks



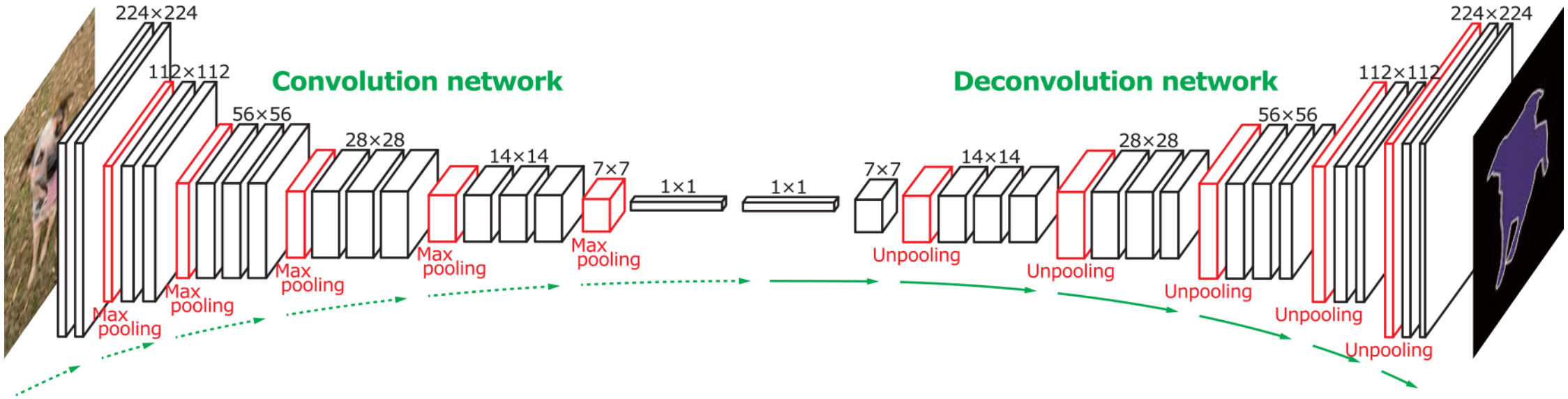
Long, Shelhamer, & Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR2015

Fully Convolutional Networks



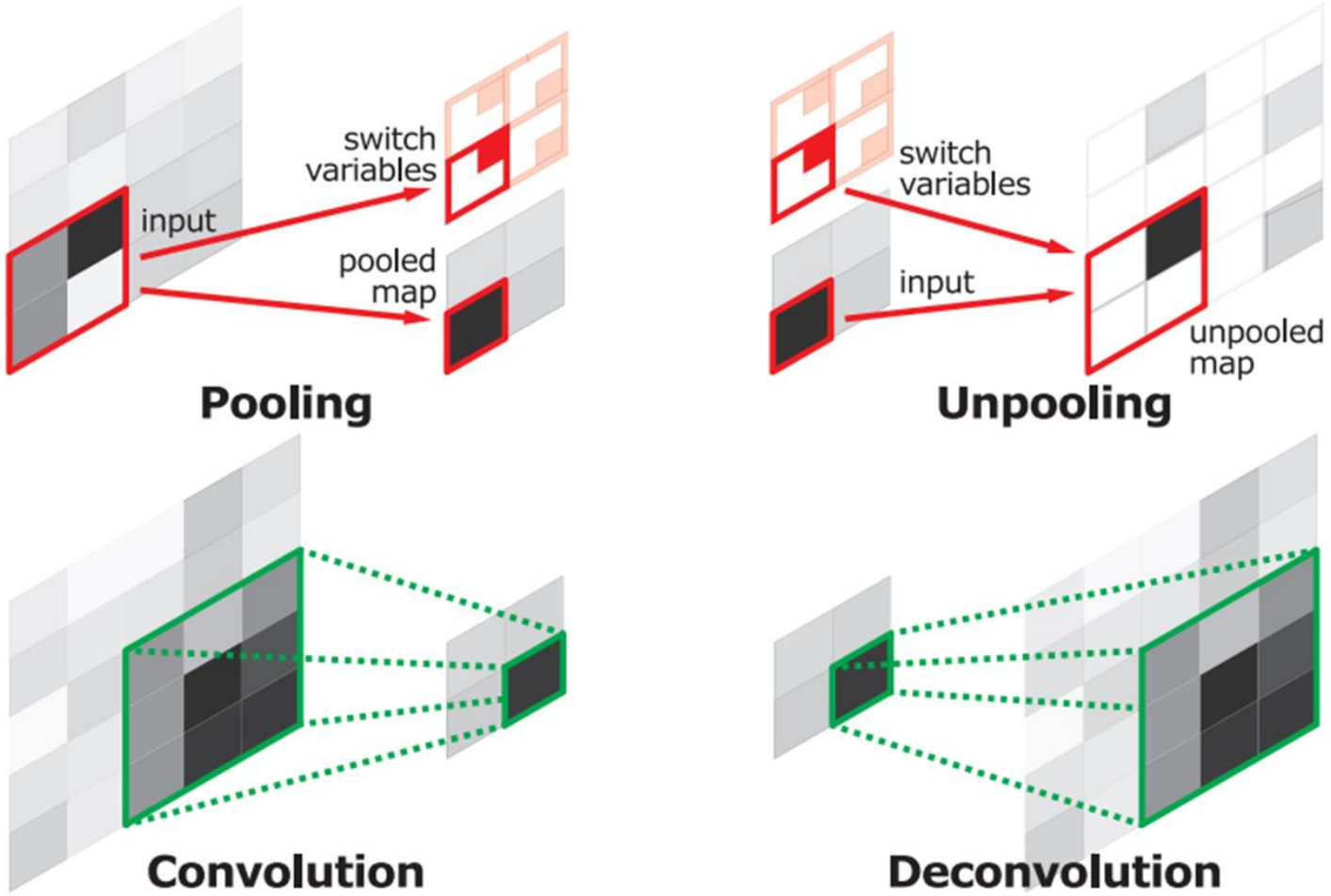
Long, Shelhamer, & Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR2015

Deconvolutional Networks



Noh, Hong, & Han, "Learning Deconvolution Network for Semantic Segmentation", ICCV2015

Deconvolutional Networks



Noh, Hong, & Han, "Learning Deconvolution Network for Semantic Segmentation", ICCV2015

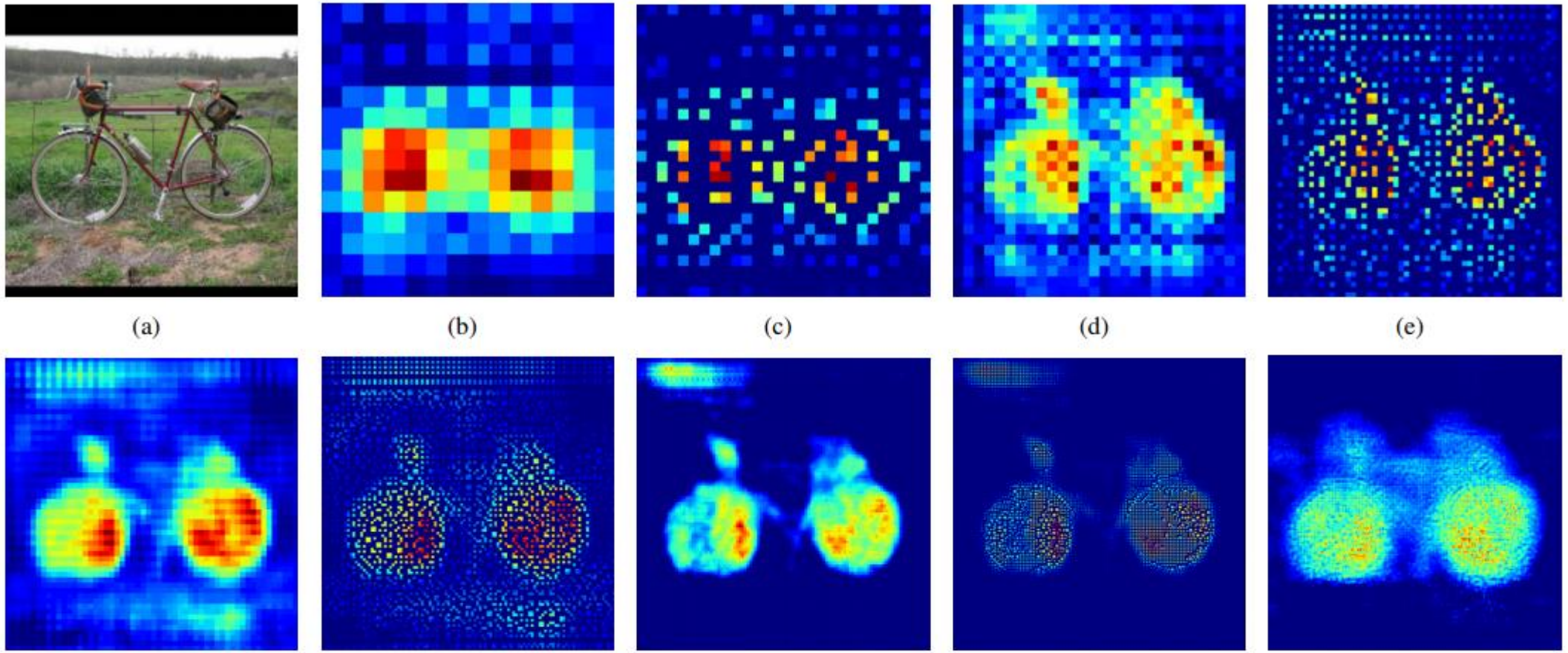
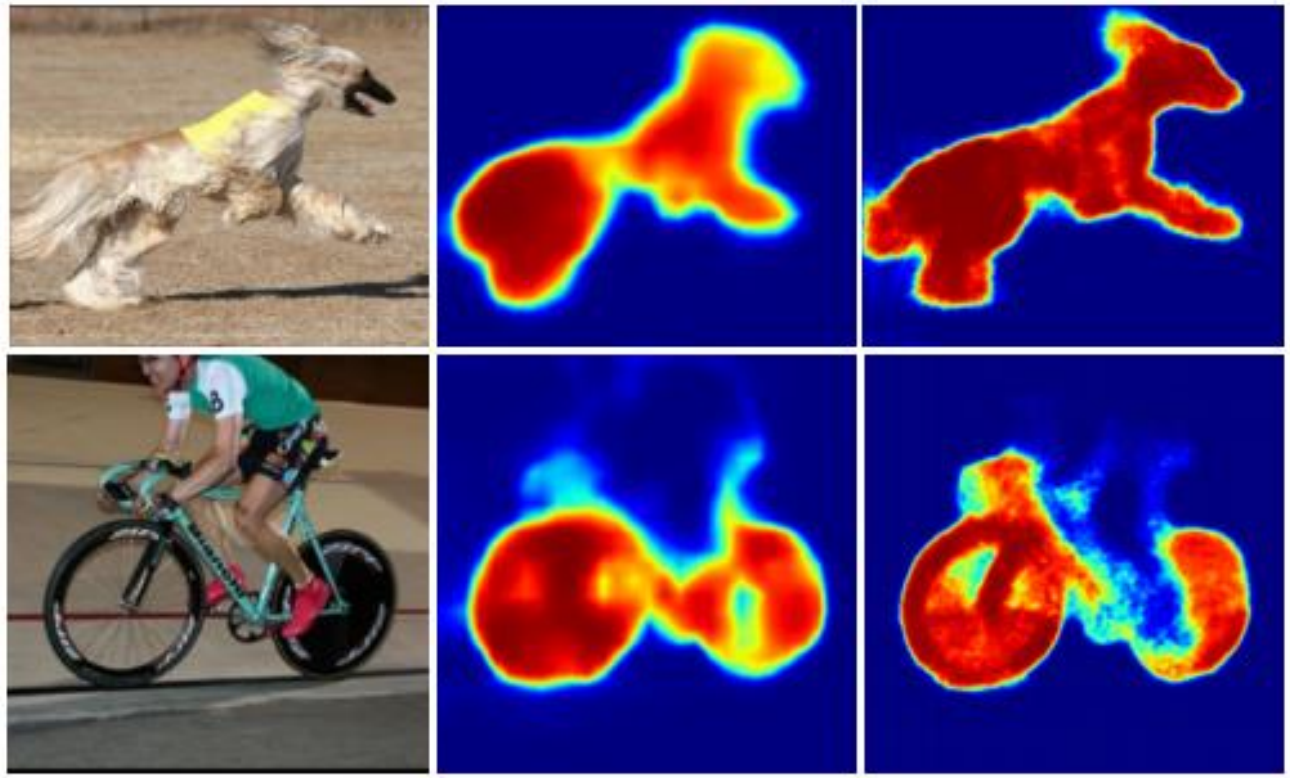


Figure 4. Visualization of activations in our deconvolution network. The activation maps from top left to bottom right correspond to the output maps from lower to higher layers in the deconvolution network. We select the most representative activation in each layer for effective visualization. The image in (a) is an input, and the rest are the outputs from (b) the last 14×14 deconvolutional layer, (c) the 28×28 unpooling layer, (d) the last 28×28 deconvolutional layer, (e) the 56×56 unpooling layer, (f) the last 56×56 deconvolutional layer, (g) the 112×112 unpooling layer, (h) the last 112×112 deconvolutional layer, (i) the 224×224 unpooling layer and (j) the last 224×224 deconvolutional layer. The finer details of the object are revealed, as the features are forward-propagated through the layers in the deconvolution network. Note that noisy activations from background are suppressed through propagation while the activations closely related to the target classes are amplified. It shows that the learned filters in higher deconvolutional layers tend to capture class-specific shape information.

