DS-GA.3001 Embodied Learning and Vision

Mengye Ren

NYU

Spring 2025

embodied-learning-vision-course.github.io

Lecture Slides for Note Taking

NYU

Multi-Sensor Fusion

- LiDAR is precise in depth perception, but the point cloud format is sparse and non-uniform (dense around the ego-car and sparse in long distance.)

Multi-Sensor Fusion

- LiDAR is precise in depth perception, but the point cloud format is sparse and non-uniform (dense around the ego-car and sparse in long distance.)
- Camera provides high resolution 2D view and good for long distance but lacks 3D. Can we achieve the best of both worlds?

Multi-Sensor Fusion

- LiDAR is precise in depth perception, but the point cloud format is sparse and non-uniform (dense around the ego-car and sparse in long distance.)
- Camera provides high resolution 2D view and good for long distance but lacks 3D. Can we achieve the best of both worlds?
- Late fusion: Generate proposals from one branch (e.g. LiDAR) and refine (e.g. using Camera).

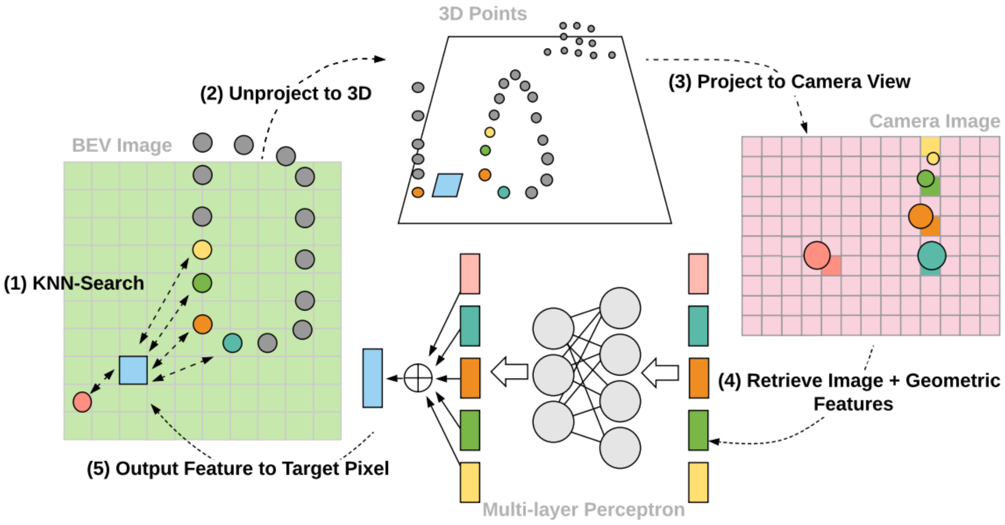
NYU

Multi-Sensor Fusion

- LiDAR is precise in depth perception, but the point cloud format is sparse and non-uniform (dense around the ego-car and sparse in long distance.)
- Camera provides high resolution 2D view and good for long distance but lacks 3D. Can we achieve the best of both worlds?
- Late fusion: Generate proposals from one branch (e.g. LiDAR) and refine (e.g. using Camera).
- Is there a way to combine the features from both modality in lower layers?

Camera-LiDAR Projection

- Unproject LiDAR points to camera view (i.e. Range View)

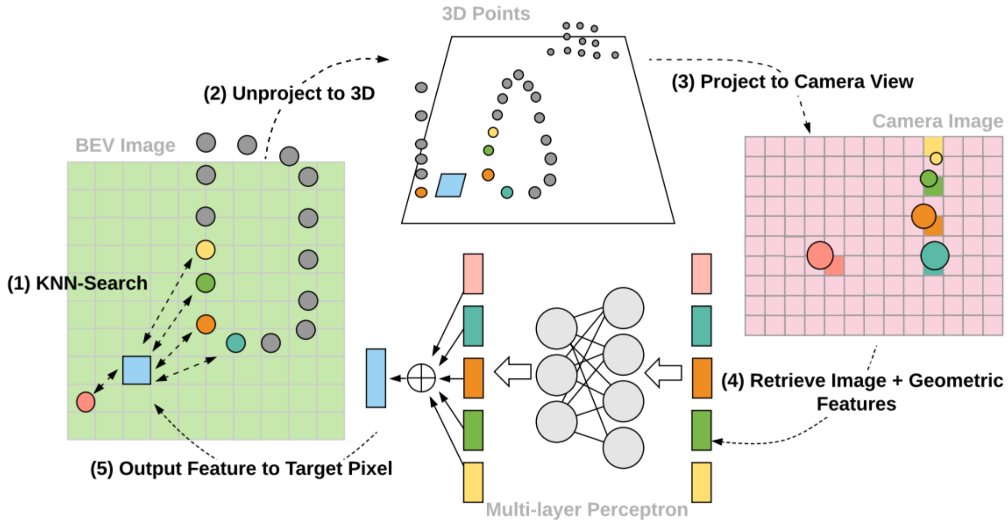


Liang et al. Deep Continuous Fusion for Multi-Sensor 3D Object Detection. ECCV 2018.

NYU

Camera-LiDAR Projection

- Unproject LiDAR points to camera view (i.e. Range View)
- Query the closest camera RGB features for each LiDAR point.

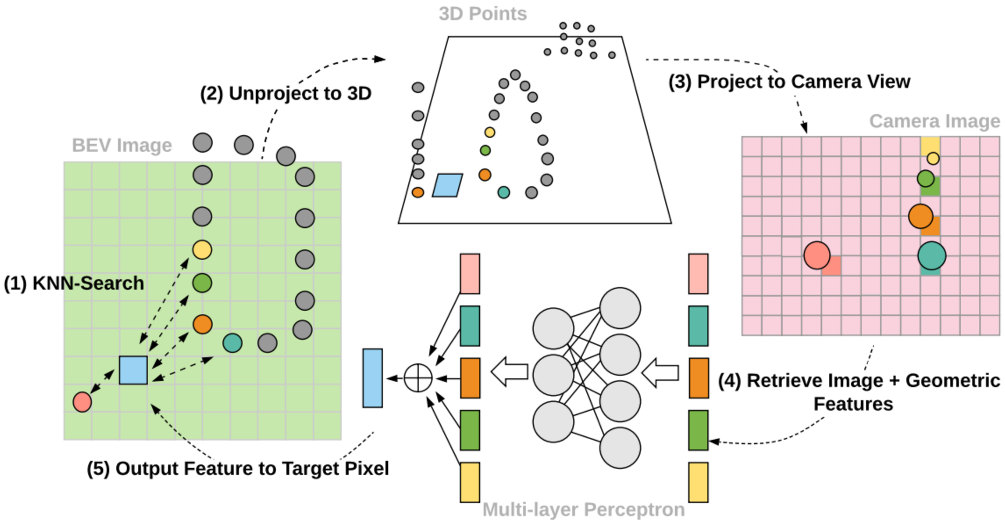


Liang et al. Deep Continuous Fusion for Multi-Sensor 3D Object Detection. ECCV 2018.

NYU

Camera-LiDAR Projection

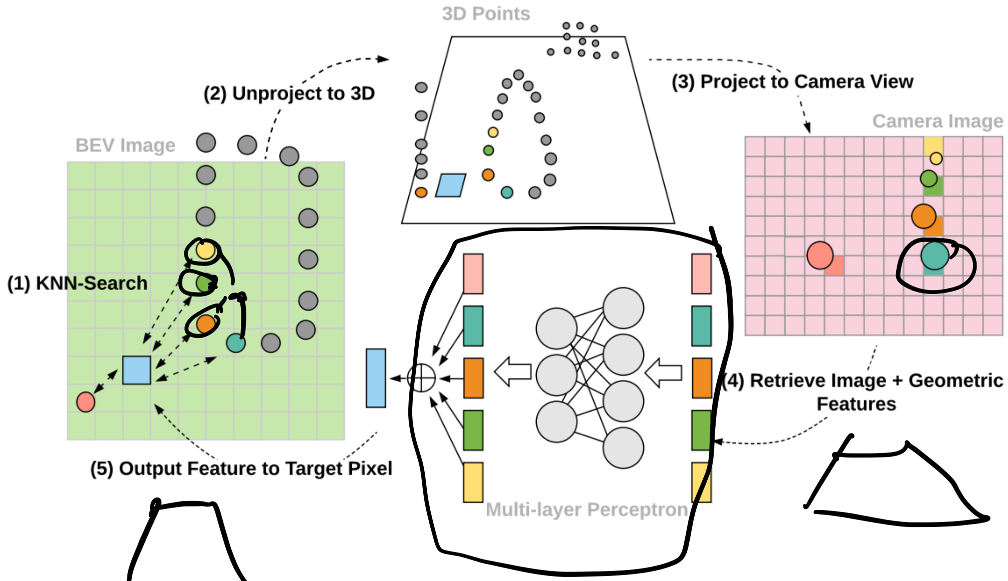
- Unproject LiDAR points to camera view (i.e. Range View)
- Query the closest camera RGB features for each LiDAR point.
- For empty space in BEV, we can interpolate from neighboring points using kNN.



