# DS-GA.3001 Embodied Learning and Vision

**Mengye Ren**

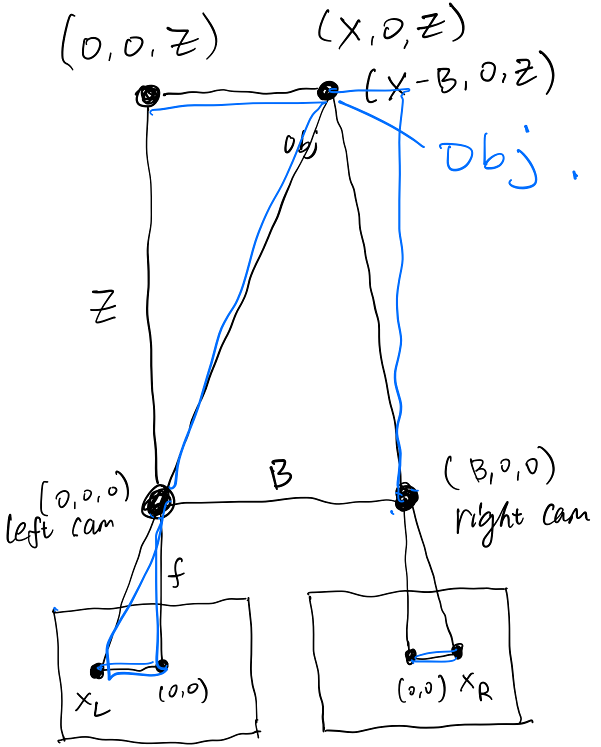
NYU

Spring 2026

elvcourse.org

# Classic Vision on Depth and Disparity

- One source of depth is from the displacement of pixels in a stereo setup.
- But we need to estimate disparity.



$$x_L = \frac{Xf}{Z}$$

$$x_R = \frac{(X-B)f}{Z}$$

$$\underline{\delta = x_L - x_R}$$

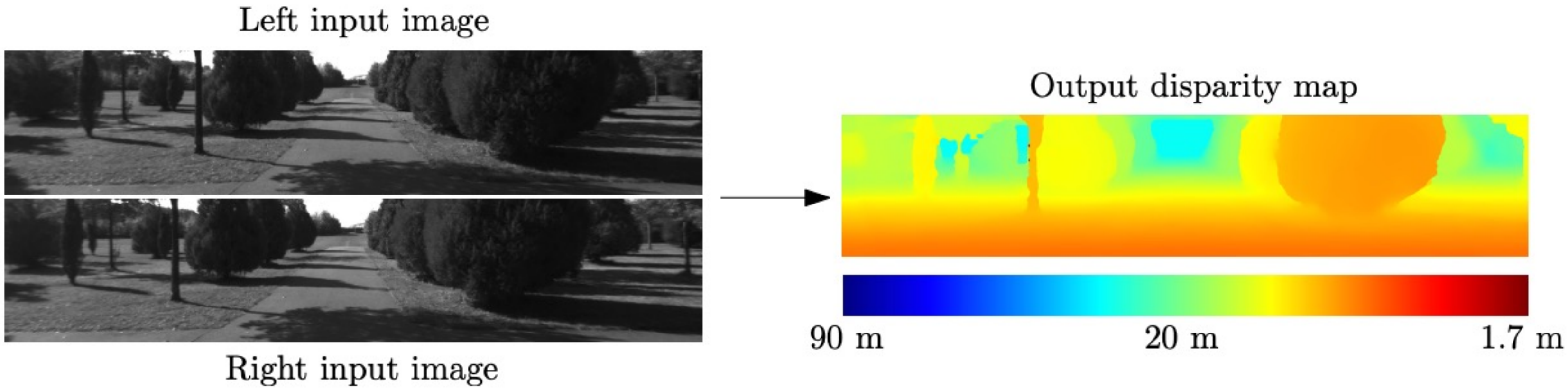
$$= \frac{[X-(X-B)]f}{Z}$$

$$= \frac{Bf}{Z}$$

$$\boxed{Z = \frac{Bf}{\delta}}$$

# From 2D to 3D: Depth Network

- A network that can output disparity.
- Using LiDAR or depth camera as groundtruth supervision.



*Zbontar & LeCun. Computing the Stereo Matching Cost with a Convolutional Neural Network. CVPR 2015.*

# The Energy-Based Approach

- The energy penalize matching with high cost (unary), and when neighboring pixels have disparity differences greater or equal to one (pairwise).
- Cost network: Train with binary classification

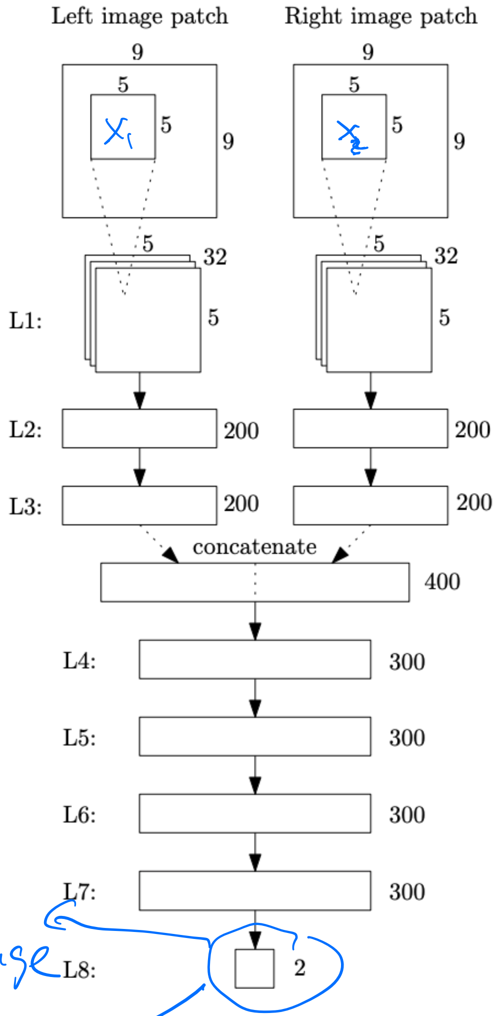
Energy $E(D) = \sum_{\mathbf{p}} \Big( \overset{\text{unary.}}{C^4_{\text{CBCA}}(\mathbf{p}, D(\mathbf{p}))}$

neighbors. $\left\{ \begin{aligned} &+ \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} P_1 \times 1\{|D(\mathbf{p}) - D(\mathbf{q})| = 1\} \\ &+ \sum_{\mathbf{q}\in\mathcal{N}_\mathbf{p}} P_2 \times 1\{|\underline{D(\mathbf{p})} - \underline{D(\mathbf{q})}| > 1\}\Big), \end{aligned} \right.$

$$D(\mathbf{p}) = \operatorname*{argmin}_d C(\mathbf{p}, d).$$

$\boxed{\phantom{x}} \longrightarrow$ sample same image

GT depth ← supervise.



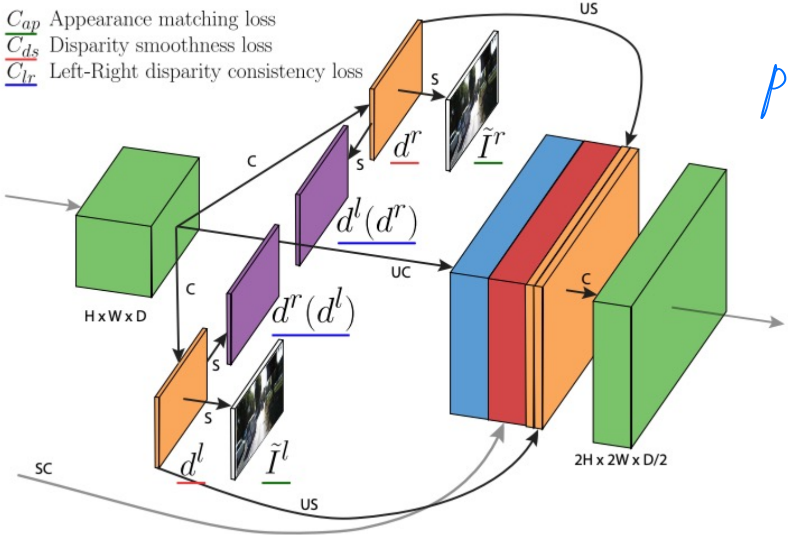
*Zbontar & LeCun. Computing the Stereo Matching Cost with a Convolutional Neural Network. CVPR 2015.*

# Self-Supervised Depth

- Appearance matching loss

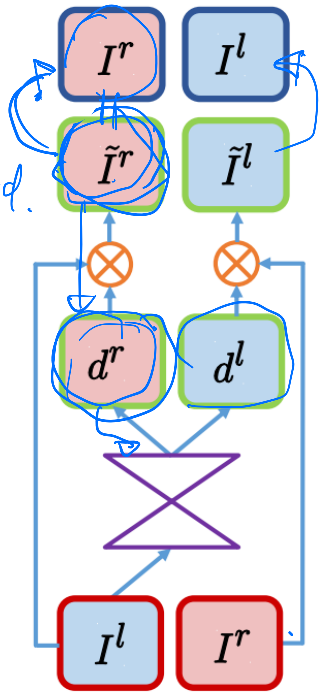
$$C^l_{ap} = \frac{1}{N}\sum_{i,j} \underline{\underline{\alpha \frac{1-\mathrm{SSIM}(I^l_{ij}, \tilde{I}^l_{ij})}{2}}} + (1-\alpha)\underline{\underline{\left\| I^l_{ij} - \tilde{I}^l_{ij} \right\|}}.$$

pixel.


$C_{ap}$ Appearance matching loss
$C_{ds}$ Disparity smoothness loss
$C_{lr}$ Left-Right disparity consistency loss
US
C
S
$d^r$
$\tilde{I}^r$
$d^l(d^r)$
UC
C
$d^r(d^l)$
S
$d^l$
$\tilde{I}^l$
H x W x D
C
2H x 2W x D/2
SC
US

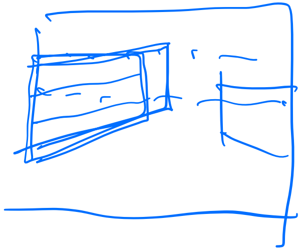
target.

pred.



Godard et al. Unsupervised Monocular Depth Estimation with Left-Right Consistency. CVPR 2017.
Wang et al. Image Quality Assessment: From Error Visibility to Structural Similarity. TIP 2004.

# Self-Supervised Depth



- Appearance matching loss

$$C^l_{ap} = \frac{1}{N}\sum_{i,j} \alpha \frac{1-\mathrm{SSIM}(I^l_{ij}, \tilde{I}^l_{ij})}{2} + (1-\alpha)\left\| I^l_{ij} - \tilde{I}^l_{ij} \right\|.$$

$\rightarrow \partial x$

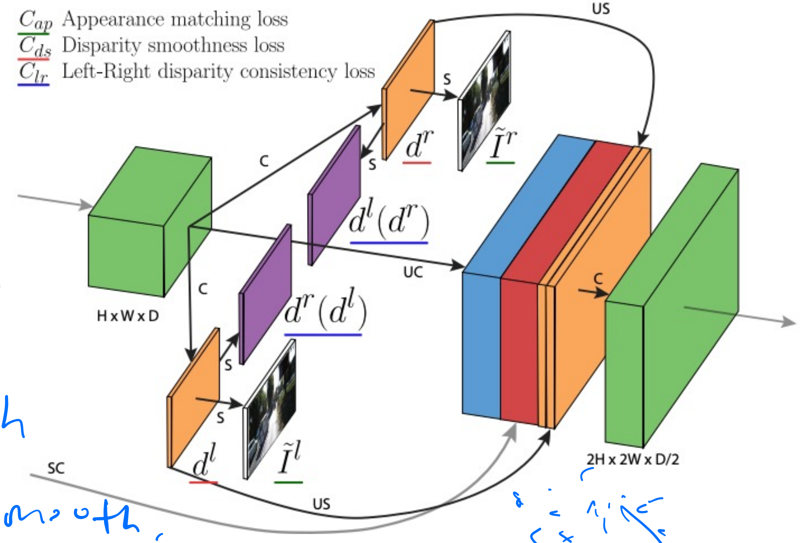
$\downarrow dy.$

- Disparity smoothness loss

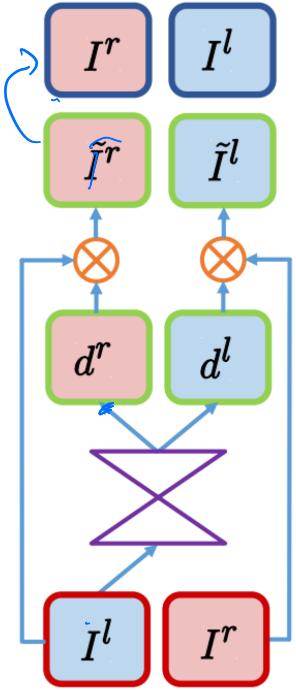
$$C^l_{ds} = \frac{1}{N}\sum_{i,j} \boxed{\left|\partial_x d^l_{ij}\right|} e^{-\left\|\boxed{\partial_x I^l_{ij}}\right\|} + \left|\partial_y d^l_{ij}\right| e^{-\left\|\partial_y I^l_{ij}\right\|}.$$

if image is smooth

then depth should also be smooth.



$C_{ap}$ Appearance matching loss
$C_{ds}$ Disparity smoothness loss
$C_{lr}$ Left-Right disparity consistency loss
US
S
C
$d^r$
$\tilde{I}^r$
$d^l(d^r)$
UC
$d^r(d^l)$
H x W x D
C
S
S
$d^l$
$\tilde{I}^l$
SC
US
2H x 2W x D/2



*Godard et al. Unsupervised Monocular Depth Estimation with Left-Right Consistency. CVPR 2017.*
*Wang et al. Image Quality Assessment: From Error Visibility to Structural Similarity. TIP 2004.*

# Self-Supervised Depth

- Appearance matching loss

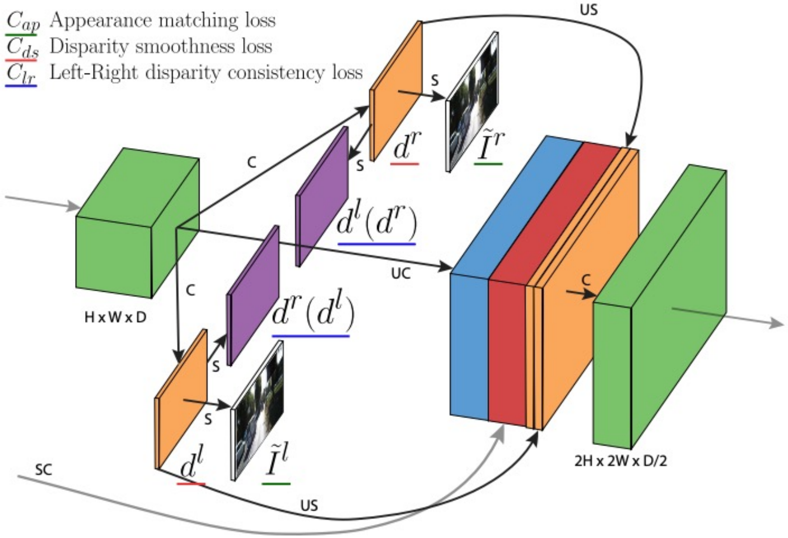
$$C^l_{ap} = \frac{1}{N}\sum_{i,j} \alpha \frac{1-\mathrm{SSIM}(I^l_{ij},\tilde{I}^l_{ij})}{2} + (1-\alpha)\left\| I^l_{ij} - \tilde{I}^l_{ij} \right\|.$$

- Disparity smoothness loss

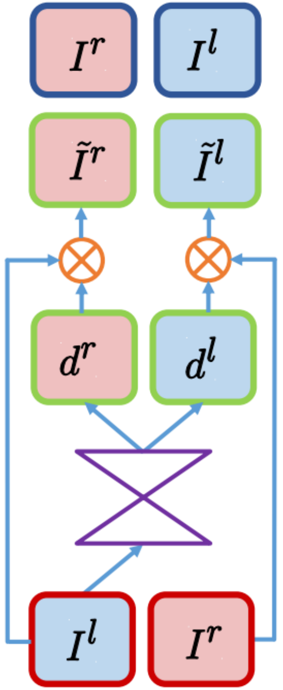
$$C^l_{ds} = \frac{1}{N}\sum_{i,j} \left|\partial_x d^l_{ij}\right| e^{-\left\|\partial_x I^l_{ij}\right\|} + \left|\partial_y d^l_{ij}\right| e^{-\left\|\partial_y I^l_{ij}\right\|}.$$

- Left-right disparity consistency loss

$$C^l_{lr} = \frac{1}{N}\sum_{i,j} \left| d^l_{ij} - d^r_{ij+d^l_{ij}} \right|.$$



$C_{ap}$ Appearance matching loss
$C_{ds}$ Disparity smoothness loss
$C_{lr}$ Left-Right disparity consistency loss
US
S
C
S
$d^r$
$\tilde{I}^r$
$d^l(d^r)$
UC
H x W x D
C
$d^r(d^l)$
S
S
C
$d^l$
$\tilde{I}^l$
SC
US
2H x 2W x D/2



Godard et al. Unsupervised Monocular Depth Estimation with Left-Right Consistency. CVPR 2017.
Wang et al. Image Quality Assessment: From Error Visibility to Structural Similarity. TIP 2004.

# Motion, Optical Flow

- Optical Flow: Estimate the motion of pixels across two consecutive video frames.

# Motion, Optical Flow

- Optical Flow: Estimate the motion of pixels across two consecutive video frames.
- Classic method uses brightness constancy assumption.

# Motion, Optical Flow

- Optical Flow: Estimate the motion of pixels across two consecutive video frames.
- Classic method uses brightness constancy assumption.

$$I(\underline{\underline{x}}, \underline{\underline{y}}, \underline{\underline{t}}) = I(\underbrace{x + \underline{\Delta x}, y + \underline{\Delta y}}_{\text{unk} \qquad \text{unk.}}, t + \underline{\underline{\Delta t}}).$$

# Motion, Optical Flow

- Optical Flow: Estimate the motion of pixels across two consecutive video frames.
- Classic method uses brightness constancy assumption.

Taylor expansion.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t).$$

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = \underline{I(x, y, t)} + \boxed{\frac{\partial I}{\partial x}}\underline{\Delta x} + \boxed{\frac{\partial I}{\partial y}}\underline{\Delta y} + \boxed{\frac{\partial I}{\partial t}}\underline{\Delta t}.$$

# Motion, Optical Flow

- Optical Flow: Estimate the motion of pixels across two consecutive video frames.
- Classic method uses brightness constancy assumption.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t).$$

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t.$$

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0.$$

# Motion, Optical Flow

- Optical Flow: Estimate the motion of pixels across two consecutive video frames.
- Classic method uses brightness constancy assumption.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t).$$

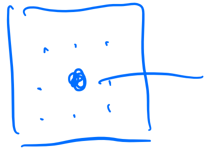
$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t.$$

$$I_x = \frac{\partial I}{\partial x}$$

$$I_y = \frac{\partial I}{\partial y}$$

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0.$$

# Motion, Optical Flow



1 point

- Optical Flow: Estimate the motion of pixels across two consecutive video frames.
- Classic method uses brightness constancy assumption.

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t).$$

$$I(x + \Delta x, y + \Delta y, t + \Delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t.$$

$$I_x = \frac{\partial I}{\partial x}$$

$$I_y = \frac{\partial I}{\partial y}$$

$$\frac{\partial I}{\partial x}\Delta x + \frac{\partial I}{\partial y}\Delta y + \frac{\partial I}{\partial t}\Delta t = 0.$$

2 unk.

1 equation

$$\boxed{I_x u + I_y v + I_t = 0.}$$

vertical.

horizonal displacement

# Classical Approach

- Under-constrained system

$$I_x u + I_y v + I_t = 0.$$

# Classical Approach

- Under-constrained system
- Use a local patch and assume smooth motion

$$I_x u + I_y v + I_t = 0.$$

$$\mathbf{A}\mathbf{u} = \mathbf{b}$$

$$\begin{pmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{N^2}) & I_y(\mathbf{p}_{N^2}) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} I_t(\mathbf{p}_1) \\ \vdots \\ I_t(\mathbf{p}_{N^2}) \end{pmatrix}$$

# Classical Approach

- Under-constrained system
- Use a local patch and assume smooth motion
- Rigid, contains many assumptions

$$I_x u + I_y v + I_t = 0.$$

lighting

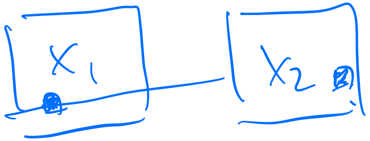
brightness — occlusion.

patch smoothness.

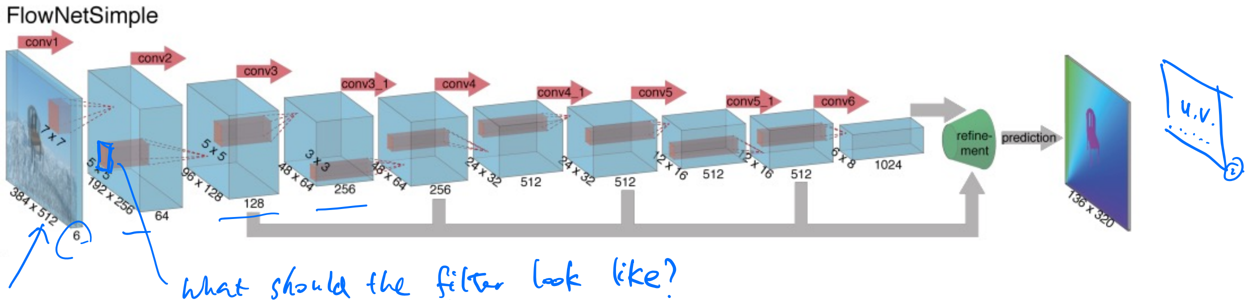
$$\mathbf{A}\mathbf{u} = \mathbf{b}$$

$$\begin{pmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{N^2}) & I_y(\mathbf{p}_{N^2}) \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = - \begin{pmatrix} I_t(\mathbf{p}_1) \\ \vdots \\ I_t(\mathbf{p}_{N^2}) \end{pmatrix}$$

# Correlation Volume Approach



- Simple Approach: Concatenate the two images together.



what should the filter look like?

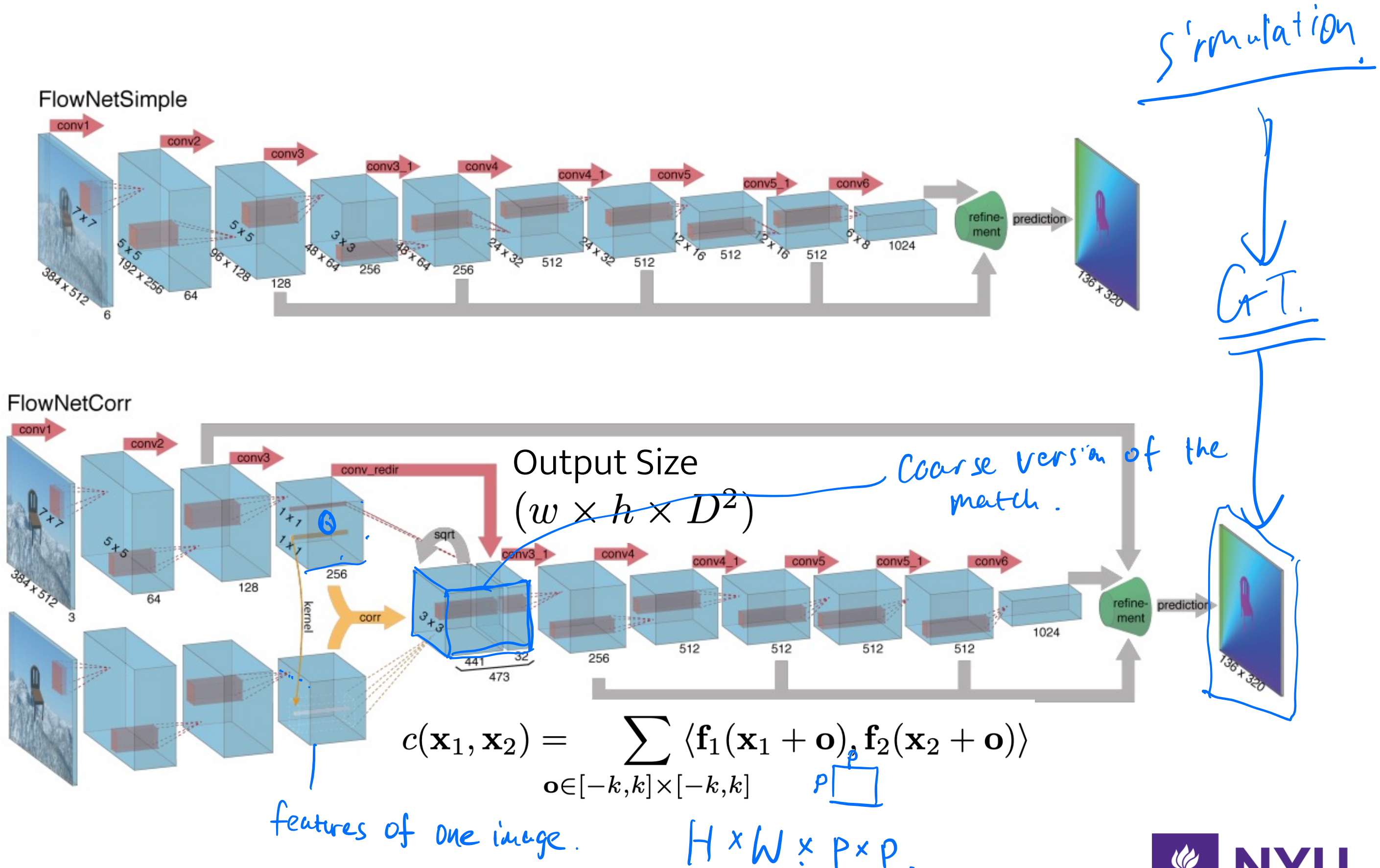
red circle.

chair

*Fischer et al. FlowNet: Learning Optical Flow with Convolutional Networks. ICCV 2015.*

# Correlation Volume Approach

- Simple Approach: Concatenate the two images together.
- Correlation: Extract some levels of features, and convolve one feature on top of another.



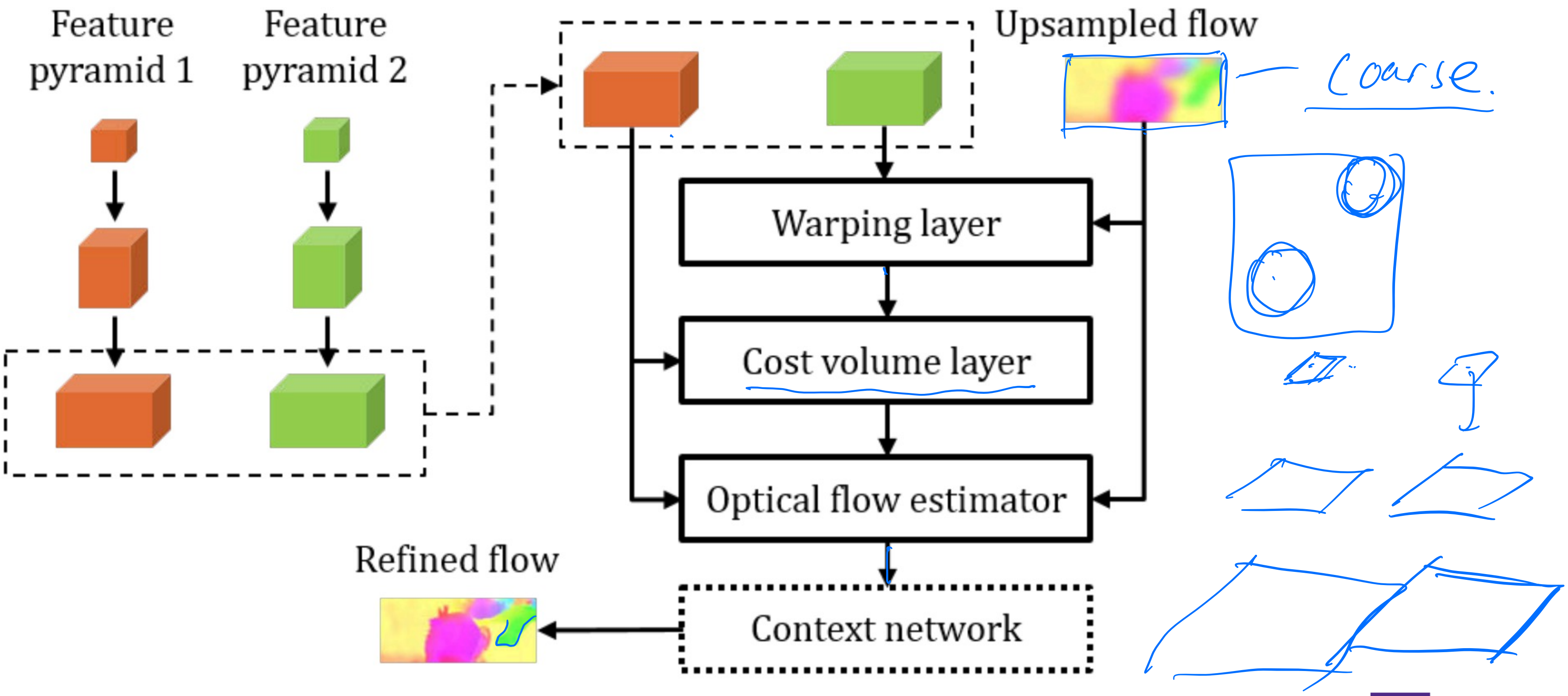
$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{\mathbf{o} \in [-k,k] \times [-k,k]} \langle \mathbf{f}_1(\mathbf{x}_1 + \mathbf{o}), \mathbf{f}_2(\mathbf{x}_2 + \mathbf{o}) \rangle$$

*Fischer et al. FlowNet: Learning Optical Flow with Convolutional Networks. ICCV 2015.*

NYU

# Iterative Refining through Feature Pyramid

Δx. Δy.



Coarse.

Sun et al. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume.

# Unsupervised Flow

- Photometric Consistency (Appearance)

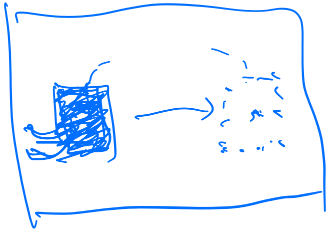
*Jonschkowski et al. What Matters in Unsupervised Optical Flow. ECCV 2020*

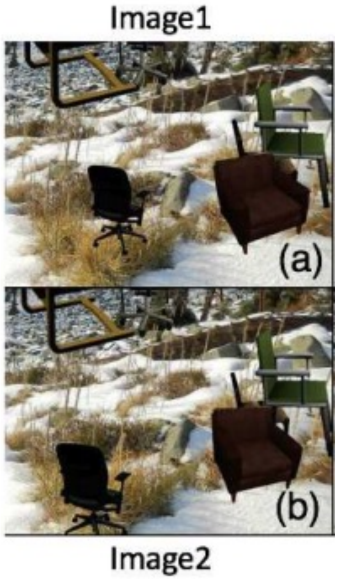
# Unsupervised Flow

$warp(I_t \rightarrow \hat{I}_{t+1})$.

$I_{t+1}$

- Photometric Consistency (Appearance)
- Occlusion Estimation
  - Forward-backward consistency



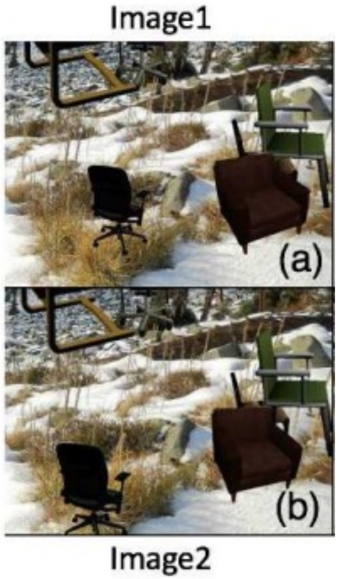


Wang et al., 2018

Jonschkowski et al. What Matters in Unsupervised Optical Flow. ECCV 2020

# Unsupervised Flow

- Photometric Consistency (Appearance)
- Occlusion Estimation
  - Forward-backward consistency
- Smoothness

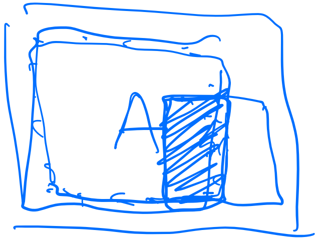


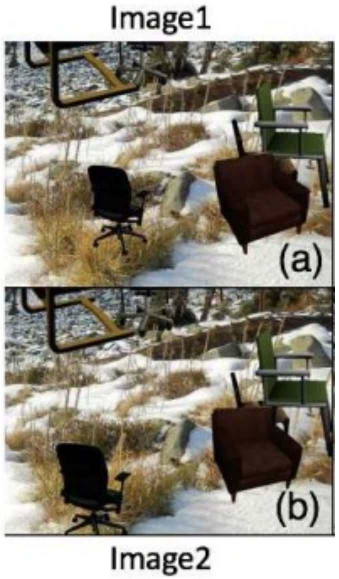
Wang et al., 2018

*Jonschkowski et al. What Matters in Unsupervised Optical Flow. ECCV 2020*

# Unsupervised Flow

- Photometric Consistency (Appearance)
- Occlusion Estimation
  - Forward-backward consistency
- Smoothness
- Self-supervision: Ensure consistent flow at different augmentation (e.g. crops)



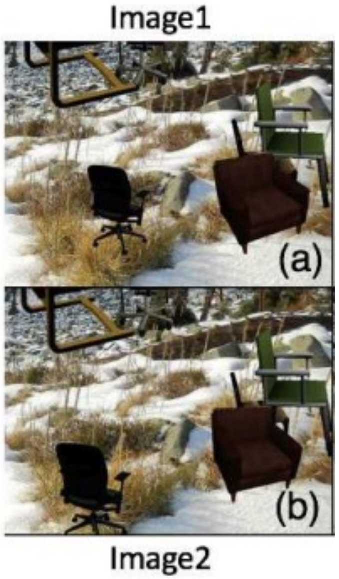


Wang et al., 2018

*Jonschkowski et al. What Matters in Unsupervised Optical Flow. ECCV 2020*

# Unsupervised Flow

- Photometric Consistency (Appearance)
- Occlusion Estimation
  - Forward-backward consistency
- Smoothness
- Self-supervision: Ensure consistent flow at different augmentation (e.g. crops)
- Can 3D information help us reason about motion?