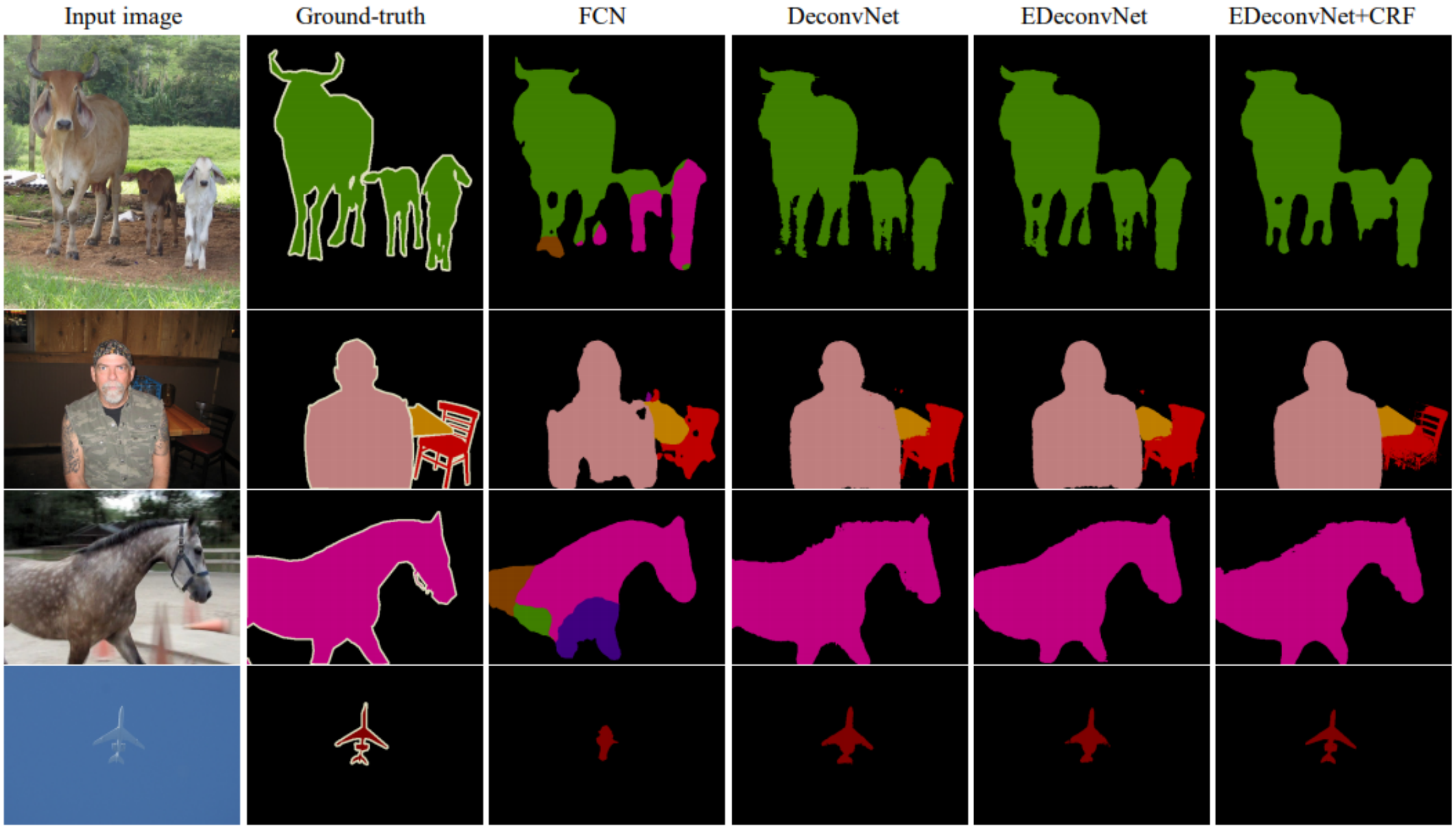
Noh, Hong, & Han, "Learning Deconvolution Network for Semantic Segmentation", ICCV2015

Deconvolutional Networks



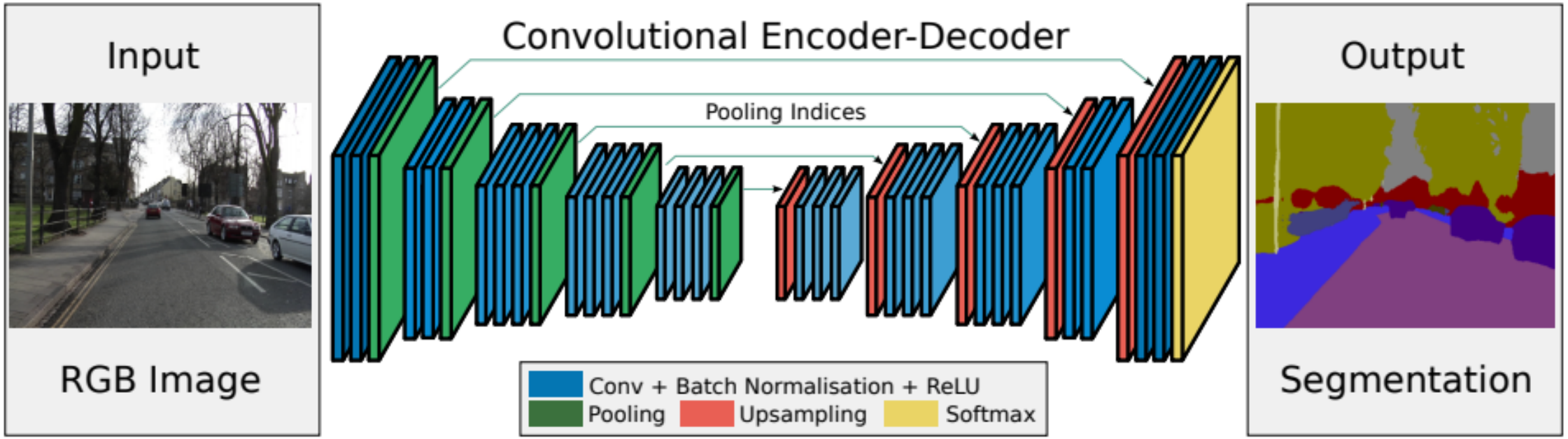
(a) Input image (b) FCN-8s (c) Ours

Noh, Hong, & Han, "Learning Deconvolution Network for Semantic Segmentation", ICCV2015



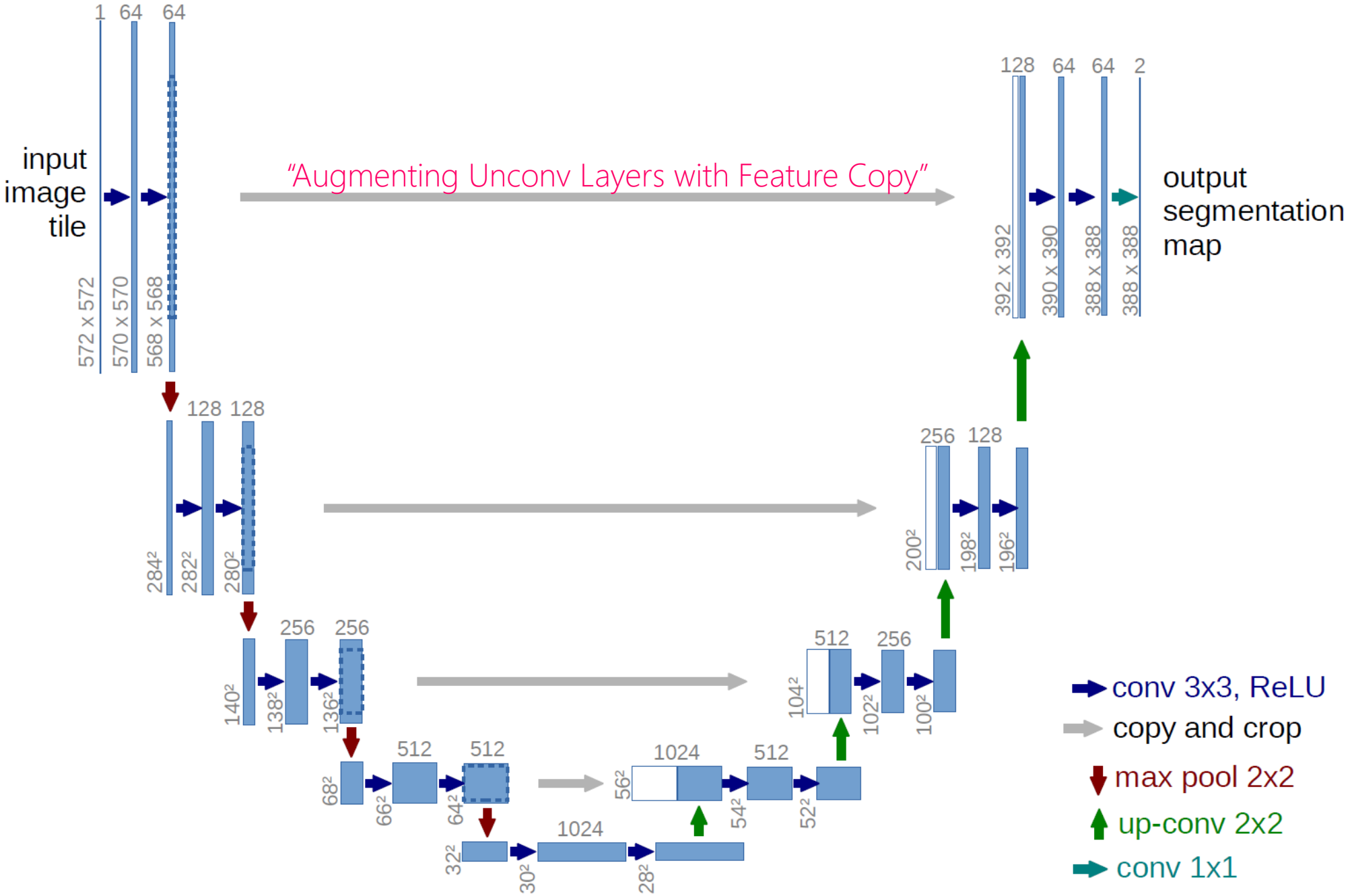
Noh, Hong, & Han, "Learning Deconvolution Network for Semantic Segmentation", ICCV2015

SegNet



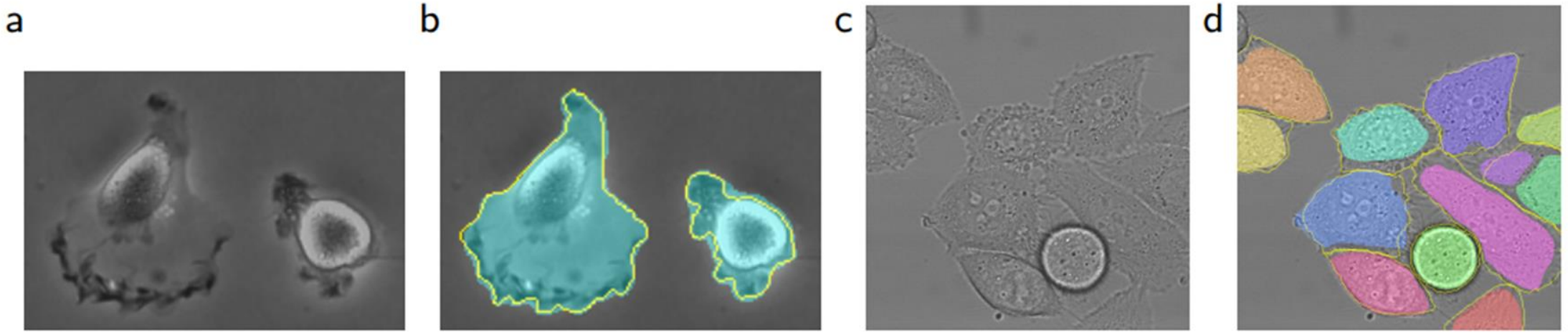
Badrinarayanan, Handa, & Cipolla "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling", arXiv 2015

U-Net



Ronneberger, Fischer, & Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", MICCAI2015

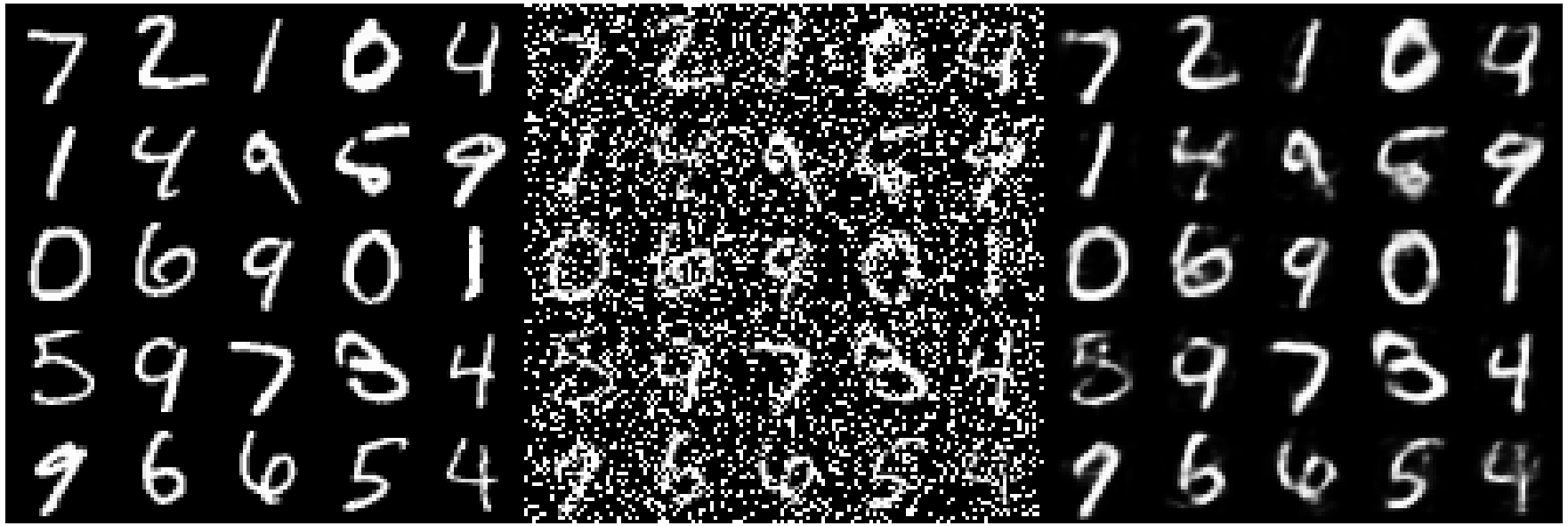
U-Net



Ronneberger, Fischer, & Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", MICCAI2015