Camera-LiDAR Projection

- Unproject LiDAR points to camera view (i.e. Range View)
- Query the closest camera RGB features for each LiDAR point.
- For empty space in BEV, we can interpolate from neighboring points using kNN.
- Continuous Fusion: $h_i = \sum_j MLP([f_j, x_j - x_i])$.

kernel



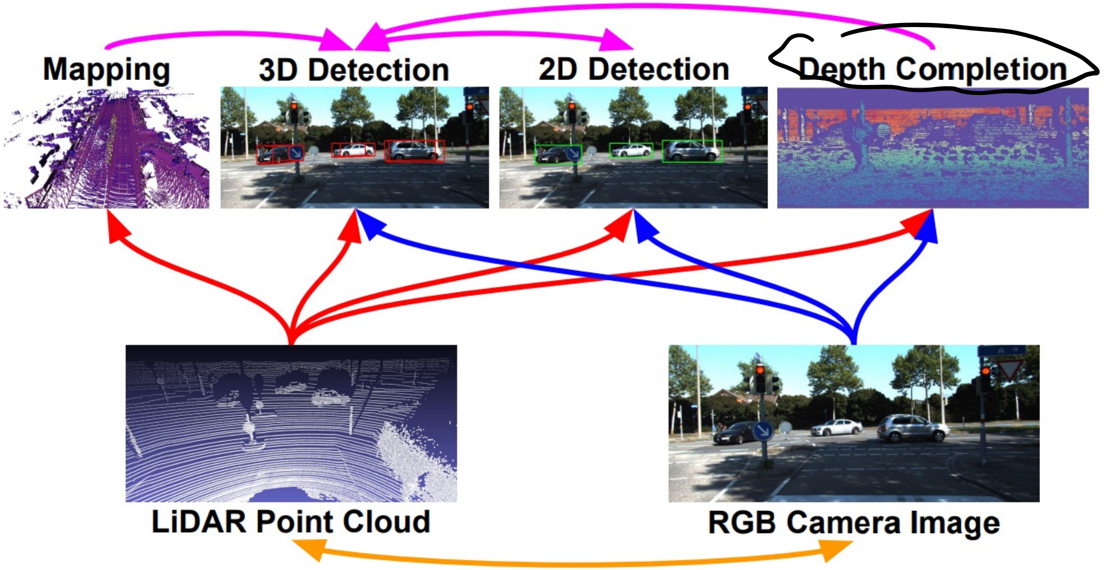
Liang et al. Deep Continuous Fusion for Multi-Sensor 3D Object Detection. ECCV 2018.

NYU

Supervised Dense Depth

- Drawback of continuous fusion: Sparse LiDAR can cause the fusion process to be less accurate. Relies on kNN.

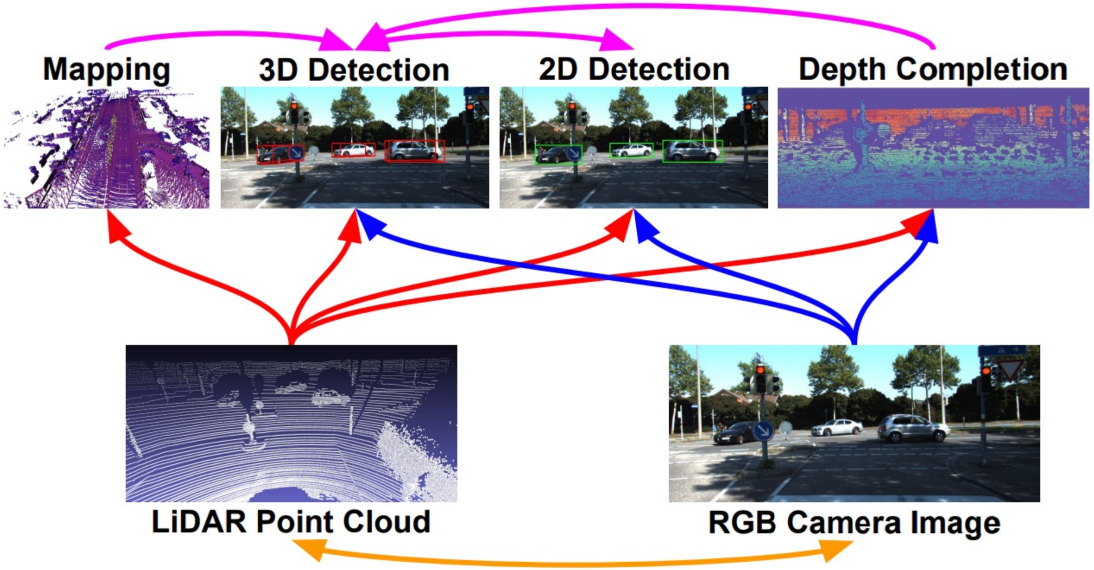
auxiliary



Liang et al. Multi-Task Multi-Sensor Fusion for 3D Object Detection. CVPR 2019.

Supervised Dense Depth

- Drawback of continuous fusion: Sparse LiDAR can cause the fusion process to be less accurate. Relies on kNN.
- Why not predict a dense depth to pair with the camera image?

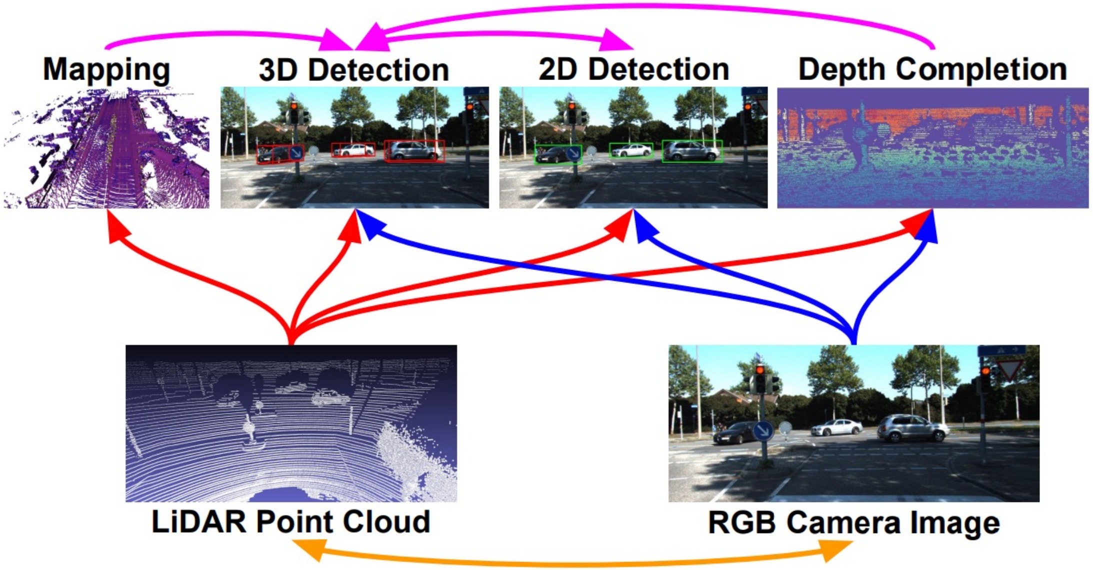


Liang et al. Multi-Task Multi-Sensor Fusion for 3D Object Detection. CVPR 2019.

NYU

Supervised Dense Depth

- Drawback of continuous fusion: Sparse LiDAR can cause the fusion process to be less accurate. Relies on kNN.
- Why not predict a dense depth to pair with the camera image?
- Depth completion module is supervised by sparse LiDAR and is used for dense fusion.



Liang et al. Multi-Task Multi-Sensor Fusion for 3D Object Detection. CVPR 2019.

NYU

3D Perception

- With the ease of use of automatic differentiation libraries, we can compose a computation graph in millions of ways.