Wang et al., 2018

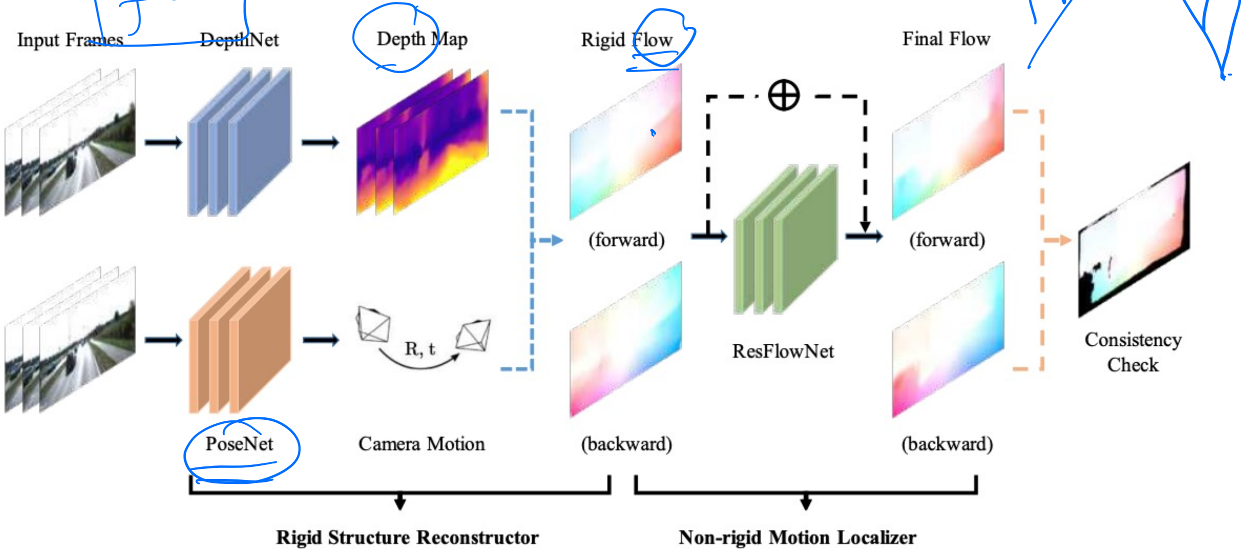
*Jonschkowski et al. What Matters in Unsupervised Optical Flow. ECCV 2020*

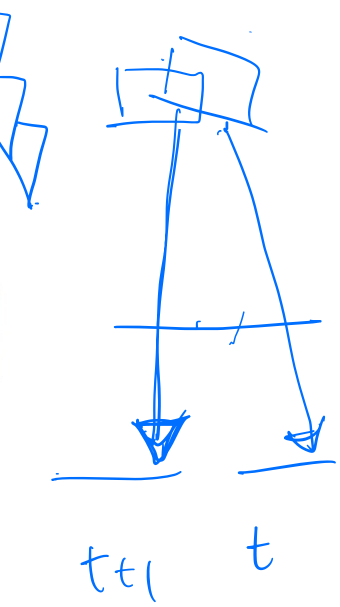
# Depth, Flow, and Pose Movement

- The static objects follow rigid flow: determined by camera motion and depth.

$$f^{rig}_{t\mapsto s}(p_t) = KT_{t\mapsto s}D_t(p_t)K^{-1}p_t - p_t.$$

f full





Yin & Shi. GeoNet: Unsupervised Learning of Dense Depth, Optical Flow and Camera Pose. CVPR 2018.

# Training Losses

rigid.

- Appearance Loss (Warping):

$$\mathcal{L}_{rw} = \alpha \frac{1-SSIM(I_t, \tilde{I}_s^{rig})}{2} + (1-\alpha)\|I_t - \tilde{I}_s^{rig}\|_1.$$

$$\mathcal{L}_{fw} = \alpha \frac{1-SSIM(I_t, \tilde{I}_s^{full})}{2} + (1-\alpha)\|I_t - \tilde{I}_s^{full}\|_1.$$

full.

# Training Losses

- Appearance Loss (Warping):

$$\mathcal{L}_{rw} = \alpha \frac{1 - SSIM(I_t, \tilde{I}_s^{rig})}{2} + (1 - \alpha) \| I_t - \tilde{I}_s^{rig} \|_1.$$

$$\mathcal{L}_{fw} = \alpha \frac{1 - SSIM(I_t, \tilde{I}_s^{full})}{2} + (1 - \alpha) \| I_t - \tilde{I}_s^{full} \|_1.$$

- Smoothness Loss:

$$\mathcal{L} = \sum_{p_t} |\underline{\nabla D(p_t)}| \cdot (\exp(-|\underline{\nabla I(p(t)}|))^T.$$

# Training Losses

- Appearance Loss (Warping):

$$\mathcal{L}_{rw} = \alpha \frac{1 - SSIM(I_t, \tilde{I}_s^{rig})}{2} + (1-\alpha)\|I_t - \tilde{I}_s^{rig}\|_1.$$

$$\mathcal{L}_{fw} = \alpha \frac{1 - SSIM(I_t, \tilde{I}_s^{full})}{2} + (1-\alpha)\|I_t - \tilde{I}_s^{full}\|_1.$$

- Smoothness Loss:

$$\mathcal{L} = \sum_{p_t} |\nabla D(p_t)| \cdot (\exp(-|\nabla I(p(t)|))^T.$$

- Forward-Backward Consistency:

$$\mathcal{L} = \sum_{p_t} [\delta(p_t)] \cdot \|\Delta f_{t \mapsto s}^{full}(p_t)\|_1.$$

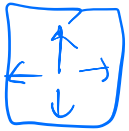
$$\delta(p_t) = \|f_{t \mapsto s}^{full}(p_t)\|_2 \max\{\alpha, \beta\|f_{t \mapsto s}^{full}(p_t)\|_2\}.$$

if $\Delta$ flow is too big, then ignore.

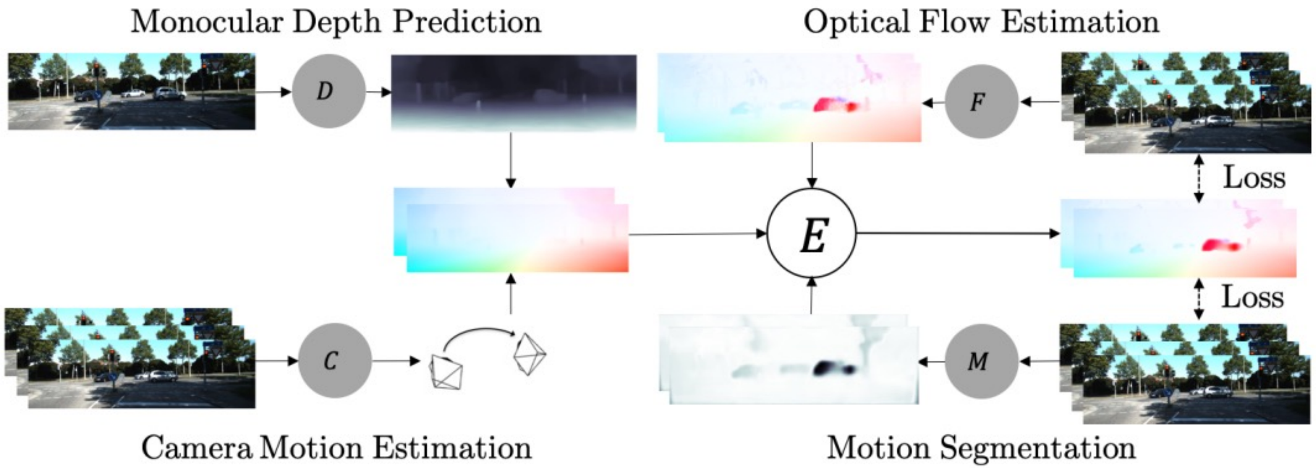
# Summary

→ horizontal → depth

flow



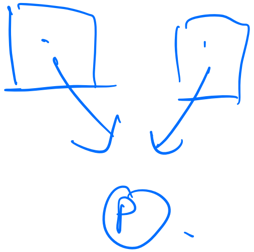
- Leverage cross correlation structure for spatial similarity matching.

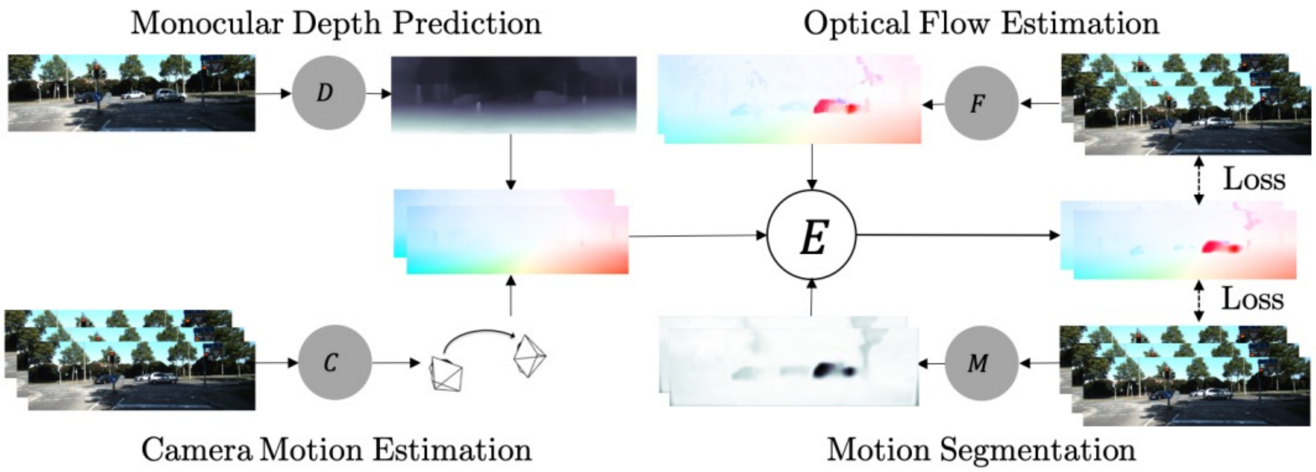


*Ranjan et al. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera, Motion, Optical Flow and Motion Segmentation. CVPR 2019*

# Summary

- Leverage cross correlation structure for spatial similarity matching.
- Can be used towards: depth, flow, and pose prediction.

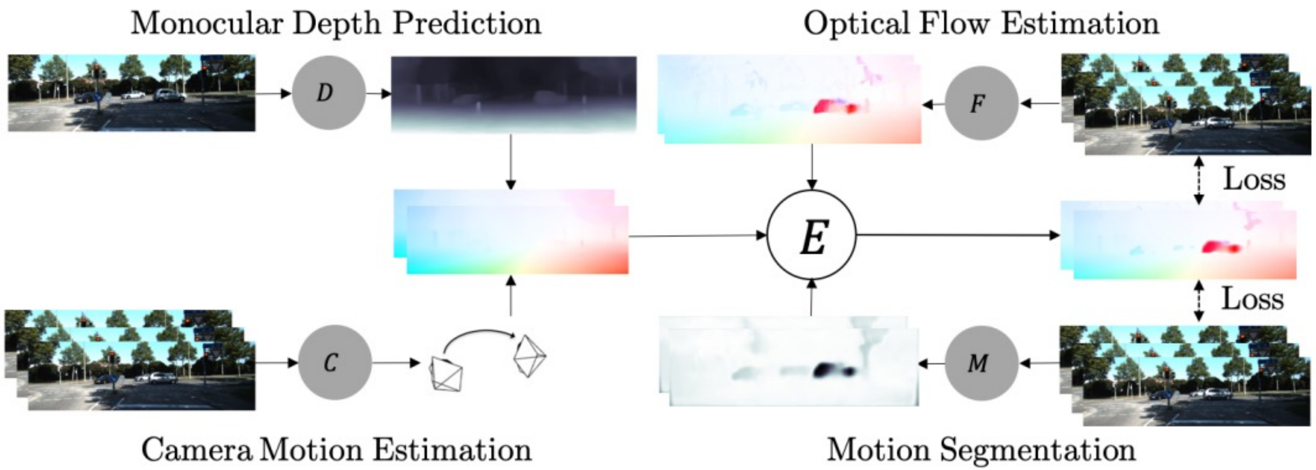




*Ranjan et al. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera, Motion, Optical Flow and Motion Segmentation. CVPR 2019*

# Summary

- Leverage cross correlation structure for spatial similarity matching.
- Can be used towards: depth, flow, and pose prediction.
- Can form triangulation for self-supervision. — static object.



*Ranjan et al. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera, Motion, Optical Flow and Motion Segmentation. CVPR 2019*

# Classical Mapping

Depth

Motion.

- Estimating 3D structure and location from 2D observations.

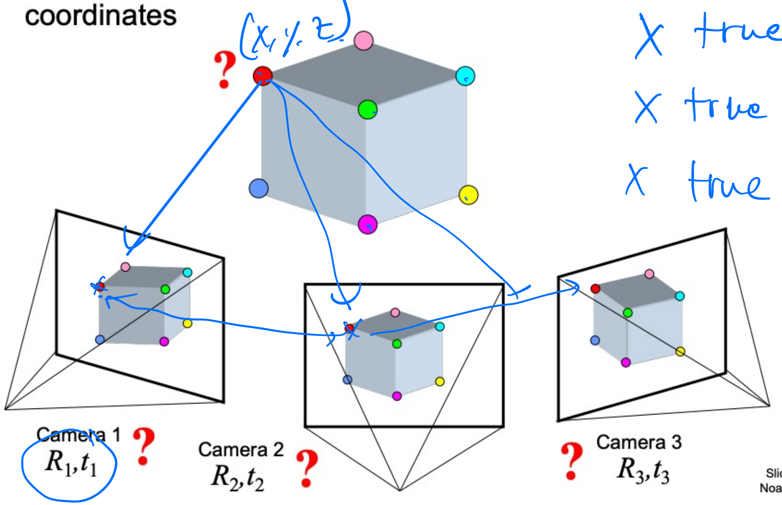
× GPS

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates

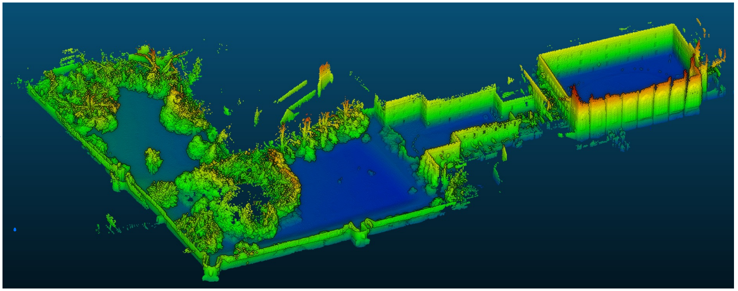
× true cam pose

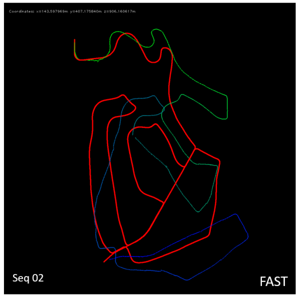
× true key points

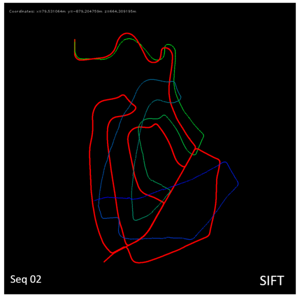
× true 3d.



Slide credit: Noah Snavely







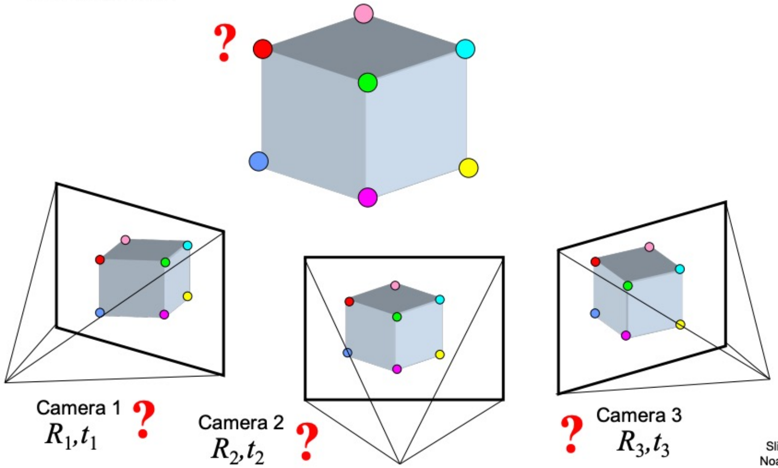
Garg & Jain

# Classical Mapping

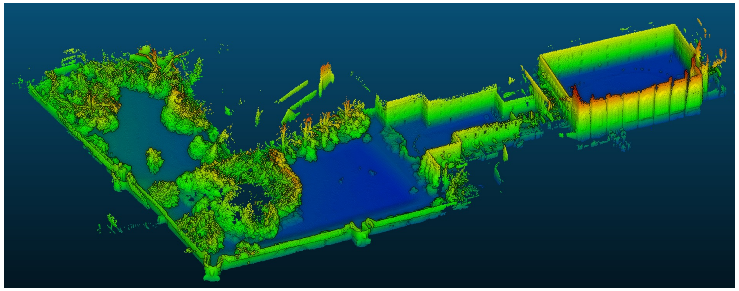
- Estimating 3D structure and location from 2D observations.
- Simultaneous Localization and Mapping. → 3d structure.

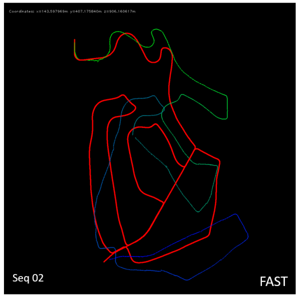
↓ pose .

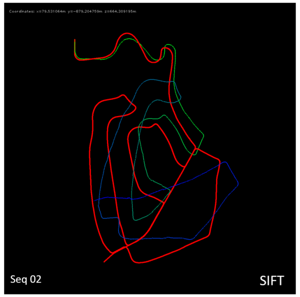
- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



Slide credit:
Noah Snavely





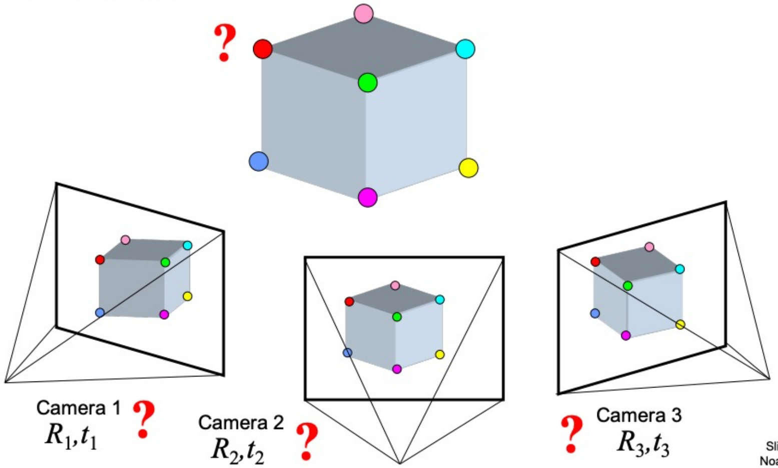


Garg & Jain

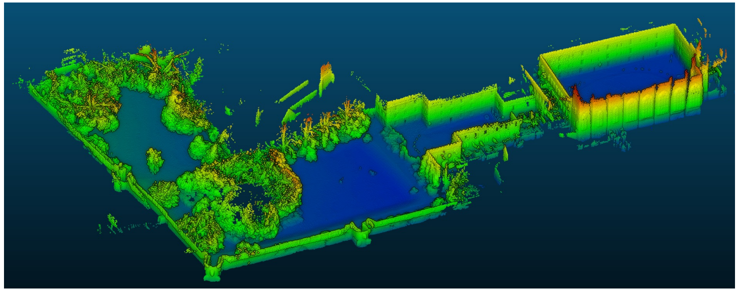
# Classical Mapping

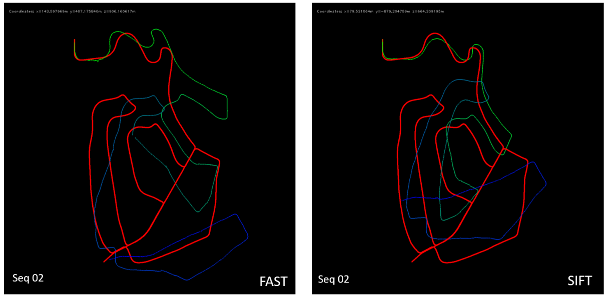
- Estimating 3D structure and location from 2D observations.
- Simultaneous Localization and Mapping.
- Common Techniques: Extended Kalman Filter, GraphSLAM

- Given a set of corresponding points in two or more images, compute the camera parameters and the 3D point coordinates



Slide credit:
Noah Snavely





Garg & Jain

# Common Drawbacks

- Probabilistic inference can take long to compute, and mapping takes a large memory storage.

# Common Drawbacks

- Probabilistic inference can take long to compute, and mapping takes a large memory storage.
- Great for 3D reconstruction but downstream tasks may not need a full precision explicit map.

# Common Drawbacks

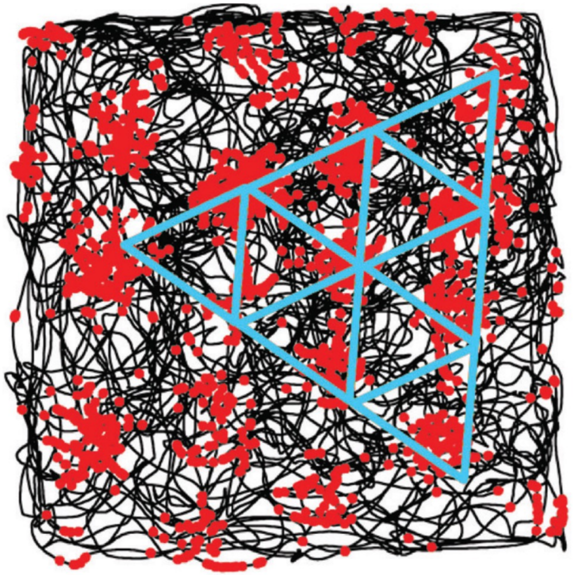
- Probabilistic inference can take long to compute, and mapping takes a large memory storage.
- Great for 3D reconstruction but downstream tasks may not need a full precision explicit map.
- May not fully understand dynamic objects (averaging across multiple scans).

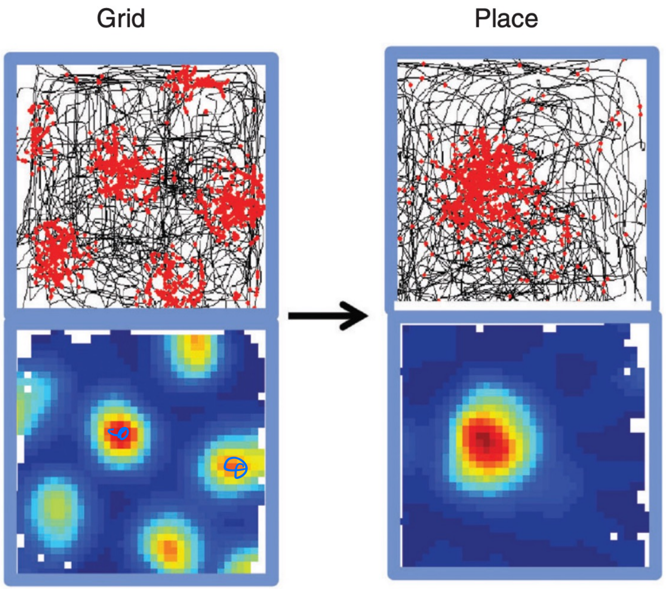
# Common Drawbacks

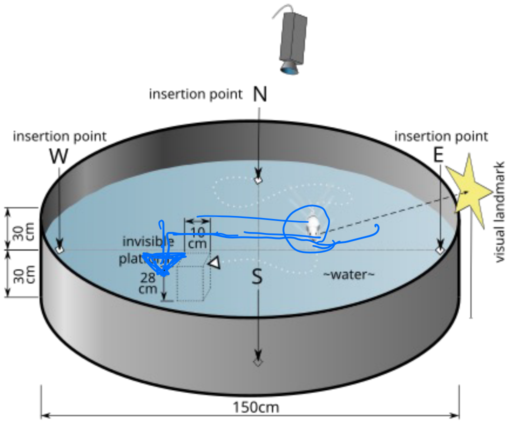
- Probabilistic inference can take long to compute, and mapping takes a large memory storage.
- Great for 3D reconstruction but downstream tasks may not need a full precision explicit map.
- May not fully understand dynamic objects (averaging across multiple scans).
- Is there a more end-to-end version?

# Mapping in the Brain: Grid and Place Cells

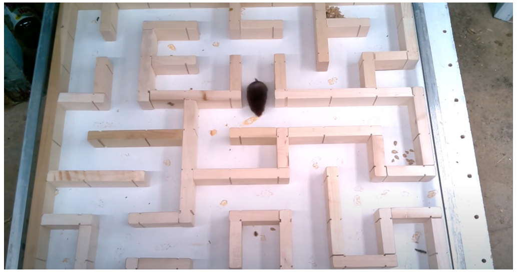
Metric







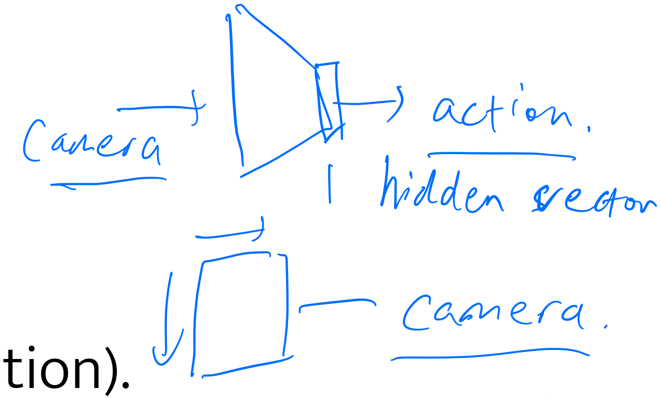
Morris water maze



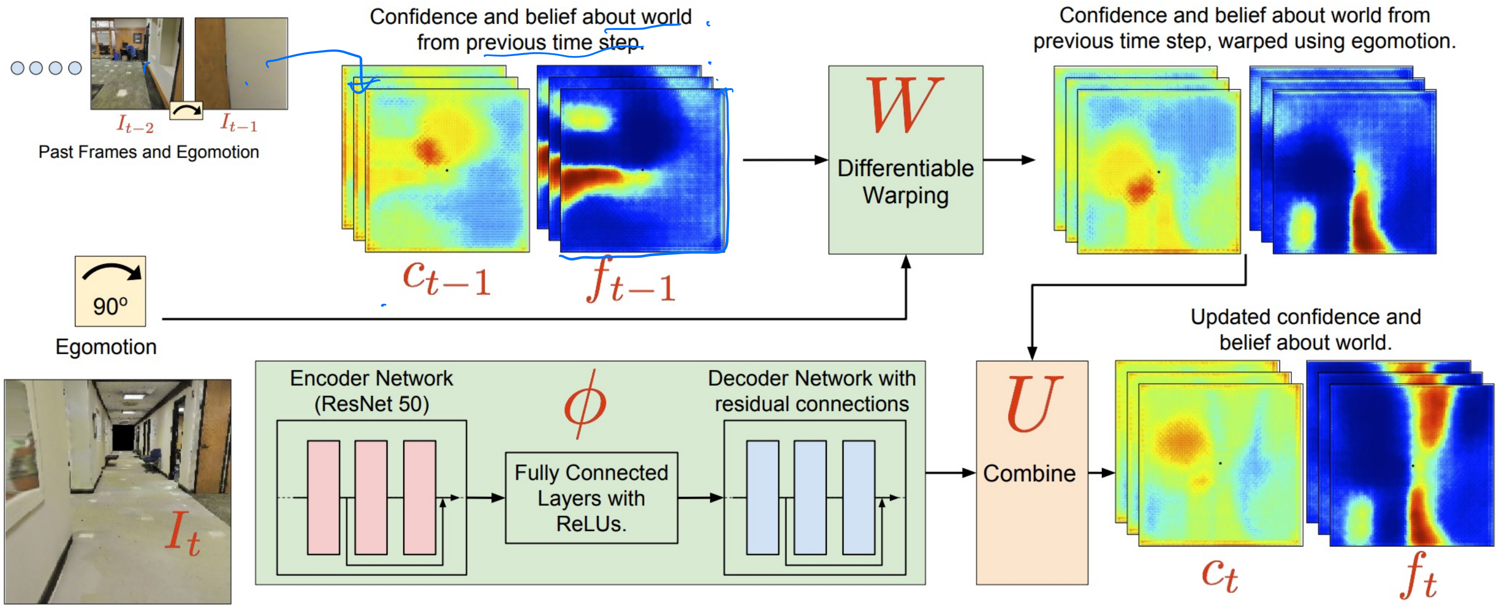
Matthias Wandel, 2018

*May-Britt Moser, David C. Rowland, and Edvard I. Moser. Place Cells, Grid Cells, and Memory.*

# Neural Mapping



- Can we learn a mapping representation?
- Metric space, top-down warping (known egomotion).

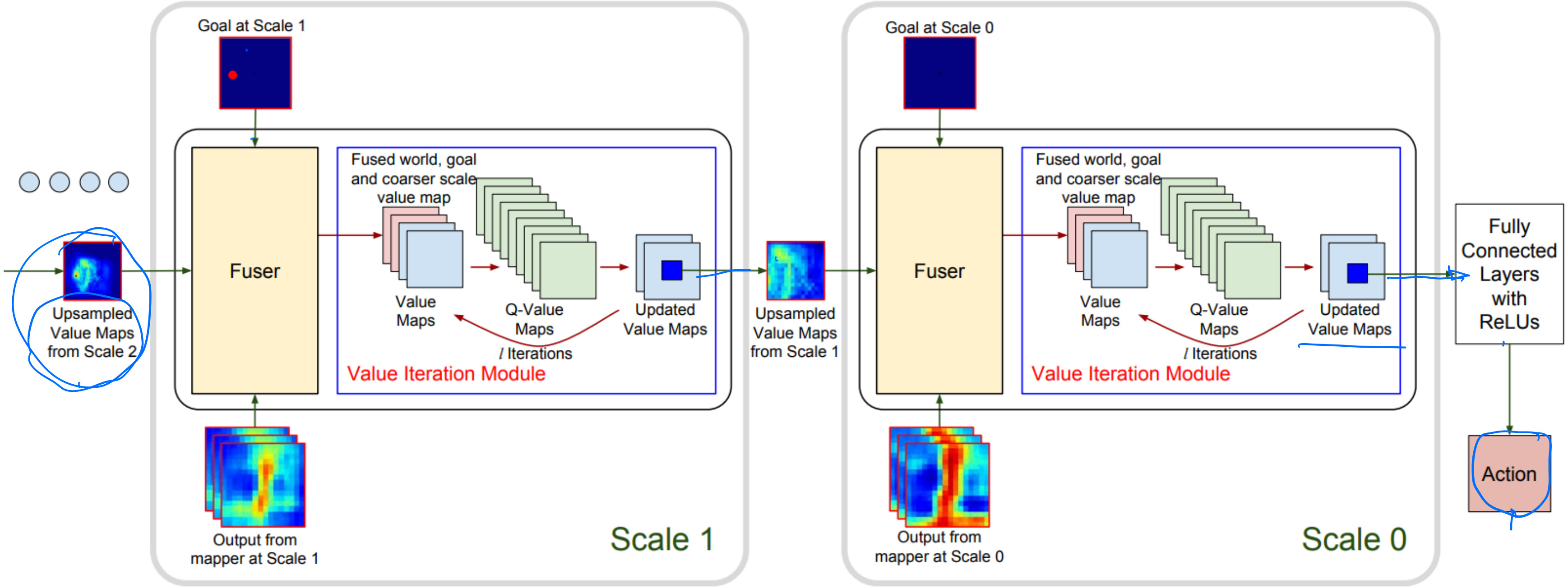

$I_{t-2}$
$I_{t-1}$
Past Frames and Egomotion
Confidence and belief about world from previous time step.
$c_{t-1}$
$f_{t-1}$
$W$
Differentiable Warping
Confidence and belief about world from previous time step, warped using egomotion.
90°
Egomotion
$I_t$
Encoder Network (ResNet 50)
$\phi$
Fully Connected Layers with ReLUs.
Decoder Network with residual connections
$U$
Combine
Updated confidence and belief about world.
$c_t$
$f_t$

Gupta et al. Cognitive Mapping and Planning for Visual Navigation. IJCV 2017.

# Hierarchical Planning

egocentric

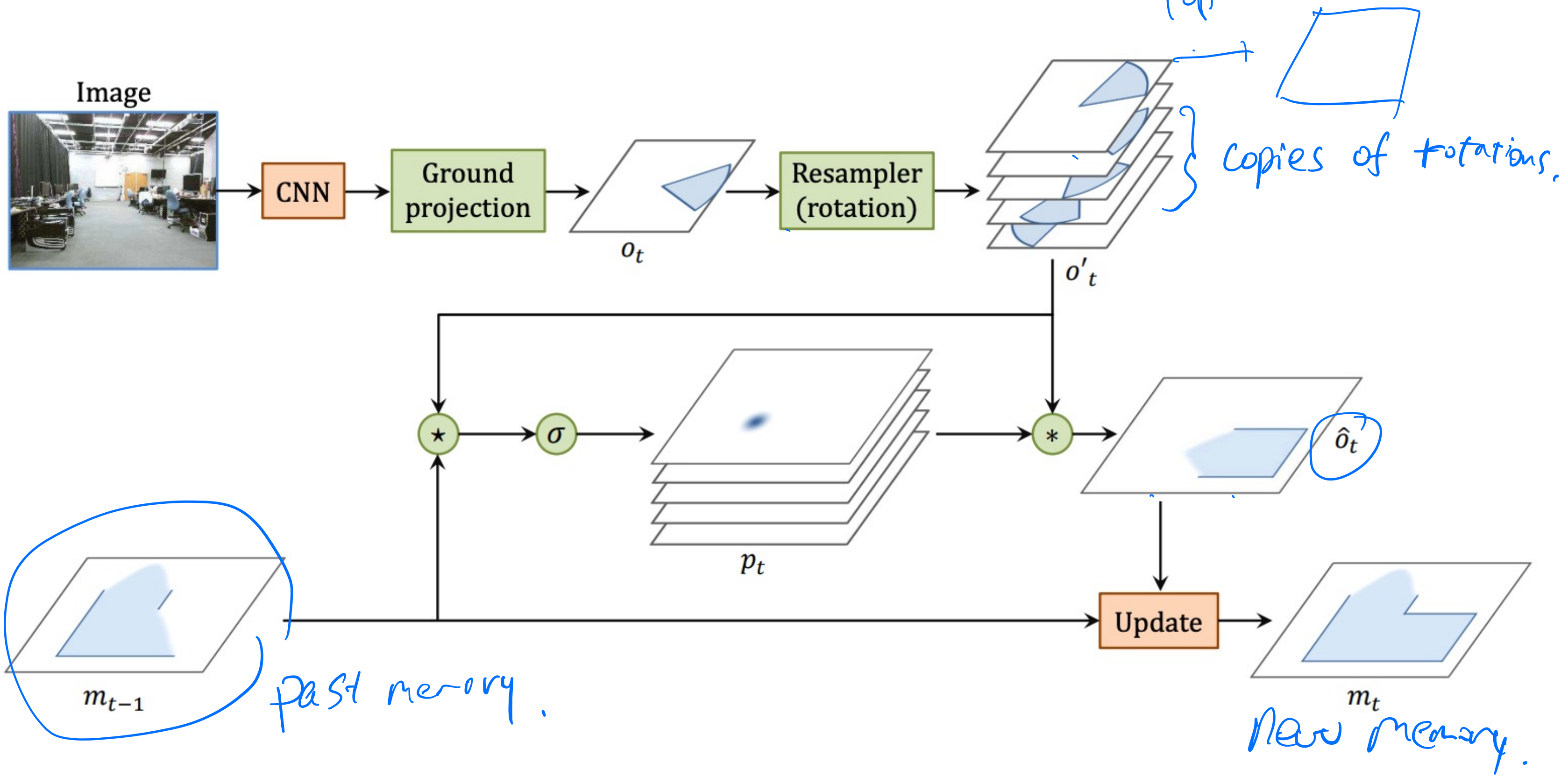
- How do we use the learned map (allocentric) feature of the world?



Goal at Scale 1
Fuser
Fused world, goal and coarser scale value map
Value Maps
Q-Value Maps
Updated Value Maps
l Iterations
Value Iteration Module
Upsampled Value Maps from Scale 2
Output from mapper at Scale 1
Scale 1
Upsampled Value Maps from Scale 1
Goal at Scale 0
Fuser
Fused world, goal and coarser scale value map
Value Maps
Q-Value Maps
Updated Value Maps
l Iterations
Value Iteration Module
Output from mapper at Scale 0
Scale 0
Fully Connected Layers with ReLUs
Action

Gupta et al. Cognitive Mapping and Planning for Visual Navigation. IJCV 2017.

# Simultaneous Localization and Registration



top-down view.

copies of rotations.

past memory.