Denoising Autoencoder (DAE)

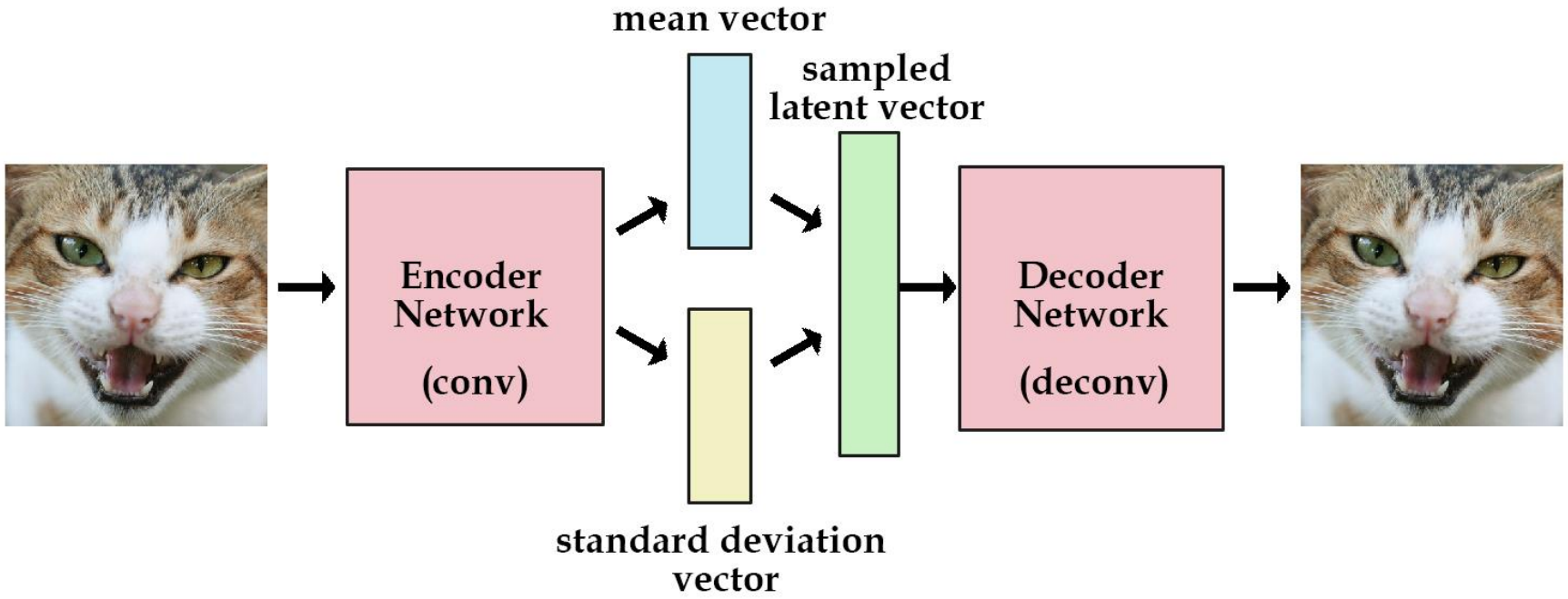
- Reconstruct data from a "corrupted" input
- Way of forcing the hidden layer to learn only more robust features



Vincent et al. "Extracting and Composing Robust Features with Denoising Autoencoders", ICML2008
Bengio et al. "Generalized Denoising Auto-Encoders as Generative Models", NIPS2013

Variational Autoencoder (VAE)

- Stochastic version of AE.
- Learn the statistical distribution of the latent variables, instead of latent variables themselves directly.



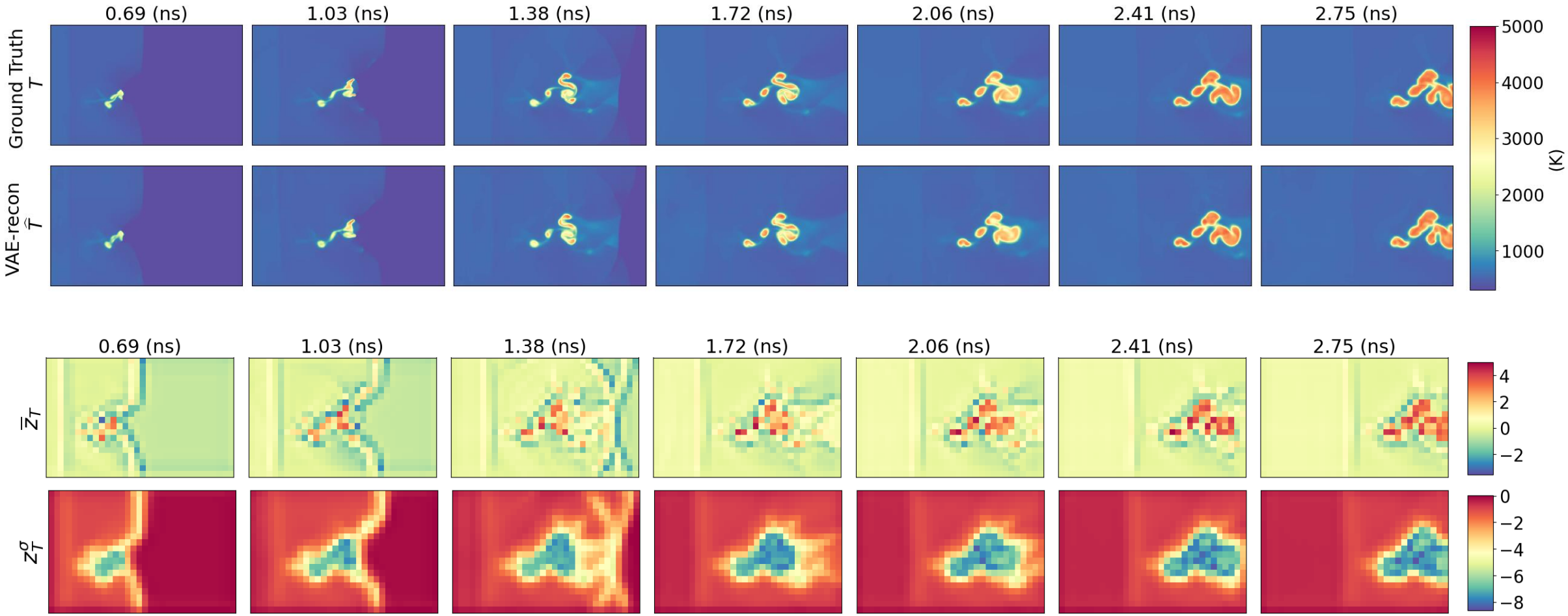
Latent representation of physical fields

- Goal: Learn the underlying dynamics of the three coupled fields:
  1) Temperature (T)
  2) Pressure (P)
  3) Microstructure (μ)

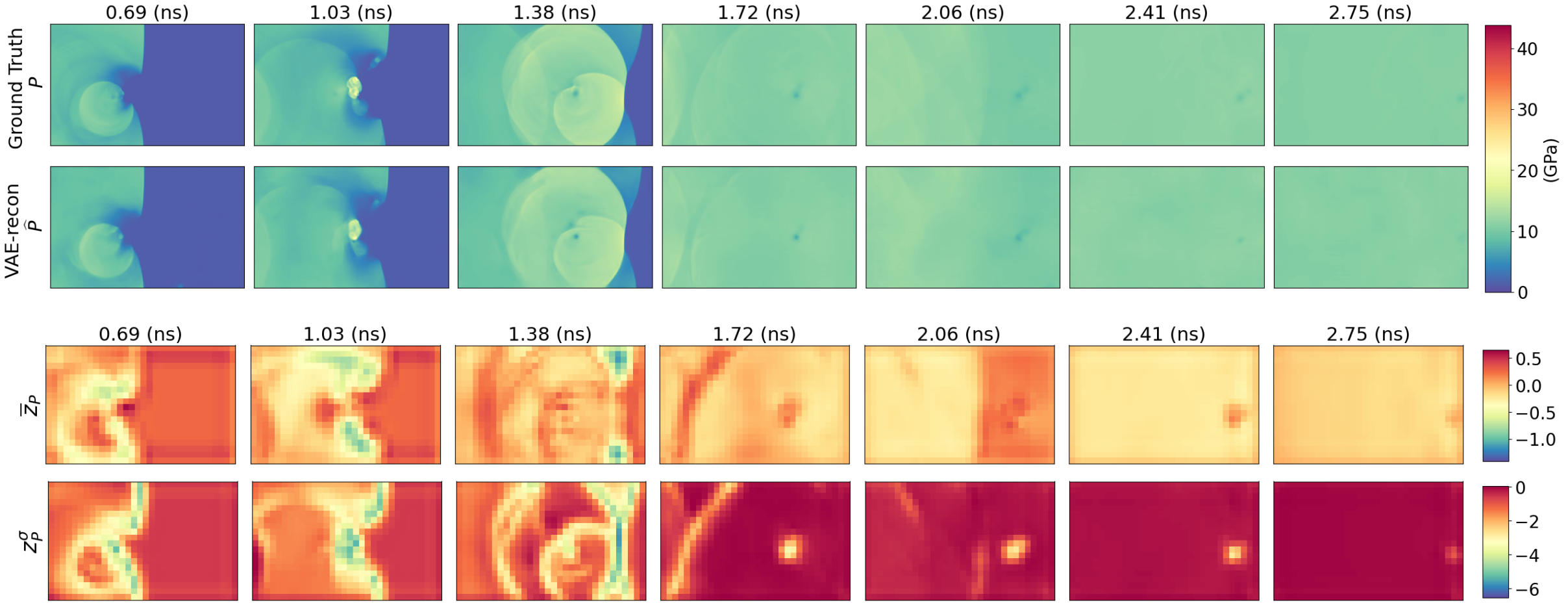
- Encoding physical fields into latent space using VAE

- VAE-based latent mean and log-variance fields:
  1) Temperature latent representation $(\overline{Z_T}, Z_T^{\sigma})$
  2) Pressure latent representation $(\overline{Z_P}, Z_P^{\sigma})$
  3) Microstructure latent representation $\left(\overline{Z_\mu}, Z_\mu^{\sigma}\right)$