3D Perception

- With the ease of use of automatic differentiation libraries, we can compose a computation graph in millions of ways.
- We can design layers and operators to accomodate different types of inputs and outputs. 3D, point cloud, sparse data, etc.

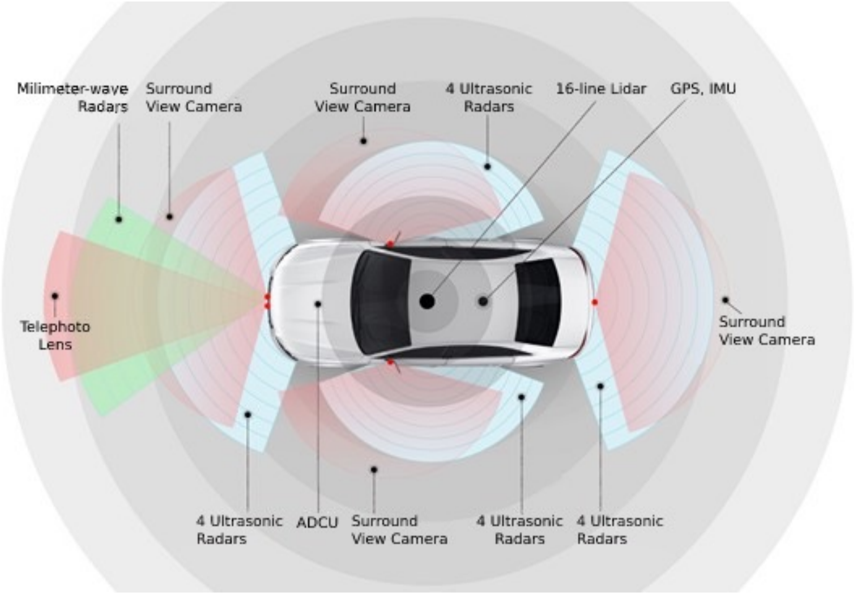
3D Perception

- With the ease of use of automatic differentiation libraries, we can compose a computation graph in millions of ways.
- We can design layers and operators to accomodate different types of inputs and outputs. 3D, point cloud, sparse data, etc.
- We can fuse different modalities together too, by leveraging geometric relationships.

NYU

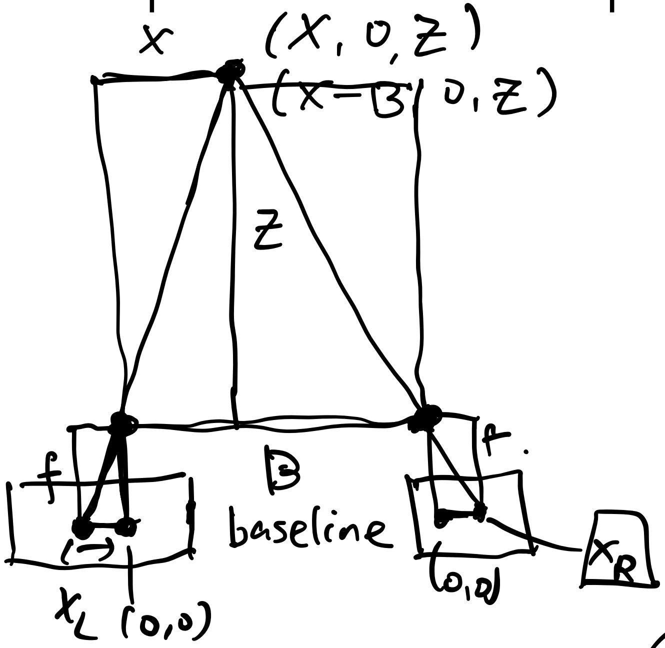
2D to 3D

- Not all embodied agents have the luxury to have a full set of sensors.
- Can we infer the geometric structure with 2D perception?



Classic Vision on Depth and Disparity

- One source of depth is from the displacement of pixels in a stereo setup.



$$\delta = x_L - x_R.$$

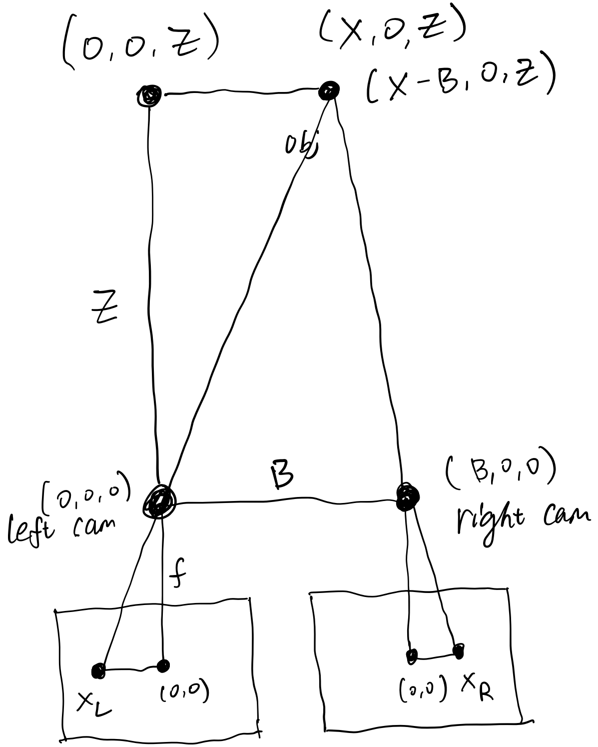
$$x_L = \frac{Xf}{Z}$$

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

$$\delta = x_L - x_R = \frac{Bf}{Z}$$

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.



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

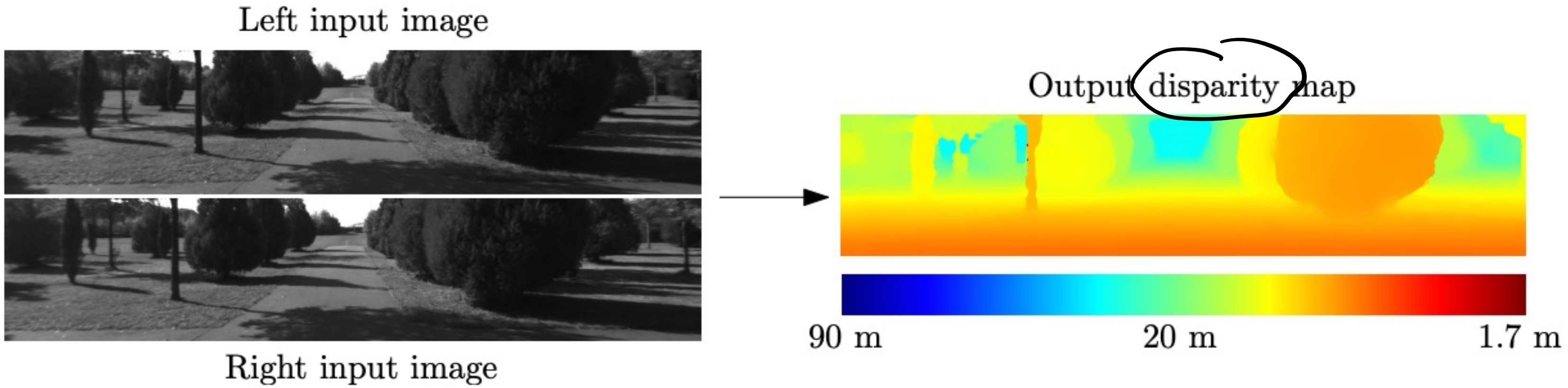
$$x_R = \frac{(X-B)f}{Z}$$

$$\delta = x_L - x_R = \frac{[X-(X-B)]f}{Z} = \frac{Bf}{Z}$$

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

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.