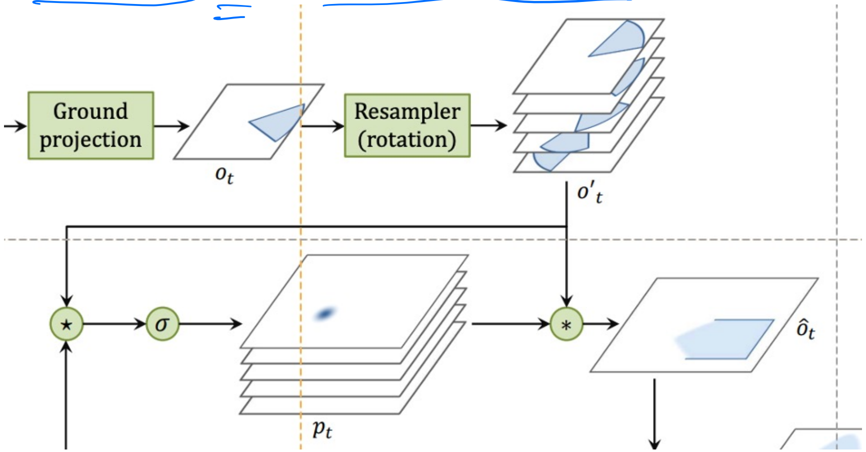
new memory.

*Henriques & Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. CVPR 2018.*

# Simultaneous Localization and Registration

- The observations $\boxed{o_t}$ are transformed into a stack $o_t'$ by applying a rotation resampler.

$$o'_{ijkl} = [R(o, 2\pi l/r)]_{ijk}.$$



Henriques & Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. CVPR 2018.

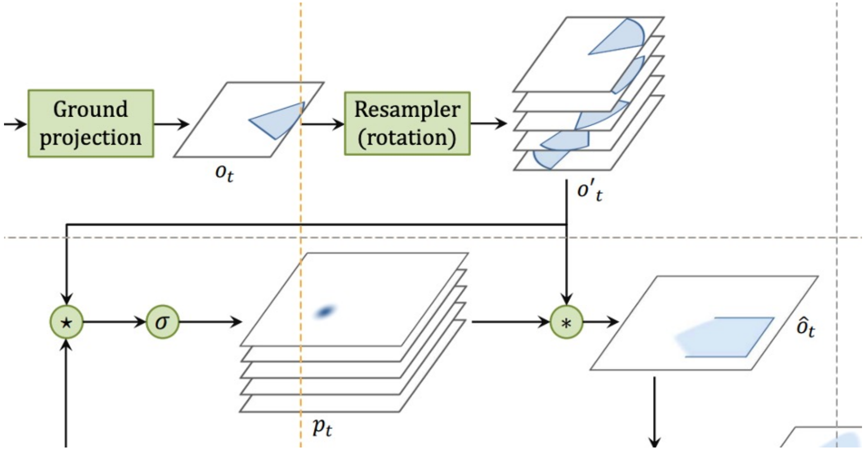
# Simultaneous Localization and Registration

- The observations $o_t$ are transformed into a stack $o'_t$ by applying a rotation resampler.

$$o'_{ijkl} = [R(o, 2\pi l/r)]_{ijk}.$$

- $o'_t$ convolve with the base feature.

$$\boxed{p_t} = \underline{\text{Softmax}}(\underline{m_{t-1}} * \underline{o'_t}).$$



Henriques & Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. CVPR 2018.

# Simultaneous Localization and Registration

- The observations $o_t$ are transformed into a stack $o'_t$ by applying a rotation resampler.

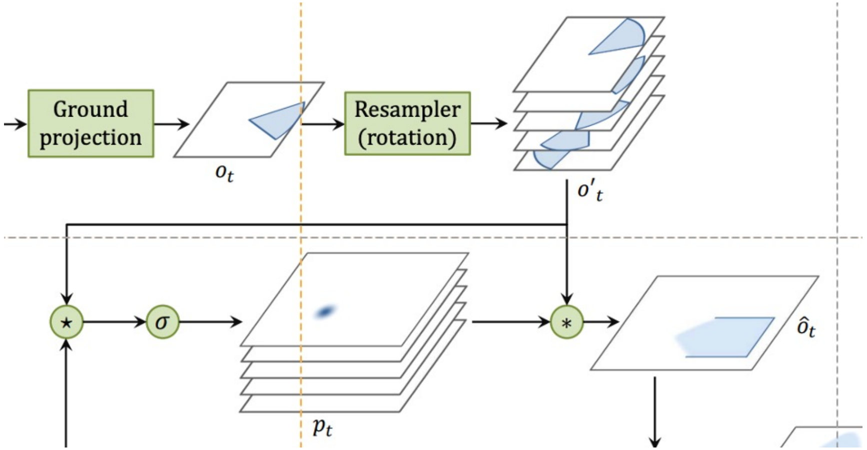
$$o'_{ijkl} = [R(o, 2\pi l/r)]_{ijk}.$$

- $o'_t$ convolve with the base feature.

$$p_t = \mathrm{Softmax}(m_{t-1} * o'_t).$$

- Transform observations into allocentric

$$\hat{o}_t = \sum_{uvw} \underline{\boxed{p_{uvw}}} T(o|u, v, w).$$



Henriques & Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. CVPR 2018.

# Simultaneous Localization and Registration

- The observations $o_t$ are transformed into a stack $o'_t$ by applying a rotation resampler.

$$o'_{ijkl} = [R(o, 2\pi l/r)]_{ijk}.$$

- $o'_t$ convolve with the base feature.

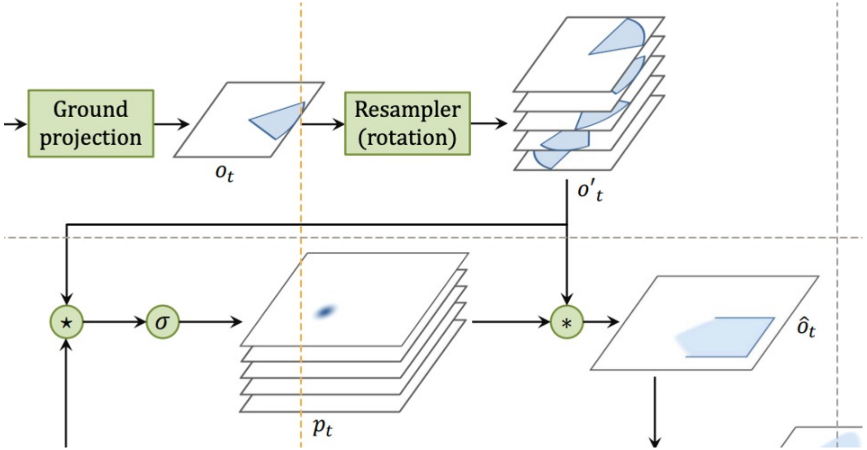
$$p_t = \mathrm{Softmax}(m_{t-1} * o'_t).$$

- Transform observations into allocentric

$$\boxed{\hat{o}_t} = \sum_{uvw} p_{uvw} T(o|u, v, w).$$

- Update belief:

$$\boxed{m_{i,j,t+1}} = \underline{\mathrm{LSTM}}(\underline{m_{i,j,t}}, \underline{\hat{o}_{i,j,t}}).$$



planner.

*Henriques & Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. CVPR 2018.*

# Simultaneous Localization and Registration

- The observations $o_t$ are transformed into a stack $o_t'$ by applying a rotation resampler.

$$o'_{ijkl} = [R(o, 2\pi l/r)]_{ijk}.$$

- $o_t'$ convolve with the base feature.

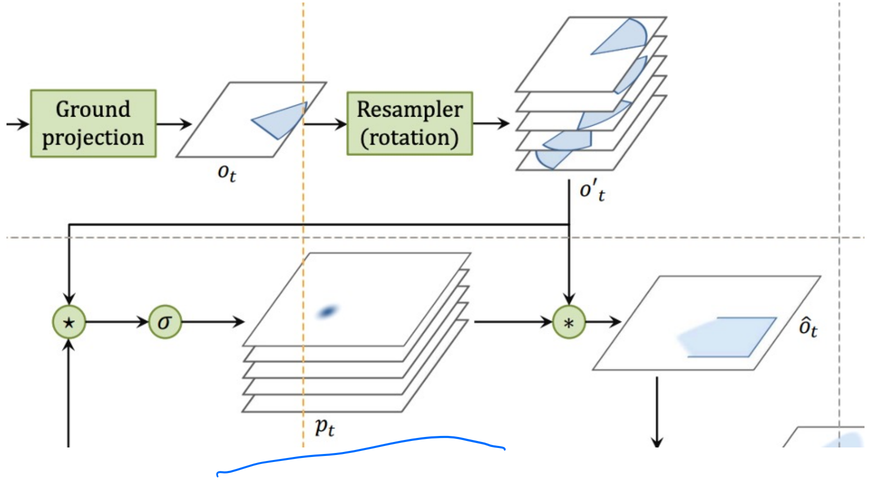
$$p_t = \mathrm{Softmax}(m_{t-1} * o_t').$$

- Transform observations into allocentric

$$\hat{o}_t = \sum\nolimits_{uvw} p_{uvw} T(o|u, v, w).$$

- Update belief:

$$m_{i,j,t+1} = \mathrm{LSTM}(m_{i,j,t}, \hat{o}_{i,j,t}).$$

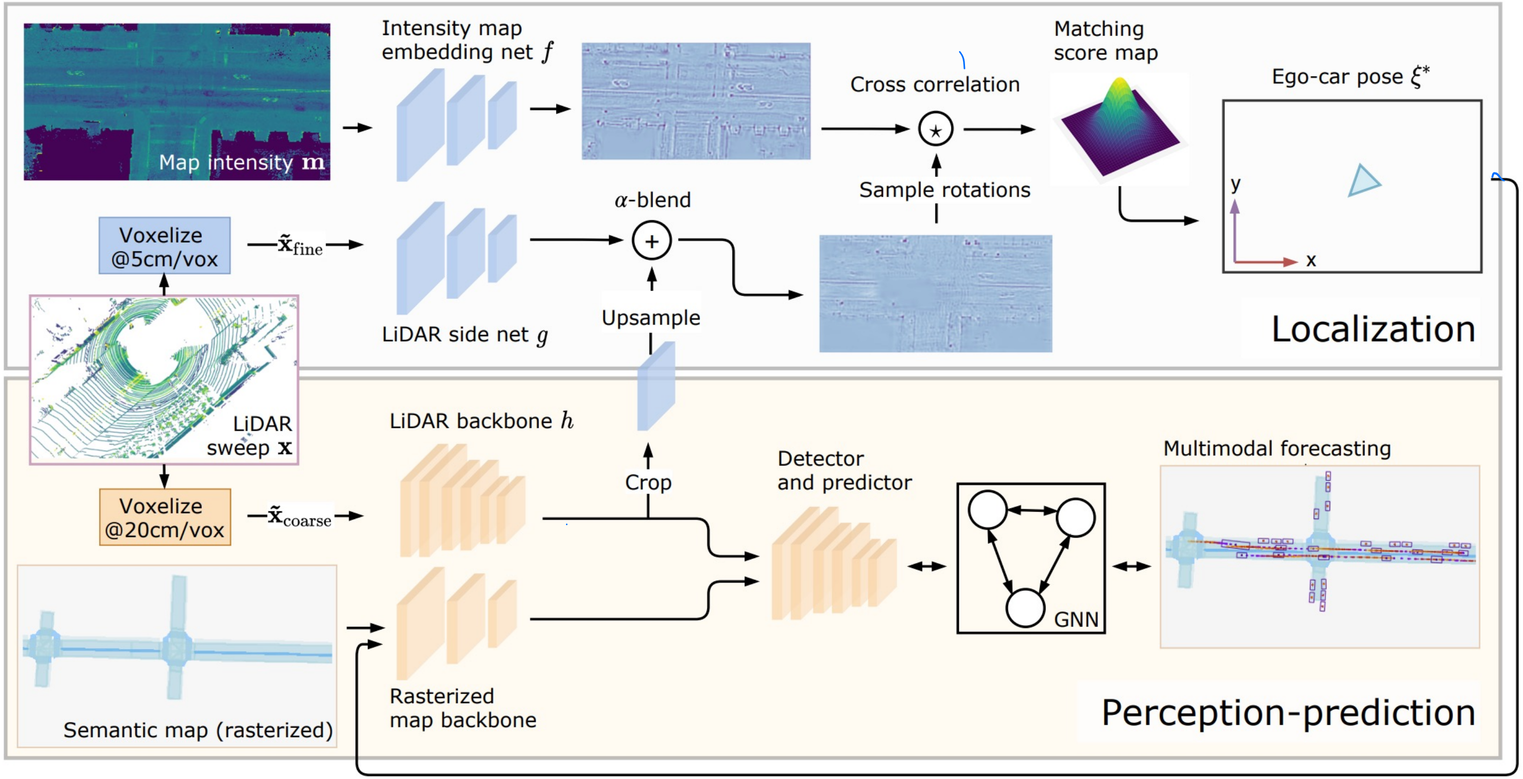


Loss:

$$\mathcal{L}(p) = \underline{-\log \sum\nolimits_t \boxed{p_{H_t W_t R_t t}}}.$$

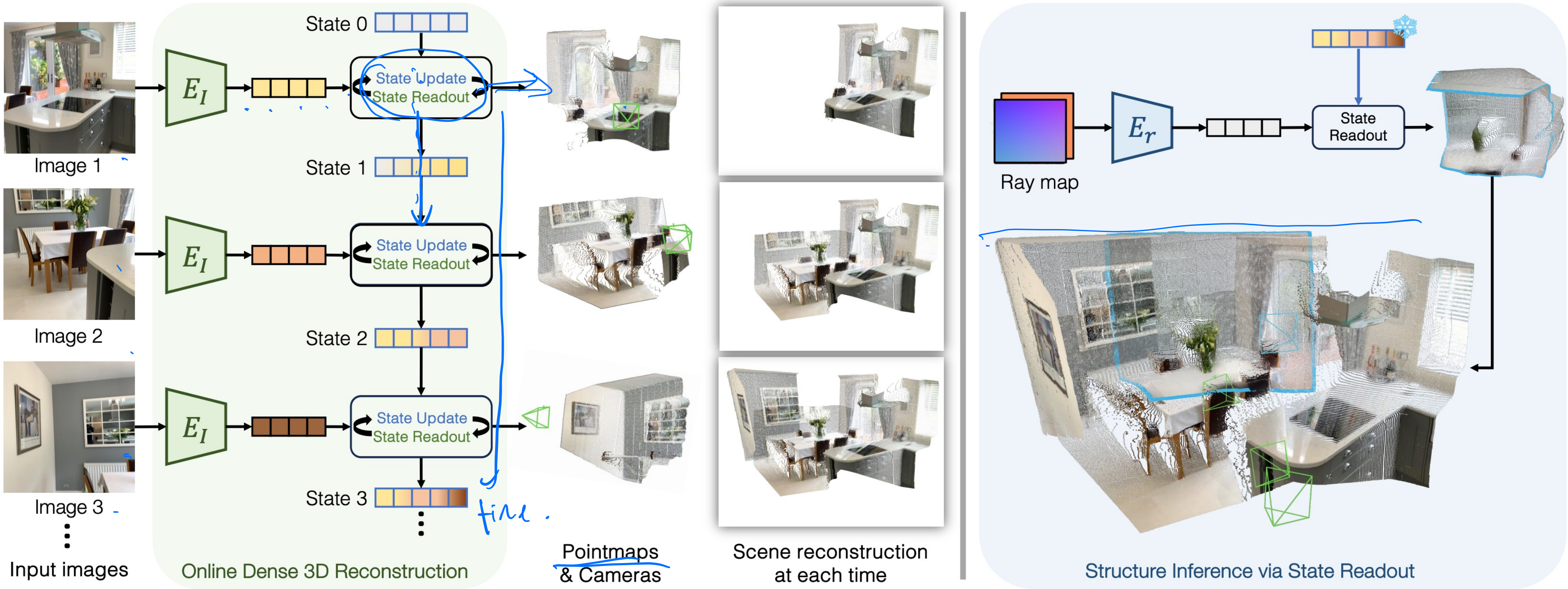
Henriques & Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. CVPR 2018.

# Joint Localization, Perception and Prediction

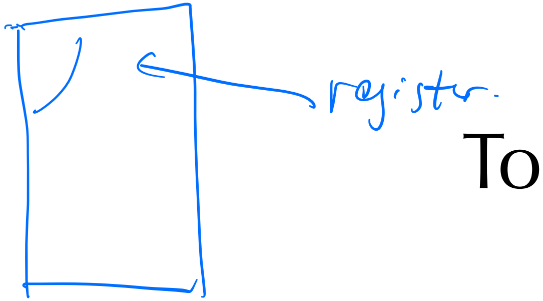


*Philips et al. Deep Multi-Task Learning for Joint Localization, Perception, and Prediction CVPR 2021.*

# Continuous 3D Perception and Mapping

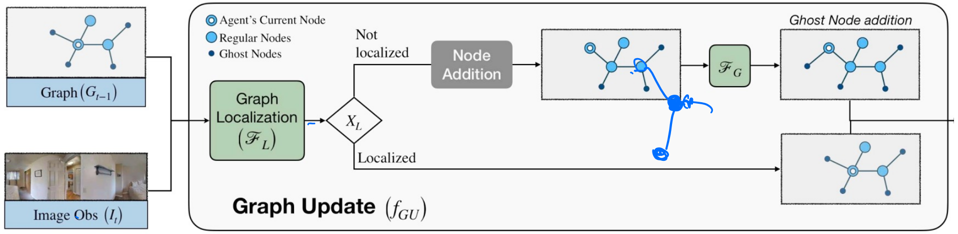


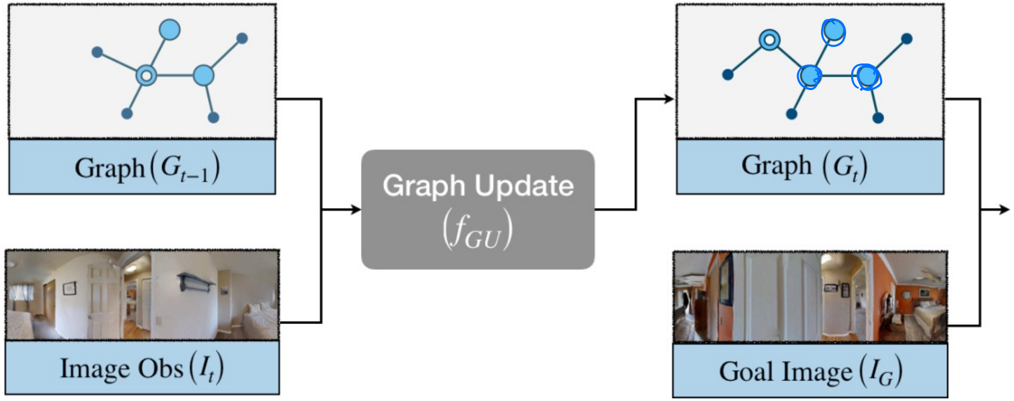
*Wang et al. Continuous 3D Perception Model with Persistent State. arXiv 2025.*

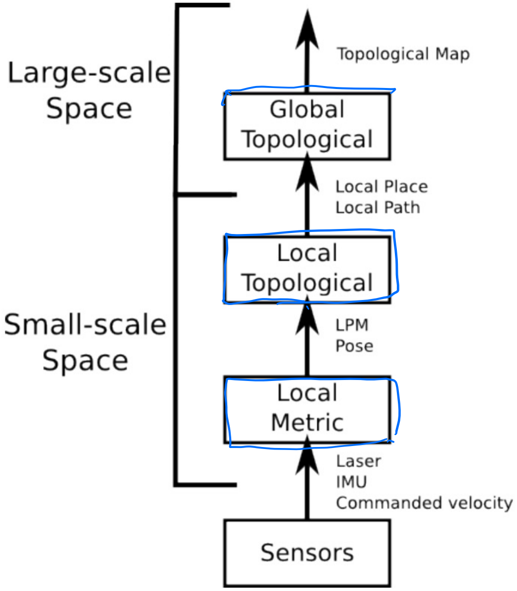


# Topological Mapping

- High-level graph representation
- Each node contains more summarized information
- Enables global planning







Johnson, 2018

*Johnson. Topological Mapping and Navigation in Real-World Environments. 2018.*
*Chaplot et al. Neural Topological SLAM for Visual Navigation. CVPR 2020.*

# Summary

- Covers 3D, motion, depth, and mapping.

# Summary

- Covers 3D, motion, depth, and mapping.
- Still needs high-level features (recognizing the object and semantics): Spatial pyramid.

# Summary

- Covers 3D, motion, depth, and mapping.
- Still needs high-level features (recognizing the object and semantics): Spatial pyramid.
- Can be made unsupervised

# Summary

- Covers 3D, motion, depth, and mapping.
- Still needs high-level features (recognizing the object and semantics): Spatial pyramid.
- Can be made unsupervised
- Design end-to-end modules that contain rich features.

# Summary

- Covers 3D, motion, depth, and mapping.
- Still needs high-level features (recognizing the object and semantics): Spatial pyramid.
- Can be made unsupervised
- Design end-to-end modules that contain rich features.
- Design joint learning frameworks.

# Summary

- Covers 3D, motion, depth, and mapping.
- Still needs high-level features (recognizing the object and semantics): Spatial pyramid.
- Can be made unsupervised
- Design end-to-end modules that contain rich features.
- Design joint learning frameworks.
- Using geometric transformation to ground representations.

# Summary

- Covers 3D, motion, depth, and mapping.
- Still needs high-level features (recognizing the object and semantics): Spatial pyramid.
- Can be made unsupervised
- Design end-to-end modules that contain rich features.
- Design joint learning frameworks.
- Using geometric transformation to ground representations.
- Useful for planning (a few weeks from now).

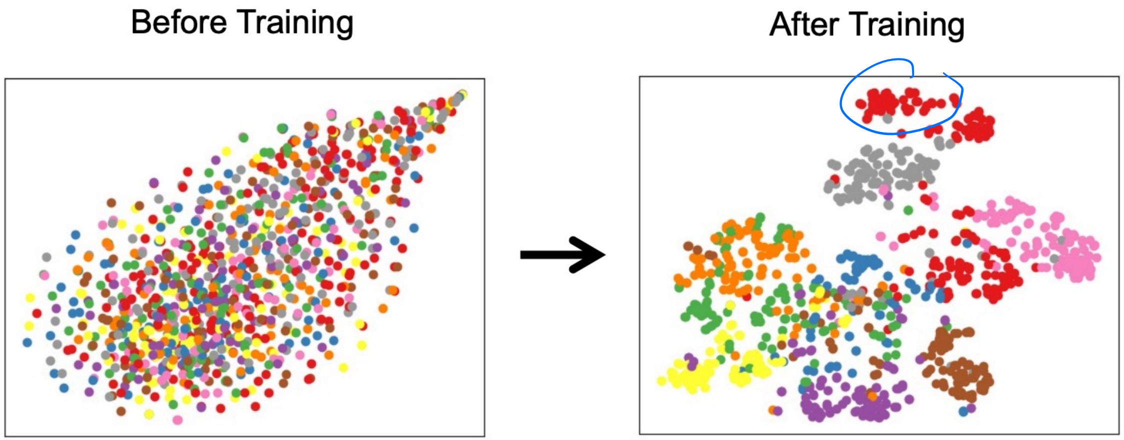
# Module 3:

Latent

# Self-Supervised Representation Learning and Object Discovery

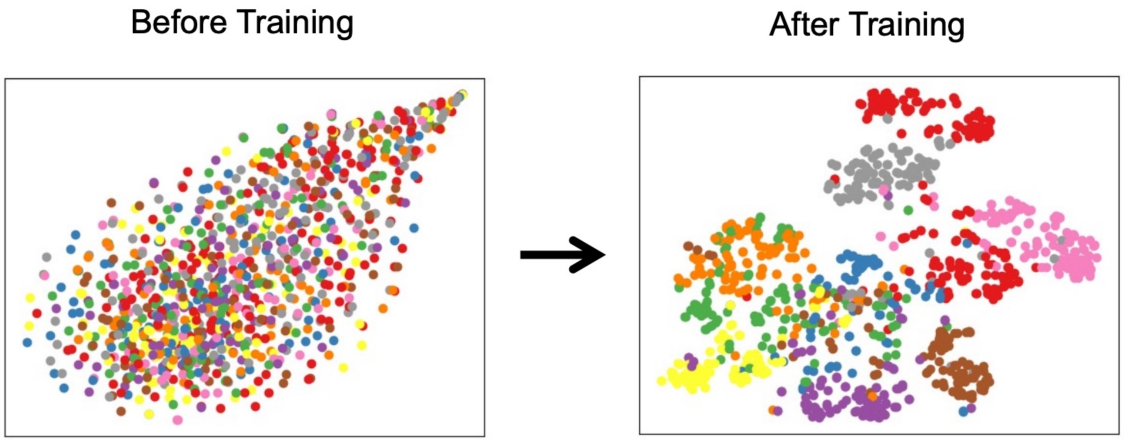
# Representation Learning

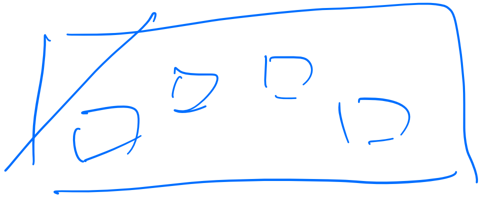
- Efficient encoding of the world that can help us recognize semantic concepts (high-level cognition).



# Representation Learning

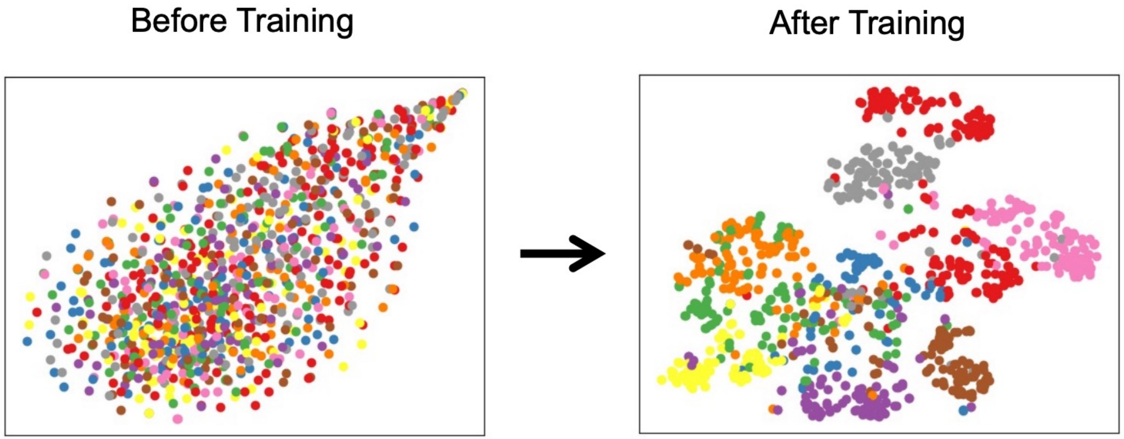
- Efficient encoding of the world that can help us recognize semantic concepts (high-level cognition).
- Efficient learning of visual data without extra supervision.





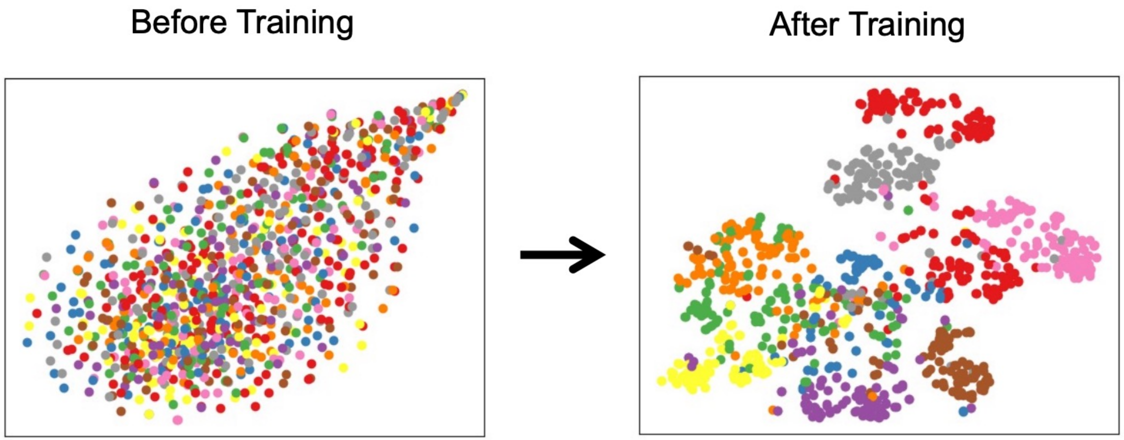
# Representation Learning

- Efficient encoding of the world that can help us recognize semantic concepts (high-level cognition).
- Efficient learning of visual data without extra supervision.
- Recognition of motion also requires global matching.



# Representation Learning

- Efficient encoding of the world that can help us recognize semantic concepts (high-level cognition).
- Efficient learning of visual data without extra supervision.
- Recognition of motion also requires global matching.
- Historically, largely driven by supervised classification.

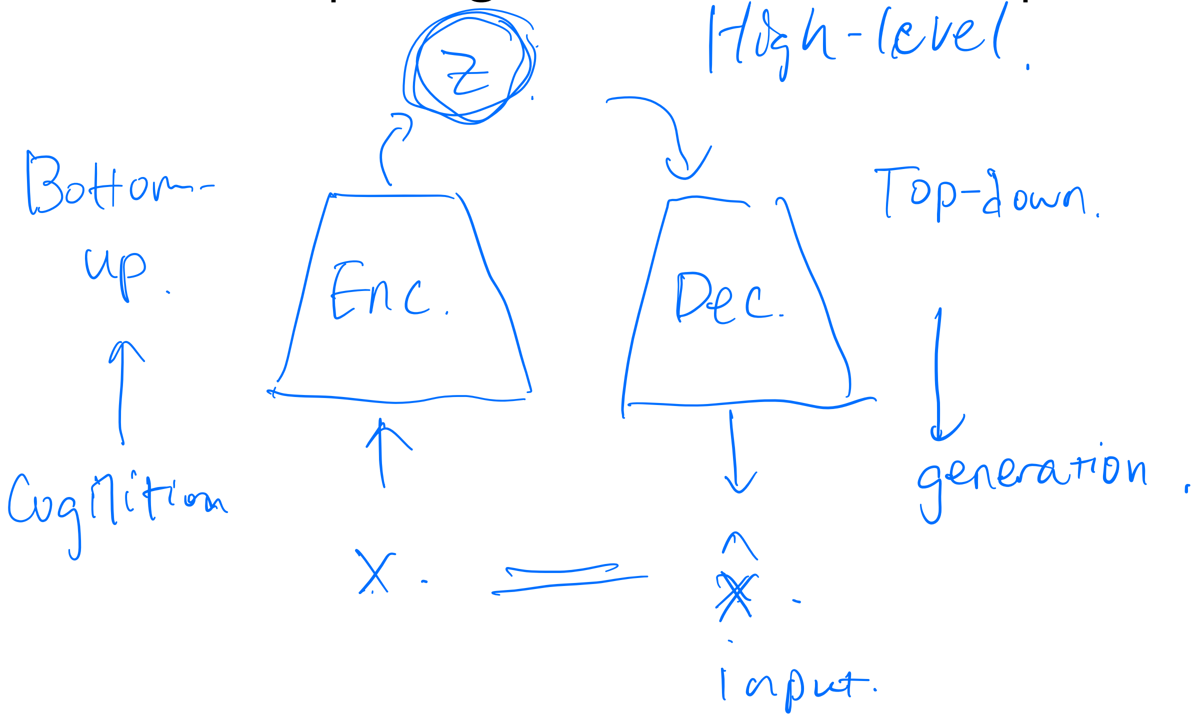


Before Training

After Training

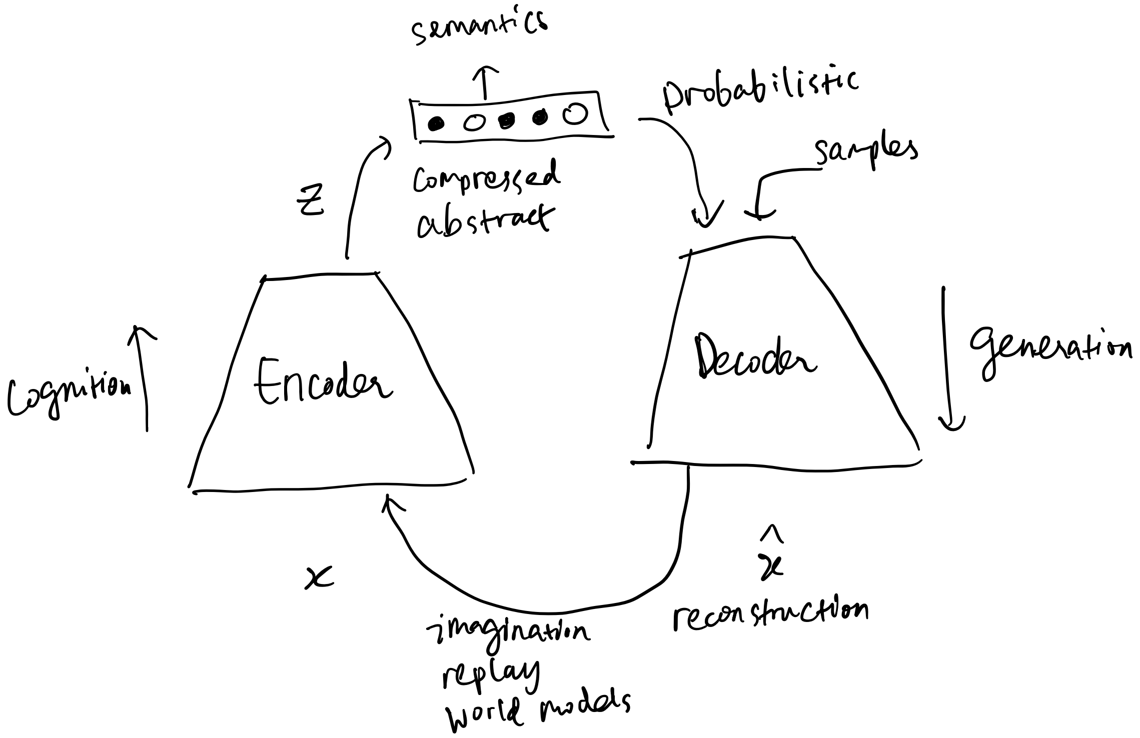
# Unsupervised Learning

- Encoder / bottom-up / cognition & decoder / top-down / generation



# Unsupervised Learning

- Encoder / bottom-up / cognition & decoder / top-down / generation



# Denoising Autoencoder (DAE)

- Making representations robust to partial corruption

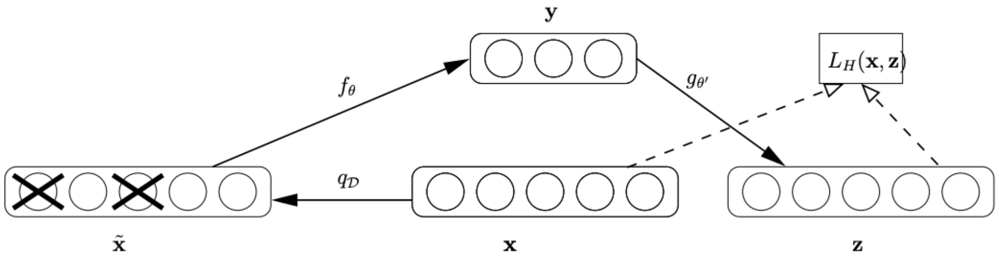
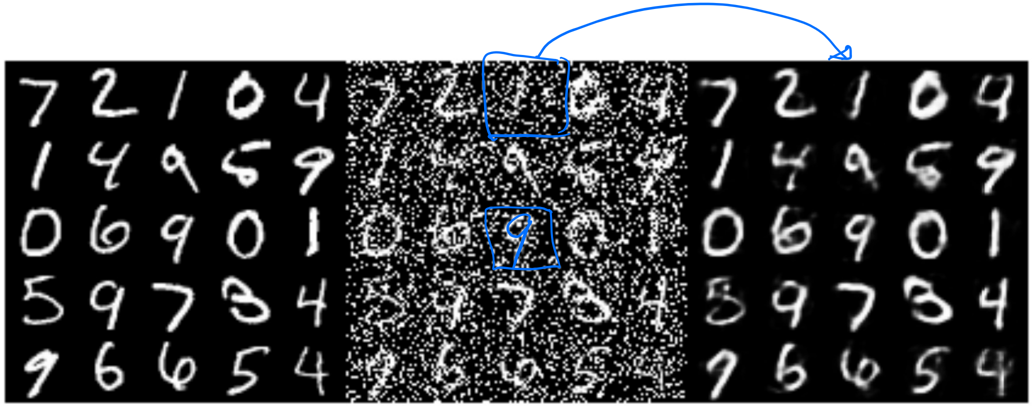


*Figure 1.* An example $\mathbf{x}$ is corrupted to $\tilde{\mathbf{x}}$. The autoencoder then maps it to $\mathbf{y}$ and attempts to reconstruct $\mathbf{x}$.