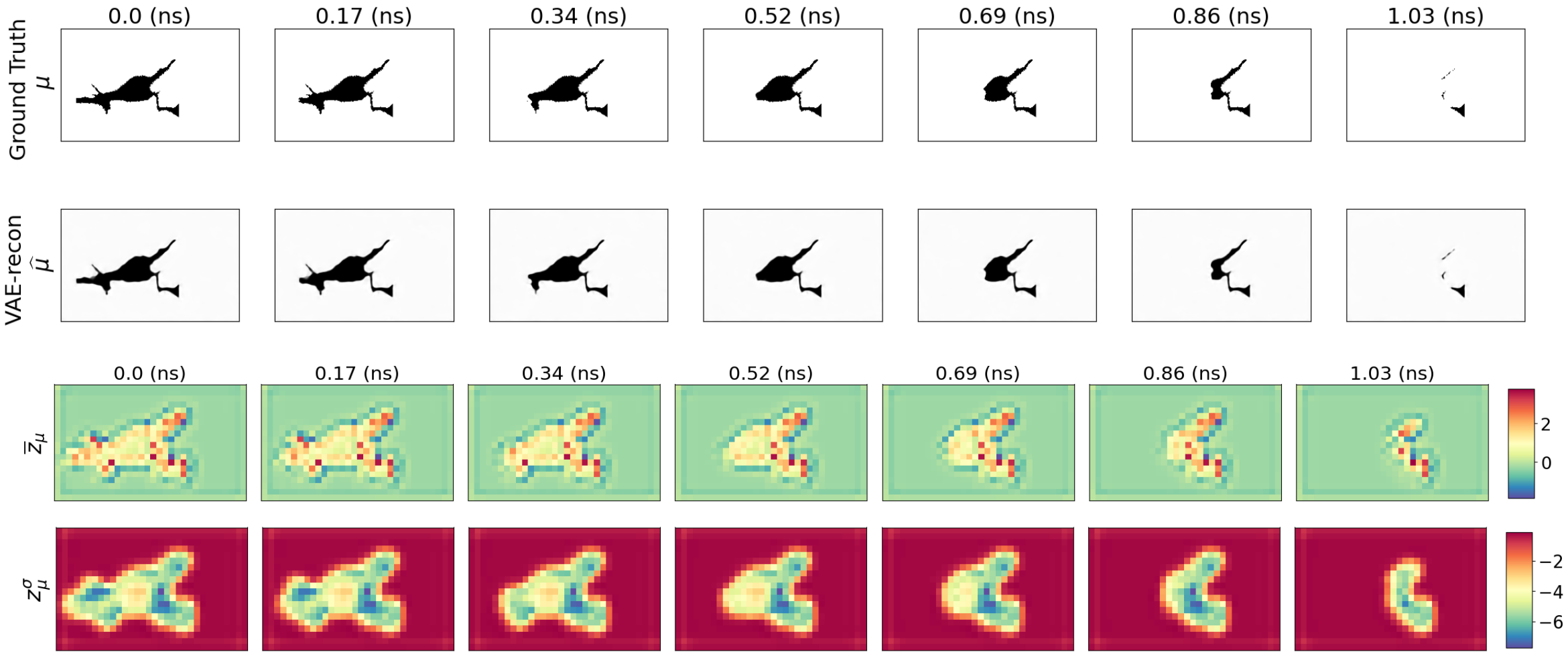
Encoding temperature field



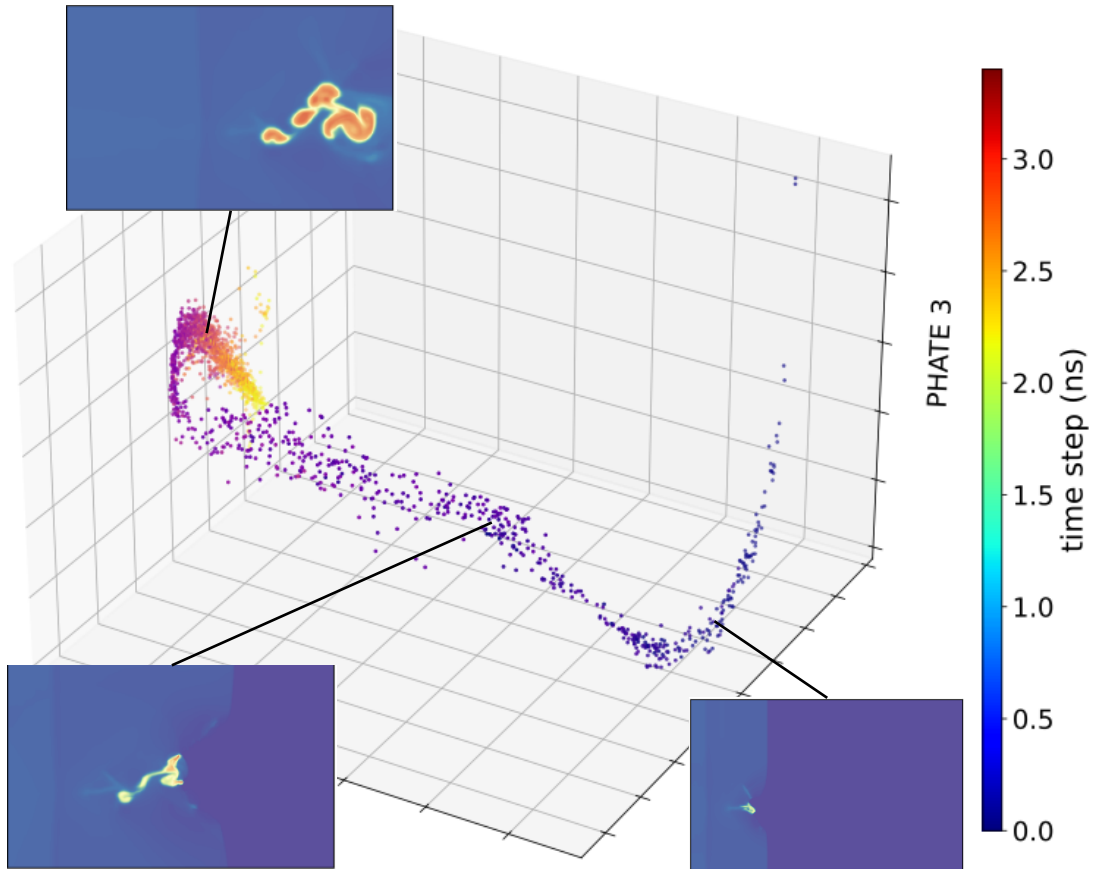
Encoding pressure field



Encoding microstructure field



3D Embedding of the Manifold of $\bar{Z}_T$ Fields



Manifold of latent fields $\bar{Z}_T$

The Coupled Evolution of the Latent Fields

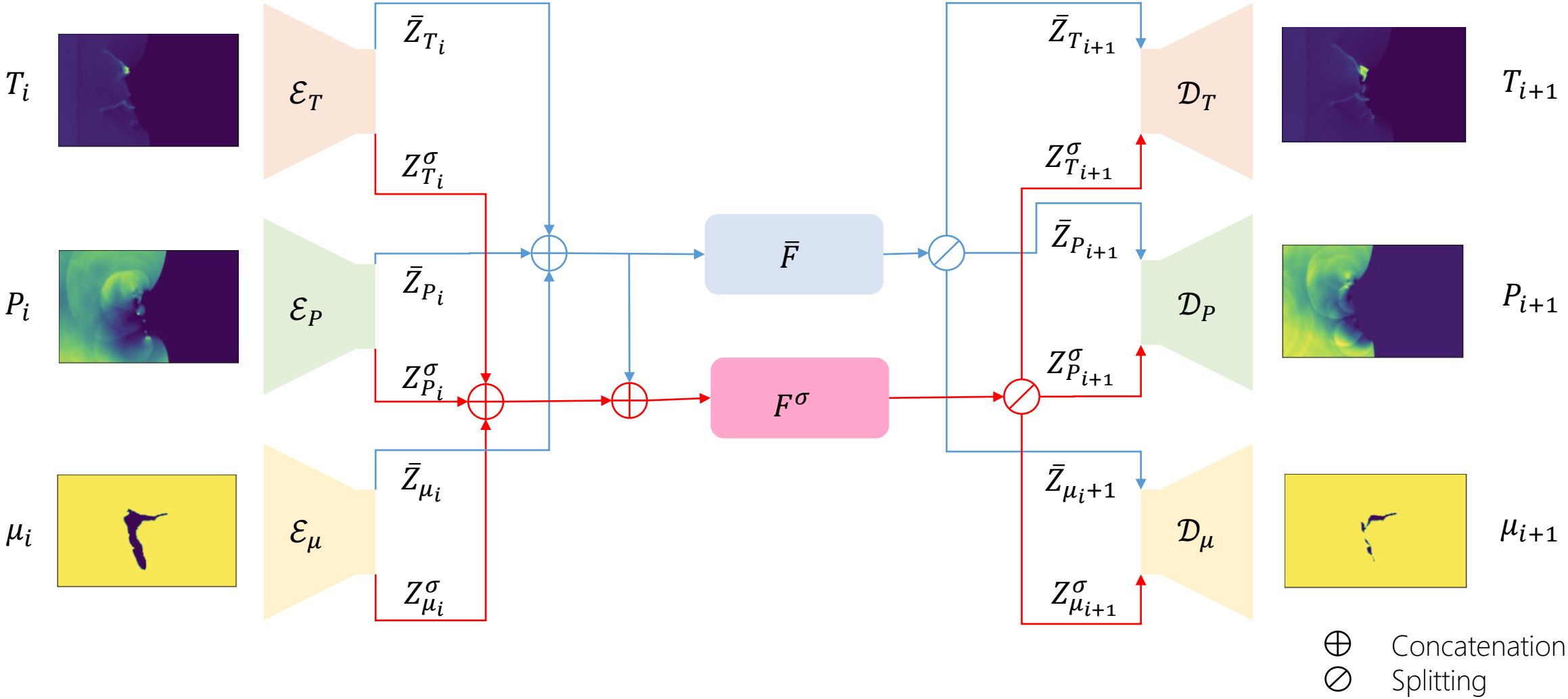
- Deterministic part of latent dynamics:

$$\begin{bmatrix} \bar{Z}_T \\ \bar{Z}_P \\ \bar{Z}_\mu \end{bmatrix}_{|t+1} = \bar{F} \begin{bmatrix} \bar{Z}_T \\ \bar{Z}_P \\ \bar{Z}_\mu \end{bmatrix}_{|t}$$

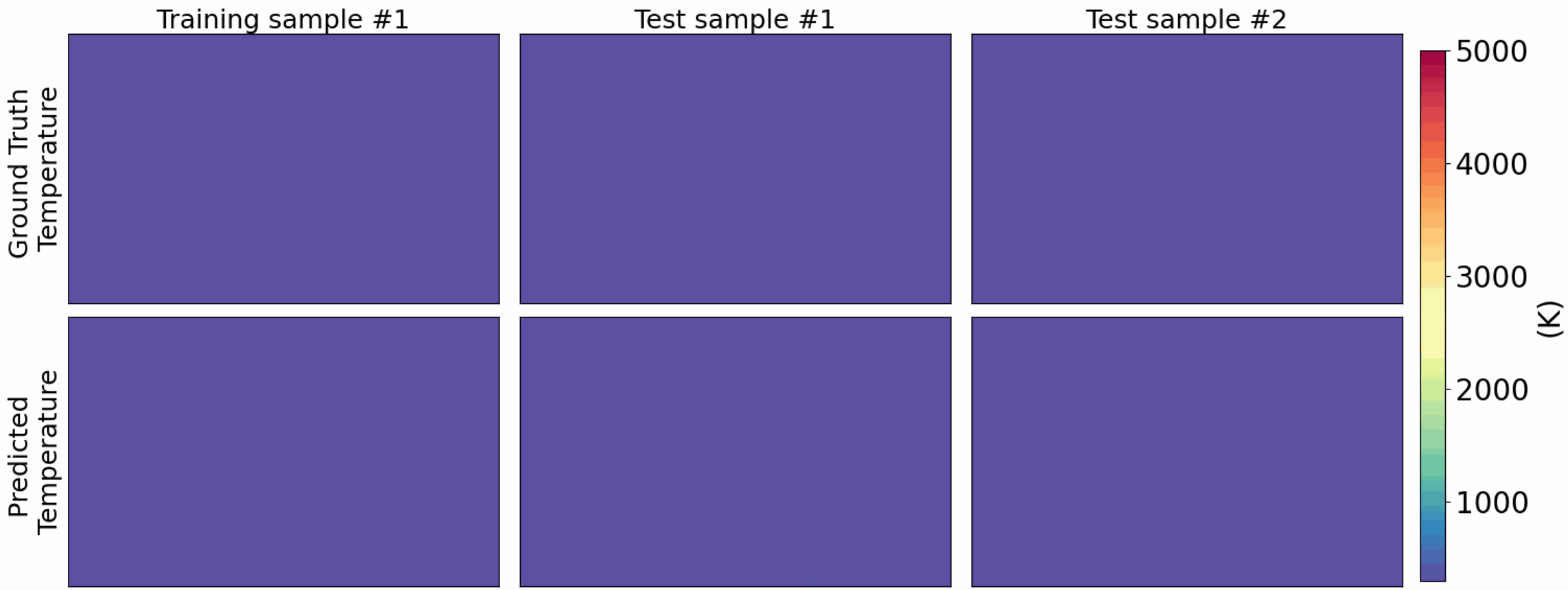
- Stochastic part of latent dynamics:

$$\begin{bmatrix} Z_T^\sigma \\ Z_P^\sigma \\ Z_\mu^\sigma \end{bmatrix}_{|t+1} = F^\sigma \begin{bmatrix} (\bar{Z}_T\,,\, Z_T^\sigma) \\ (\bar{Z}_P\,,\, Z_P^\sigma) \\ (\bar{Z}_\mu\,,\, Z_\mu^\sigma) \end{bmatrix}_{|t}$$