NYU

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

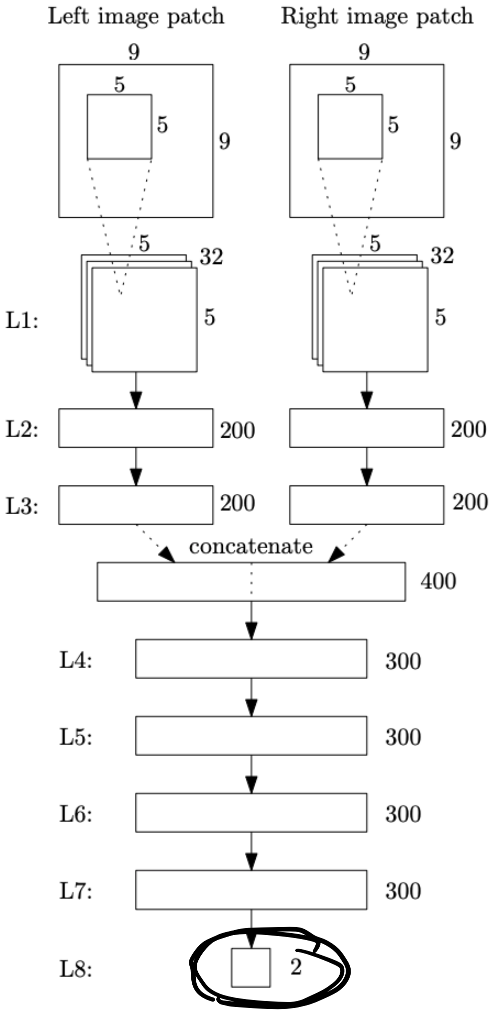
Energy
$$E(D) = \sum_{\mathbf{p}} \Big(C^4_{\text{CBCA}}(\mathbf{p}, D(\mathbf{p})) + \sum_{\mathbf{q} \in \mathcal{N}_\mathbf{p}} P_1 \times 1\{|D(\mathbf{p}) - D(\mathbf{q})| = 1\} + \sum_{\mathbf{q} \in \mathcal{N}_\mathbf{p}} P_2 \times 1\{|D(\mathbf{p}) - D(\mathbf{q})| > 1\} \Big),$$

Smoothness

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

d^*

displacement prediction



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

NYU

Self-Supervised Depth

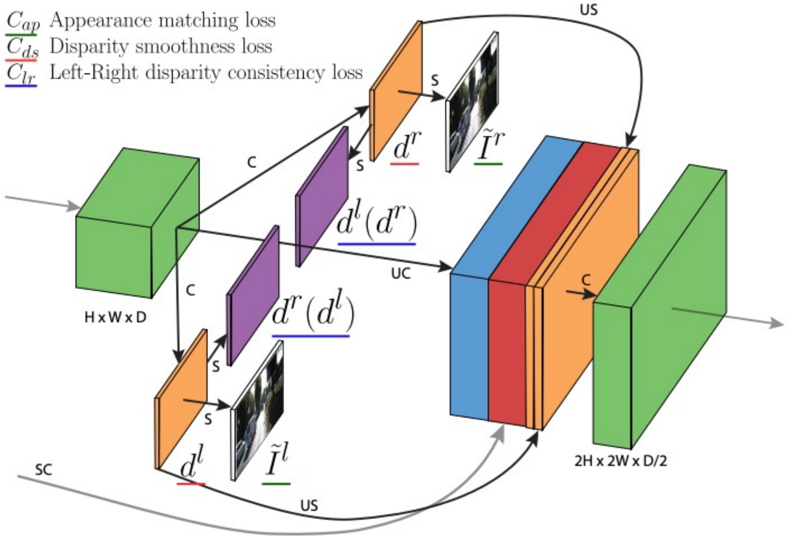
- Appearance matching loss

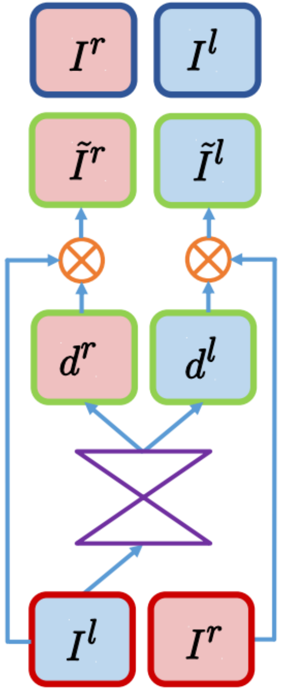
$$C_{ap}^{l} = \frac{1}{N} \sum_{i,j} \alpha \frac{1 - \mathrm{SSIM}(I_{ij}^{l}, \tilde{I}_{ij}^{l})}{2} + (1-\alpha) \left\| I_{ij}^{l} - \tilde{I}_{ij}^{l} \right\|.$$

warped.

SSIM.

pixel





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.

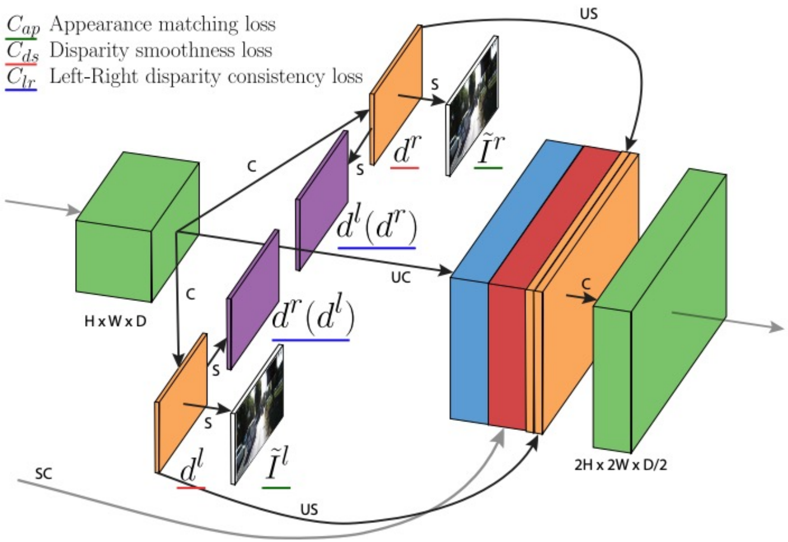
Self-Supervised Depth

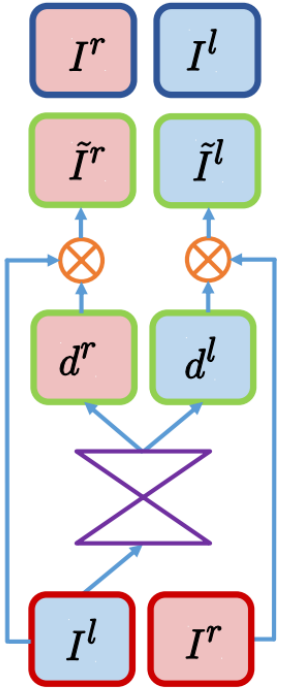
- Appearance matching loss

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

- Disparity smoothness loss

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





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

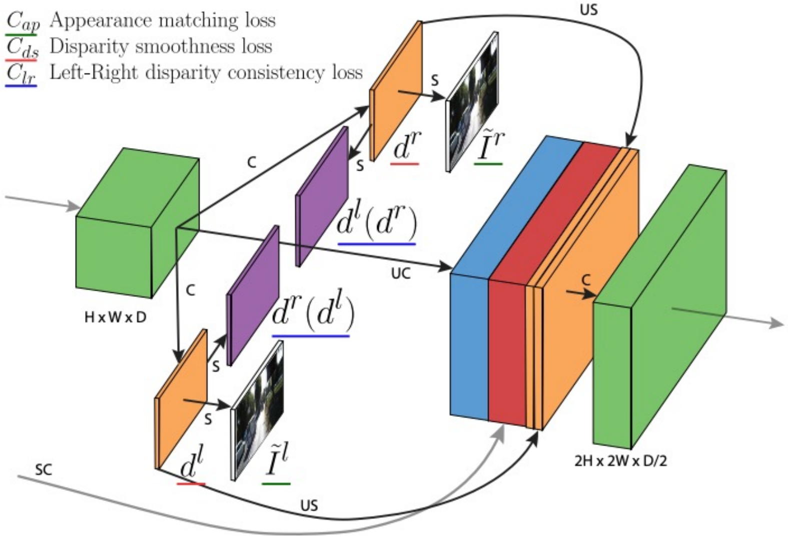
$$C_{ap}^{l}=\frac{1}{N}\sum_{i,j}\alpha\frac{1-\mathrm{SSIM}(I_{ij}^{l},\tilde{I}_{ij}^{l})}{2}+(1-\alpha)\left\|I_{ij}^{l}-\tilde{I}_{ij}^{l}\right\|.$$