*Vincent et al. Extracting and Composing Robust Features with Denoising Autoencoders. ICML 2008.*

# Denoising Autoencoder (DAE)

- Making representations robust to partial corruption
- Low-dimensional manifold near which the data concentrate: $p(x|\tilde{x}) = B_{g_{\theta'}(f_\theta)}(x)$.

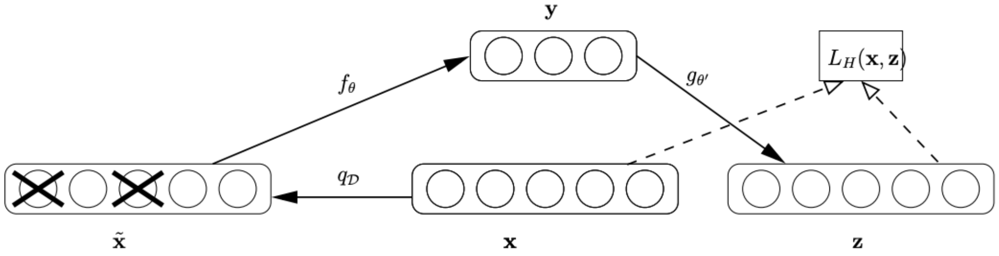
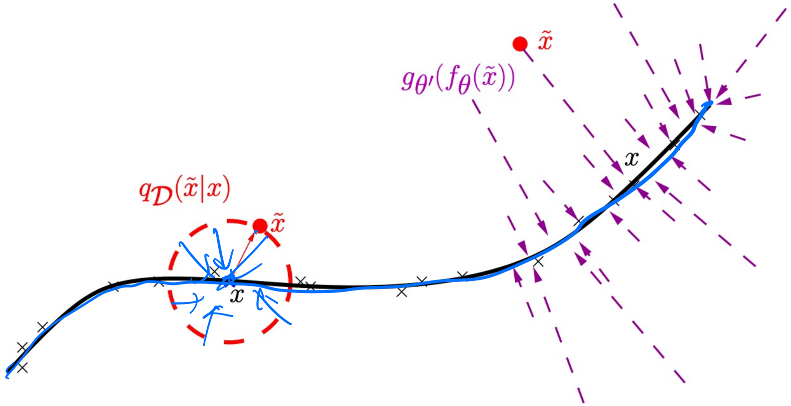
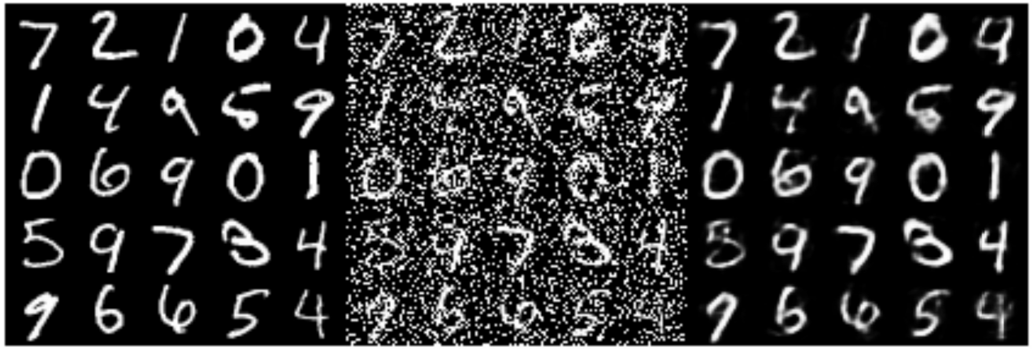


*Figure 1.* An example **x** is corrupted to $\tilde{\mathbf{x}}$. The autoencoder then maps it to **y** and attempts to reconstruct **x**.



$\tilde{x}$
$g_{\theta'}(f_\theta(\tilde{x}))$
$x$
$q_\mathcal{D}(\tilde{x}|x)$
$\tilde{x}$
$x$

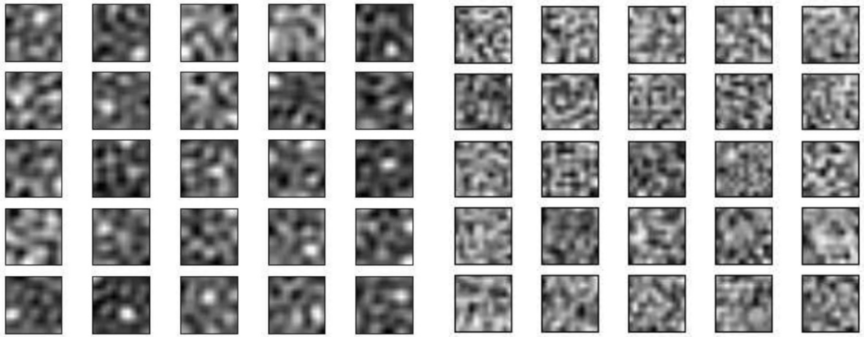


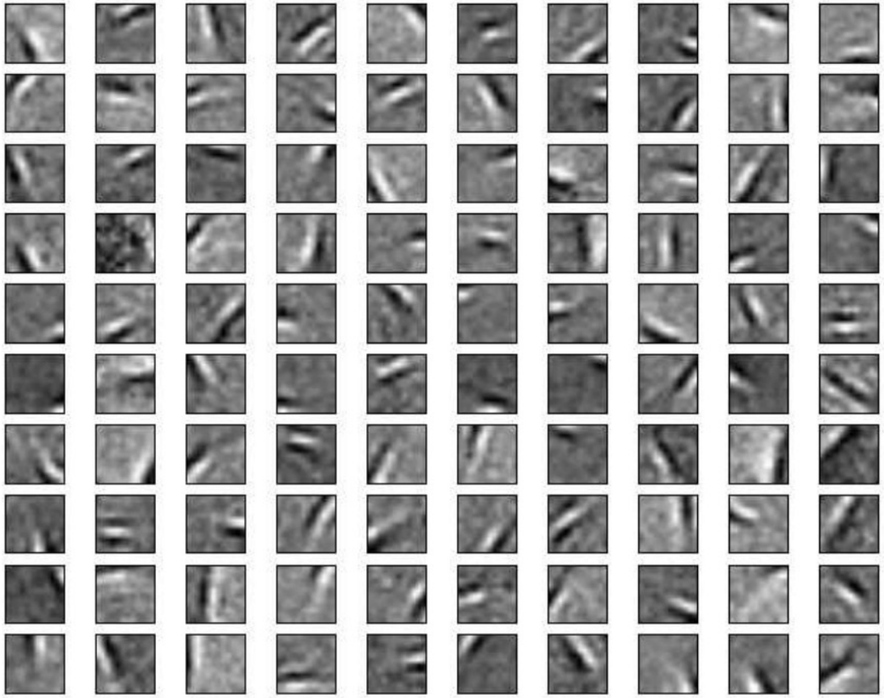
Vincent et al. Extracting and Composing Robust Features with Denoising Autoencoders. ICML 2008.

# Denoising Autoencoder (DAE)

- Regular autoencoders do not learn good filters.

preserve all information possible .





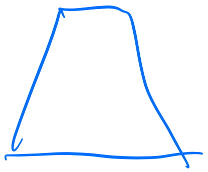
*Vincent et al. Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion. JMLR 2010.*

# Connection to Diffusion Models

- Both has denoising as learning objective.

# Connection to Diffusion Models

- Both has denoising as learning objective.
- Diffusion models – Fully generative; DAE – Locally generative, aim was to learn good representations.



# Connection to Diffusion Models

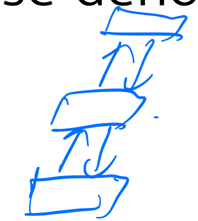
- Both has denoising as learning objective.
- Diffusion models – Fully generative; DAE – Locally generative, aim was to learn good representations.
- Not straightforward to extract good representations.

# Connection to Diffusion Models

- Both has denoising as learning objective.
- Diffusion models – Fully generative; DAE – Locally generative, aim was to learn good representations.
- Not straightforward to extract good representations.
- DAE: Simple architecture, aims to denoise in one go, not a good generative model.

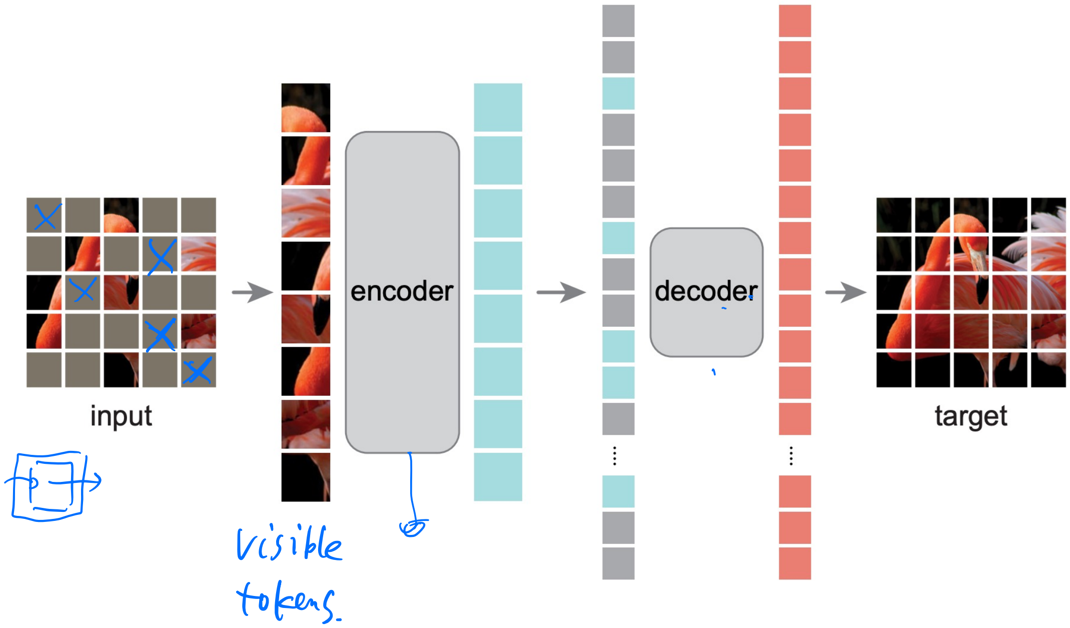
# Connection to Diffusion Models

- Both has denoising as learning objective.
- Diffusion models – Fully generative; DAE – Locally generative, aim was to learn good representations.
- Not straightforward to extract good representations.
- DAE: Simple architecture, aims to denoise in one go, not a good generative model.
- Stacked DAE: Stacked layerwise noise-denoise mechanism. Used to “pretrain” deep networks.



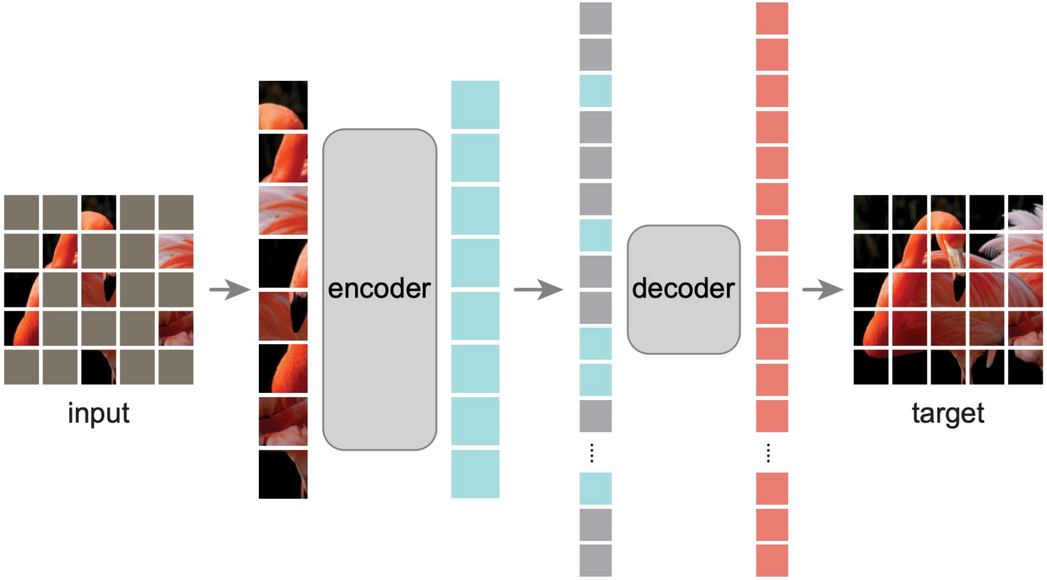
# Masked Autoencoder (MAE)

- Modernized version of denoising autoencoder.



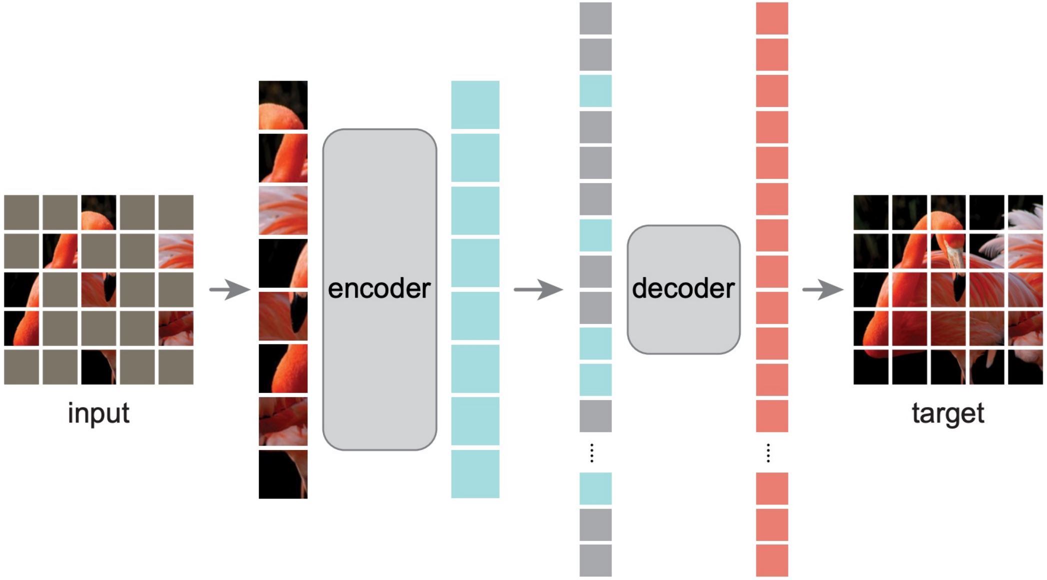
# Masked Autoencoder (MAE)

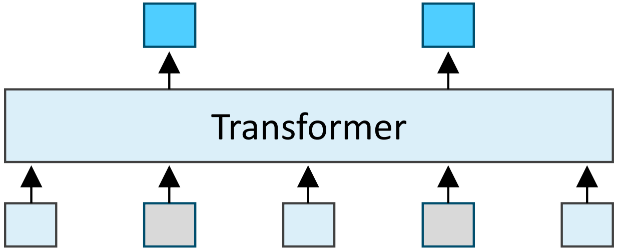
- Modernized version of denoising autoencoder.
- Mask noise: No artifacts
- ViT: No overlapping region, no empty space, no boundary.



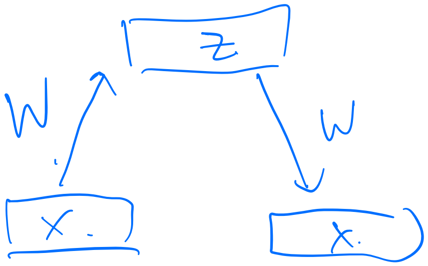
# Masked Autoencoder (MAE)

- Modernized version of denoising autoencoder.
- Mask noise: No artifacts
- ViT: No overlapping region, no empty space, no boundary.
- Idea also came from masked language models.





MLM

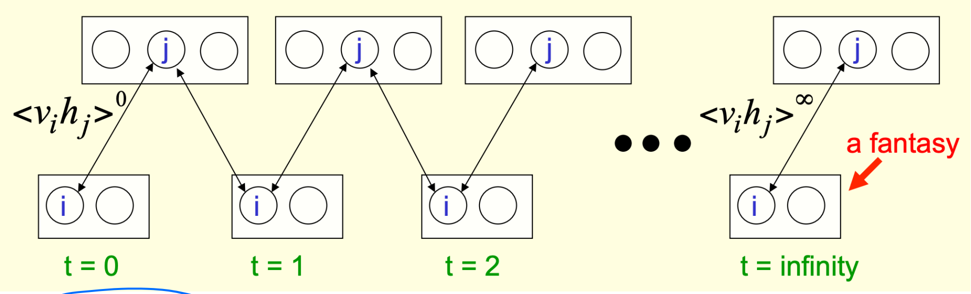


# Energy-Based Learning

xWz

- Example: RBMs
- Energy: $E(v,h) = -\sum_{i,j} v_i h_j w_{ij}.$

$$p(h_j = 1 | v_i) = \sigma(\textstyle\sum_{ij} v_i w_{ij}).$$



$$\frac{\partial \log p(v)}{\partial w_{ij}} = < v_i h_j >^0 - \boxed{< v_i h_j >^\infty}.$$

Hinton. Restricted Boltzmann Machines.

# General EBMs

- Inference requires running gradient descent and MCMC samples.

*Du & Mordatch. Implicit Generation and Modeling with Energy-Based Models. NeurIPS 2019.*

# General EBMs

- Inference requires running gradient descent and MCMC samples.

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k\breve{}1}) + \omega^k, \ \omega^k \sim \mathcal{N}(0, \lambda)$$

*Du & Mordatch. Implicit Generation and Modeling with Energy-Based Models. NeurIPS 2019.*

# General EBMs

- Inference requires running gradient descent and MCMC samples.

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k\breve{}1}) + \omega^k,\ \omega^k \sim \mathcal{N}(0, \lambda)$$

$$\nabla_\theta \mathcal{L}_{\mathrm{ML}} \approx \mathbb{E}_{\mathbf{x}^+ \sim p_D}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^- \sim q_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right].$$

*Du & Mordatch. Implicit Generation and Modeling with Energy-Based Models. NeurIPS 2019.*

# General EBMs

- Inference requires running gradient descent and MCMC samples.

$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k\,\breve{}\,1}) + \omega^k,\ \omega^k \sim \mathcal{N}(0,\lambda)$$

$$\nabla_\theta \mathcal{L}_{\mathrm{ML}} \approx \mathbb{E}_{\mathbf{x}^+ \sim p_D}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^- \sim q_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right].$$

- Can be applied on hand manipulation trajectory generation.

*Du & Mordatch. Implicit Generation and Modeling with Energy-Based Models. NeurIPS 2019.*

# General EBMs

- Inference requires running gradient descent and MCMC samples.

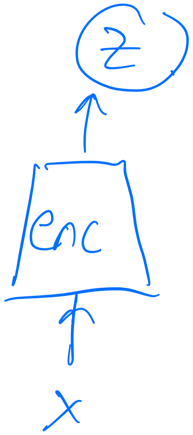
$$\tilde{\mathbf{x}}^k = \tilde{\mathbf{x}}^{k-1} - \frac{\lambda}{2}\nabla_{\mathbf{x}} E_\theta(\tilde{\mathbf{x}}^{k\breve{}1}) + \omega^k,\ \omega^k \sim \mathcal{N}(0, \lambda)$$

$$\nabla_\theta \mathcal{L}_{\mathrm{ML}} \approx \mathbb{E}_{\mathbf{x}^+ \sim p_D}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^- \sim q_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right].$$

- Can be applied on hand manipulation trajectory generation.
- Good results in generation but still not a generalized representation learning algorithm.

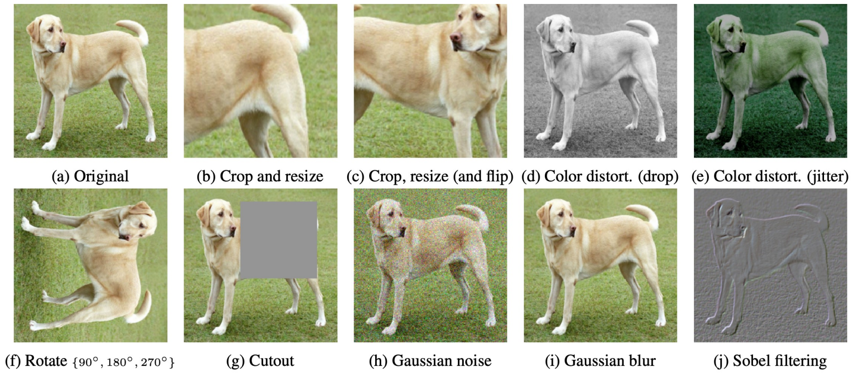
*Du & Mordatch. Implicit Generation and Modeling with Energy-Based Models. NeurIPS 2019.*

# Self-Supervised Visual Learning

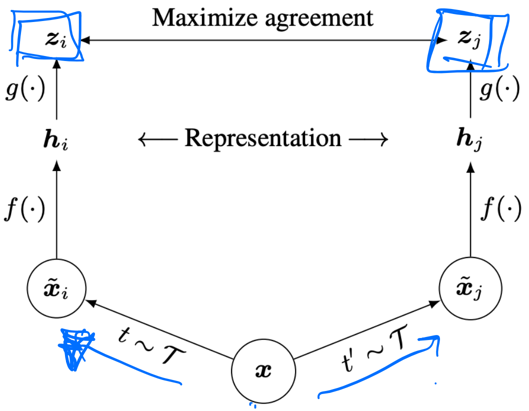


- Match the same image (with severe augmentation)

Augmentation

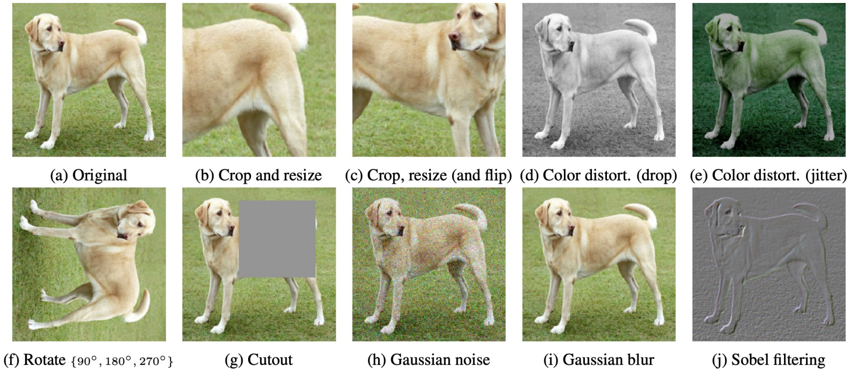

(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering



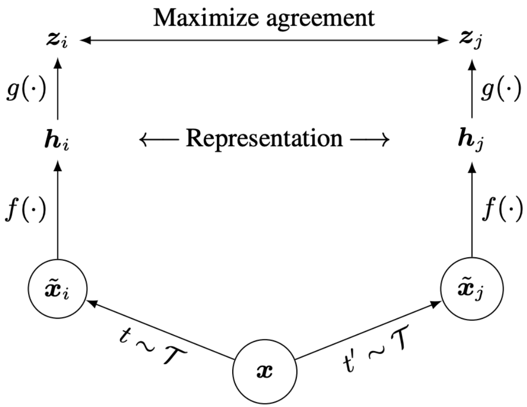
# Self-Supervised Visual Learning

- Match the same image (with severe augmentation)
- Joint embedding approach: Apply loss on the embedding level.



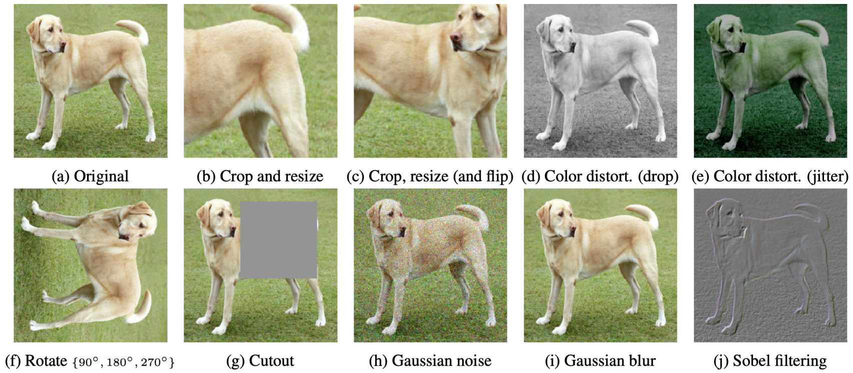
(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering



# Self-Supervised Visual Learning

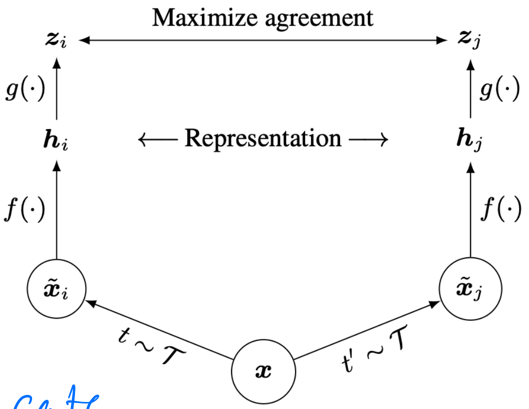
- Match the same image (with severe augmentation)
- Joint embedding approach: Apply loss on the embedding level.
- Use negative examples (contrastive) or not (non-contrastive).



(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

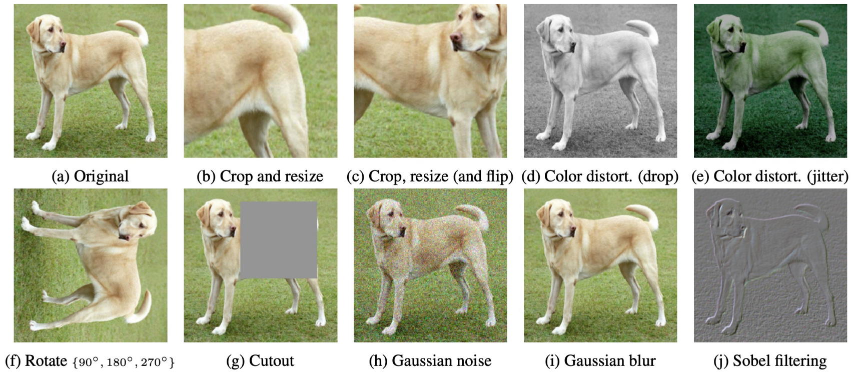
(f) Rotate {90°, 180°, 270°} (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

classify. $z_2$ based on $z_1$ among other negative. $z$'s → cats boats.

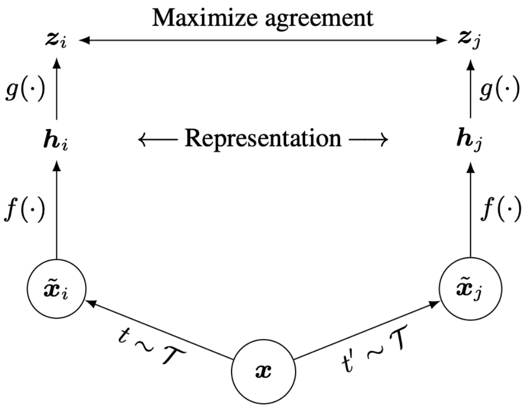


# Self-Supervised Visual Learning

- Match the same image (with severe augmentation)
- Joint embedding approach: Apply loss on the embedding level.
- Use negative examples (contrastive) or not (non-contrastive).
- Energy is defined between a pair of images.



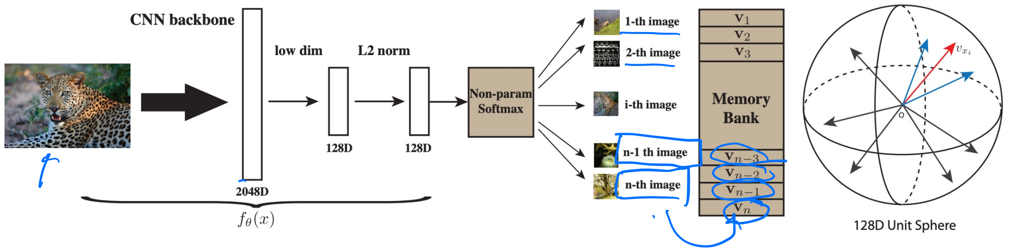
(a) Original
(b) Crop and resize
(c) Crop, resize (and flip)
(d) Color distort. (drop)
(e) Color distort. (jitter)
(f) Rotate {90°, 180°, 270°}
(g) Cutout
(h) Gaussian noise
(i) Gaussian blur
(j) Sobel filtering



# Several Embedding Loss Formulations

Wu et al., 2018

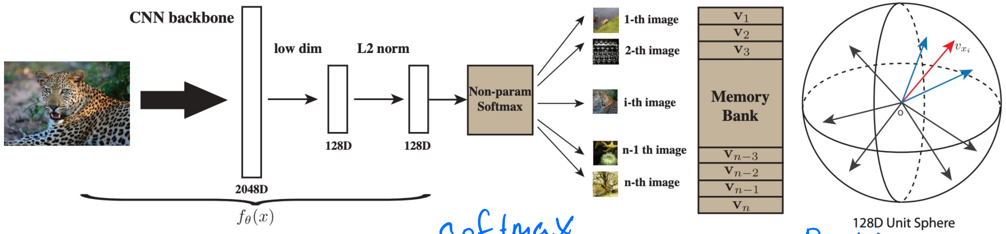
- Instance Classification:



# Several Embedding Loss Formulations

Wu et al., 2018

- Instance Classification:



softmax.

positives.

- Contrastive Learning: Cross entropy on pairs

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

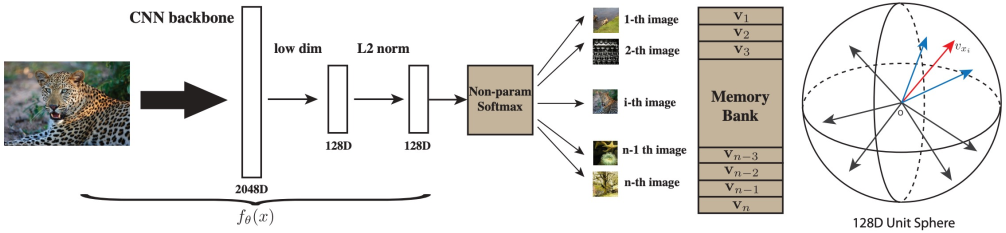
Chen et al. 2020

Cross-entroy.

# Several Embedding Loss Formulations

- Instance Classification:

Wu et al., 2018



- Contrastive Learning: Cross entropy on pairs

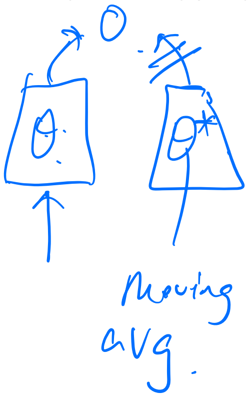
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Chen et al. 2020

- Non-contrastive Learning (Positive Only)
  - Moving Average [Grill et al., 2020]
  - Stop Gradient [Chen & He, 2020]

$\|Z_1 - Z_2\|_2^2$ ..

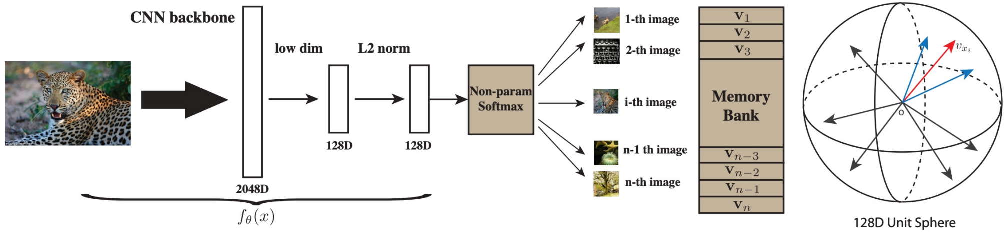
↳ collapse.