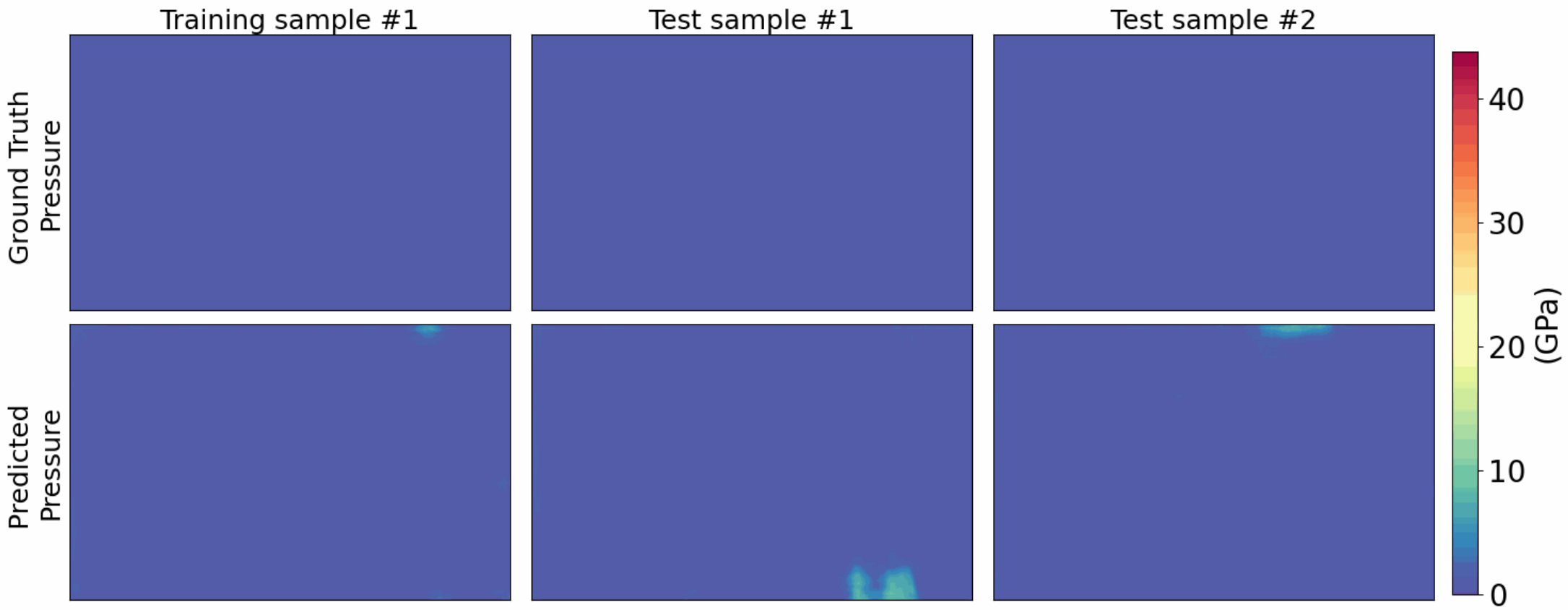
Model Architecture



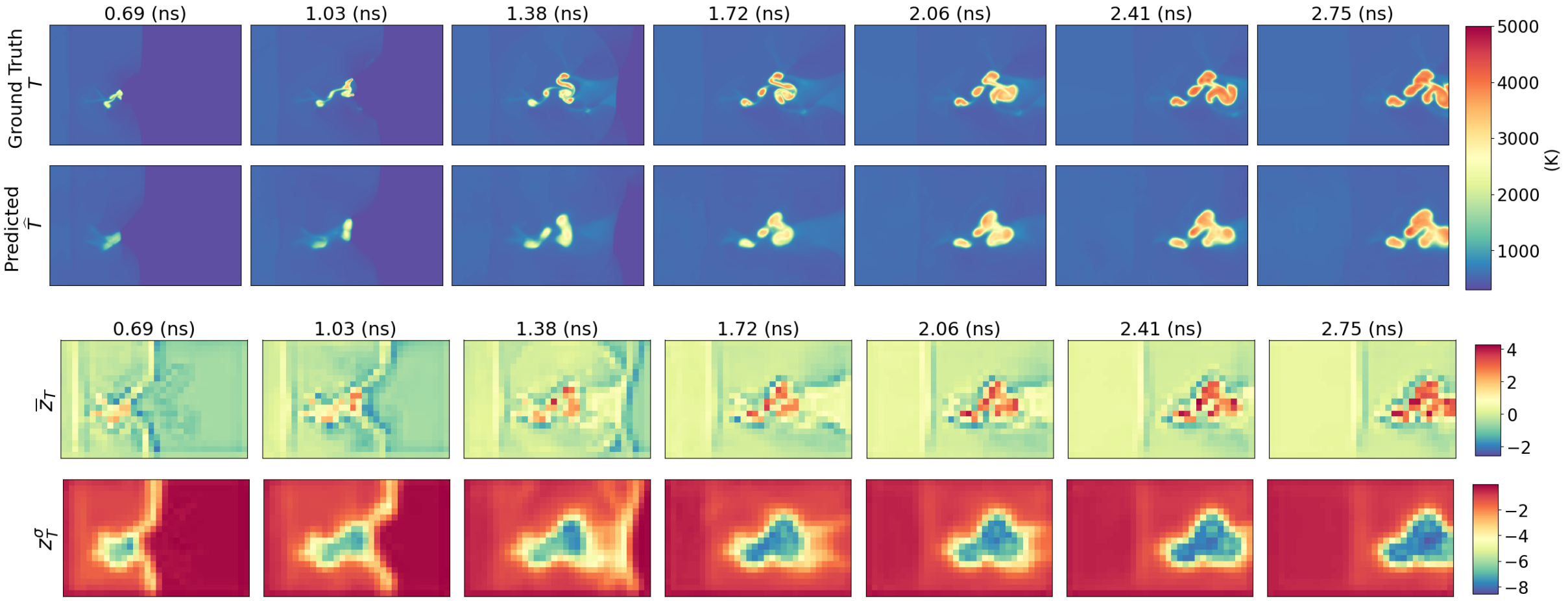
Direct numerical simulation vs.
Model prediction of temperature evolution



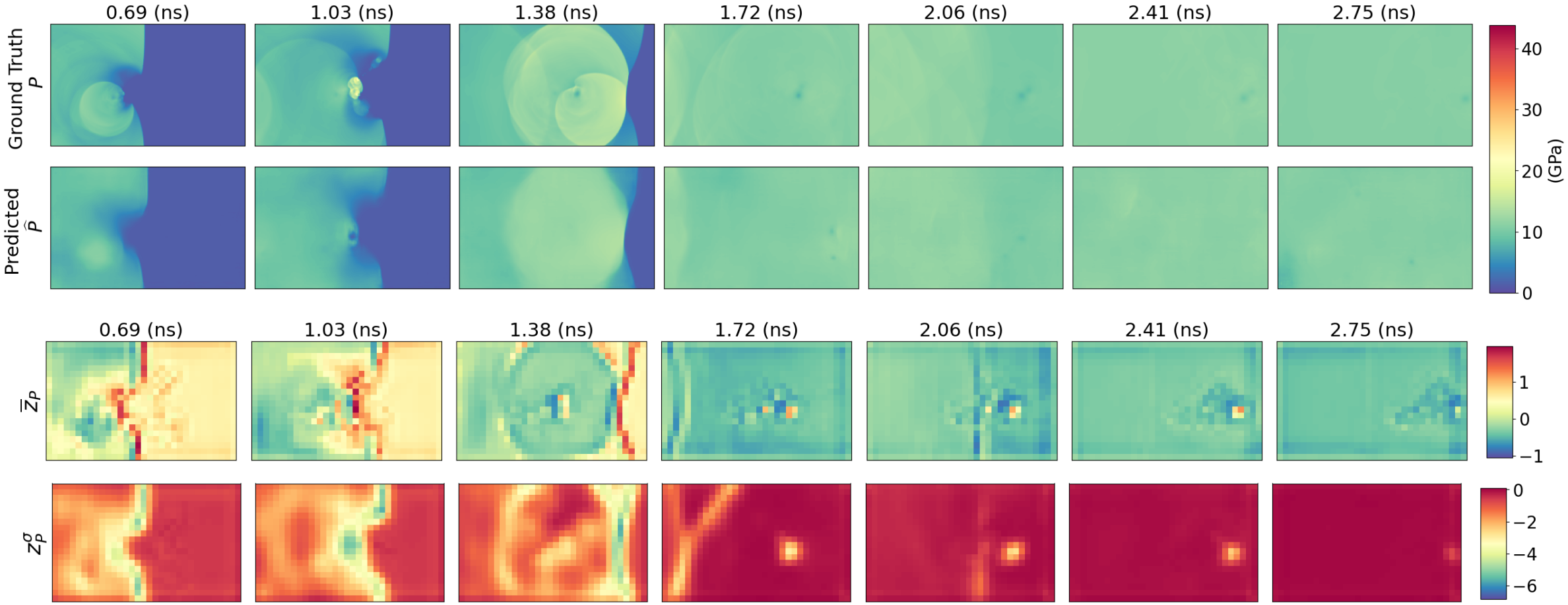
Direct numerical simulation vs. Model prediction of pressure evolution