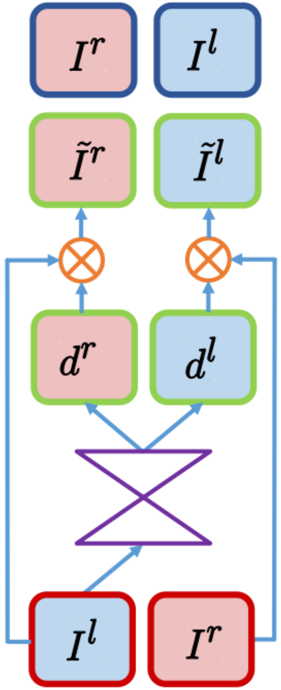
- Disparity smoothness loss

$$C_{ds}^{l}=\frac{1}{N}\sum_{i,j}\left|\partial_x d_{ij}^{l}\right|e^{-\left\|\partial_x I_{ij}^{l}\right\|}+\left|\partial_y d_{ij}^{l}\right|e^{-\left\|\partial_y I_{ij}^{l}\right\|}.$$

- Left-right disparity consistency loss

$$C_{lr}^{l}=\frac{1}{N}\sum_{i,j}\left|d_{ij}^{l}-d_{ij+d_{ij}^{l}}^{r}\right|.$$

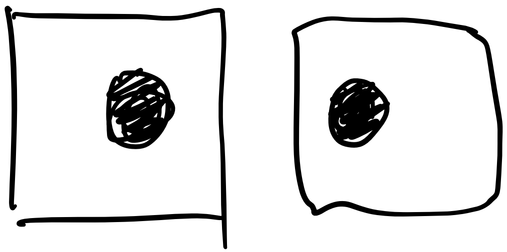




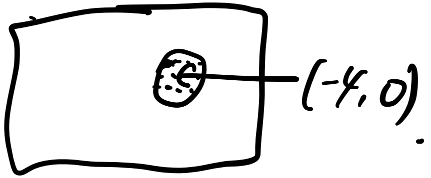
Godard et al. Unsupervised Monocular Depth Estimation with Left-Right Consistency. CVPR 2017.
Wang et al. Image Quality Assessment: From Error Visibility to Structural Similarity. TIP 2004.

NYU

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.

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

Motion, Optical Flow

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

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

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

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.

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

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

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

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

Motion, Optical Flow

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

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

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

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

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

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

x direction y direction

NYU

Classical Approach

- Under-constrained system

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

Classical Approach

2 unk.

- Under-constrained system

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

- Use a local patch and assume smooth motion

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

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

3×3.

unknown.

NYU

Classical Approach

- Under-constrained system

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

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

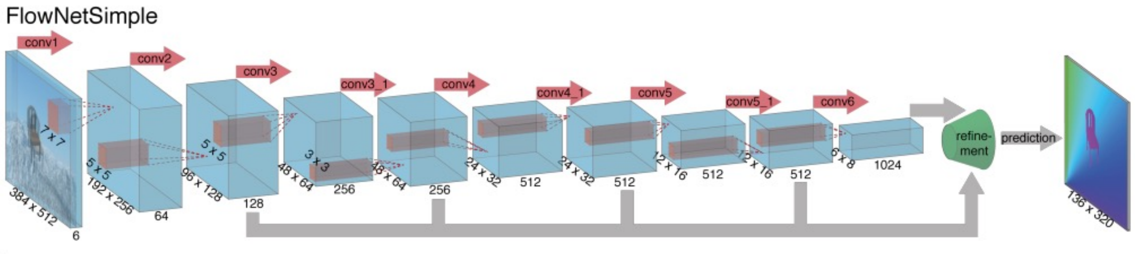
$$\mathbf{A}\mathbf{u} = \mathbf{b}$$

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

NYU

Correlation Volume Approach

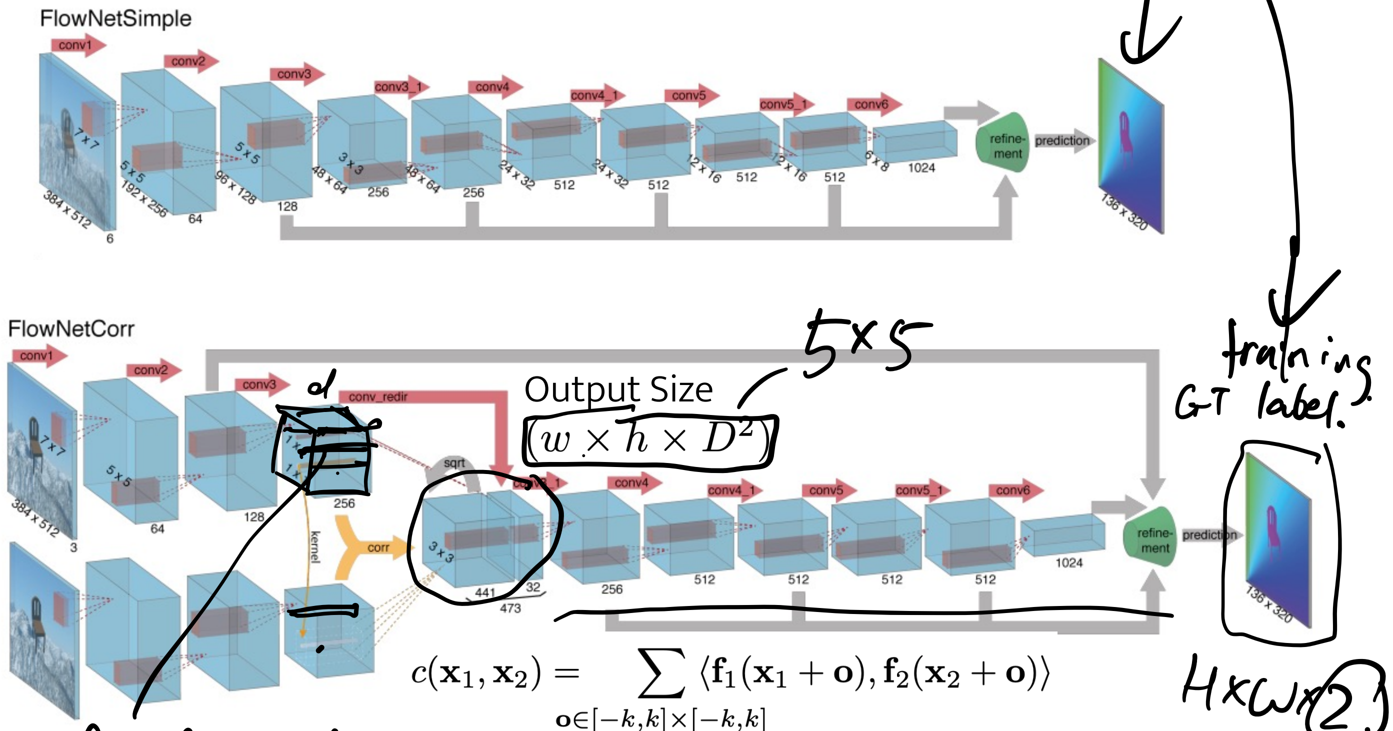
- Simple Approach: Concatenate the two images together.



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.