# Several Embedding Loss Formulations

Wu et al., 2018

- Instance Classification:



- Contrastive Learning: Cross entropy on pairs

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

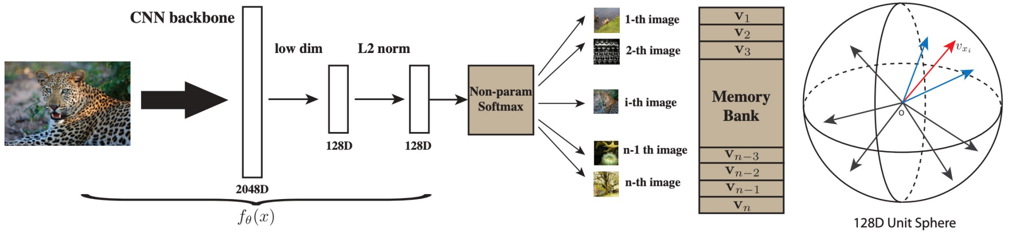
Chen et al. 2020

- Non-contrastive Learning (Positive Only)
  - Moving Average [Grill et al., 2020]
  - Stop Gradient [Chen & He, 2020]
- Use of projectors and predictors

# Several Embedding Loss Formulations

- Instance Classification:

Wu et al., 2018

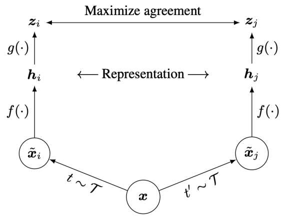


- Contrastive Learning: Cross entropy on pairs

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Chen et al. 2020

- Non-contrastive Learning (Positive Only)
  - Moving Average [Grill et al., 2020]
  - Stop Gradient [Chen & He, 2020]
- Use of projectors and predictors

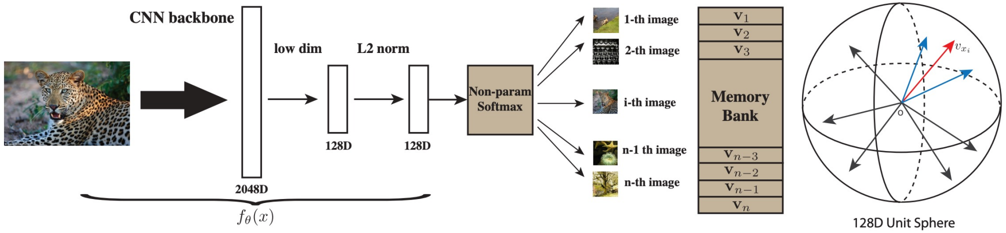


Chen et al. 2020

# Several Embedding Loss Formulations

Wu et al., 2018

- Instance Classification:

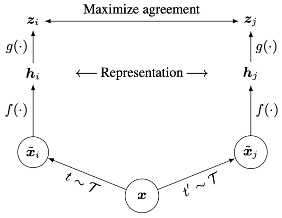


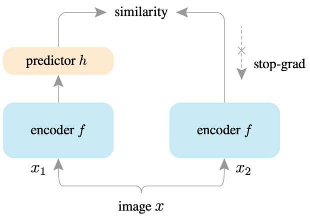
- Contrastive Learning: Cross entropy on pairs

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Chen et al. 2020

- Non-contrastive Learning (Positive Only)
  - Moving Average [Grill et al., 2020]
  - Stop Gradient [Chen & He, 2020]
- Use of projectors and predictors



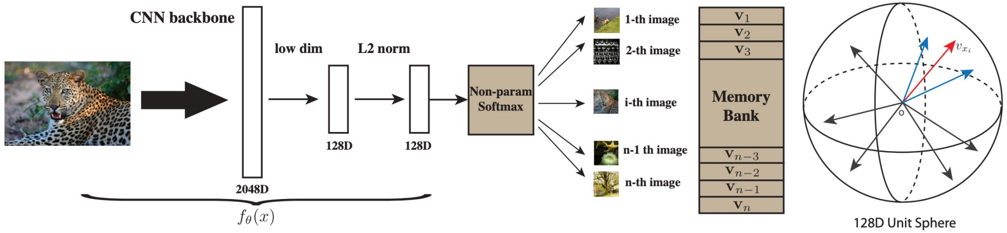


Chen et al. 2020 Chen & He, 2021

# Several Embedding Loss Formulations

- Instance Classification:

Wu et al., 2018

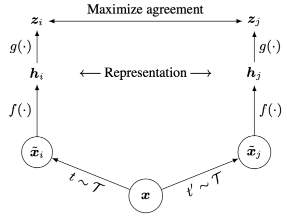


- Contrastive Learning: Cross entropy on pairs

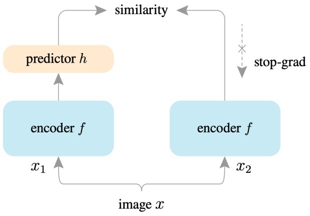
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Chen et al. 2020

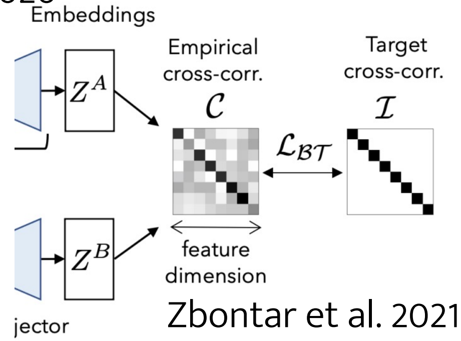
- Non-contrastive Learning (Positive Only)
  - Moving Average [Grill et al., 2020]
  - Stop Gradient [Chen & He, 2020]
- Use of projectors and predictors
- Use of co-variance regularization



Chen et al. 2020



Chen & He, 2021

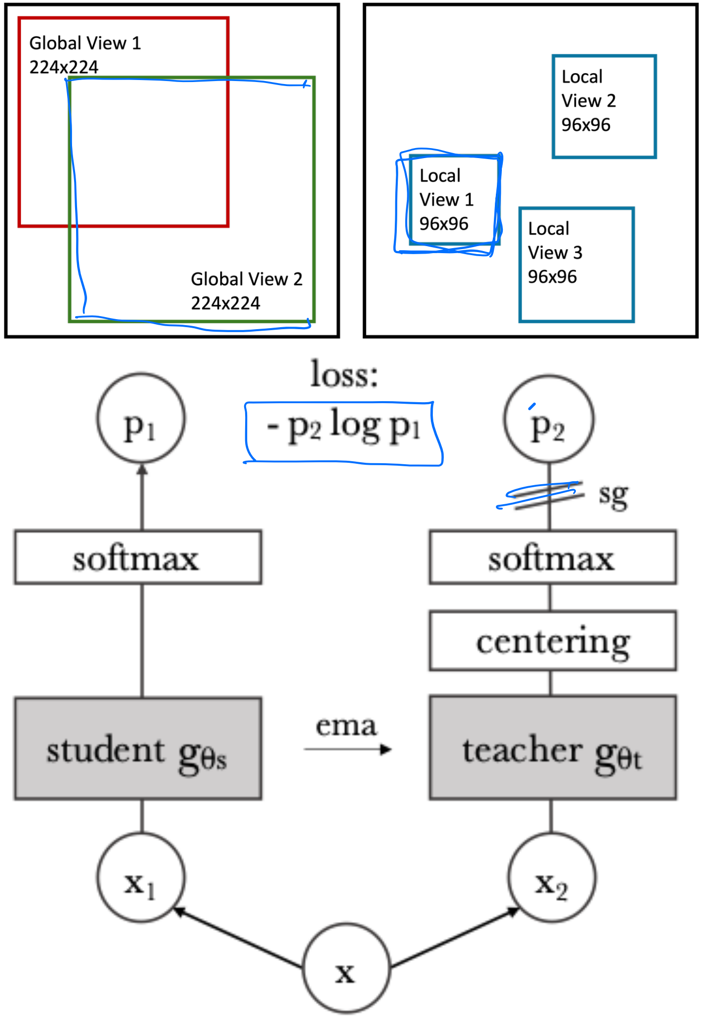


Zbontar et al. 2021
Bardes et al. 2021

Coviance feature dimensions.

# DINO

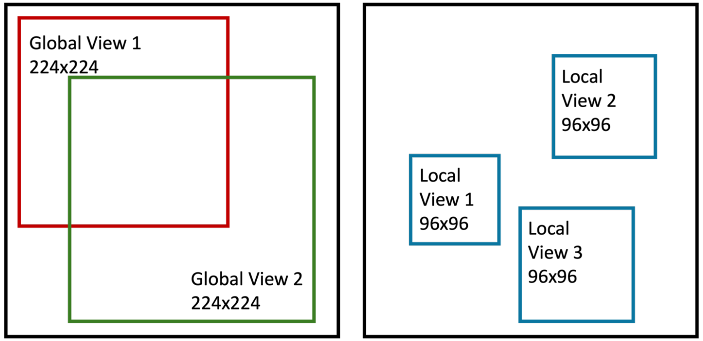
- Knowledge distillation between a student and a teacher network.

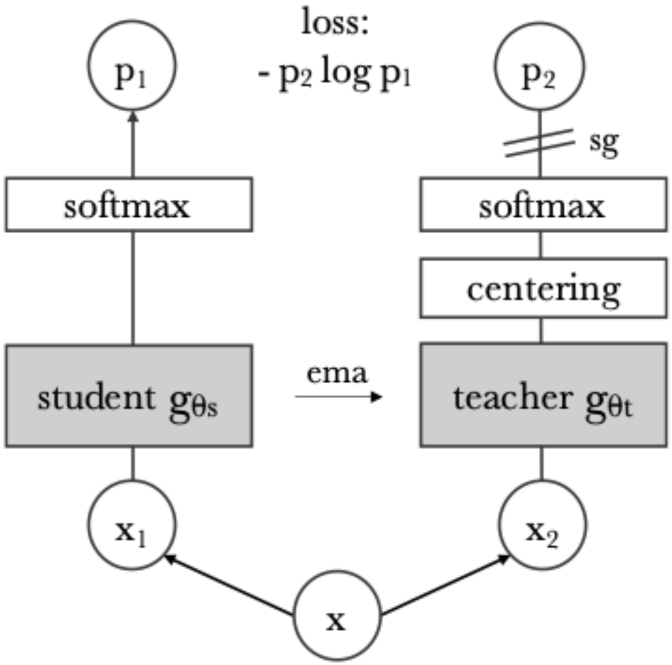


*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

# DINO

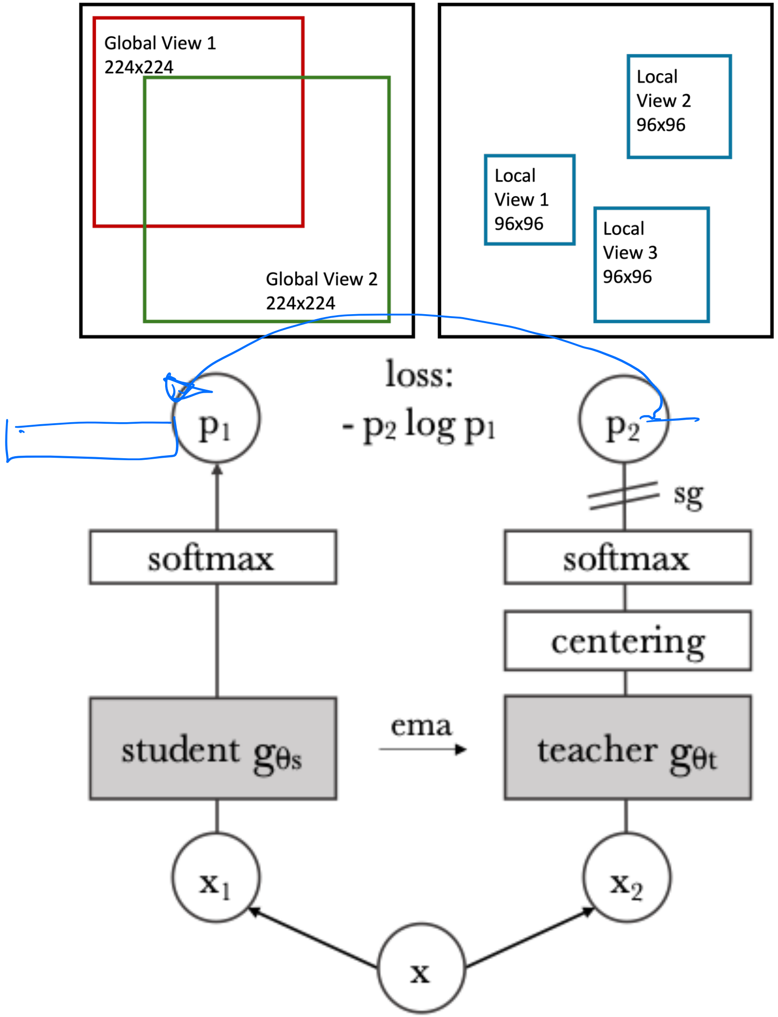
- Knowledge distillation between a student and a teacher network.
- Student: $p_s(x) = \frac{\exp(g_{\theta_s}(x)_i / \tau_s)}{\sum_k \exp(g_{\theta_s}(x)_k / \tau_s)}.$





*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

# DINO

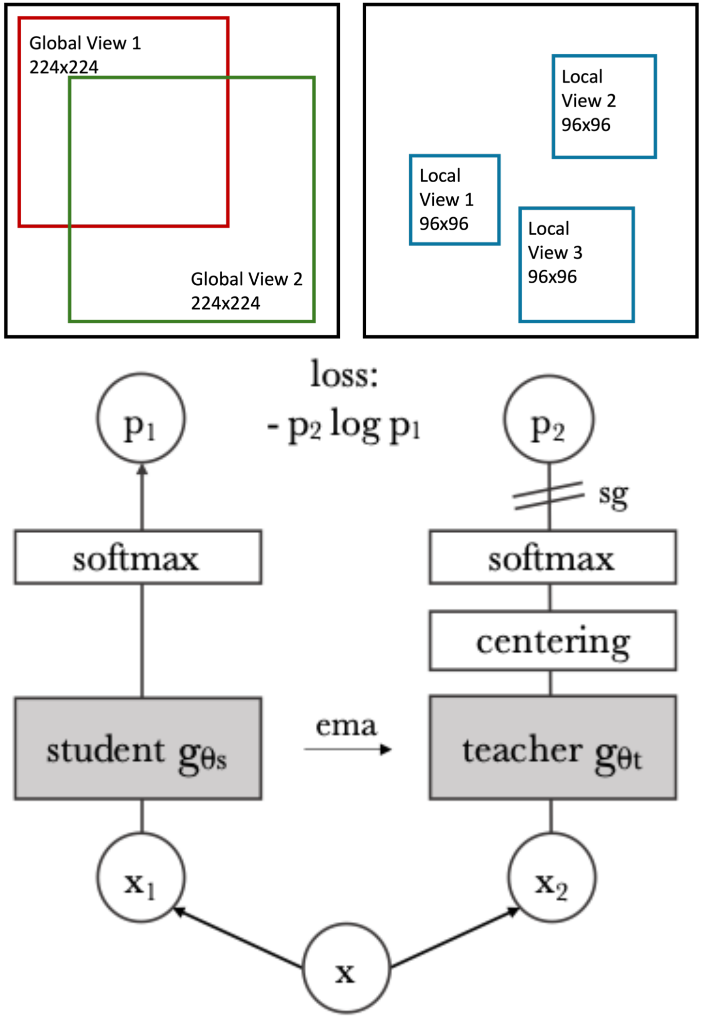


- Knowledge distillation between a student and a teacher network.
- Student: $p_s(x) = \frac{\exp(g_{\theta_s}(x)_i / \tau_s)}{\sum_k \exp(g_{\theta_s}(x)_k / \tau_s)}.$
- Minimize CE: $\min_\theta H(p_t(x), p_s(x)).$

*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

# DINO

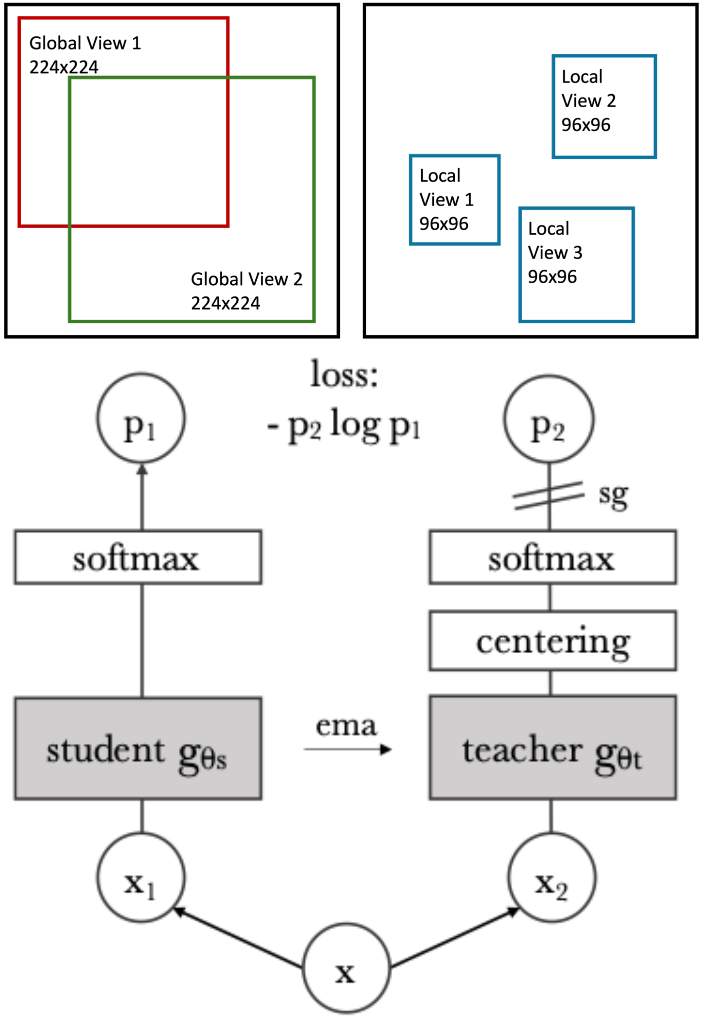
- Knowledge distillation between a student and a teacher network.
- Student: $p_s(x) = \frac{\exp(g_{\theta_s}(x)_i/\tau_s)}{\sum_k \exp(g_{\theta_s}(x)_k/\tau_s)}.$
- Minimize CE: $\min_\theta H(p_t(x), p_s(x)).$
- Stop gradient on the teacher (no true label).



*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

# DINO

- Knowledge distillation between a student and a teacher network.
- Student: $p_s(x) = \frac{\exp(g_{\theta_s}(x)_i / \tau_s)}{\sum_k \exp\left(g_{\theta_s}(x)_k / \tau_s\right)}.$
- Minimize CE: $\min_\theta H(p_t(x), p_s(x)).$
- Stop gradient on the teacher (no true label).
- Teacher network has EMA weights copied from student (prevent collapse).



*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

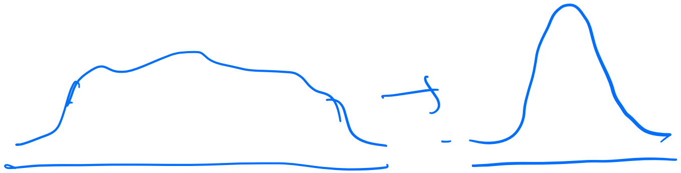
# Preventing Collapse

- Cross entropy objective can make both sides collapse to uniform distribution.
  - Apply sharpening, apply a temperature term on both teacher and student.
  - $\mathrm{softmax}(g/\tau)$ The higher the temperature, the more uniform.

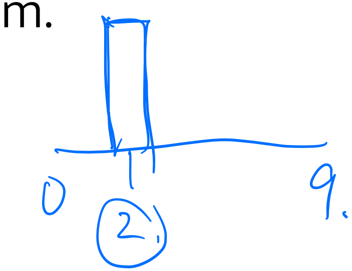
*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

NYU

# Preventing Collapse

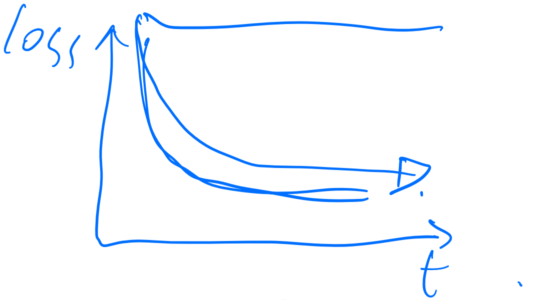


- Cross entropy objective can make both sides collapse to uniform distribution.
  - Apply sharpening, apply a temperature term on both teacher and student.
  - $\mathrm{softmax}(g/\tau)$ The higher the temperature, the more uniform.
- It can also collapse into always activating a single unit.
  - Mean statistics: $c_t = mc_{t-1} + (1-m)\frac{1}{B}\sum_{i=1}^{B} g_{\theta_t}(x_i)$
  - Center teacher prediction: $p_t(x) = \frac{\exp((g_{\theta_t}(x)_i - c_t)/\tau_t)}{\sum_k \exp((g_{\theta_t}(x)_k - c_t)/\tau_t)}$.

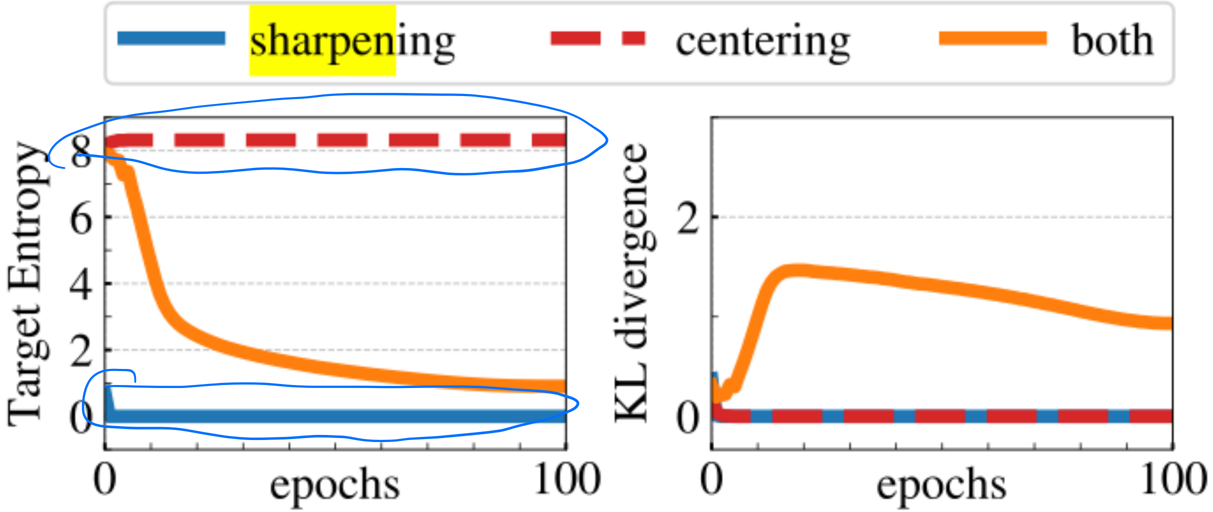


*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

# Centering and Sharpening

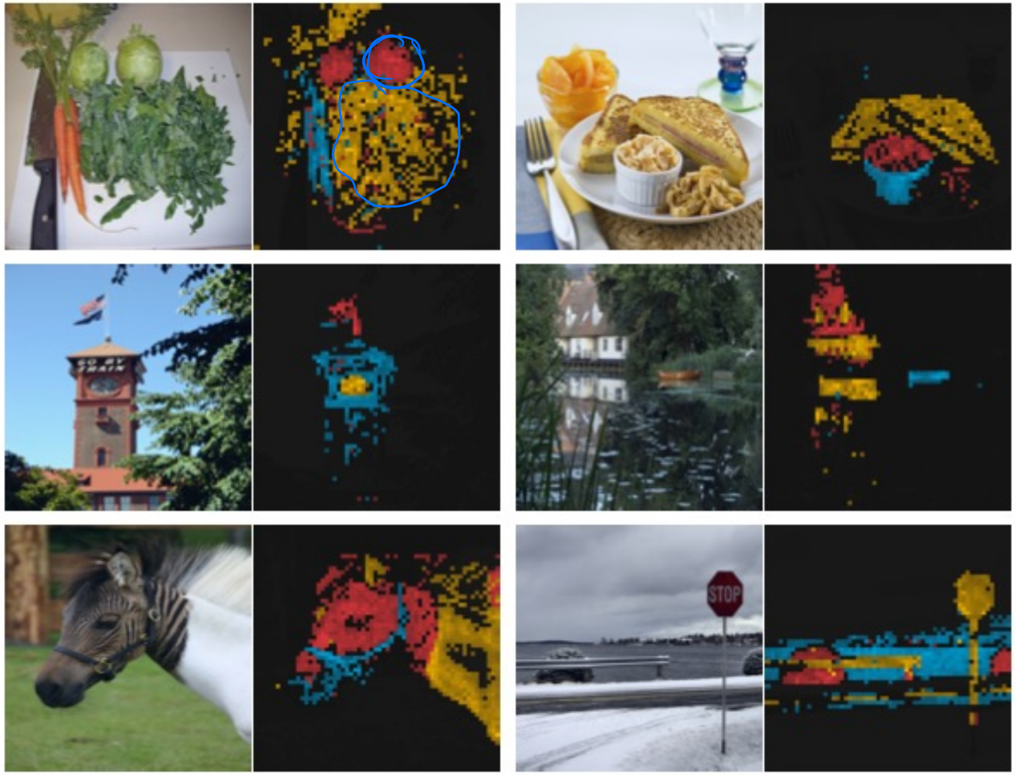


- Only centering: Always uniform distribution, high entropy, easy to guess.
- Only sharpening: Collapsed into one unit, easy to guess, low loss, but no real learning.



# Visualizing Attention

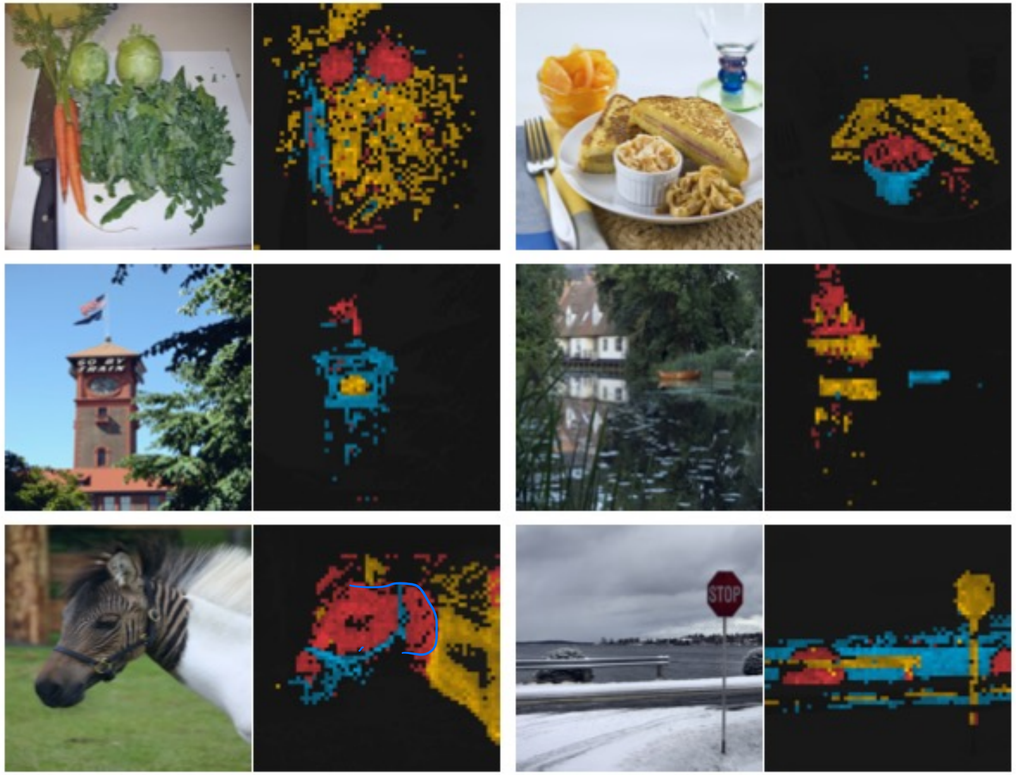
- The [CLS] token is an extra token added to summarize the whole image into a vector.



Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.

# Visualizing Attention

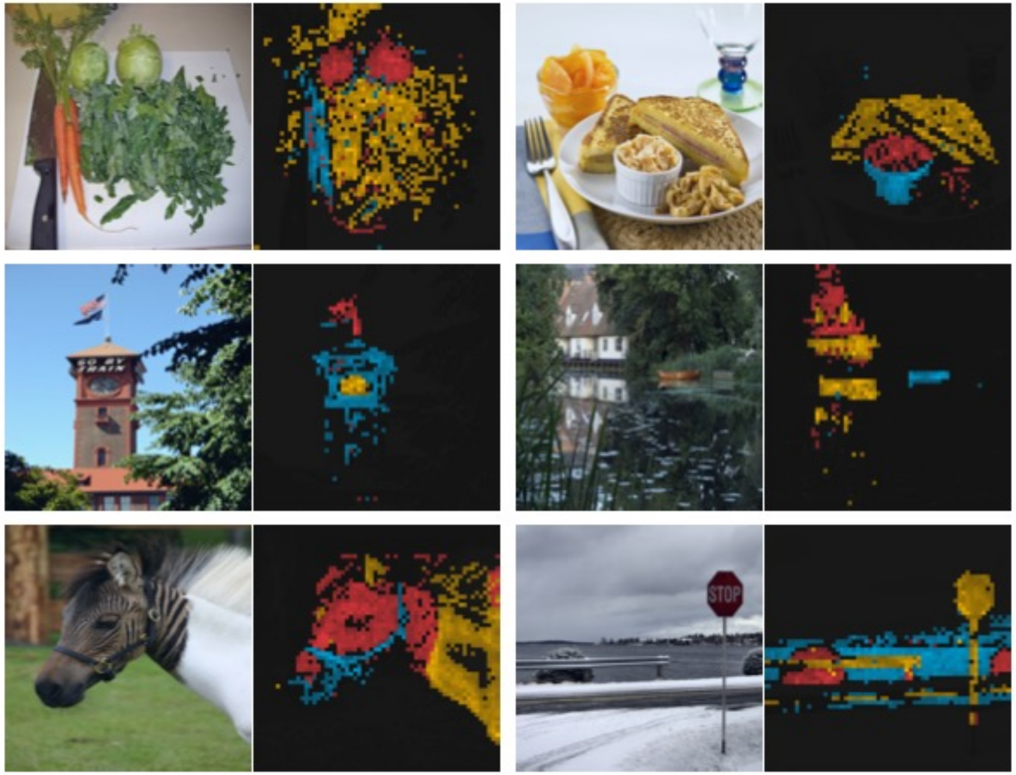
- The [CLS] token is an extra token added to summarize the whole image into a vector.
- Visualize the attention map of different attention heads using different colors.



*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

# Visualizing Attention

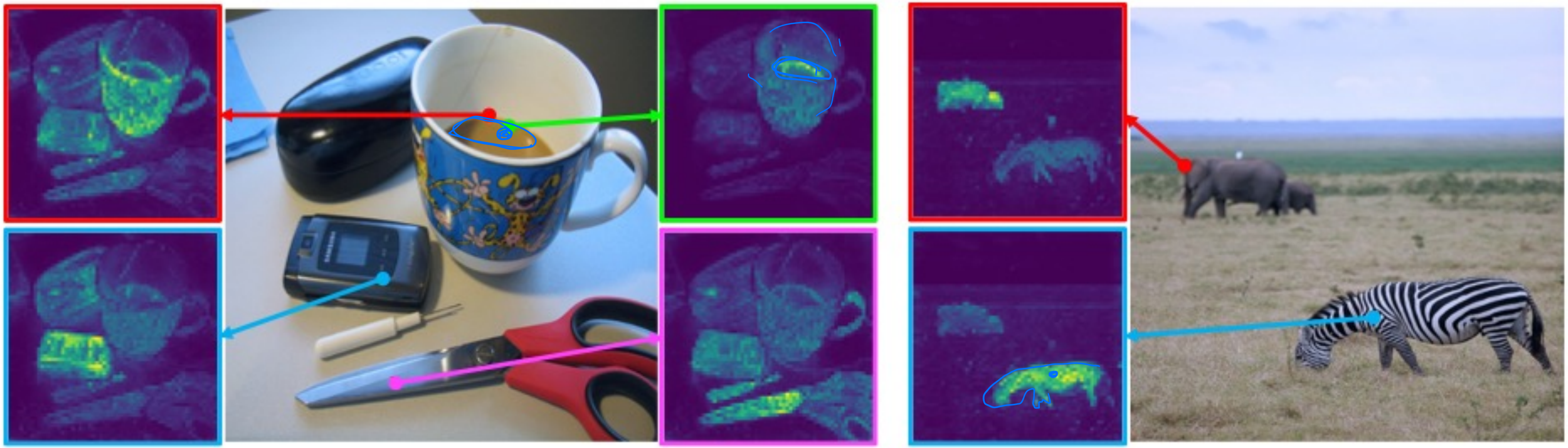
- The [CLS] token is an extra token added to summarize the whole image into a vector.
- Visualize the attention map of different attention heads using different colors.
- Showing understanding of different objects and parts.



*Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.*

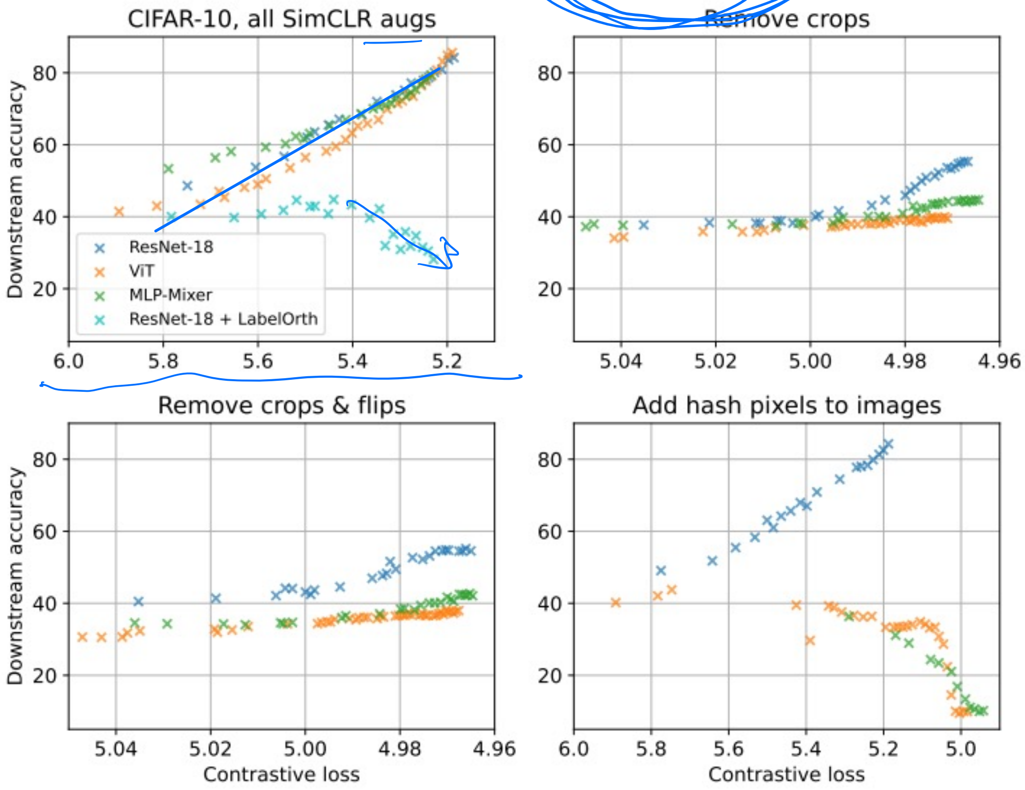
# Visualizing Attention

- We can also visualize the attention by querying from a location.
- Weak separation of objects.



# Why Does SSL Work?

- The unsupervised loss is a surrogate. If an image belongs to a similarity class, it also belongs to the same semantic class.
- The choice of similarity class matters.

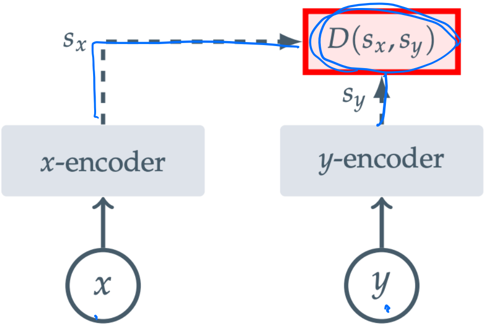


Aurora et al. A Theoretical Analysis of Contrastive Unsupervised Representation Learning.
Saunshi et al. Understanding Contrastive Learning Requires Incorporating Inductive Biases. 2022.

NYU

# Joint Embedding Predictive Architecture (JEPA)

- Predict what changes from one view to another.

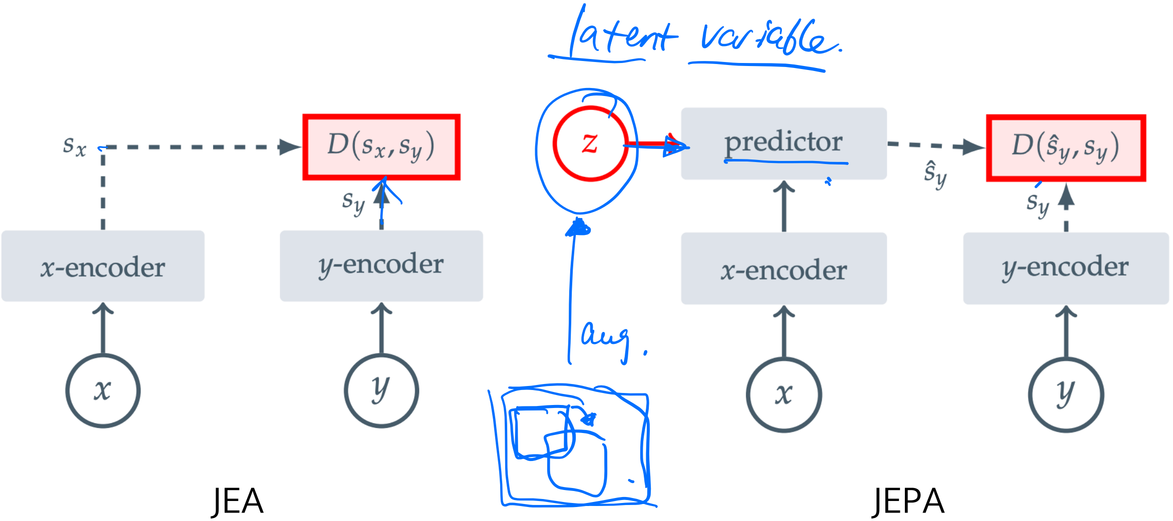


JEA

Assran et al. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. ICCV 2023.

# Joint Embedding Predictive Architecture (JEPA)

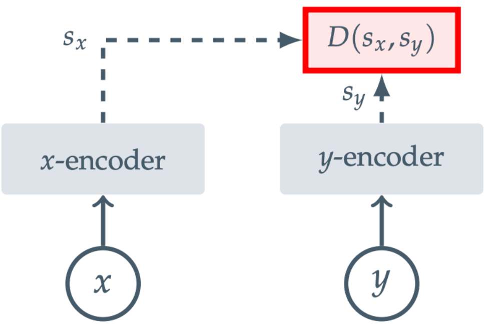
- Predict what changes from one view to another.