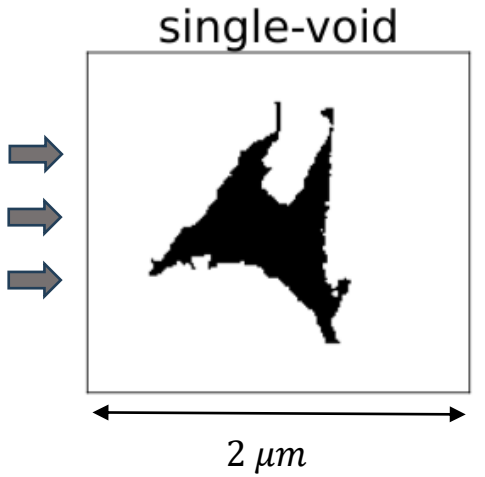
Induced dynamics over latent space

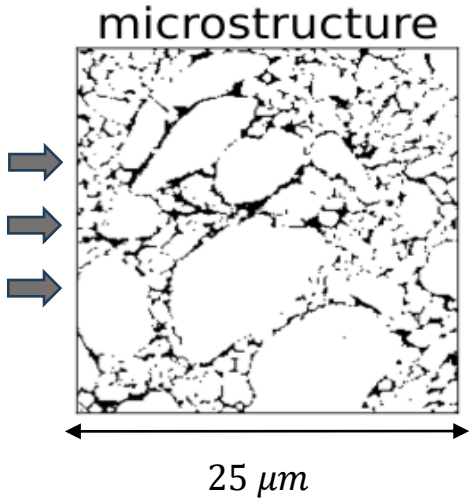


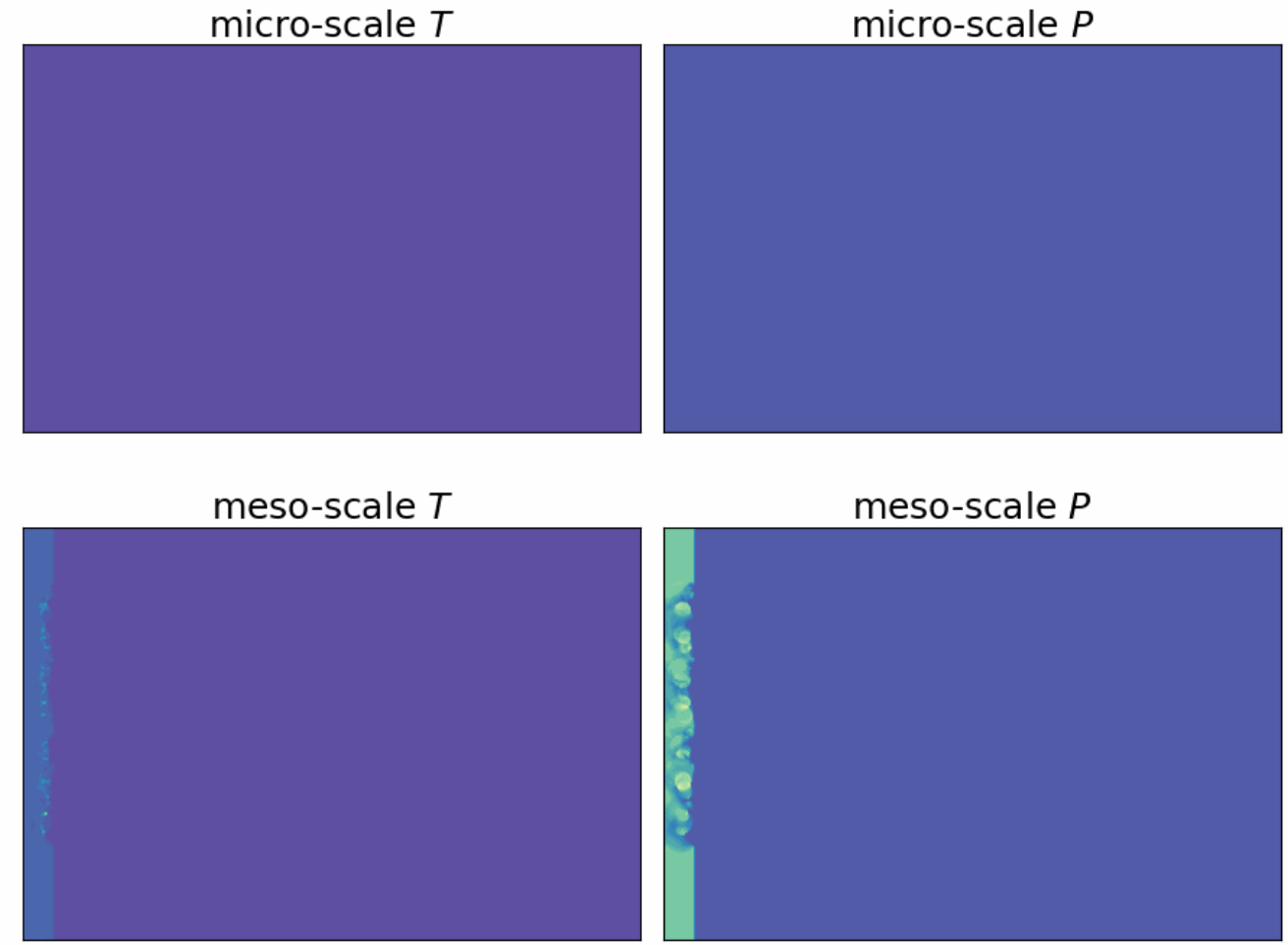
Induced dynamics over latent space



Scale-bridging between micro- and meso-scale dynamics

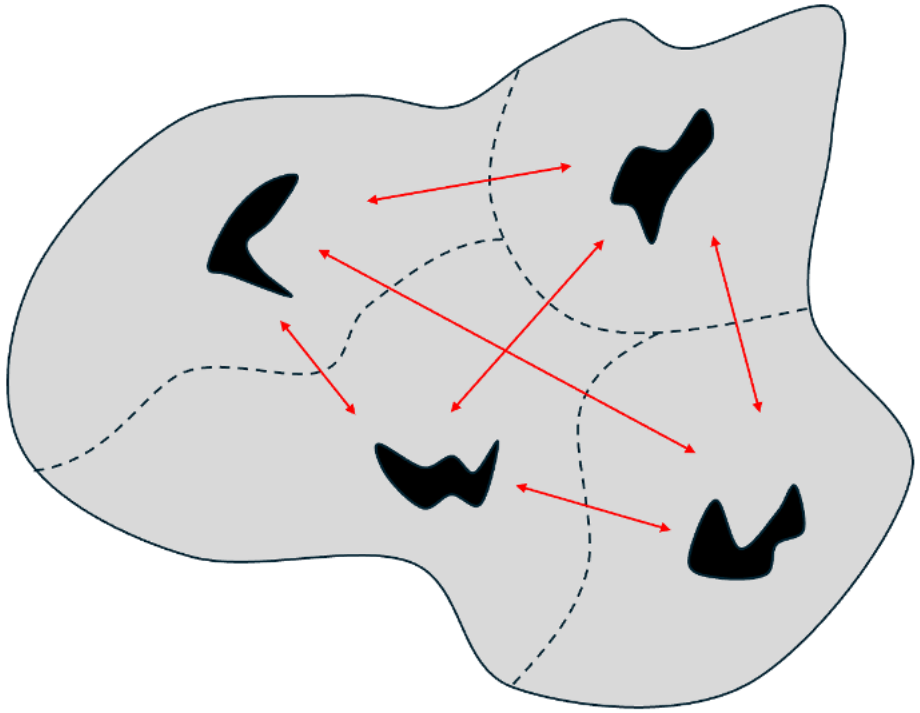


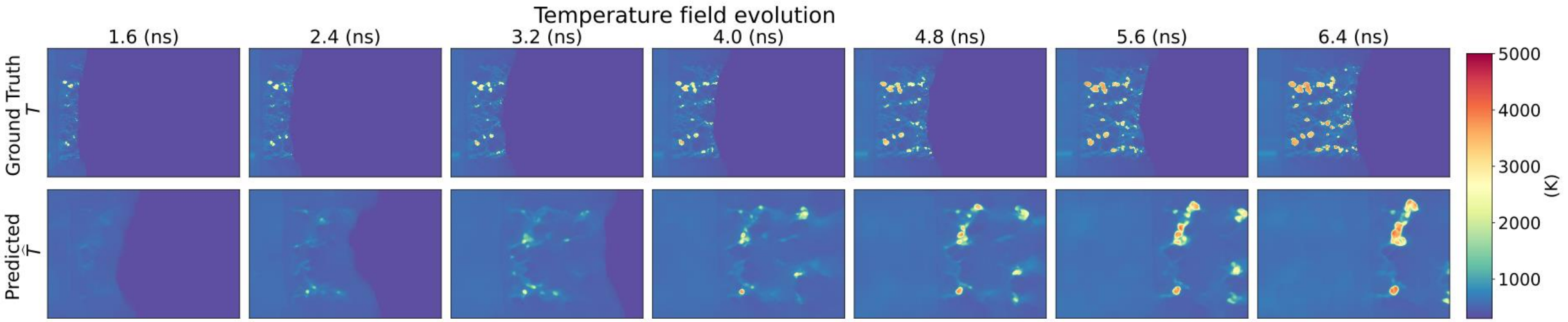




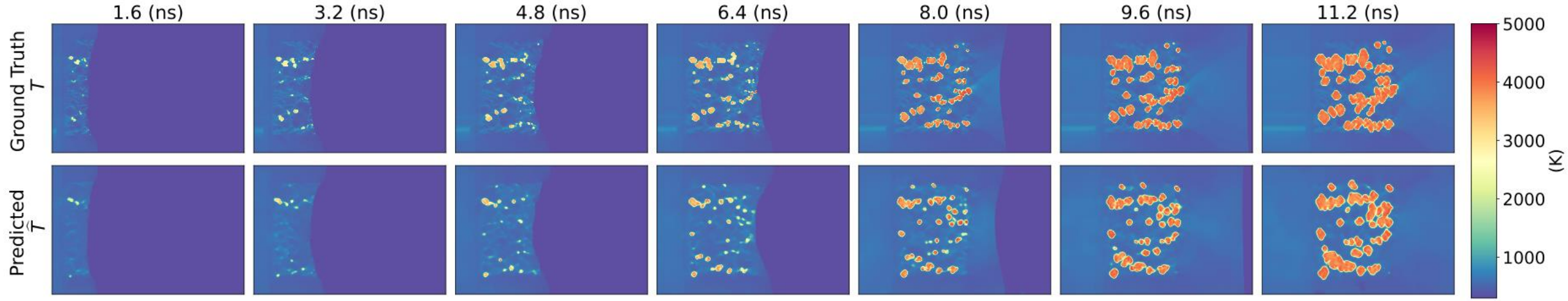
Meta-learning approach

1) Learn a reduced representation of micro-scale dynamics as building blocks of meso-scale dynamics
2) Learn the correlation between micro-scale building blocks



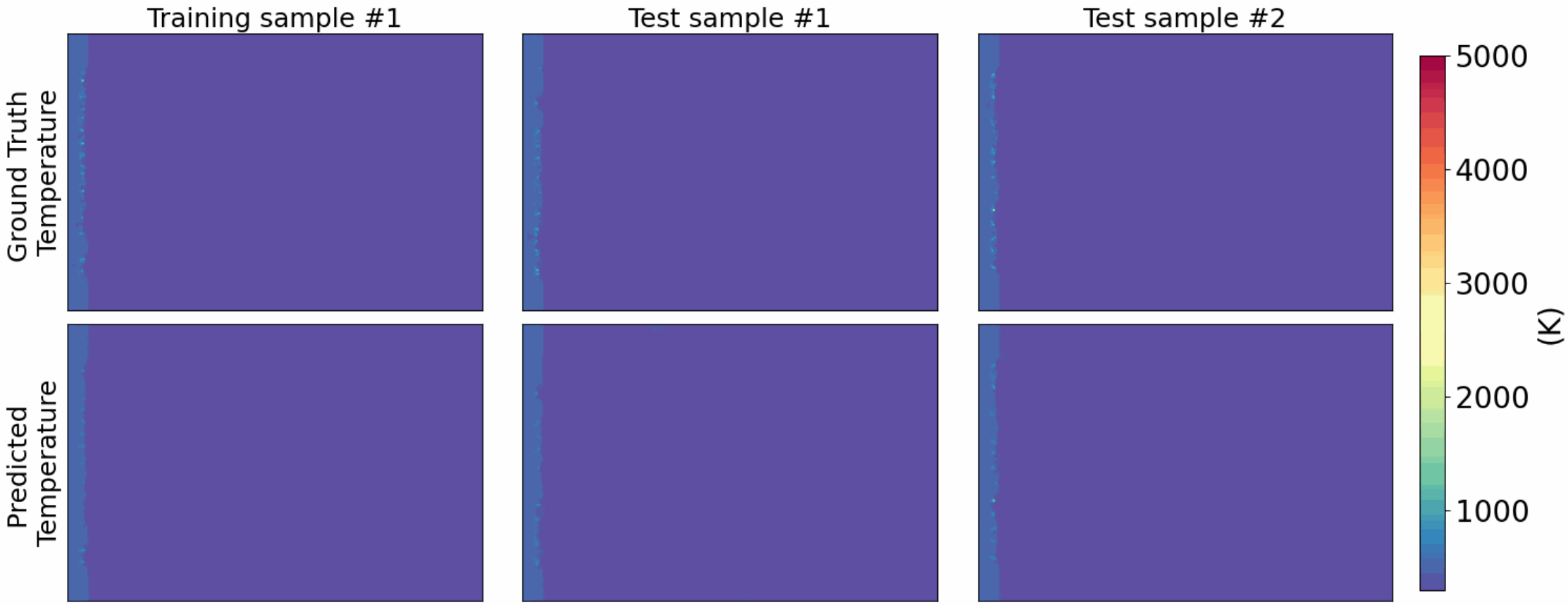


Before learning the correlation

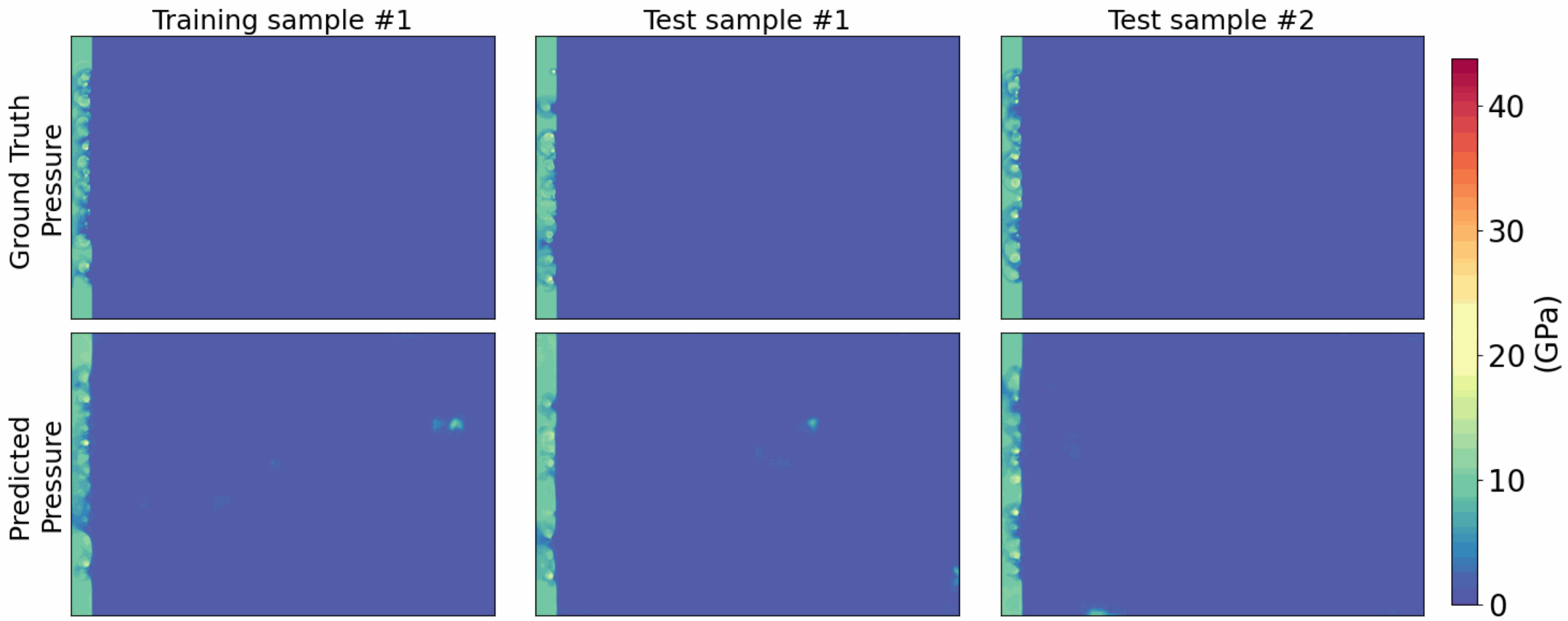


After learning the correlation

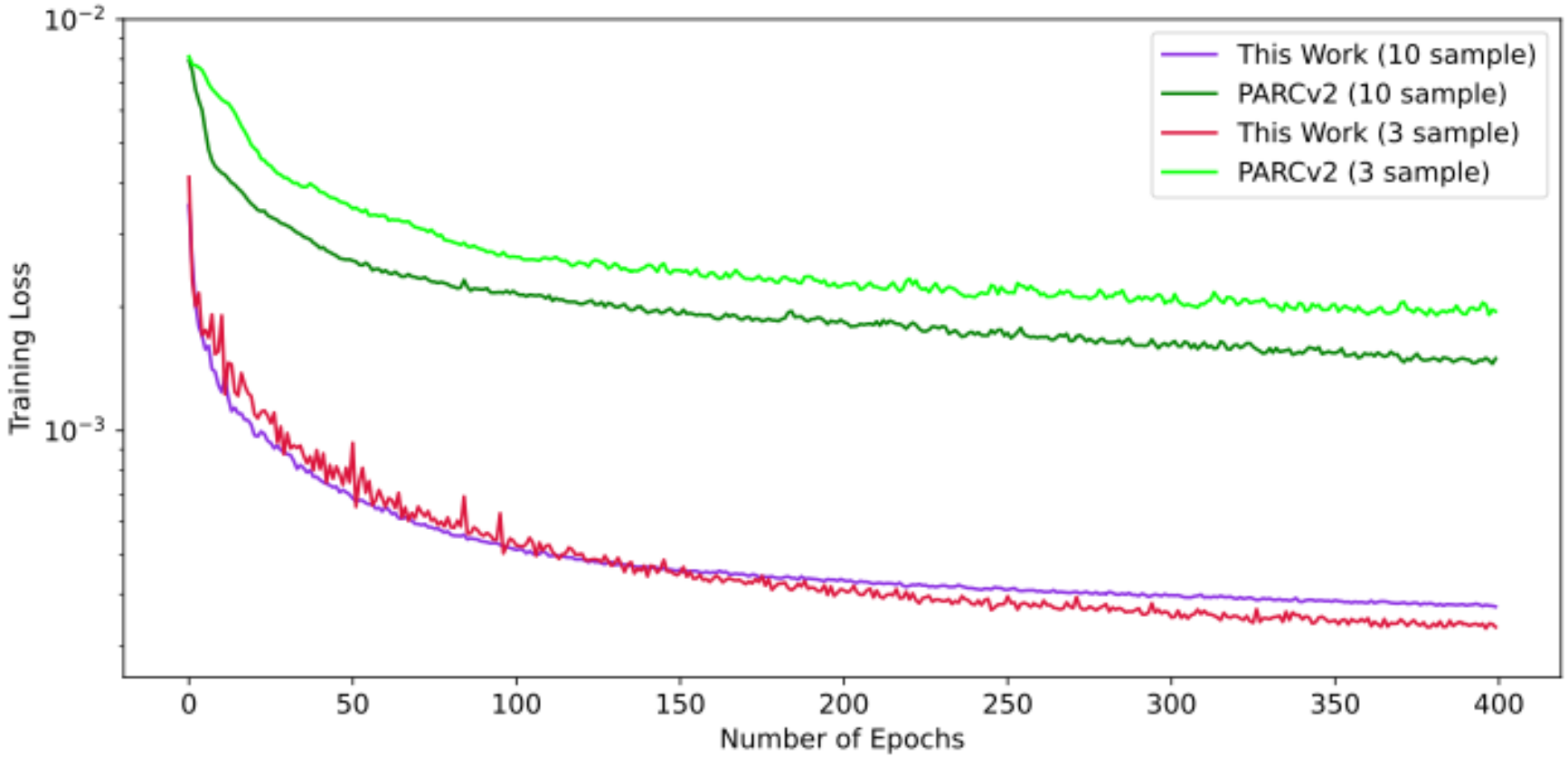
Direct numerical simulation vs. Model prediction of temperature evolution