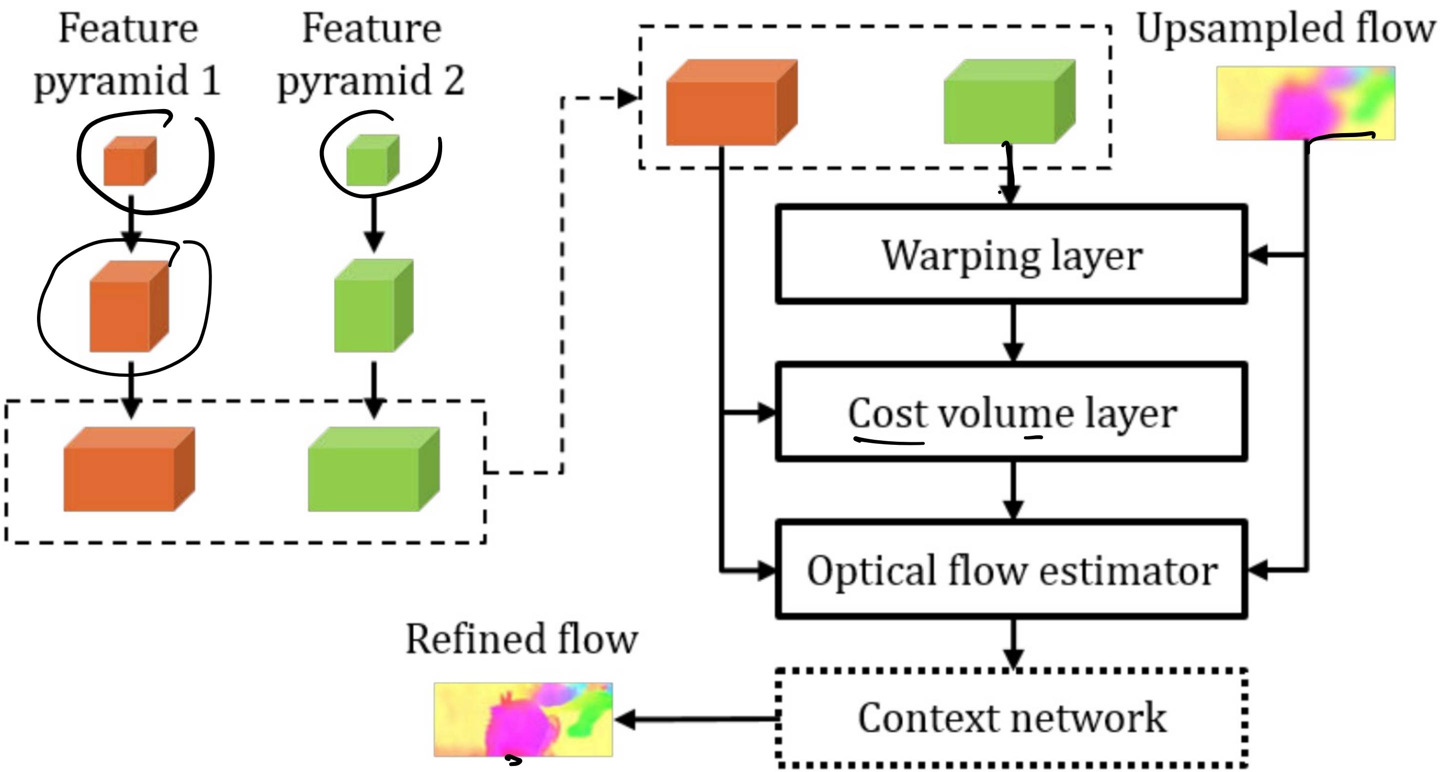
Output Size $(w \times h \times D^2)$

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

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

NYU

Iterative Refining through Feature Pyramid



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

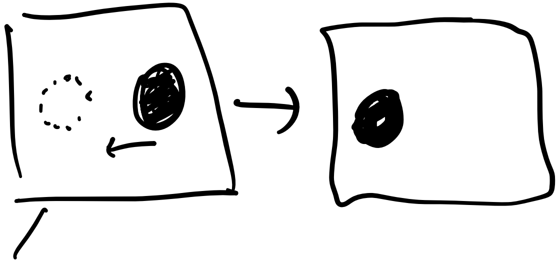
Unsupervised Flow

- Photometric Consistency (Appearance)

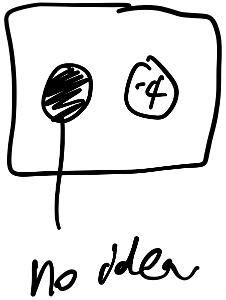
Jonschkowski et al. What Matters in Unsupervised Optical Flow. ECCV 2020

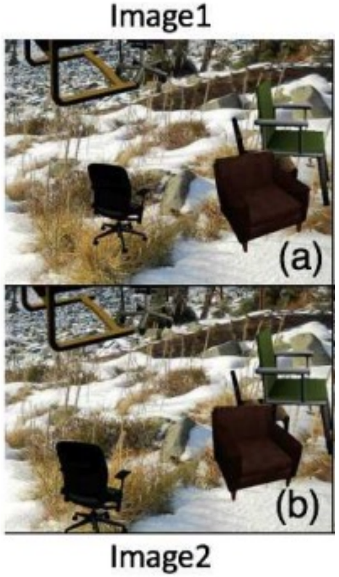
NYU

Unsupervised Flow



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



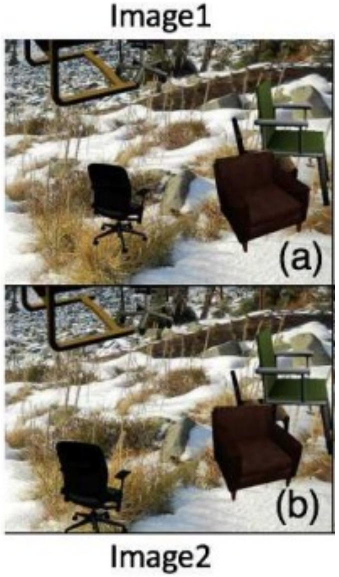


Wang et al., 2018

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

Unsupervised Flow

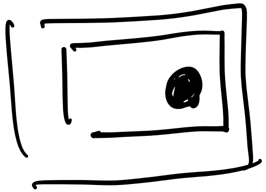
- Photometric Consistency (Appearance)
- Occlusion Estimation
  - Forward-backward consistency
- Smoothness



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)

Image1

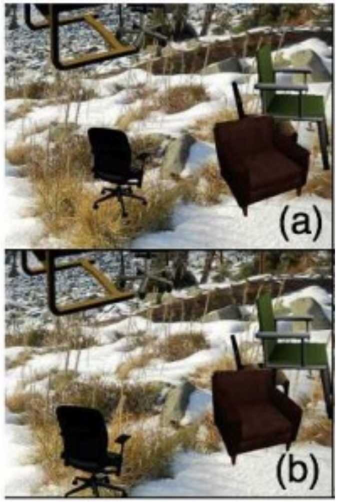


Image2

Wang et al., 2018

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

NYU

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?

Image1

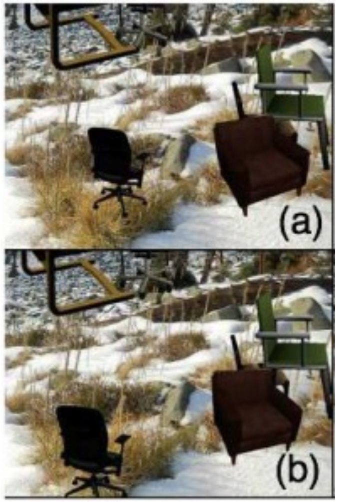


Image2

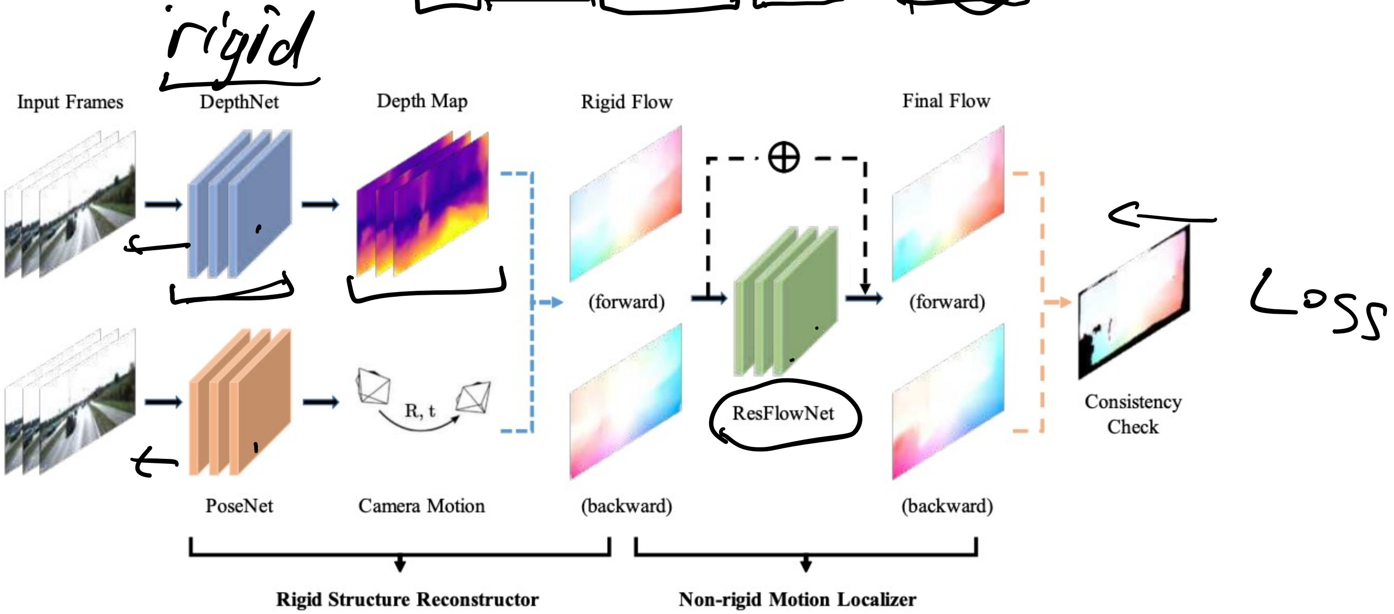
Wang et al., 2018

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.