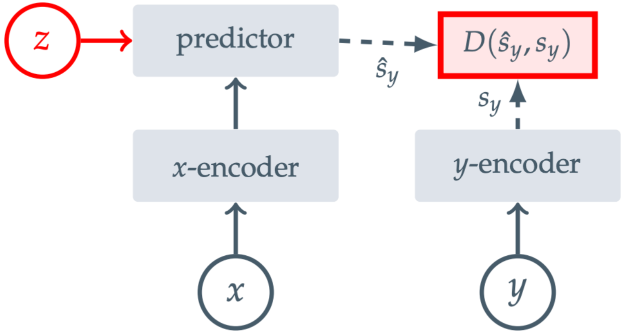
*Assran et al. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. ICCV 2023.*

# Joint Embedding Predictive Architecture (JEPA)

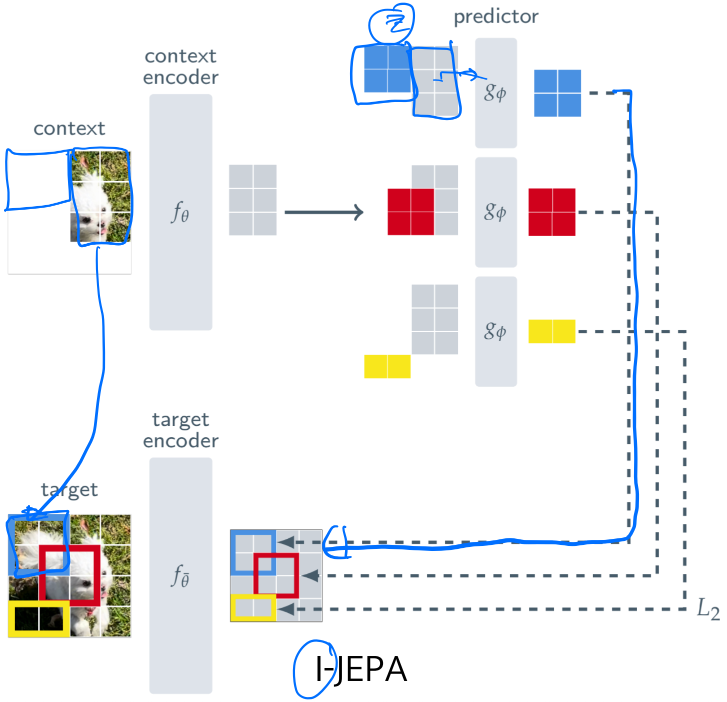
- Predict what changes from one view to another.



JEA



JEPA

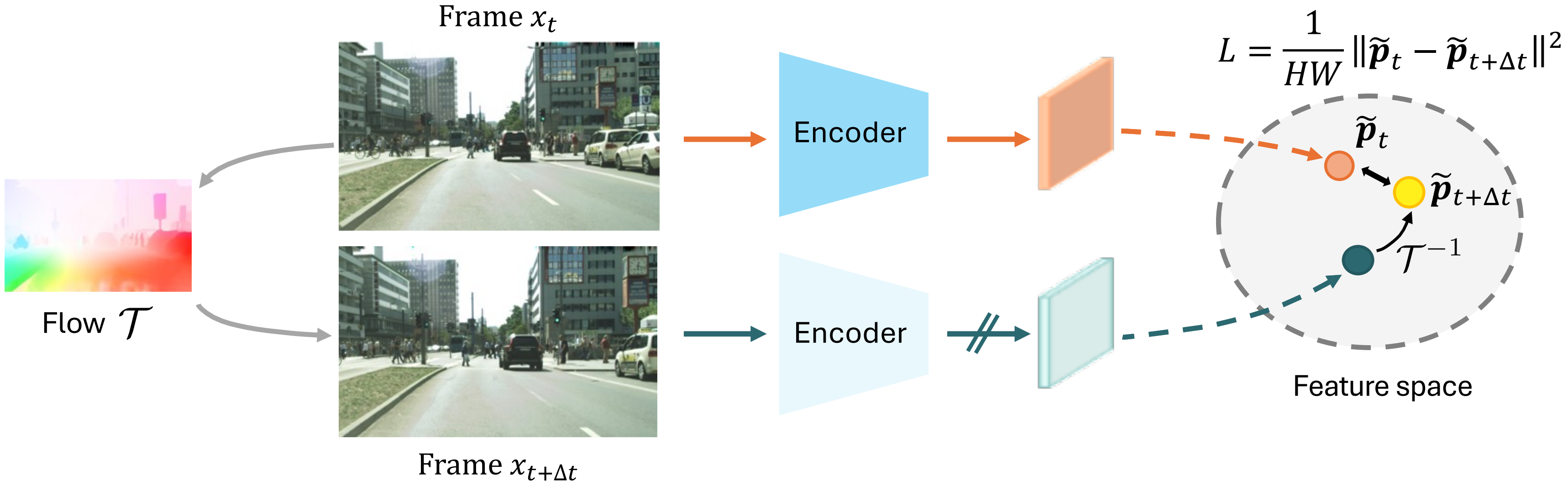


I-JEPA

*Assran et al. Self-Supervised Learning from Images with a Joint-Embedding Predictive Architecture. ICCV 2023.*

# SSL with Dense Motion

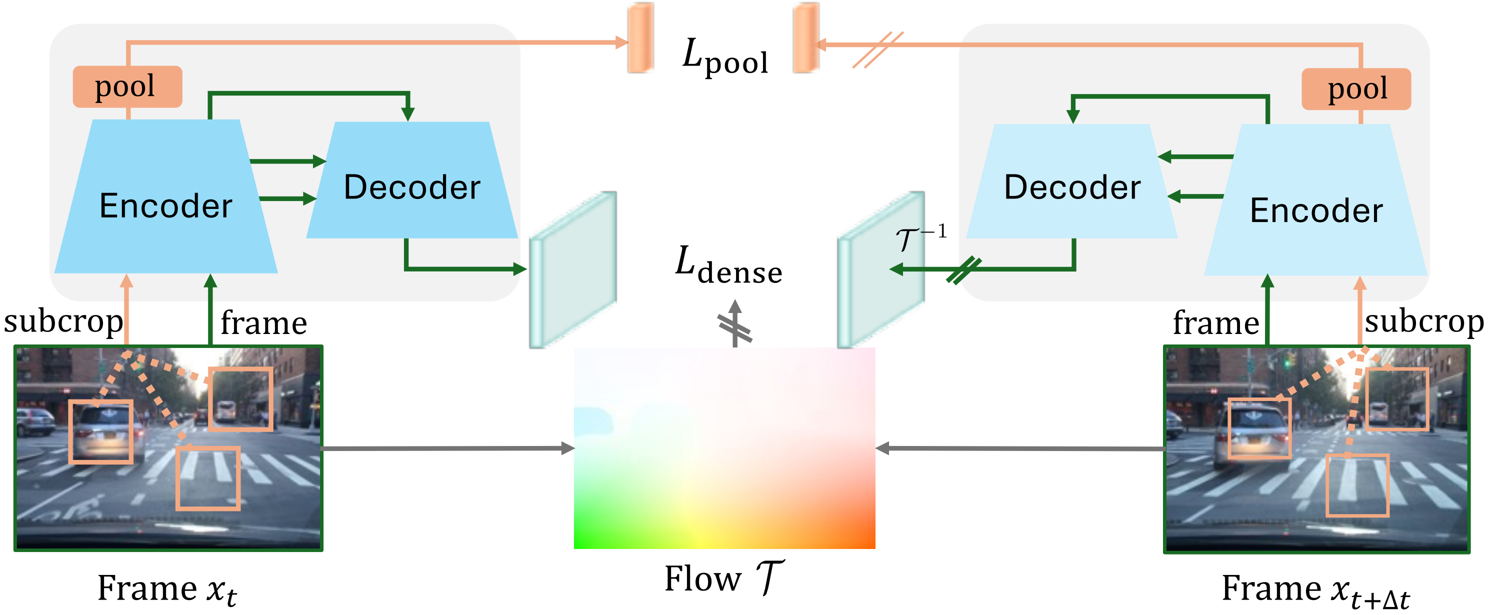
- Can we use adjacent video frames as self-supervision?
- Objects move densely throughout the image.



Frame $x_t$
Flow $\mathcal{T}$
Encoder
Encoder
Frame $x_{t+\Delta t}$
$L = \frac{1}{HW} \|\tilde{\boldsymbol{p}}_t - \tilde{\boldsymbol{p}}_{t+\Delta t}\|^2$
$\tilde{\boldsymbol{p}}_t$
$\tilde{\boldsymbol{p}}_{t+\Delta t}$
$\mathcal{T}^{-1}$
Feature space

# SSL with Dense Motion

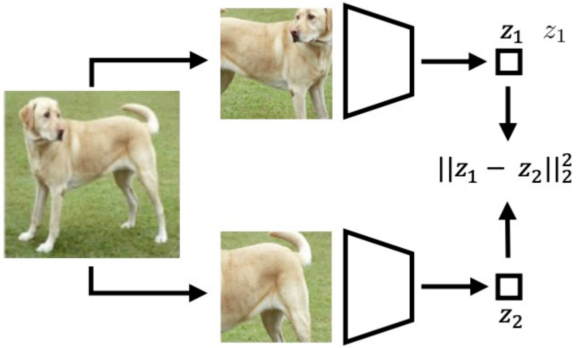
- Perform SSL in multiple scales (small objects vs. big regions).



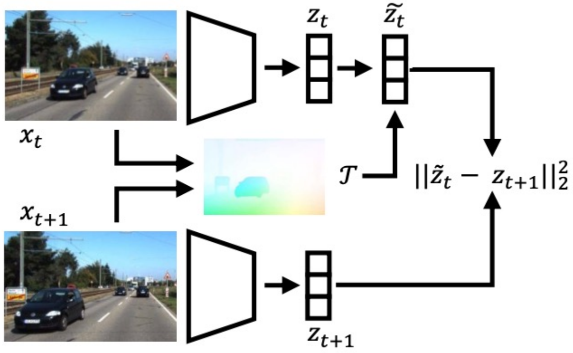
Wang et al. PooDLe: Pooled and Dense Self-Supervised Learning from Naturalistic Videos. ICLR 2025.

# Joint Motion and Semantic Learning

- We either create or assume some external motion / action information to send to predictors.
- Additional motion needs to be inferred.



JEA

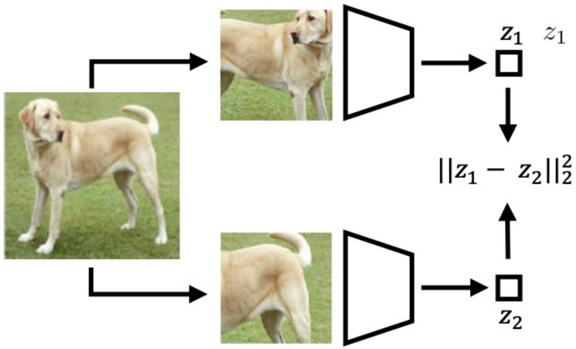


JEPA w/ dense motion

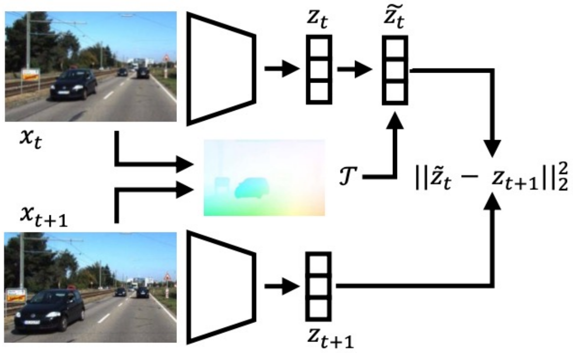
*Hoang et al. Midway Network: Learning Representations for Recognition and Motion from Latent Dynamics.*

# Joint Motion and Semantic Learning

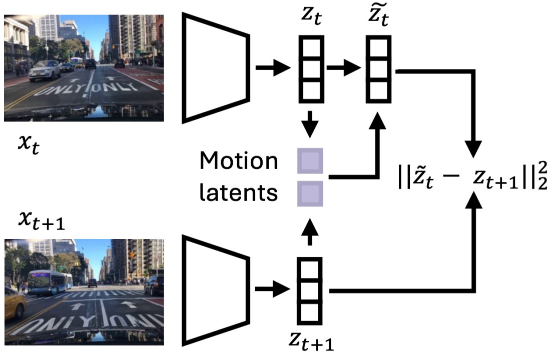
- We either create or assume some external motion / action information to send to predictors.
- Additional motion needs to be inferred.



JEA



JEPA w/ dense motion

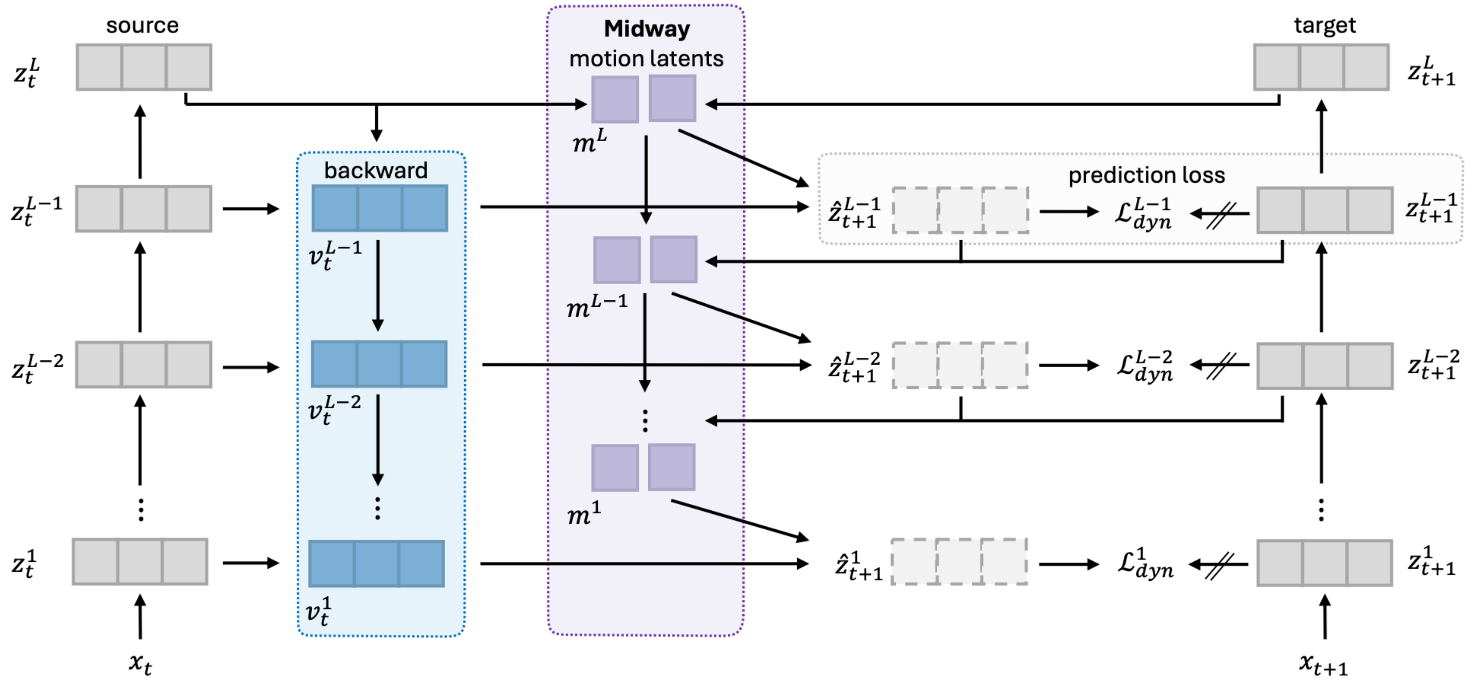


JEPA w/ inferred motion

*Hoang et al. Midway Network: Learning Representations for Recognition and Motion from Latent Dynamics.*

# Midway Networks

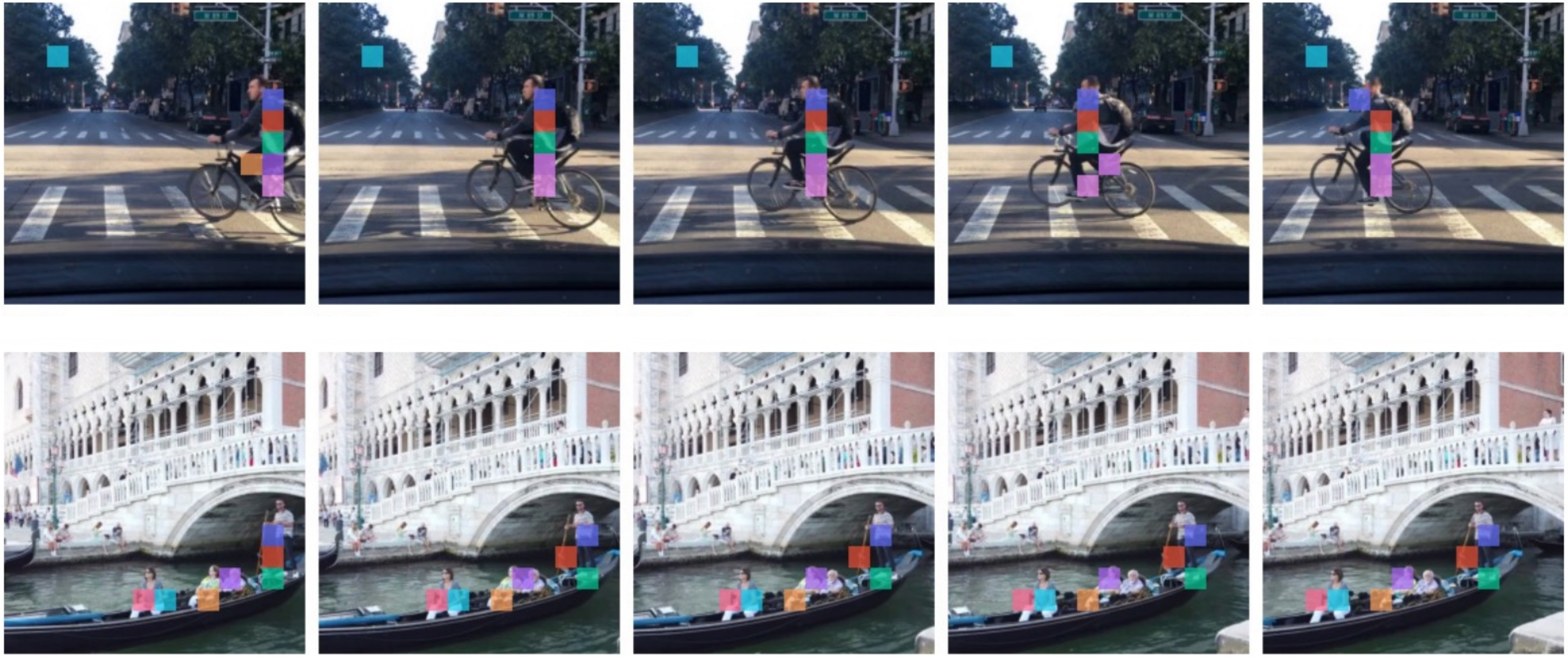
- Using backward refinement from Optical Flow networks.



*Hoang et al. Midway Network: Learning Representations for Recognition and Motion from Latent Dynamics. ICLR 2026.*

# Midway Networks

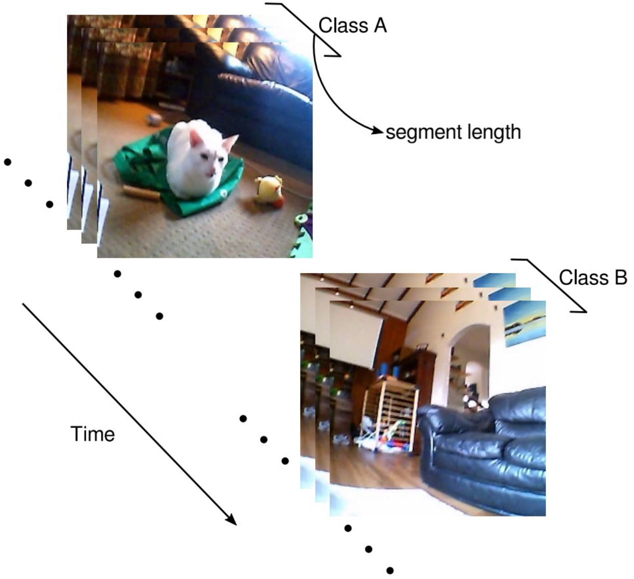
- Use perturbation to visualize the motion learned from motion tokens.

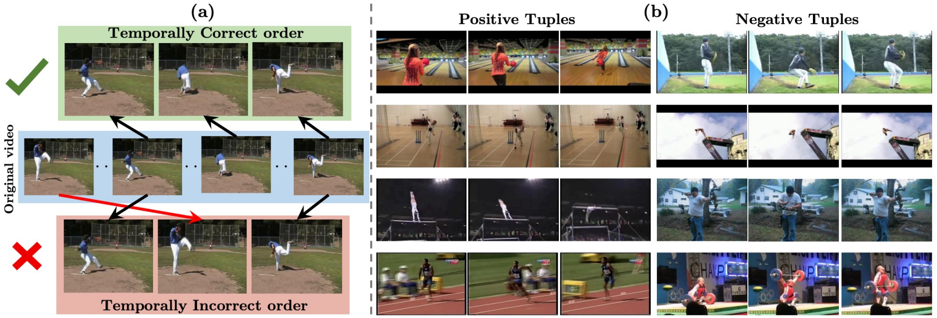


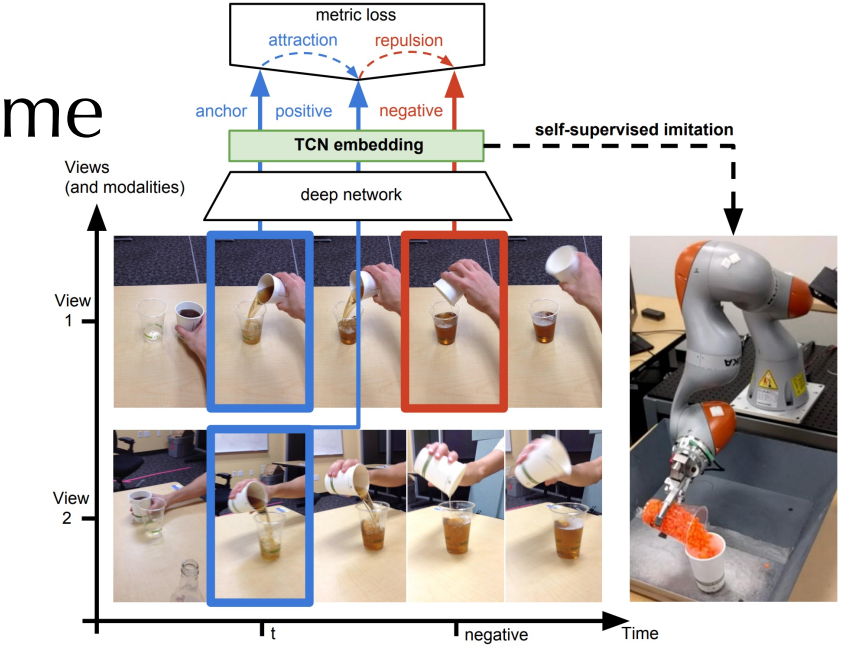
*Hoang et al. Midway Network: Learning Representations for Recognition and Motion from Latent Dynamics. ICLR 2026.*

# SSL with Time

- Use time as an additional source of supervision.



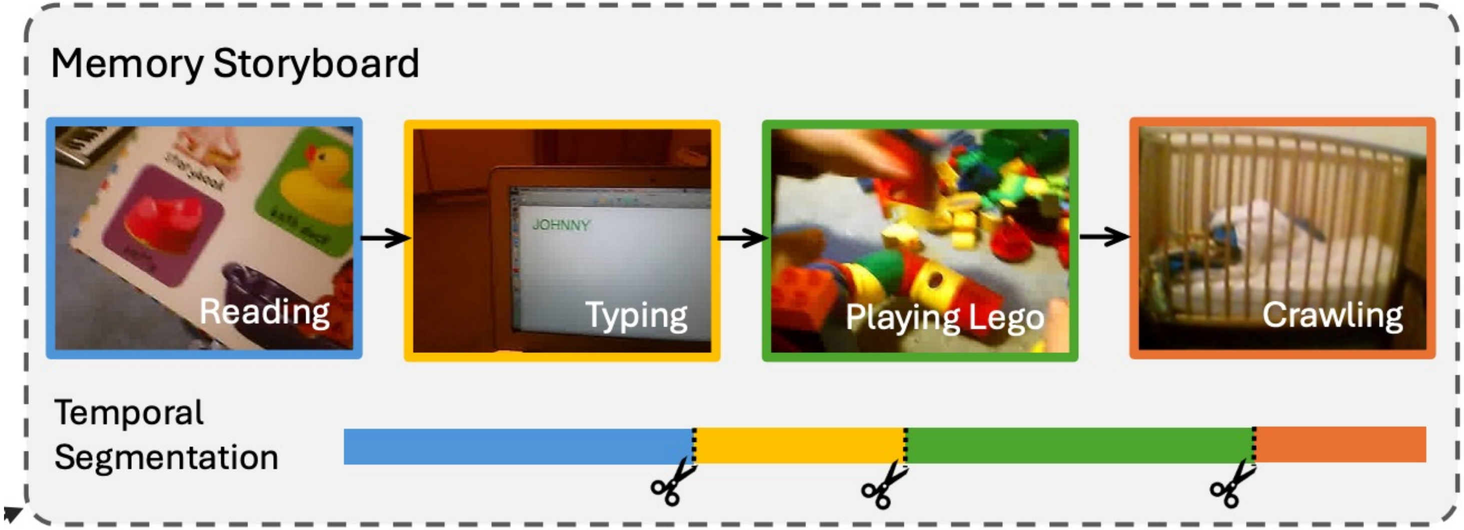




*Misra et al. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. ECCV 2016.*
*Sermanet et al. Time-Contrastive Networks: Self-Supervised Learning from Video. ICRA 2018.*
*Orhan et al. Self-Supervised Learning through the Eyes of a Child. NeurIPS 2020.*

# SSL with Time

- We can segment videos into meaningful events.
- Leverage the spatiotemporal continuity structure.



*Yang & Ren. Memory Storyboard: Leveraging Temporal Segmentation for Streaming Self-Supervised Learning from Egocentric Videos. arXiv 2025.*

# Summary

- Representation learning leverage the information in unlabeled data.

# Summary

- Representation learning leverage the information in unlabeled data.
- A foundation for sensorimotor learning.

# Summary

- Representation learning leverage the information in unlabeled data.
- A foundation for sensorimotor learning.
- Inductive biases matter.

# Summary

- Representation learning leverage the information in unlabeled data.
- A foundation for sensorimotor learning.
- Inductive biases matter.
- Possible learning objectives for egocentric videos.

# Summary

- Representation learning leverage the information in unlabeled data.
- A foundation for sensorimotor learning.
- Inductive biases matter.
- Possible learning objectives for egocentric videos.
- Incorporate 3D vision and actions for downstream planning.

# Emergent Attention, Object Discovery

- SSL representations show awareness of object classes and instance identities.

# Emergent Attention, Object Discovery

- SSL representations show awareness of object classes and instance identities.
- Why does attention show awareness of objects?

# Emergent Attention, Object Discovery

- SSL representations show awareness of object classes and instance identities.
- Why does attention show awareness of objects?
- The network is encouraged to associate different parts of the objects together in order to identify whether two inputs belong to the same image or not.

# Emergent Attention, Object Discovery

- SSL representations show awareness of object classes and instance identities.
- Why does attention show awareness of objects?
- The network is encouraged to associate different parts of the objects together in order to identify whether two inputs belong to the same image or not.
- Attending to semantically similar parts facilitates the process.

# Emergent Attention, Object Discovery

- SSL representations show awareness of object classes and instance identities.
- Why does attention show awareness of objects?
- The network is encouraged to associate different parts of the objects together in order to identify whether two inputs belong to the same image or not.
- Attending to semantically similar parts facilitates the process.
- The network is a hierarchical information processing pipeline – Lower layers integrate more granular and smaller neighborhood.

# Weak-to-Strong Supervision

- General idea: Use self-supervised learning to learn good features, which allow us to generate low-quality masks.

# Weak-to-Strong Supervision

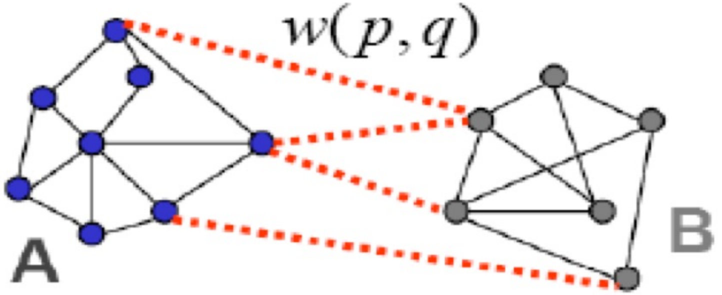
- General idea: Use self-supervised learning to learn good features, which allow us to generate low-quality masks.
- Then use these masks as pseudo labels and supervise the network to predict these low-quality masks.

# Weak-to-Strong Supervision

- General idea: Use self-supervised learning to learn good features, which allow us to generate low-quality masks.
- Then use these masks as pseudo labels and supervise the network to predict these low-quality masks.
- Question: how do we come up with masks? What loss is used to supervise the network?

# Graph Cut

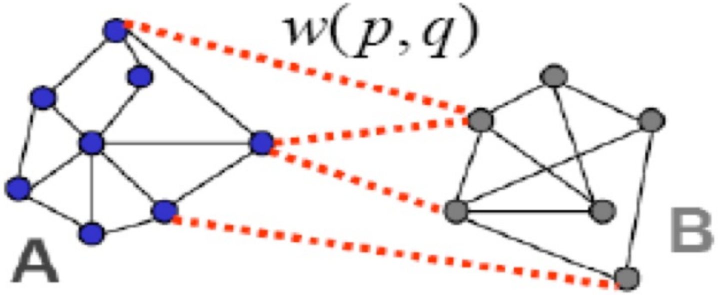
- Segmentation is essentially a clustering problem.



$$cut(A,B) = \sum_{p \in A, q \in B} w(p,q)$$

# Graph Cut

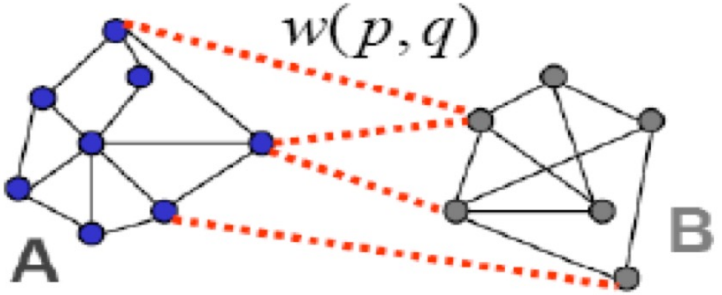
- Segmentation is essentially a clustering problem.
- We can transform the clustering problem with the graph cut problem.



$$cut(A,B) = \sum_{p \in A, q \in B} w(p,q)$$

# Graph Cut

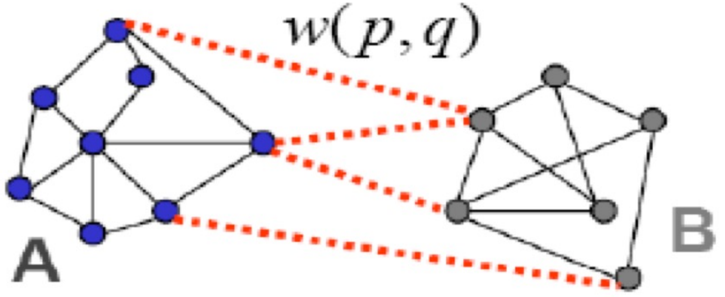
- Segmentation is essentially a clustering problem.
- We can transform the clustering problem with the graph cut problem.
- Pixel = node.



$$cut(A, B) = \sum_{p \in A, q \in B} w(p, q)$$

# Graph Cut

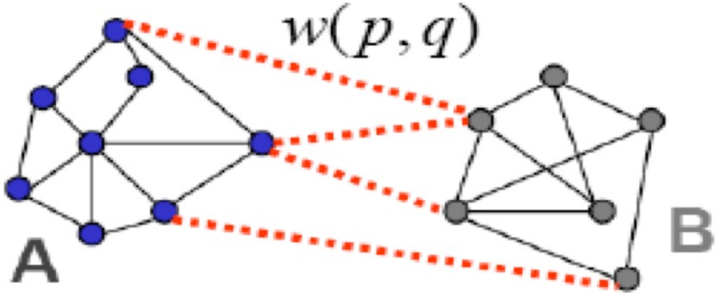
- Segmentation is essentially a clustering problem.
- We can transform the clustering problem with the graph cut problem.
- Pixel = node.
- Affinity between the two pixels = edge value (flow).



$$cut(A,B) = \sum_{p \in A, q \in B} w(p,q)$$

# Graph Cut

- Segmentation is essentially a clustering problem.
- We can transform the clustering problem with the graph cut problem.
- Pixel = node.
- Affinity between the two pixels = edge value (flow).
- Objective: Cut the graph into disconnected components with a minimum sum of edge values.

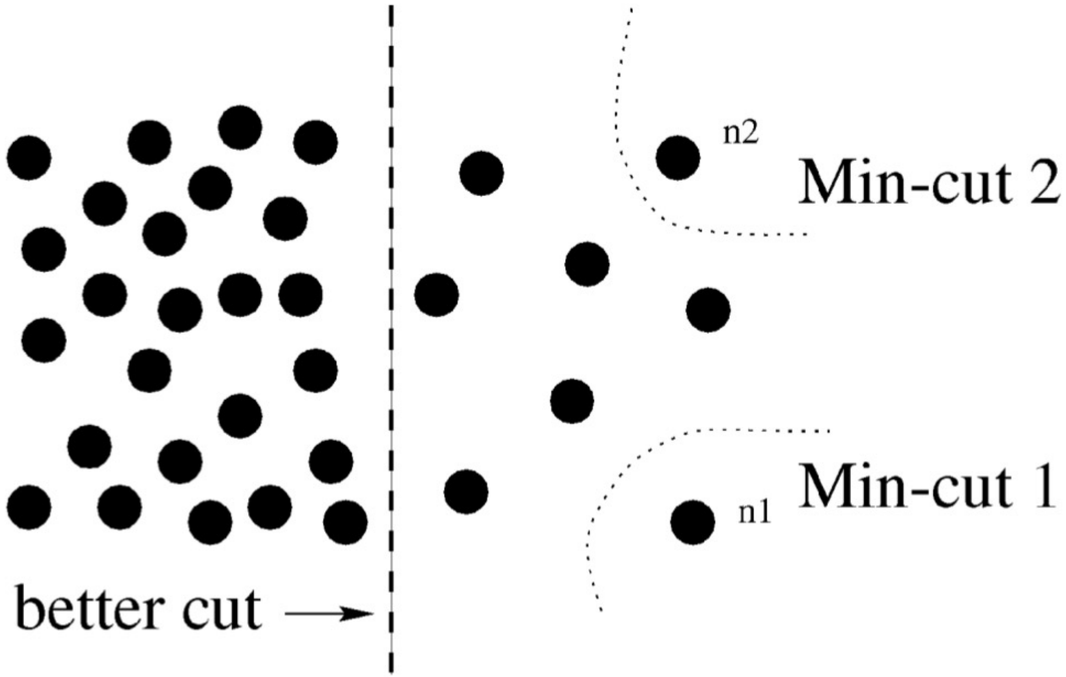


$$cut(A,B) = \sum_{p\in A, q\in B} w(p,q)$$

# Normalized Graph Cut (NCut)

- How to prevent cutting small isolated nodes?

$$Ncut(A, B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

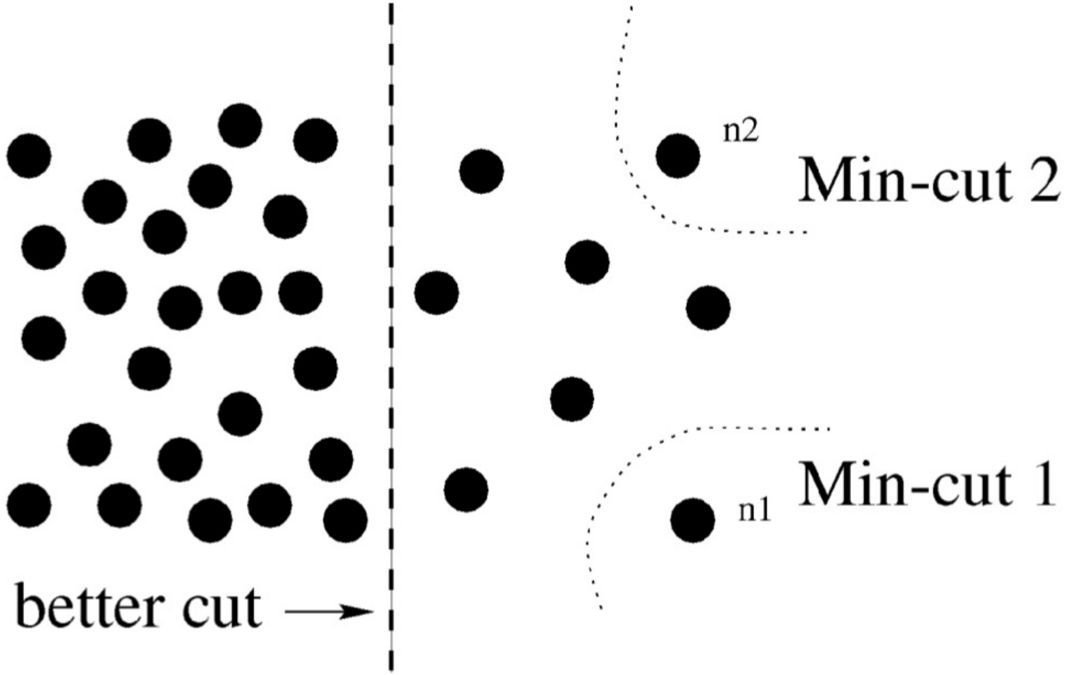


Shi and Malik. Normalized Cuts and Image Segmentation. TPAMI 2000.