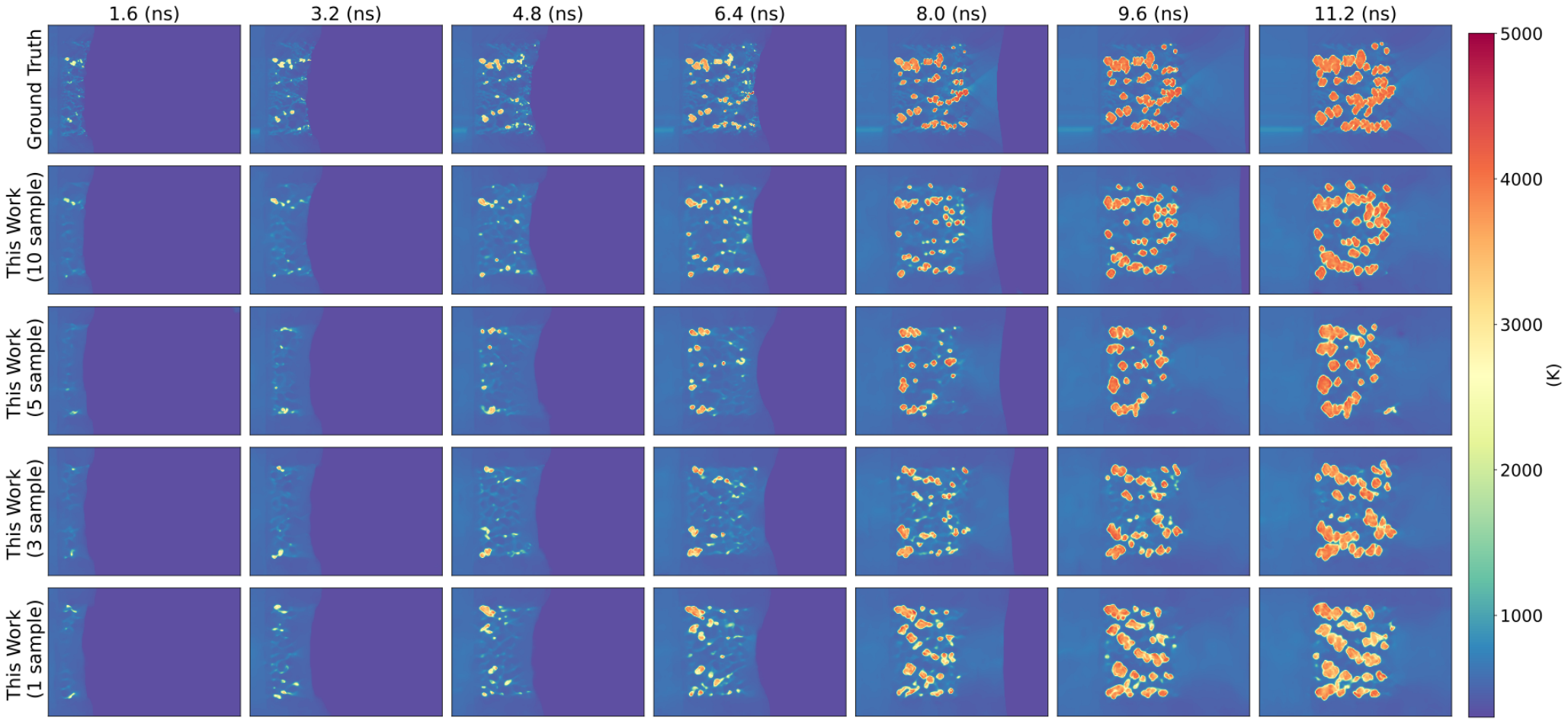
Direct numerical simulation vs. Model prediction of pressure evolution



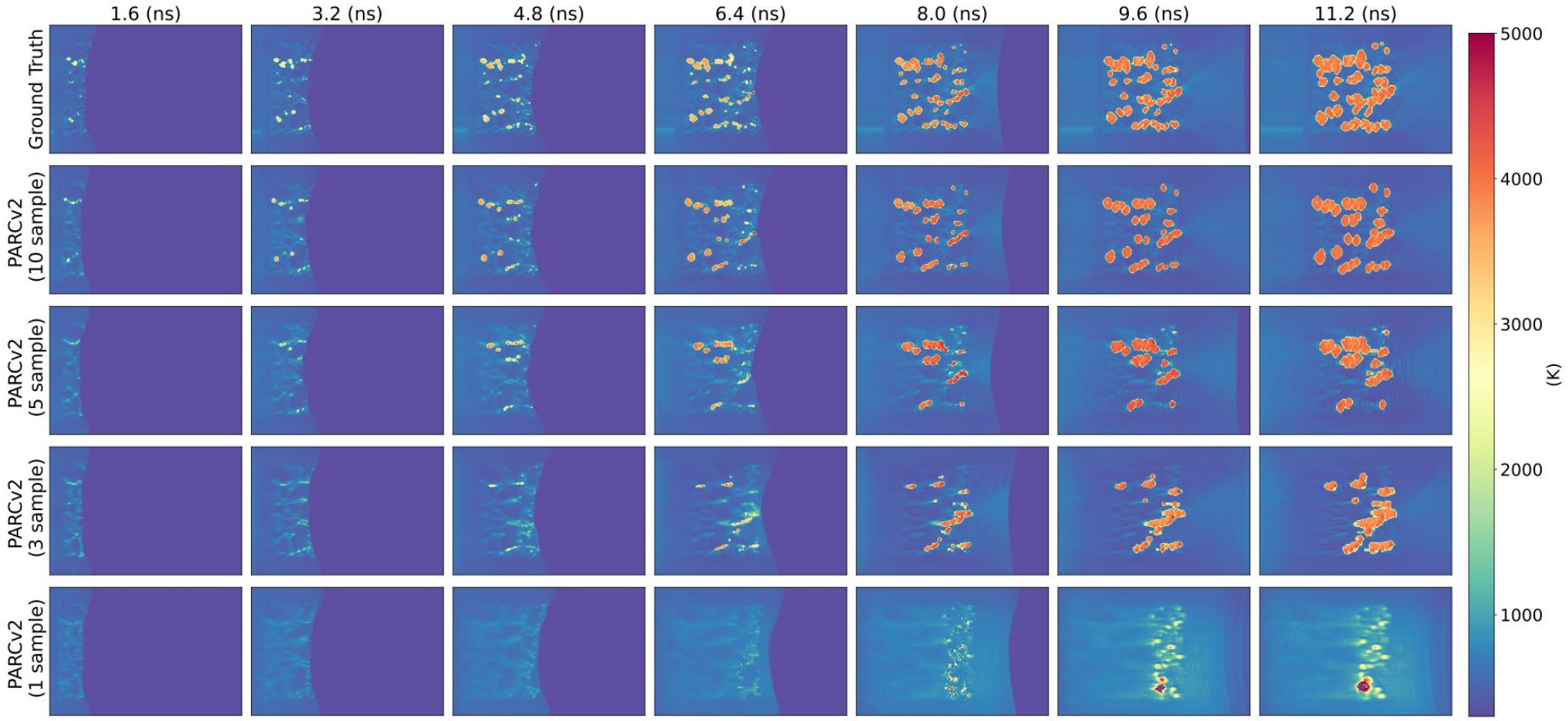
Comparison with PARC in scarce meso-scale data regime