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



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

NYU

Training Losses

- Appearance Loss (Warping):

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

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

SSIM. pixel

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:

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

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:

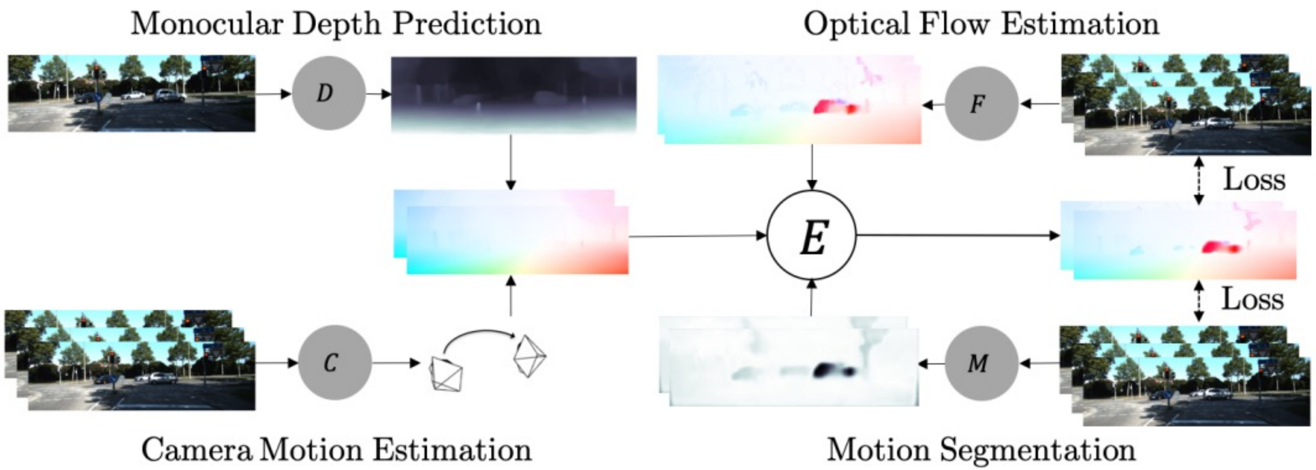
$$\mathcal{L} = \sum_{p_t} [\delta(p_t)] \cdot \|\Delta f_{t \mapsto s}^{full}(p_t)\|_1.$$

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

NYU

Summary

- Leverage cross correlation structure for spatial similarity matching.

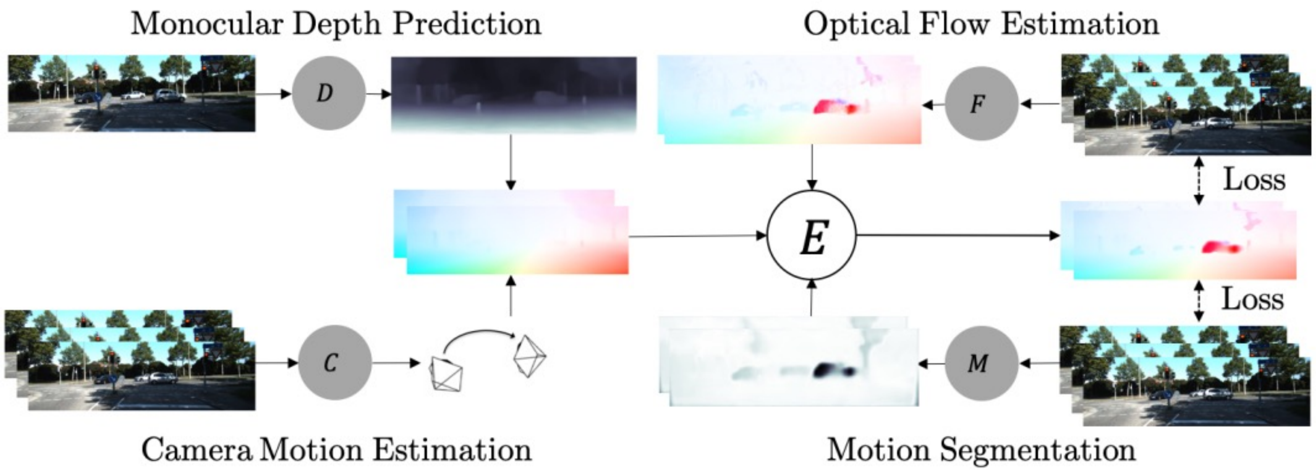


Ranjan et al. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera, Motion, Optical Flow and Motion Segmentation. CVPR 2019

NYU

Summary

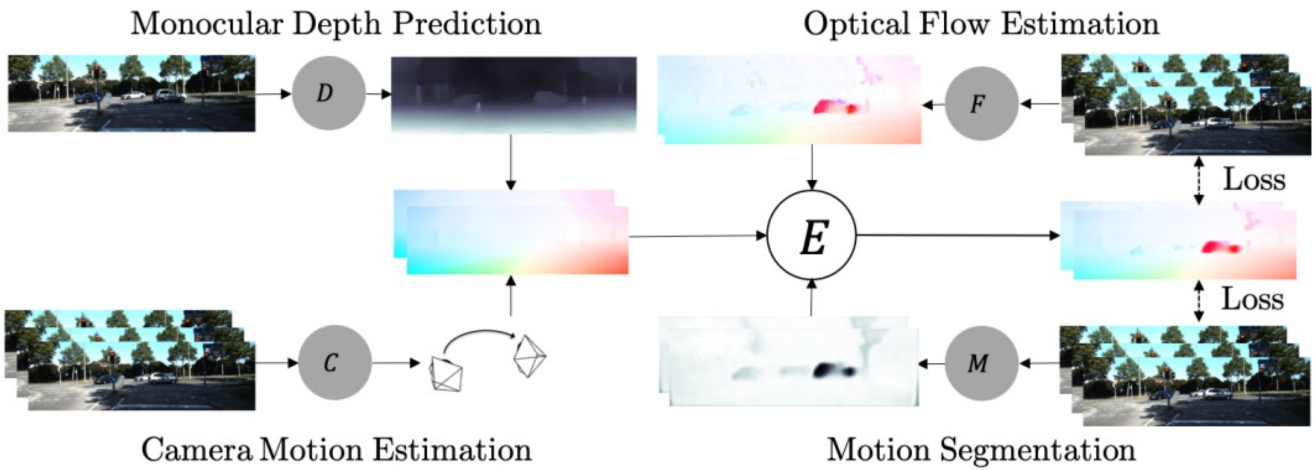
- Leverage cross correlation structure for spatial similarity matching.
- Can be used towards: depth, flow, and pose prediction.



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.



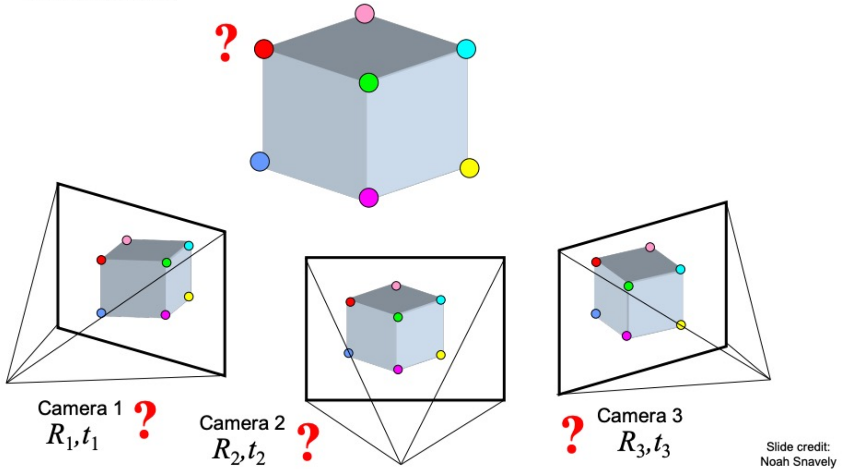
Ranjan et al. Competitive Collaboration: Joint Unsupervised Learning of Depth, Camera, Motion, Optical Flow and Motion Segmentation. CVPR 2019