# Normalized Graph Cut (NCut)

- How to prevent cutting small isolated nodes?
- Normalize by the total edge connections of a group to all the nodes.

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$



*Shi and Malik. Normalized Cuts and Image Segmentation. TPAMI 2000.*

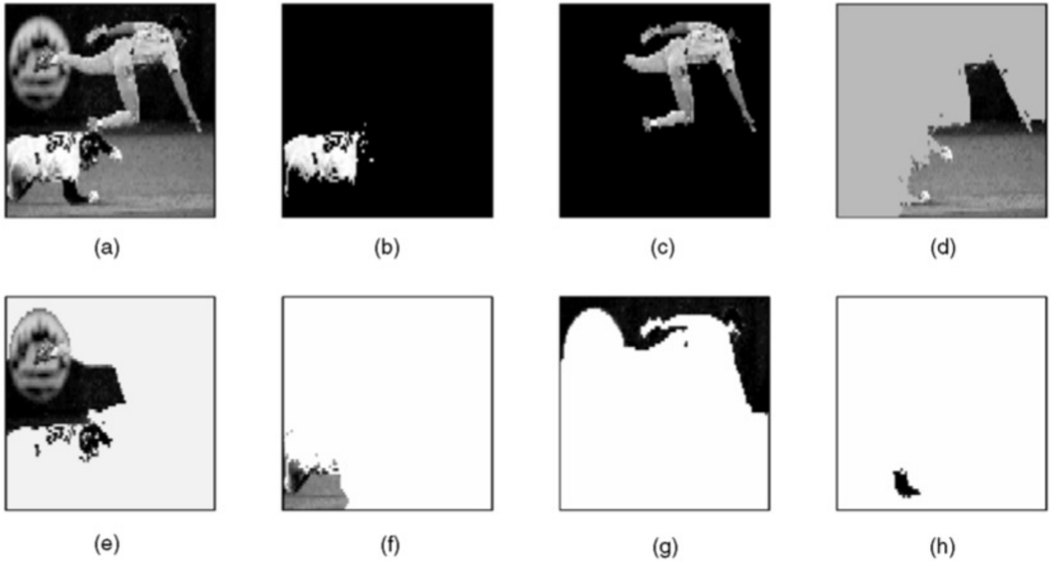
# NCut Details (Optional)

- A form of spectral clustering.
- Degree matrix $D$ $N \times N$ with $d_i$ on the diagonal.
- Weight matrix $W$ $N \times N$ symmetric $w_{ij}$.
- Selection vector $x_i = 1$ if $i \in A$ otherwise $-1$.
- Solve the minimization: $\min_y \frac{y^\top (D-W) y}{y^\top D y}$ $\quad y = (1 + x) - \frac{\sum_{i|x_i>0} d_i}{\sum_{i|x_i<0} d_i}(1 - x).$
- Generalized eigenvalue system: $(D - W)y = \lambda D y.$
- Let $z = D^{1/2} y$ $\quad D^{-\frac{1}{2}}(D - W)D^{-\frac{1}{2}} z = \lambda z.$

*Shi and Malik. Normalized Cuts and Image Segmentation. TPAMI 2000.*

# NCut

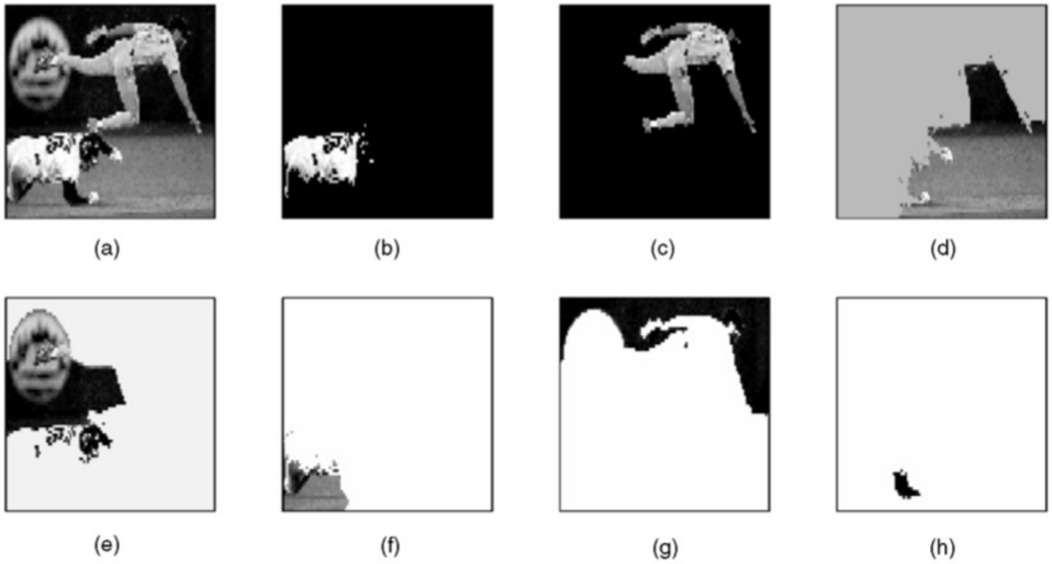
- Sort the eigenvectors from the smallest to the largest.



*Shi and Malik. Normalized Cuts and Image Segmentation. TPAMI 2000.*

# NCut

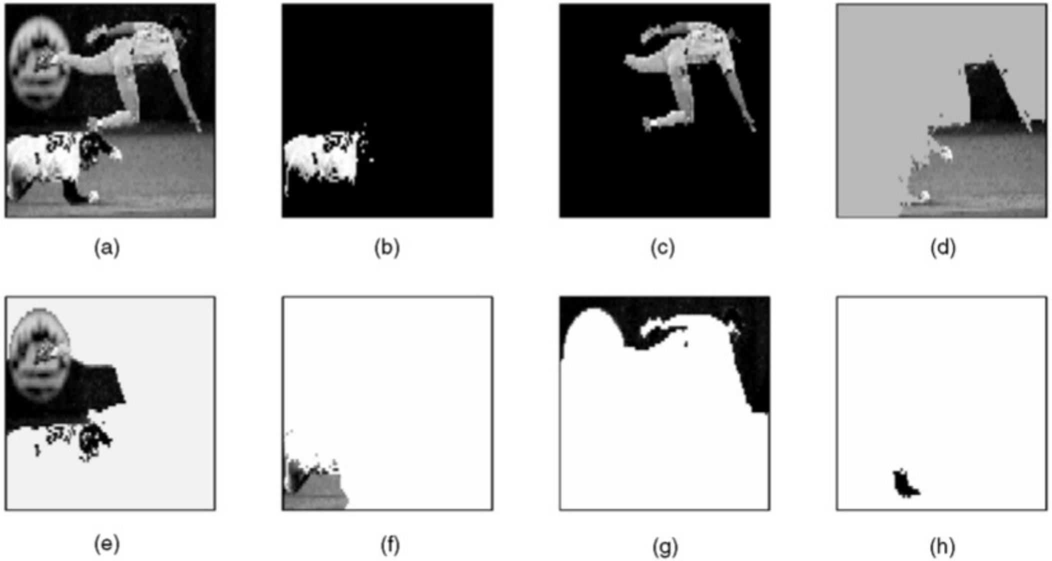
- Sort the eigenvectors from the smallest to the largest.
- This was a classic image segmentation technique operating directly on image intensity.



*Shi and Malik. Normalized Cuts and Image Segmentation. TPAMI 2000.*

# NCut

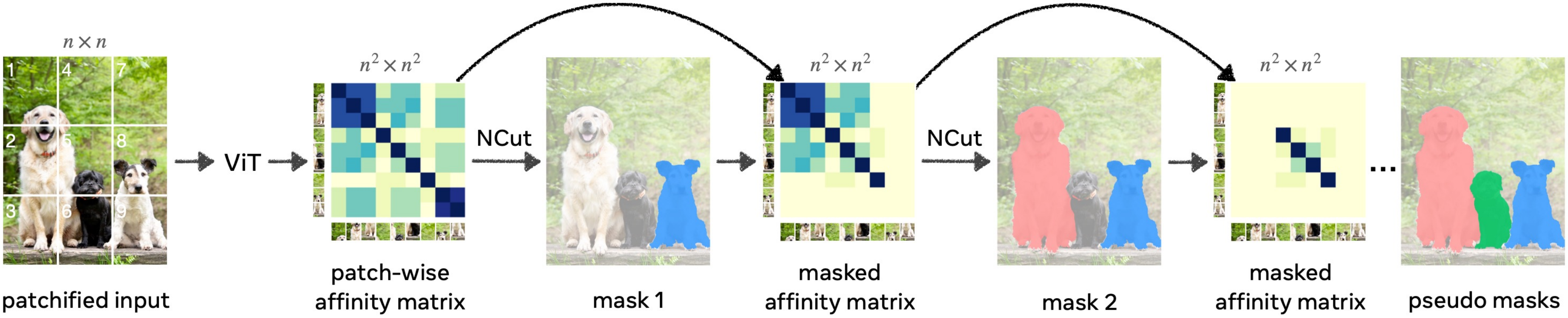
- Sort the eigenvectors from the smallest to the largest.
- This was a classic image segmentation technique operating directly on image intensity.
- Now, instead of segmenting pixels, we can directly segment semantically meaningful representations from self-supervision.



*Shi and Malik. Normalized Cuts and Image Segmentation. TPAMI 2000.*

# MaskCut

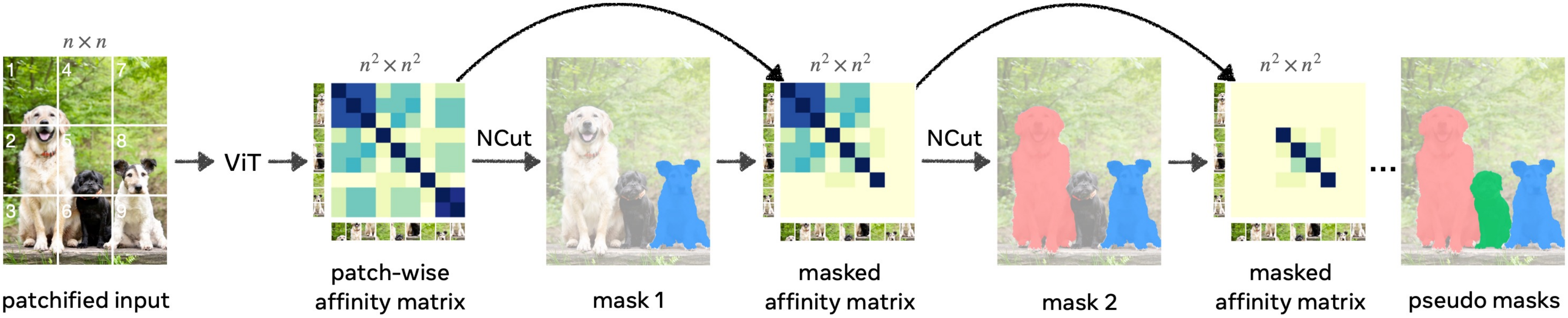
- Use a pretrained DINO ViT network.



*Wang et al. Self-supervised transformers for unsupervised object discovery using normalized cut. CVPR 2022.*
*Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.*

# MaskCut

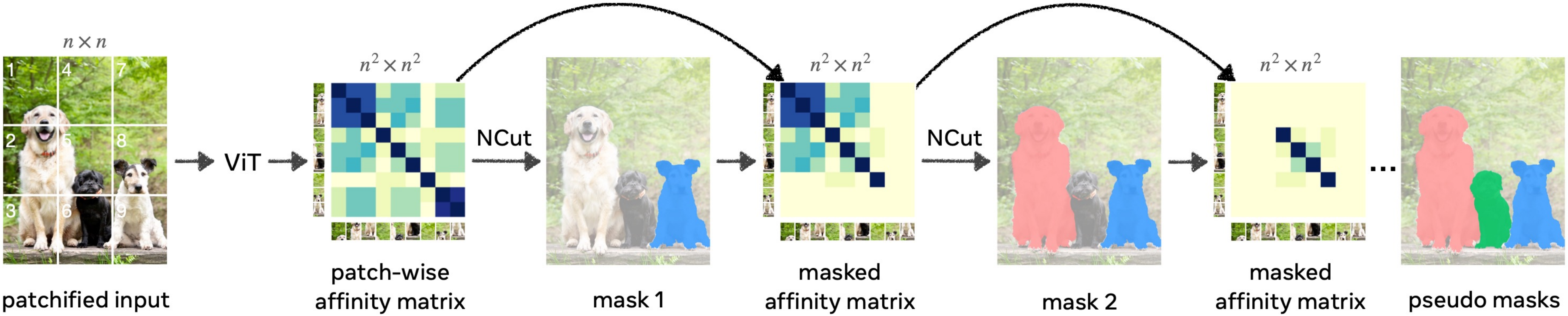
- Use a pretrained DINO ViT network.
- Use the “key” features from the last attention layer: $W_{ij} = \frac{K_i K_j}{\|K_i\|_2 \|K_j\|_2}$



*Wang et al. Self-supervised transformers for unsupervised object discovery using normalized cut. CVPR 2022.*
*Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.*

# MaskCut

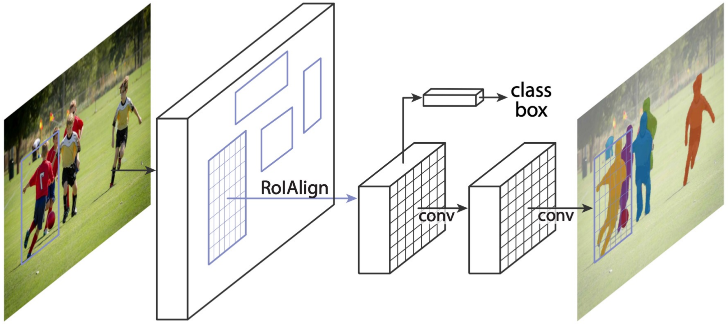
- Use a pretrained DINO ViT network.
- Use the "key" features from the last attention layer: $W_{ij} = \frac{K_i K_j}{\|K_i\|_2 \|K_j\|_2}$
- Iterative NCut on the pairwise matrix by masking out the regions from previous stages.

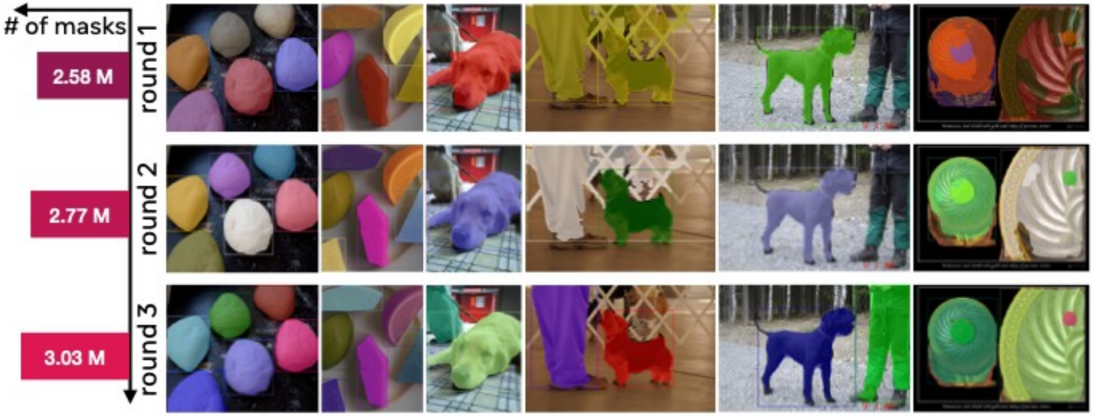


*Wang et al. Self-supervised transformers for unsupervised object discovery using normalized cut. CVPR 2022.*
*Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.*

# Iterative Self-Training

- Now add a MaskRCNN structure on top of the pretrained network.

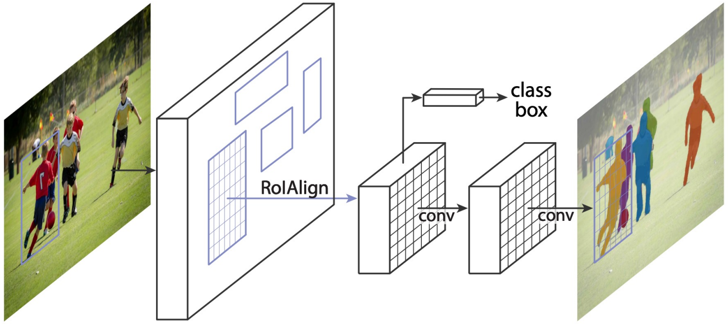


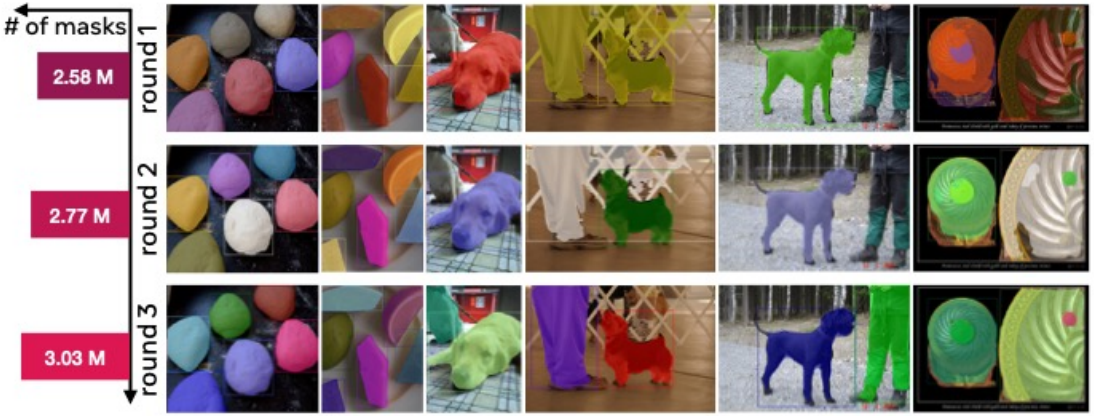


Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.

# Iterative Self-Training

- Now add a MaskRCNN structure on top of the pretrained network.
- Select the predictions with the highest confidence score and use them as labels.

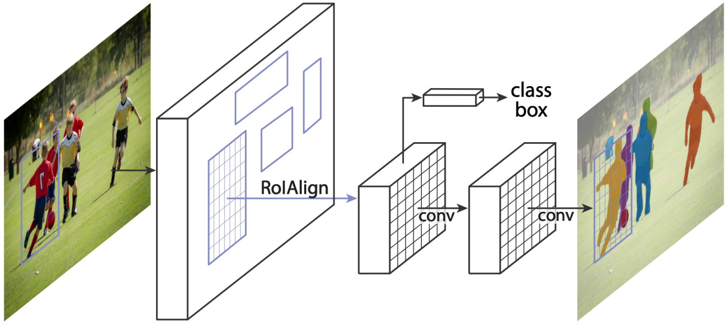


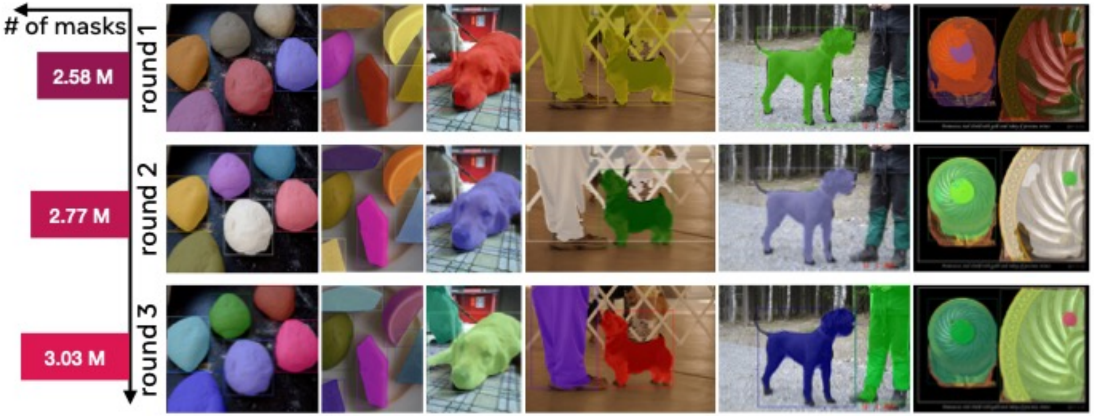


*Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.*

# Iterative Self-Training

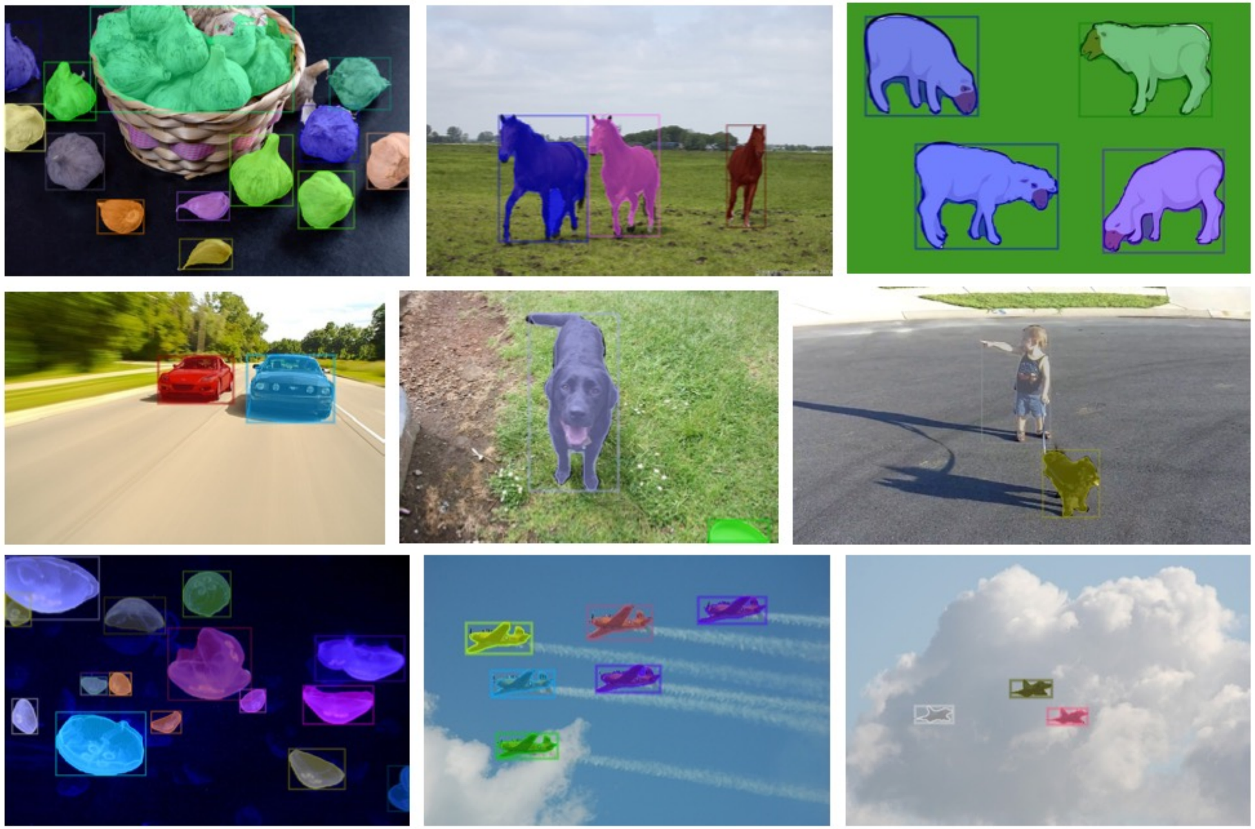
- Now add a MaskRCNN structure on top of the pretrained network.
- Select the predictions with the highest confidence score and use them as labels.
- Neural networks can learn from the noisy labels and output smoother predictions.





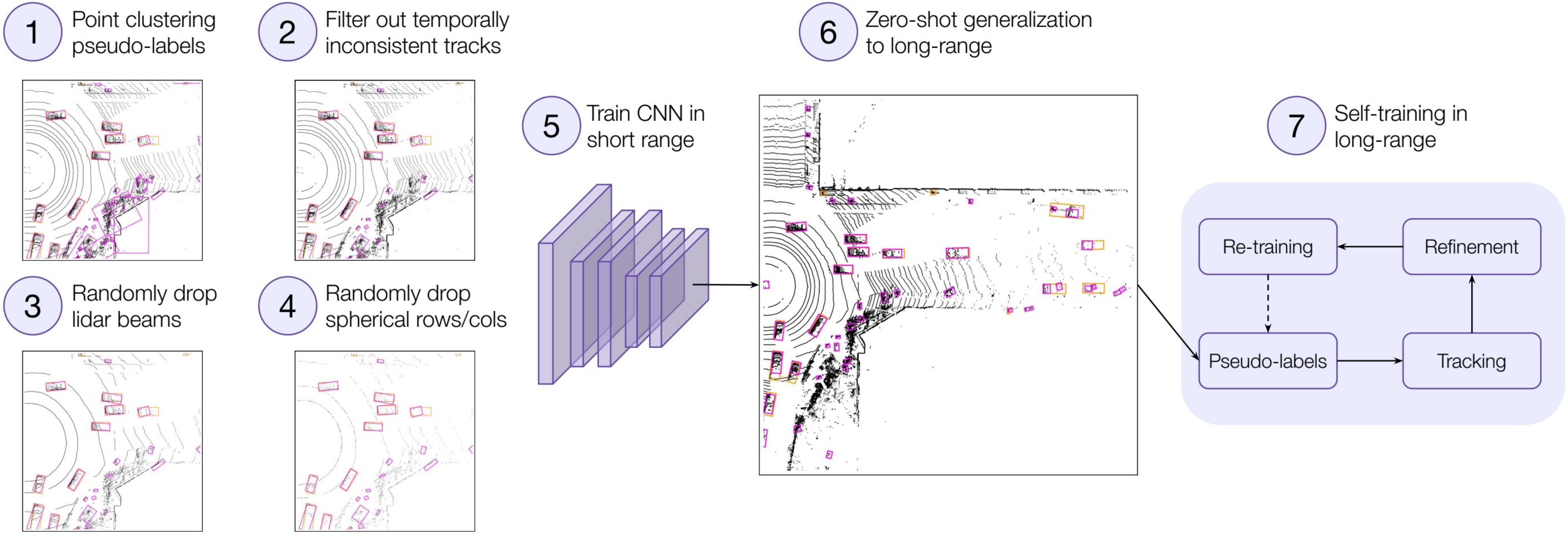
*Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.*

# More Visualization



*Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.*

# Pseudo Labels in 3D



1 Point clustering pseudo-labels

2 Filter out temporally inconsistent tracks

3 Randomly drop lidar beams

4 Randomly drop spherical rows/cols

5 Train CNN in short range

6 Zero-shot generalization to long-range

7 Self-training in long-range

Re-training

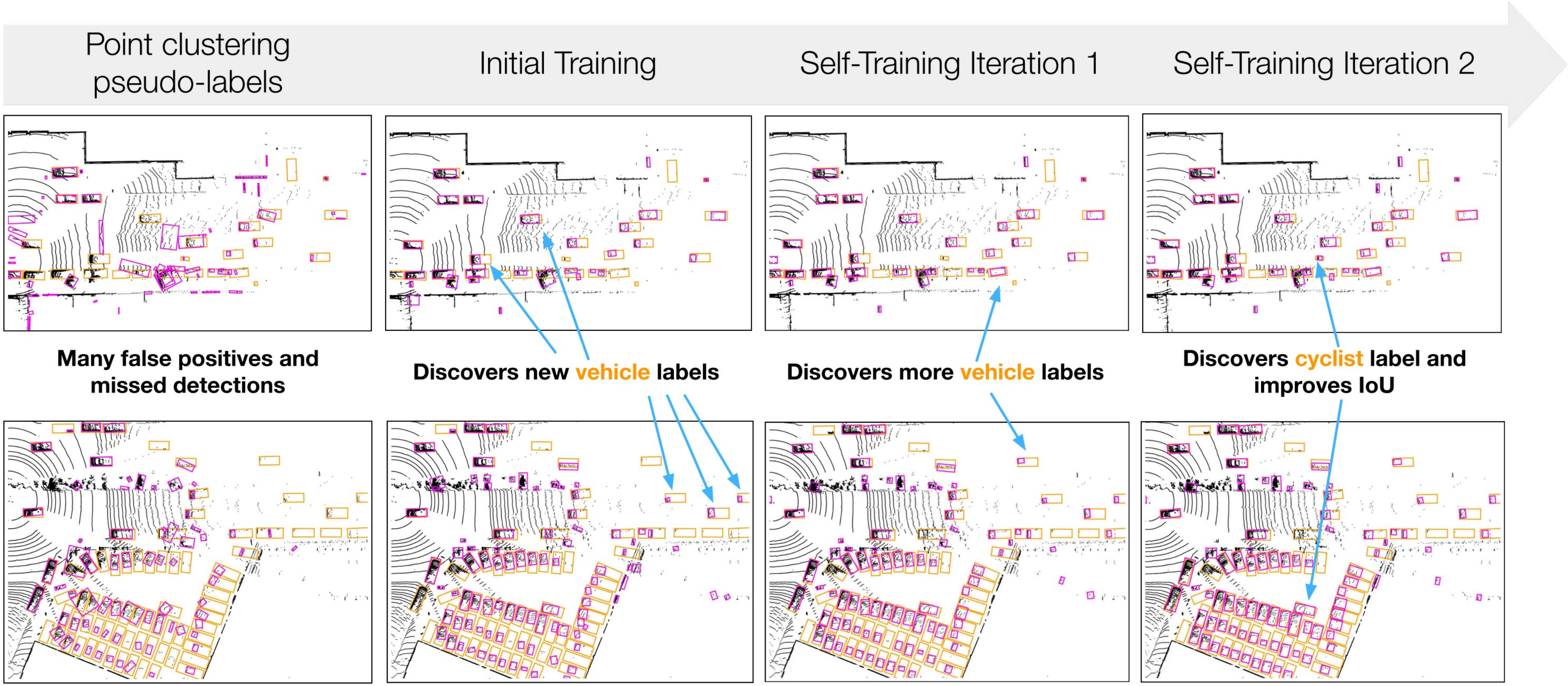
Refinement

Pseudo-labels

Tracking

*Zhang et al. Towards Unsupervised Object Detection from LiDAR Point Clouds. CVPR 2023.*

# Iterative Refinement of Pseudo Labels



Point clustering pseudo-labels

Initial Training

Self-Training Iteration 1

Self-Training Iteration 2

**Many false positives and missed detections**

**Discovers new vehicle labels**

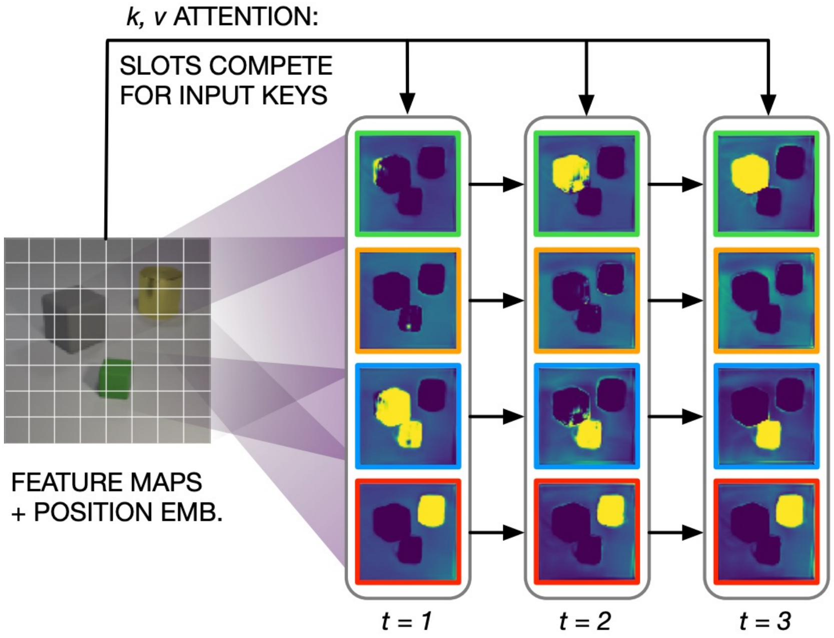
**Discovers more vehicle labels**

**Discovers cyclist label and improves IoU**

*Zhang et al. Towards Unsupervised Object Detection from LiDAR Point Clouds. CVPR 2023.*

# Slot Attention Networks

- Can we learn clustering as an end-to-end operation?

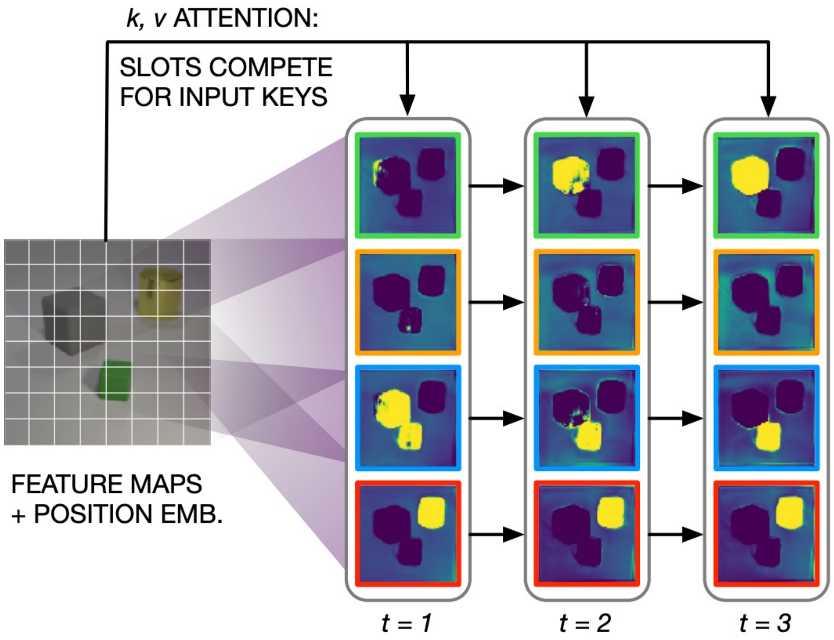


(a) Slot Attention module.

*Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.*

# Slot Attention Networks

- Can we learn clustering as an end-to-end operation?
- Slot attention is inspired by the success of the attention mechanism.

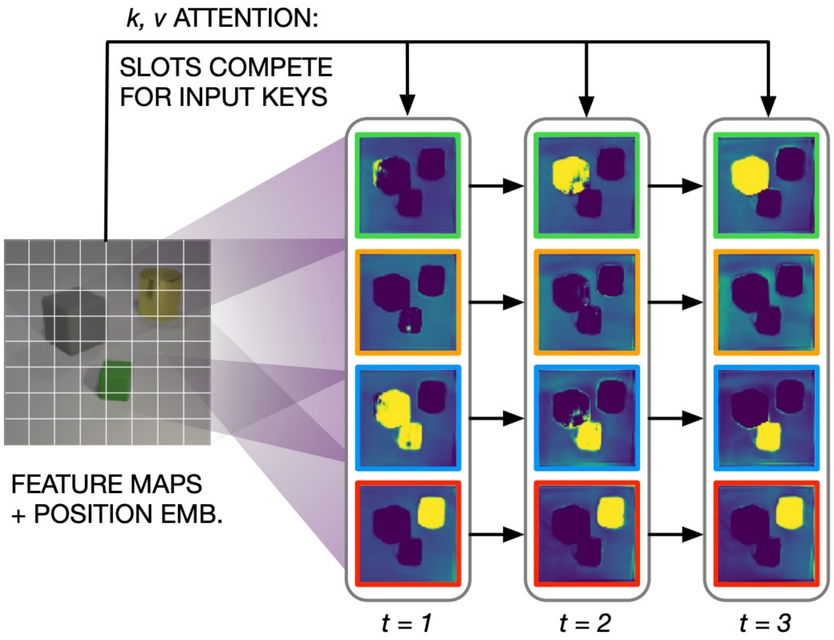


(a) Slot Attention module.

*Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.*

# Slot Attention Networks

- Can we learn clustering as an end-to-end operation?
- Slot attention is inspired by the success of the attention mechanism.
- Each "slot" attends to a region of the image and stores an object centric representation.

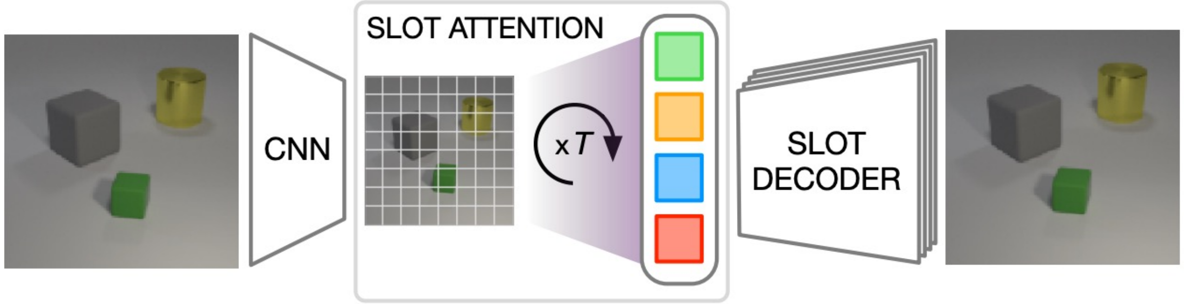


(a) Slot Attention module.

*Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.*

# Slot Attention Networks

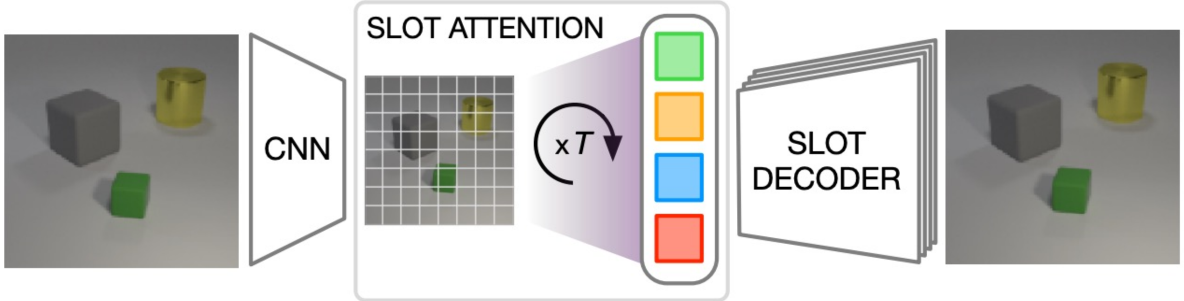
- Goal: Reconstruct the image with a concise slot-based representation.



*Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.*

# Slot Attention Networks

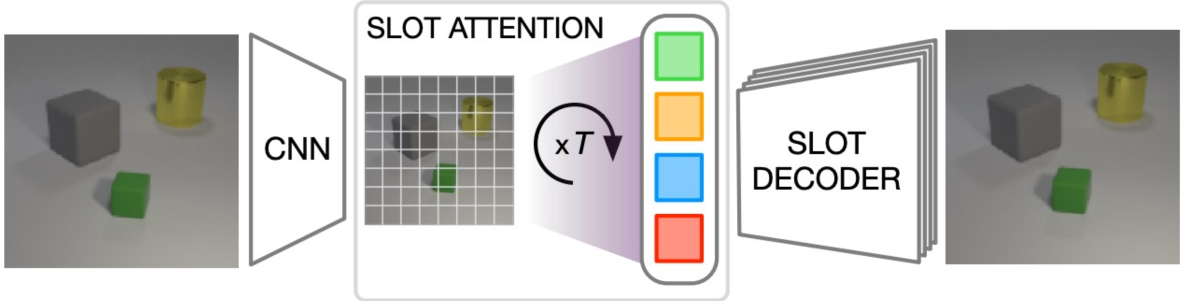
- Goal: Reconstruct the image with a concise slot-based representation.
- Input: $x \in \mathbb{R}^{N \times D}$ (after encoder), Slots: $m \in \mathbb{R}^{M \times D}$. Normalize: $\widetilde{m}_{t-1} = LN(m_{t-1})$.



Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.

# Slot Attention Networks

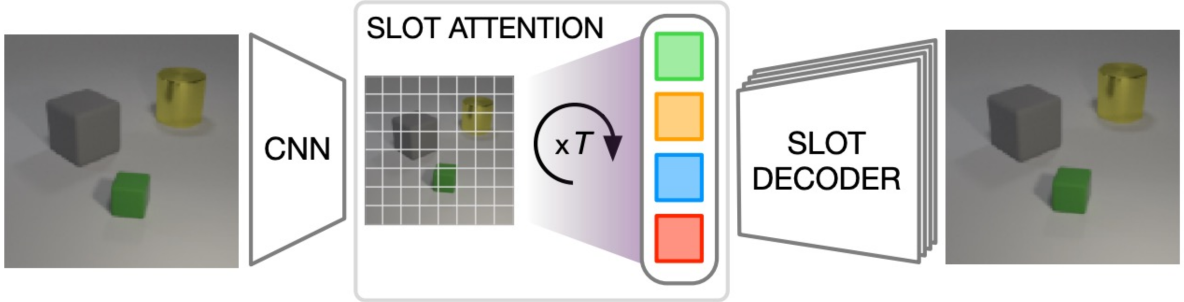
- Goal: Reconstruct the image with a concise slot-based representation.
- Input: $x \in \mathbb{R}^{N\times D}$ (after encoder), Slots: $m \in \mathbb{R}^{M\times D}$. Normalize: $\tilde{m}_{t-1} = LN(m_{t-1})$.
- Attention over slots: $a_{t,i,j} = \frac{\frac{1}{\sqrt{D}} k(x_i) \cdot q(\tilde{m}_j)^\top}{\sum_j \frac{1}{\sqrt{D}} k(x_i) \cdot q(\tilde{m}_j)^\top}$.



Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.

# Slot Attention Networks

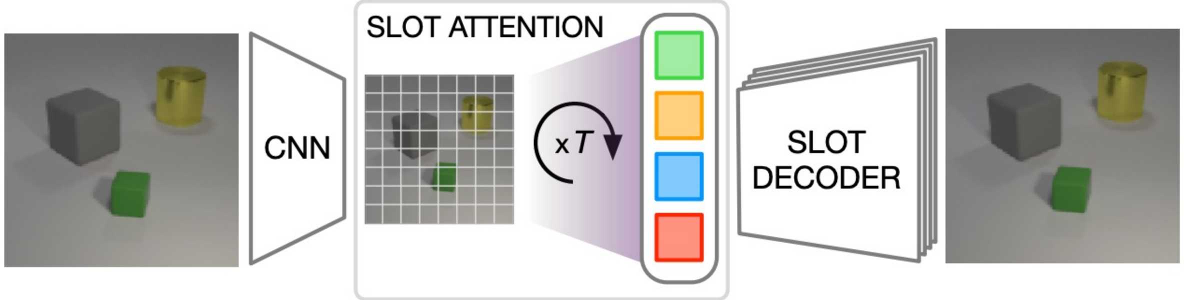
- Goal: Reconstruct the image with a concise slot-based representation.
- Input: $x \in \mathbb{R}^{N \times D}$ (after encoder), Slots: $m \in \mathbb{R}^{M \times D}$. Normalize: $\tilde{m}_{t-1} = LN(m_{t-1})$.
- Attention over slots: $a_{t,i,j} = \frac{\frac{1}{\sqrt{D}} k(x_i) \cdot q(\tilde{m}_j)^\top}{\sum_j \frac{1}{\sqrt{D}} k(x_i) \cdot q(\tilde{m}_j)^\top}$.
- Updates: $u_{tj} = \sum_i a_{tij} v(x_i)$.



*Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.*

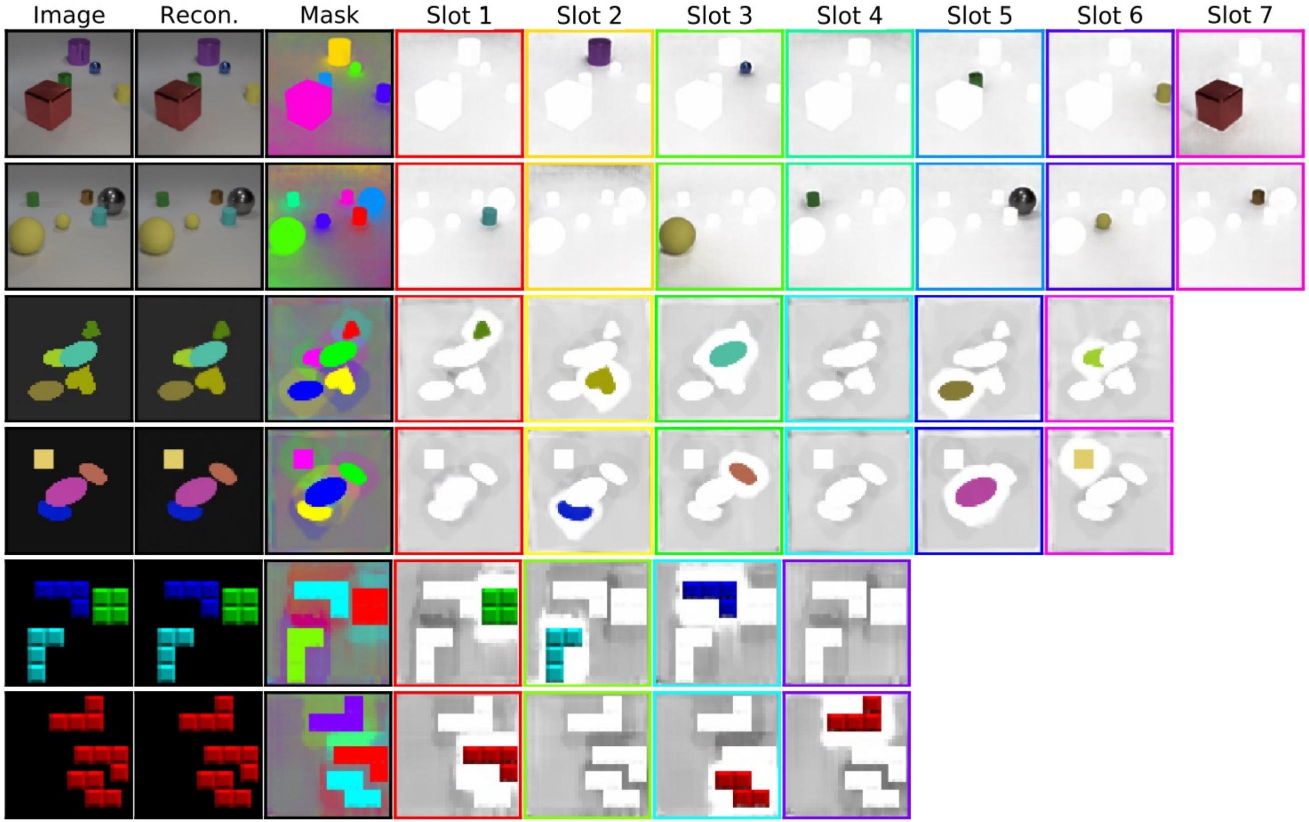
# Slot Attention Networks

- Goal: Reconstruct the image with a concise slot-based representation.
- Input: $x \in \mathbb{R}^{N \times D}$ (after encoder), Slots: $m \in \mathbb{R}^{M \times D}$. Normalize: $\widetilde{m}_{t-1} = LN(m_{t-1})$.
- Attention over slots: $a_{t,i,j} = \frac{\frac{1}{\sqrt{D}} k(x_i) \cdot q(\tilde{m}_j)^\top}{\sum_j \frac{1}{\sqrt{D}} k(x_i) \cdot q(\tilde{m}_j)^\top}$.
- Updates: $u_{tj} = \sum_i a_{tij} v(x_i)$.
- Write into slots: $m_t = GRU(m_{t-1}, u_t) + MLP(\tilde{m}_{t-1})$.



Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.

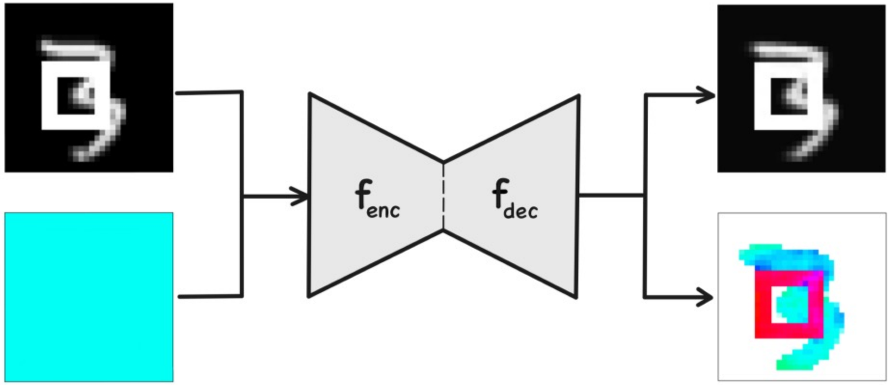
# Slot Attention Networks

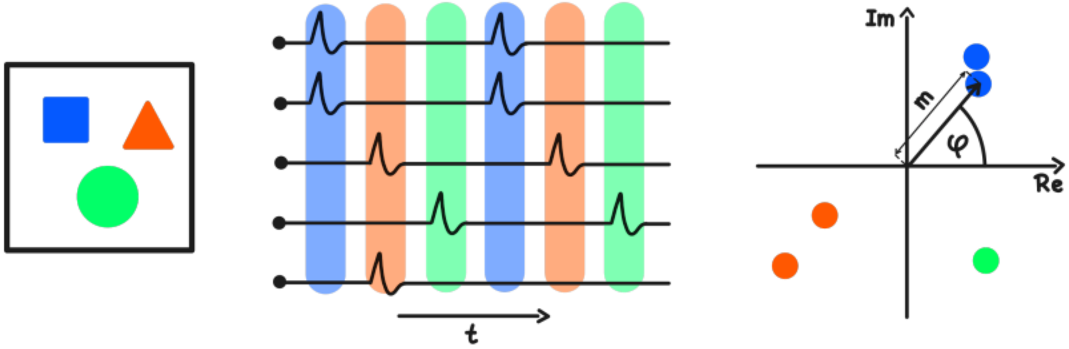


*Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.*

# Complex-Valued Autoencoders (CAEs)

- The complex number can represent magnitude and phase: $z = m \cdot e^{i\varphi} \in \mathbb{C}.$

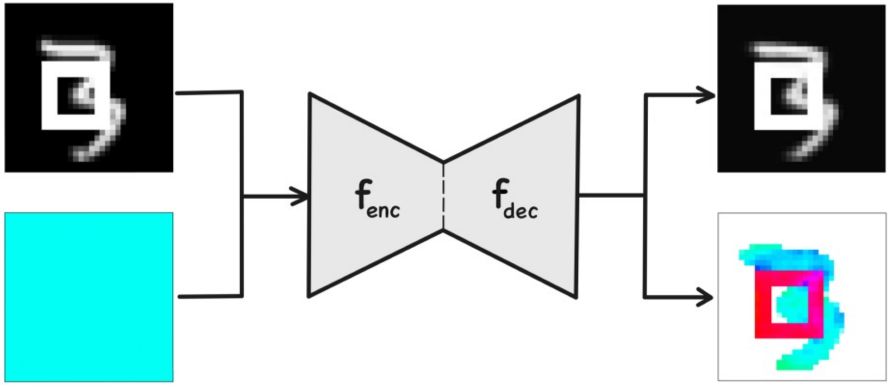


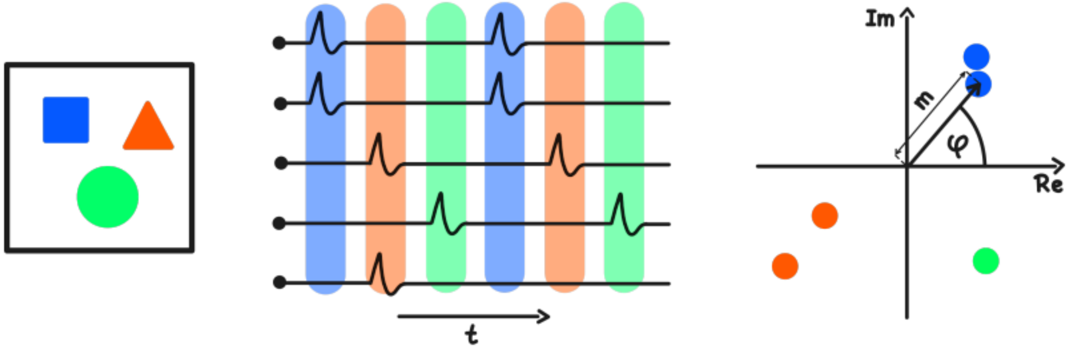


*Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.*

# Complex-Valued Autoencoders (CAEs)

- The complex number can represent magnitude and phase: $z = m \cdot e^{i\varphi} \in \mathbb{C}$.
- Each pixel starts with an initial phase 0.



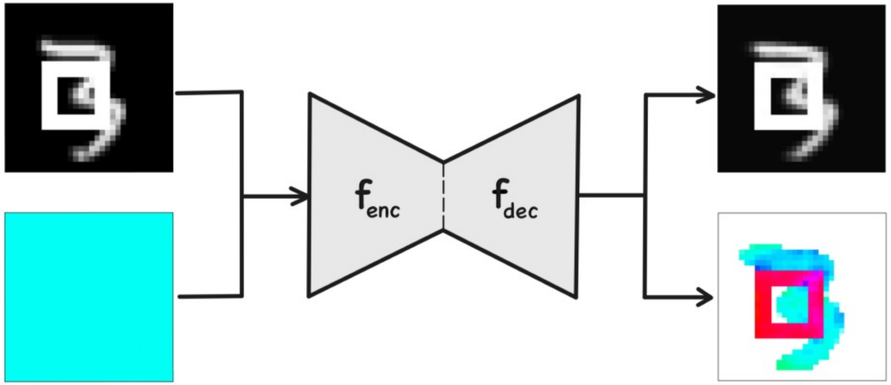


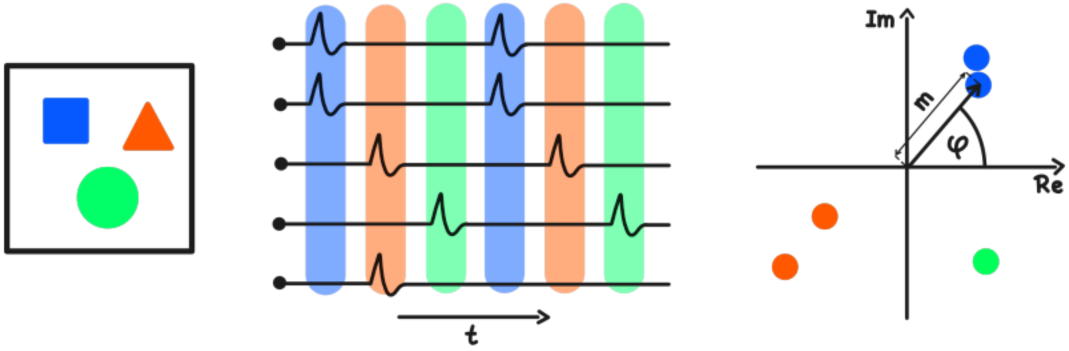
*Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.*

# Complex-Valued Autoencoders (CAEs)

- The complex number can represent magnitude and phase: $z = m \cdot e^{i\varphi} \in \mathbb{C}$.
- Each pixel starts with an initial phase 0.

$$\hat{\mathbf{z}} = f_{\text{dec}}(f_{\text{enc}}(\mathbf{x})) \in \mathbb{C}^{h \times w}.$$





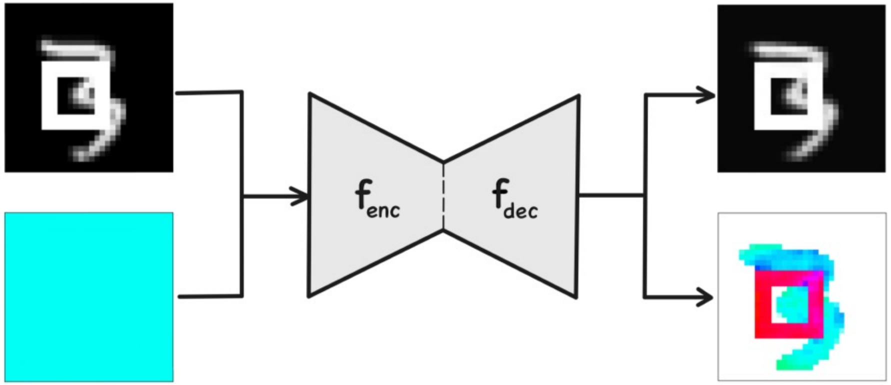
*Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.*

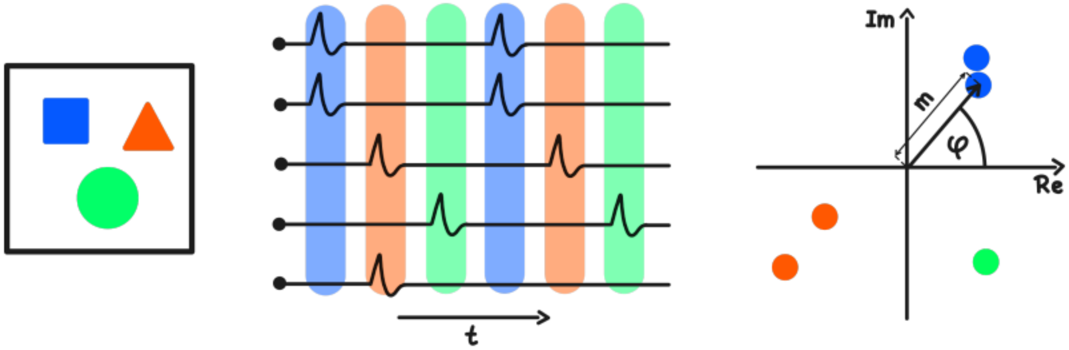
# Complex-Valued Autoencoders (CAEs)

- The complex number can represent magnitude and phase: $z = m \cdot e^{i\varphi} \in \mathbb{C}$.
- Each pixel starts with an initial phase 0.

$$\hat{\mathbf{z}} = f_{\text{dec}}(f_{\text{enc}}(\mathbf{x})) \in \mathbb{C}^{h \times w}.$$

$$\hat{\mathbf{x}} = f_{\text{out}}(\hat{\mathbf{z}}) \in \mathbb{R}^{h \times w}.$$





Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.

# CAE: More Details

- Apply weights separately to real and imaginary:

$$\boldsymbol{\psi} = f_{\mathbf{w}}(\mathbf{z}) = f_{\mathbf{w}}(\mathrm{Re}(\mathbf{z})) + f_{\mathbf{w}}(\mathrm{Im}(\mathbf{z})) \cdot i \quad \in \mathbb{C}^{d_{\text{out}}}$$

[Reichert & Serre, 2014]

*Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.*
*Reichert & Serre. Neuronal Synchrony in Complex-Valued Deep Networks. ICLR 2014.*

# CAE: More Details

- Apply weights separately to real and imaginary:

$$\boldsymbol{\psi} = f_{\mathbf{w}}(\mathbf{z}) = f_{\mathbf{w}}(\mathrm{Re}(\mathbf{z})) + f_{\mathbf{w}}(\mathrm{Im}(\mathbf{z})) \cdot i \quad \in \mathbb{C}^{d_{\mathrm{out}}}$$

- Bias on magnitude and phase:

$$\boldsymbol{m}_{\boldsymbol{\psi}} = |\boldsymbol{\psi}| + \boldsymbol{b}_{\boldsymbol{m}} \quad \in \mathbb{R}^{d_{\mathrm{out}}} \; \boldsymbol{\varphi}_{\boldsymbol{\psi}} = \arg(\boldsymbol{\psi}) + \boldsymbol{b}_{\boldsymbol{\varphi}} \quad \in \mathbb{R}^{d_{\mathrm{out}}}$$

[Reichert & Serre, 2014]

*Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.*
*Reichert & Serre. Neuronal Synchrony in Complex-Valued Deep Networks. ICLR 2014.*

# CAE: More Details

- Apply weights separately to real and imaginary:

$$\boldsymbol{\psi} = f_{\mathbf{w}}(\mathbf{z}) = f_{\mathbf{w}}(\mathrm{Re}(\mathbf{z})) + f_{\mathbf{w}}(\mathrm{Im}(\mathbf{z})) \cdot i \quad \in \mathbb{C}^{d_{\text{out}}}$$

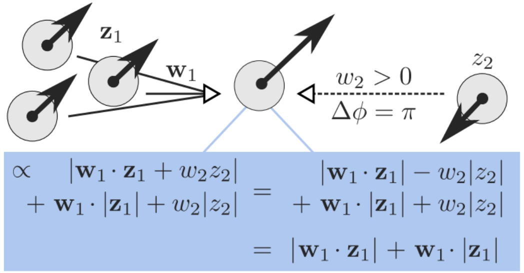
- Bias on magnitude and phase:

$$\boldsymbol{m}_{\boldsymbol{\psi}} = |\boldsymbol{\psi}| + \boldsymbol{b}_{\boldsymbol{m}} \quad \in \mathbb{R}^{d_{\text{out}}} \quad \boldsymbol{\varphi}_{\boldsymbol{\psi}} = \arg(\boldsymbol{\psi}) + \boldsymbol{b}_{\boldsymbol{\varphi}} \quad \in \mathbb{R}^{d_{\text{out}}}$$

- Gating:

$$\boldsymbol{\chi} = f_{\mathbf{w}}(|\mathbf{z}|) + \boldsymbol{b}_{\boldsymbol{m}} \qquad \in \mathbb{R}^{d_{\text{out}}}$$

$$\boldsymbol{m}_{\mathbf{z}} = \underbrace{0.5 \cdot \boldsymbol{m}_{\boldsymbol{\psi}}}_{\text{Synchrony Term}} + \underbrace{0.5 \cdot \boldsymbol{\chi}}_{\text{Classic Term}} \quad \in \mathbb{R}^{d_{\text{out}}}$$



$$\begin{aligned} \propto \quad & |\mathbf{w}_1 \cdot \mathbf{z}_1 + w_2 z_2| \\ & + \mathbf{w}_1 \cdot |\mathbf{z}_1| + w_2|z_2| = \begin{array}{r} |\mathbf{w}_1 \cdot \mathbf{z}_1| - w_2|z_2| \\ + \mathbf{w}_1 \cdot |\mathbf{z}_1| + w_2|z_2| \end{array} \\ & = |\mathbf{w}_1 \cdot \mathbf{z}_1| + \mathbf{w}_1 \cdot |\mathbf{z}_1| \end{aligned}$$

[Reichert & Serre, 2014]

*Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.*
*Reichert & Serre. Neuronal Synchrony in Complex-Valued Deep Networks. ICLR 2014.*

# CAE: More Details

- Apply weights separately to real and imaginary:

$$\boldsymbol{\psi} = f_{\mathbf{w}}(\mathbf{z}) = f_{\mathbf{w}}(\mathrm{Re}(\mathbf{z})) + f_{\mathbf{w}}(\mathrm{Im}(\mathbf{z})) \cdot i \quad \in \mathbb{C}^{d_{\text{out}}}$$

- Bias on magnitude and phase:

$$\boldsymbol{m_\psi} = |\boldsymbol{\psi}| + \boldsymbol{b_m} \quad \in \mathbb{R}^{d_{\text{out}}} \quad \boldsymbol{\varphi_\psi} = \arg(\boldsymbol{\psi}) + \boldsymbol{b_\varphi} \quad \in \mathbb{R}^{d_{\text{out}}}$$

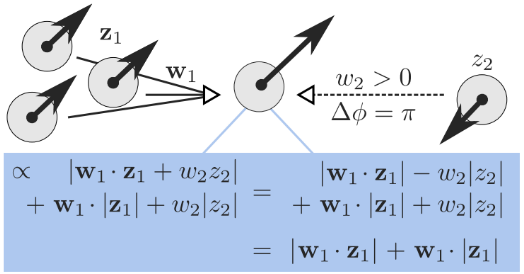
- Gating:

$$\boldsymbol{\chi} = f_{\mathbf{w}}(|\mathbf{z}|) + \boldsymbol{b_m} \quad \in \mathbb{R}^{d_{\text{out}}}$$

$$\boldsymbol{m_{\mathbf{z}}} = \underbrace{0.5 \cdot \boldsymbol{m_\psi}}_{\text{Synchrony Term}} + \underbrace{0.5 \cdot \boldsymbol{\chi}}_{\text{Classic Term}} \quad \in \mathbb{R}^{d_{\text{out}}}$$

- Activation $\mathbf{z}' = \mathrm{ReLU}(\mathrm{BatchNorm}(\boldsymbol{m_{\mathbf{z}}})) \circ e^{i\boldsymbol{\varphi_\psi}} \quad \in \mathbb{C}^{d_{\text{out}}}$

$$f_{\boldsymbol{w}}(\boldsymbol{z}) = \boldsymbol{w} \cdot \boldsymbol{z}$$

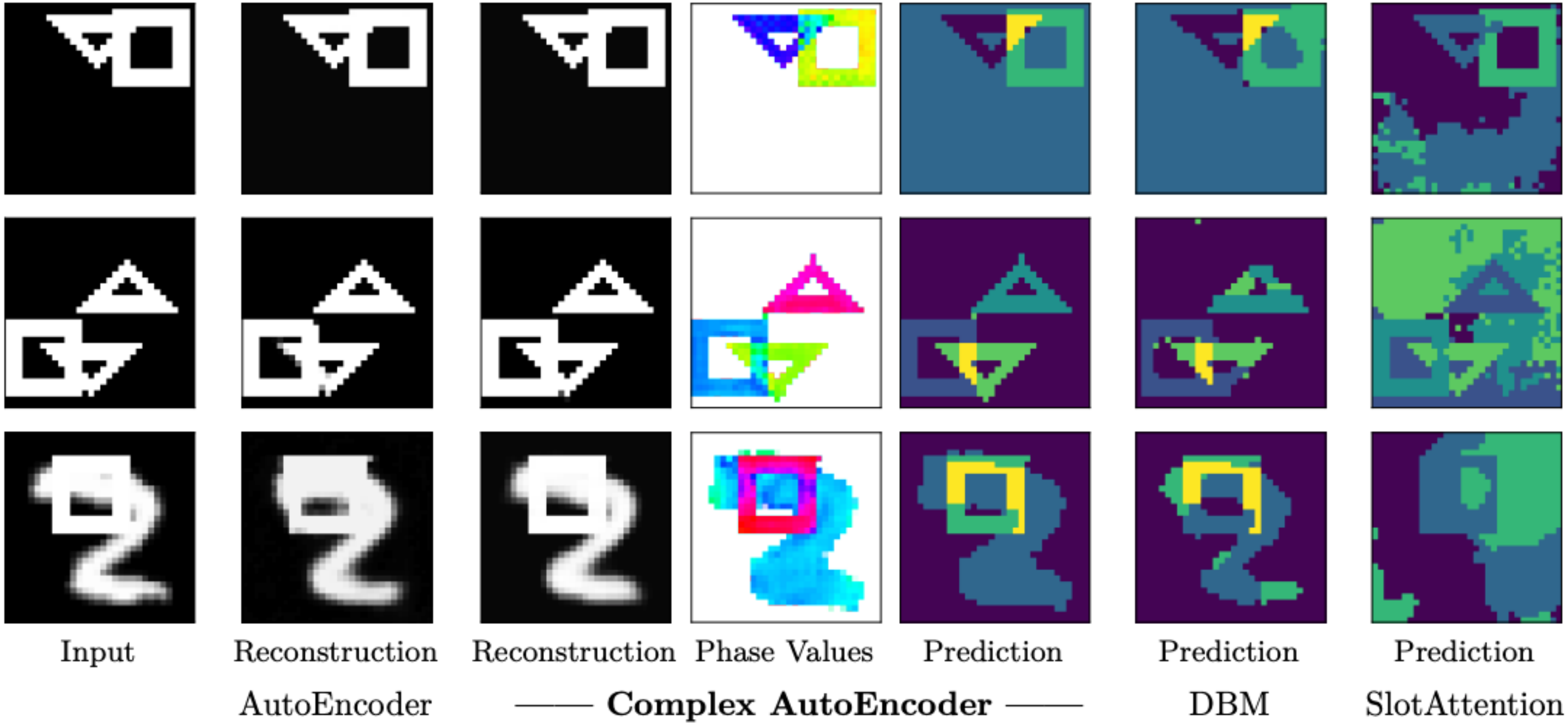


$$\begin{aligned} \propto \quad & |\mathbf{w}_1 \cdot \mathbf{z}_1 + w_2 z_2| \\ & + \mathbf{w}_1 \cdot |\mathbf{z}_1| + w_2|z_2| = \quad |\mathbf{w}_1 \cdot \mathbf{z}_1| - w_2|z_2| \\ & \qquad + \mathbf{w}_1 \cdot |\mathbf{z}_1| + w_2|z_2| \\ & = \quad |\mathbf{w}_1 \cdot \mathbf{z}_1| + \mathbf{w}_1 \cdot |\mathbf{z}_1| \end{aligned}$$

[Reichert & Serre, 2014]

*Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.*
*Reichert & Serre. Neuronal Synchrony in Complex-Valued Deep Networks. ICLR 2014.*

# Complex-Valued Autoencoders



*Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.*

# Summary: Object Discovery

- Combine deep features with clustering algorithms.

# Summary: Object Discovery

- Combine deep features with clustering algorithms.
- Pseudo-labels to train detector networks.

# Summary: Object Discovery

- Combine deep features with clustering algorithms.
- Pseudo-labels to train detector networks.
- Creative end-to-end learning-based solutions exist, but there are still plenty room for improvement.
  - Possible to train from scratch!

# Summary: Object Discovery

- Combine deep features with clustering algorithms.
- Pseudo-labels to train detector networks.
- Creative end-to-end learning-based solutions exist, but there are still plenty room for improvement.
  - Possible to train from scratch!
- What do we make use of the discovered objects? Is it better to keep the awareness in the latent space?