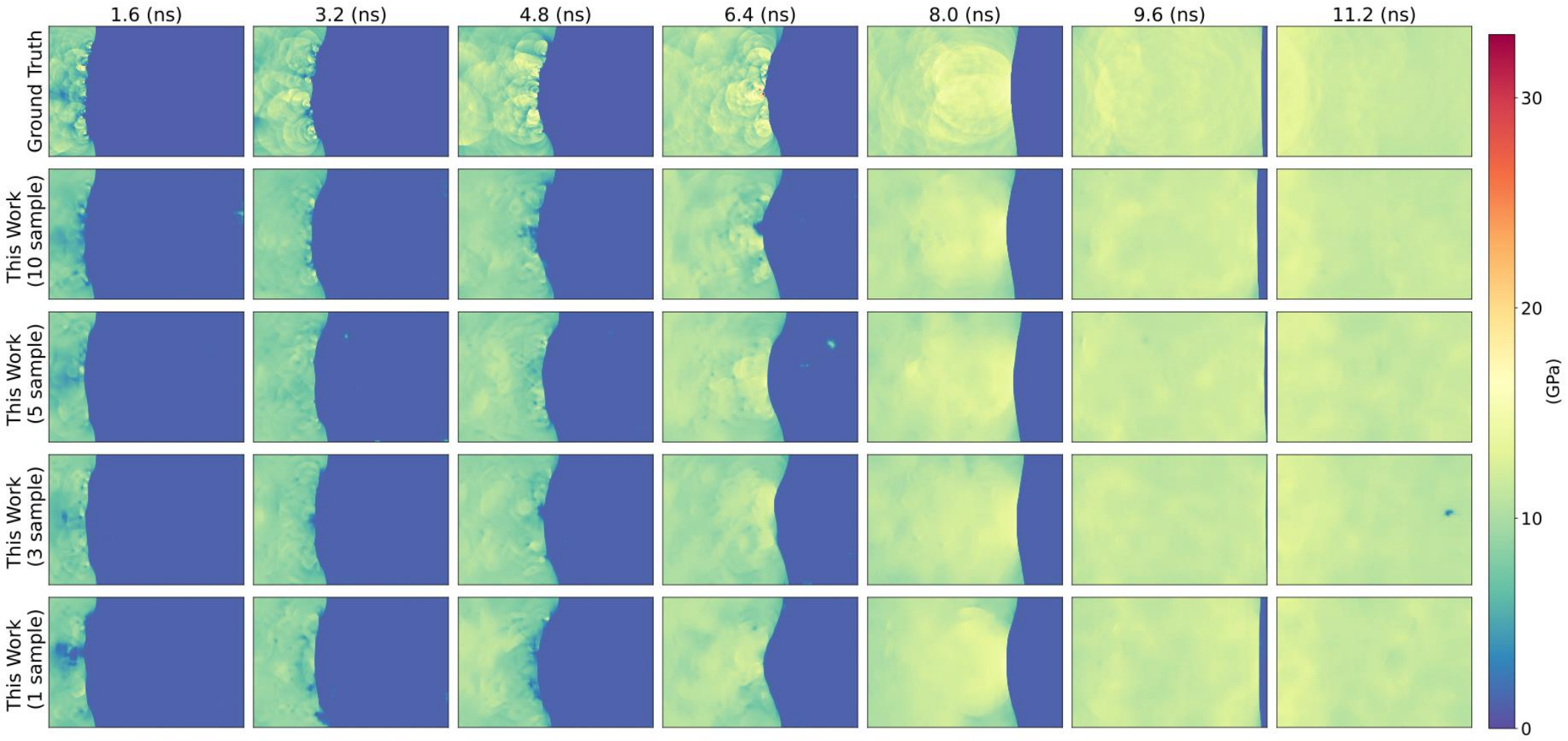
Meta learned prediction for temperature



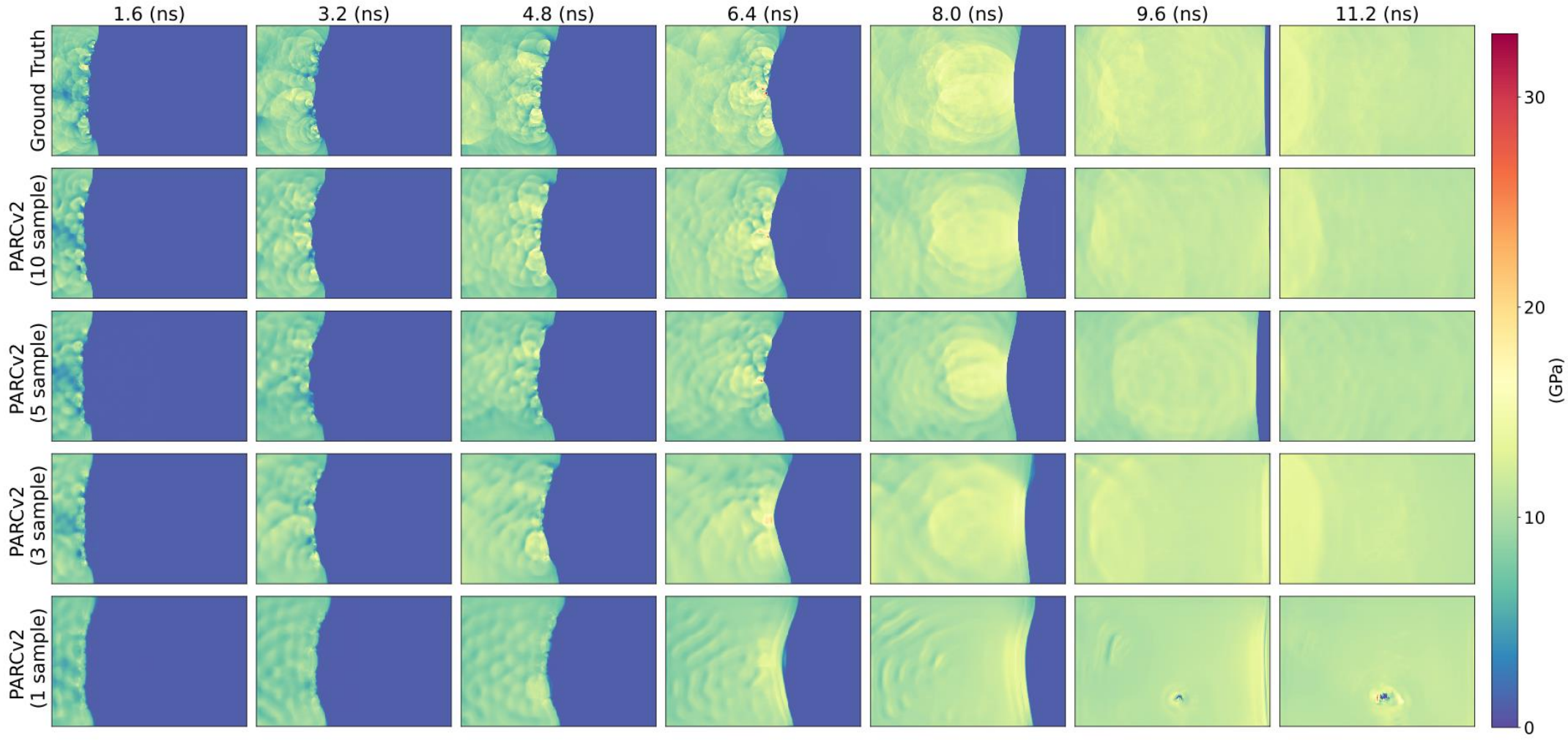
PARC prediction for temperature



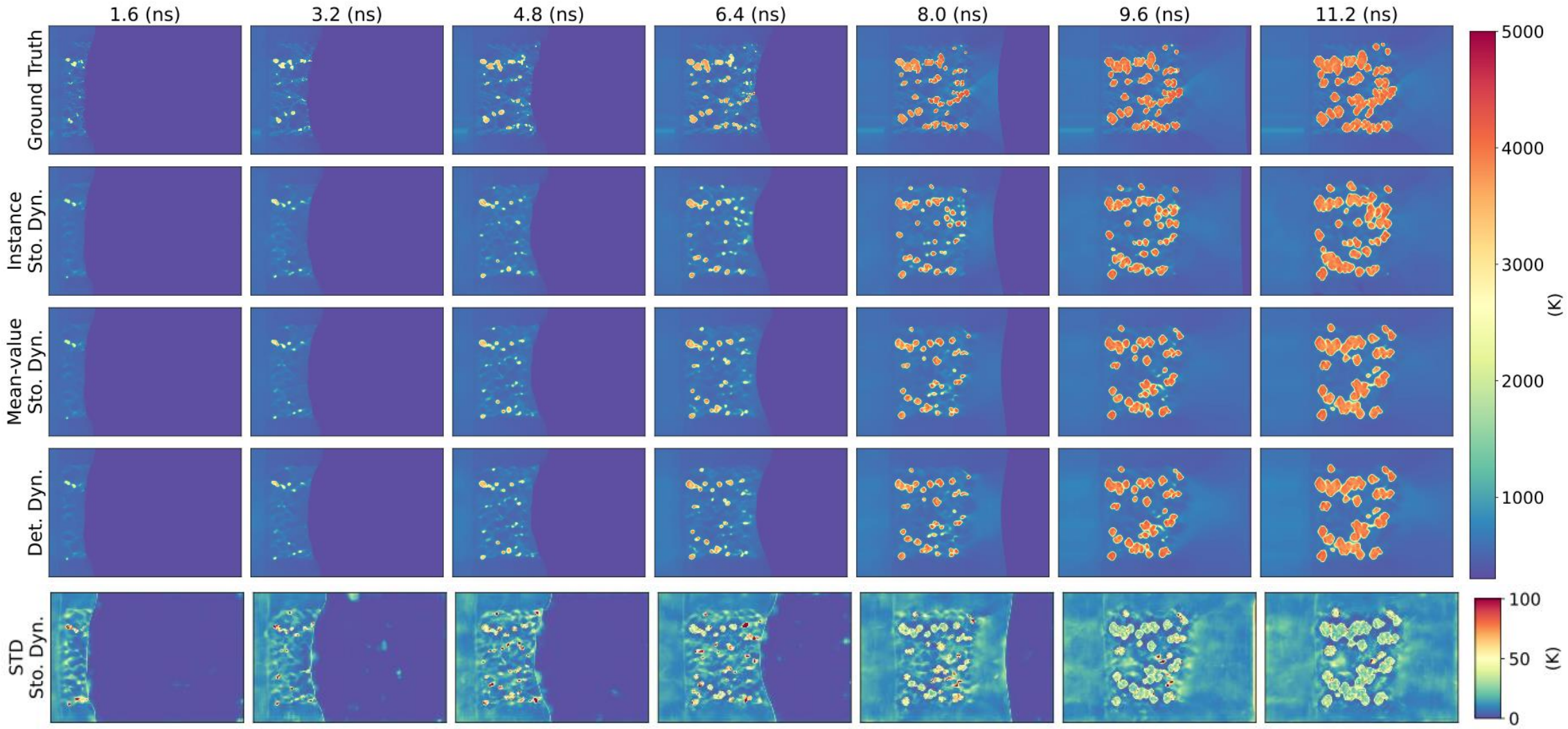
Meta learned prediction for pressure



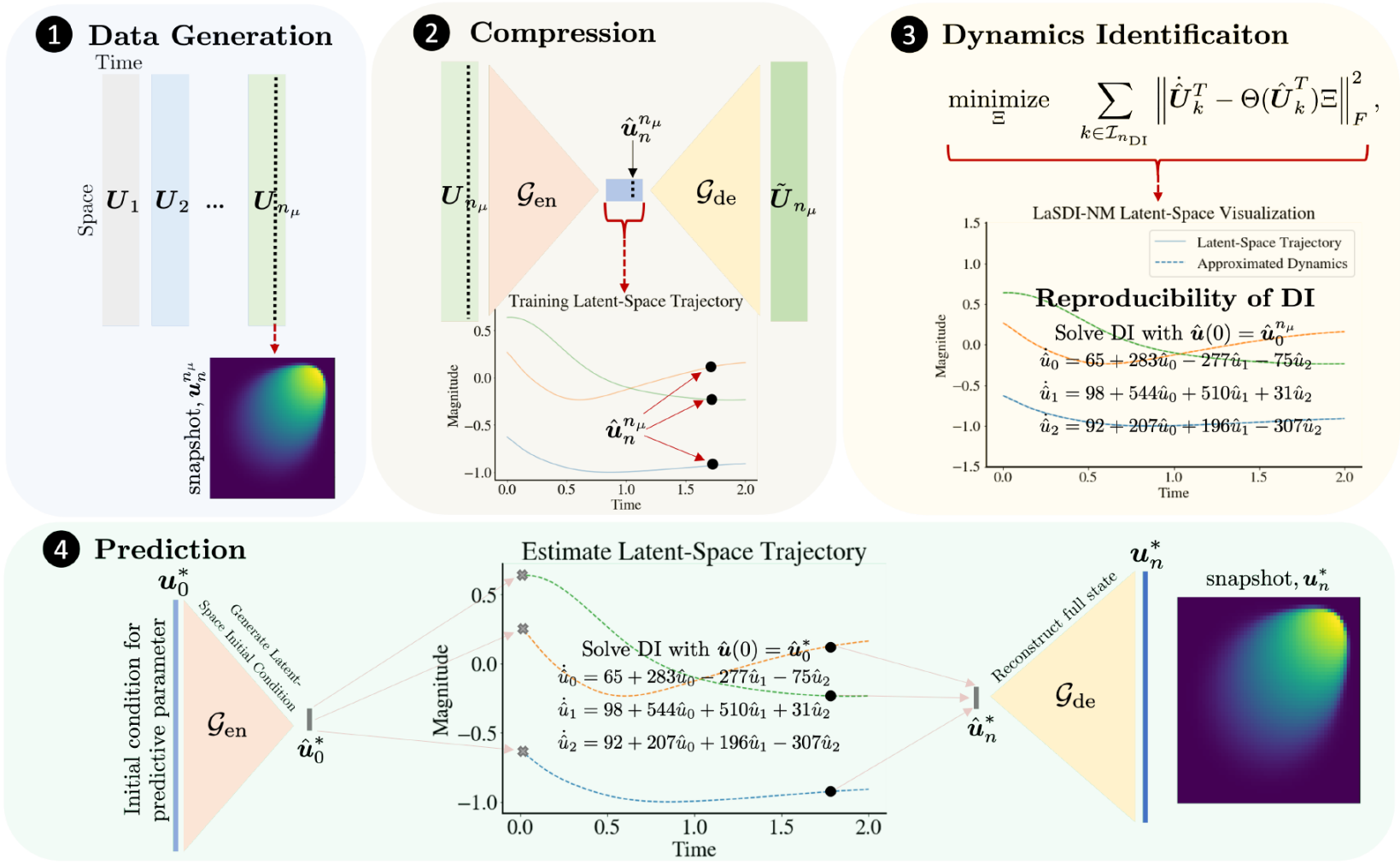
PARC prediction for pressure evolution



Uncertainty quantification

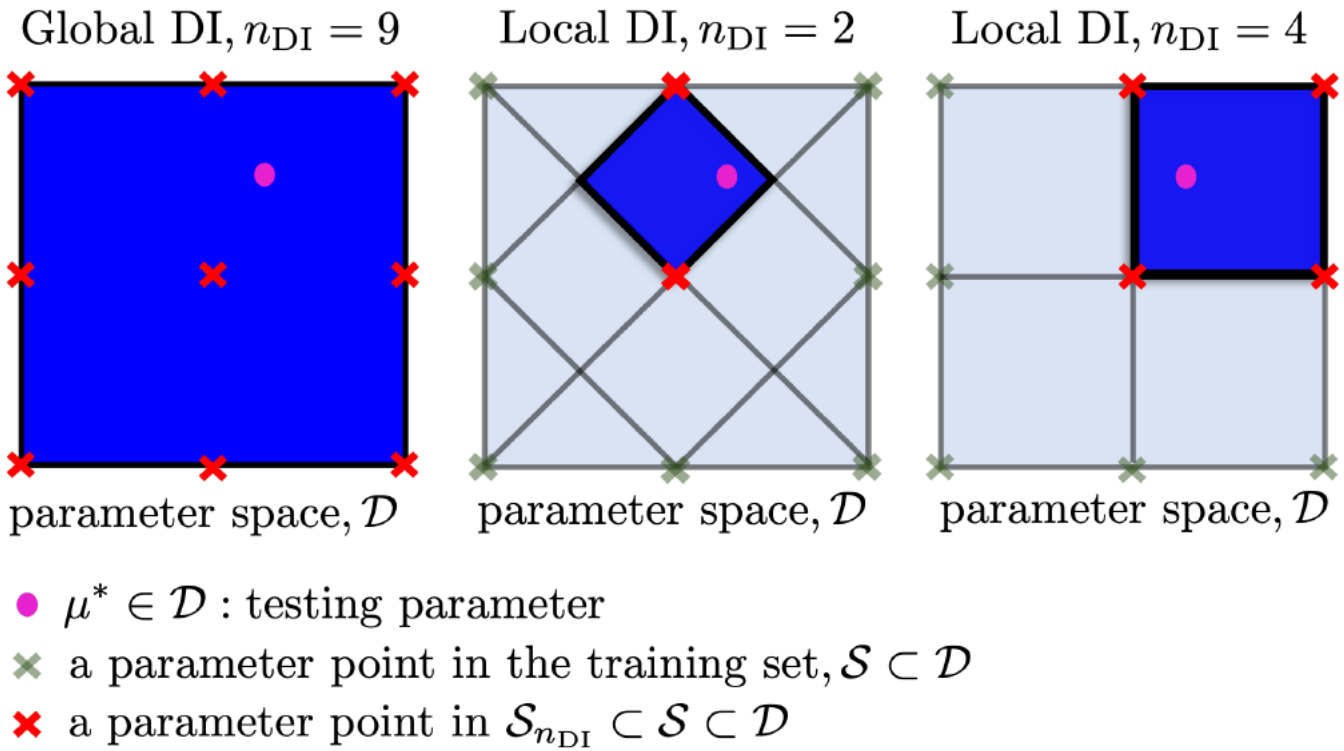


Latent Space Dynamics Identification (LaSDI)



Fries, He, & Choi, "LaSDI: Parametric Latent Space Dynamics Identification", 2022

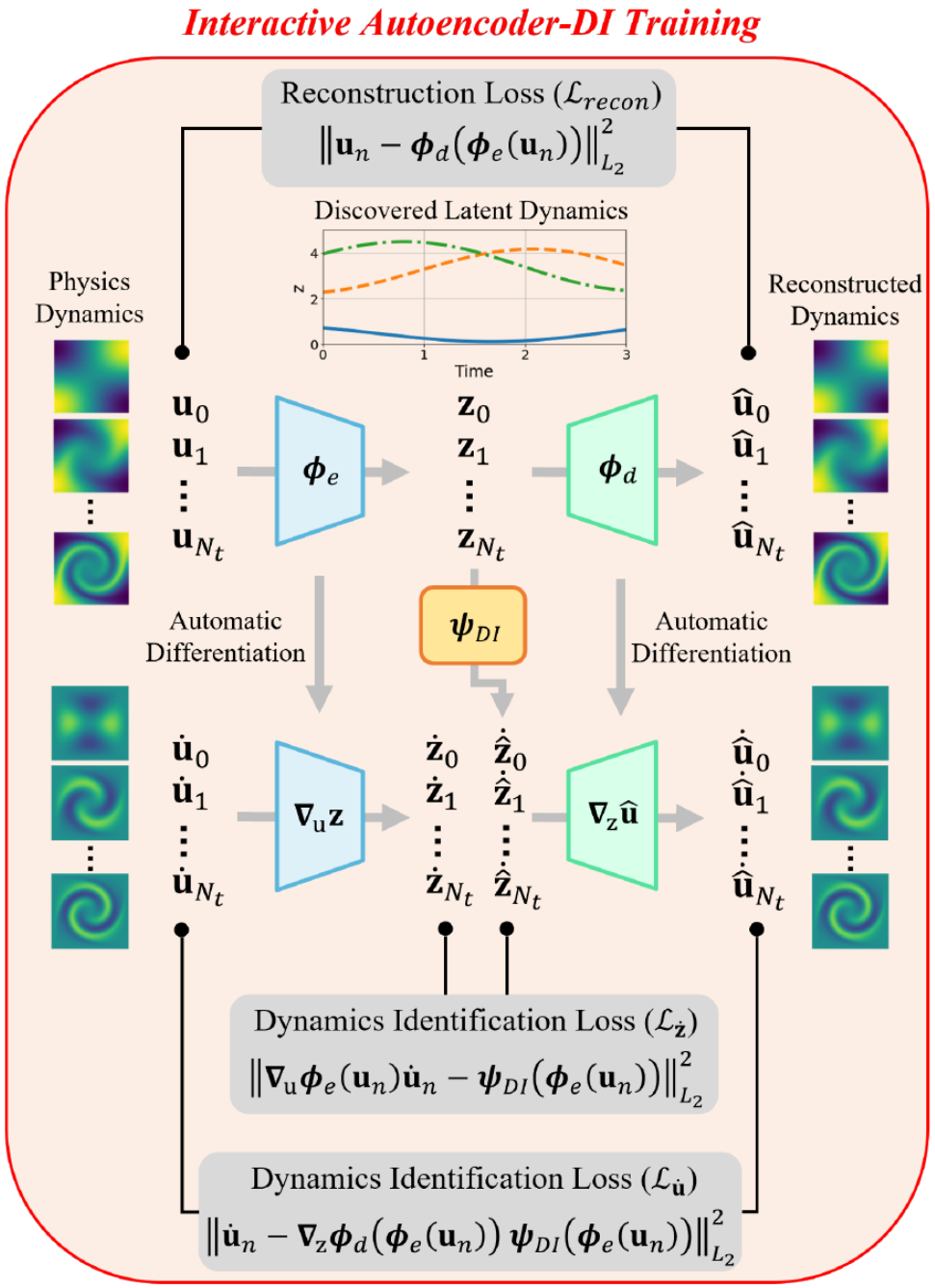
Local vs. Global Dynamics Identification



Fries, He, & Choi, "LaSDI: Parametric Latent Space Dynamics Identification", 2022

gLaSDI

- Dynamics identification loss using automatic differentiation:
  - Over latent space
  - Over input space



He et al., "gLaSDI: Parametric Physics-Informed Greedy Latent Space Dynamics Identification", 2023