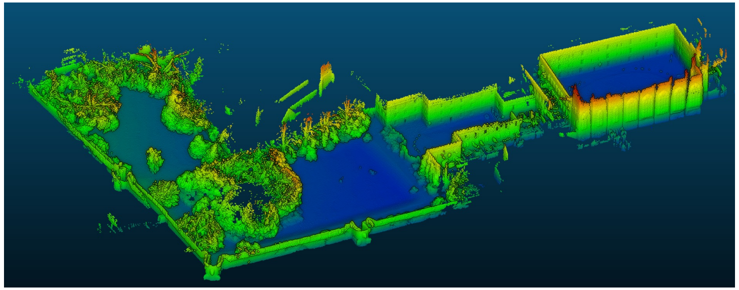
NYU

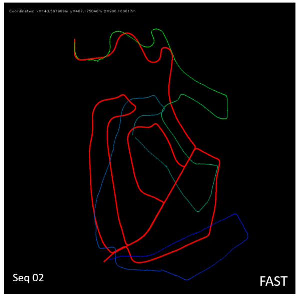
Classical Mapping

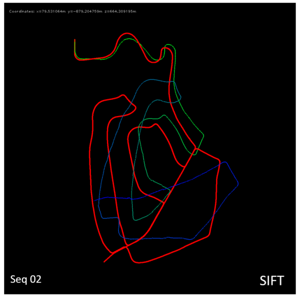
- Estimating 3D structure and location from 2D observations.

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







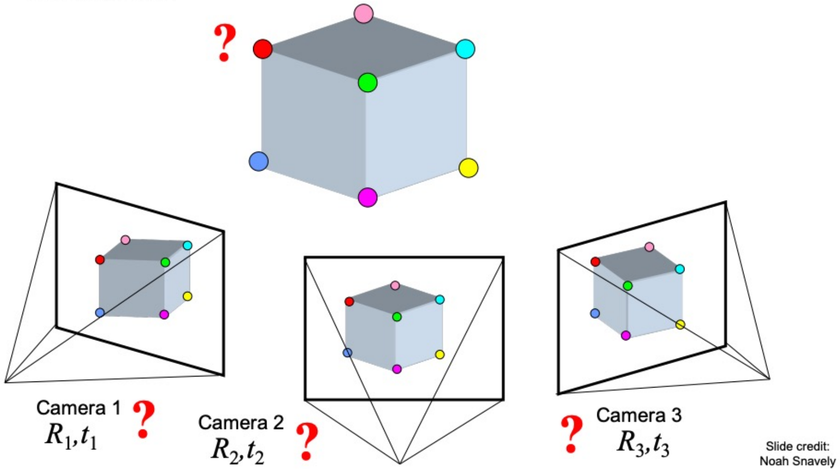


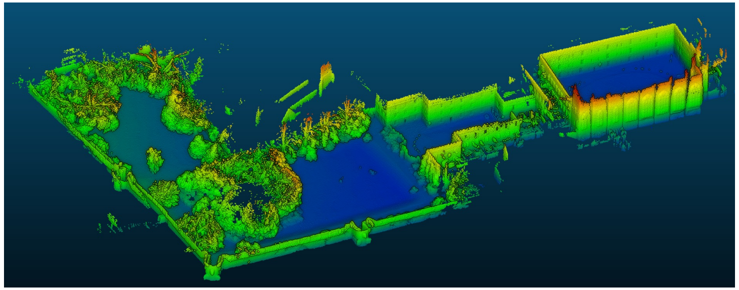
Garg & Jain

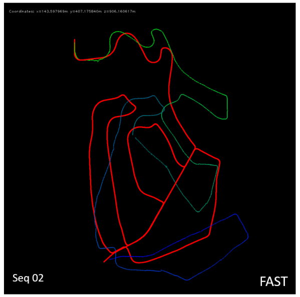
Classical Mapping

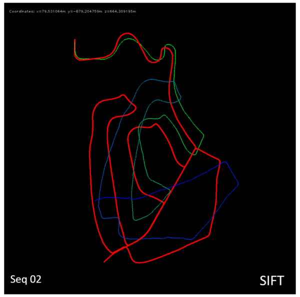
- Estimating 3D structure and location from 2D observations.
- Simultaneous Localization and Mapping.

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







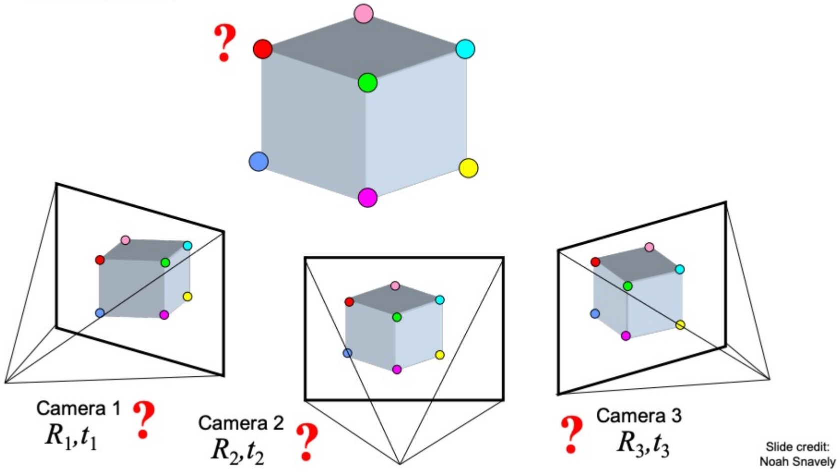


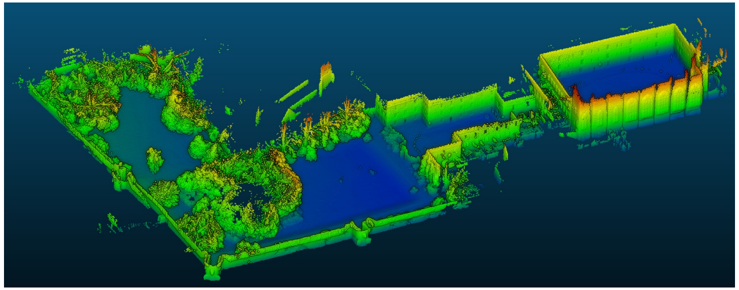
Garg & Jain

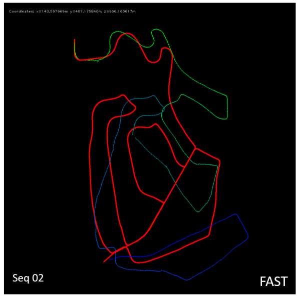
Classical Mapping

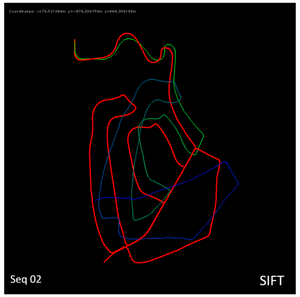
- Estimating 3D structure and location from 2D observations.
- Simultaneous Localization and Mapping.
- Common Techniques: Extended Kalman Filter, GraphSLAM

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









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

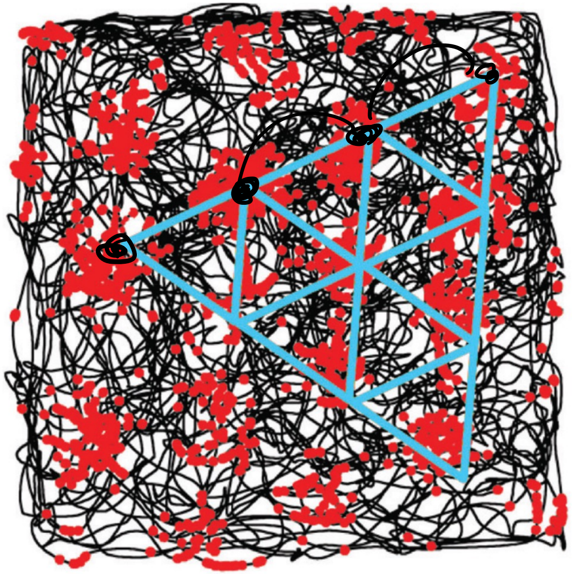
- Probabilistic inference can take long to compute, and mapping takes a large memory storage.
- Great for 3D reconstruction but downstream tasks may not need a full precision explicit map.
- May not fully understand dynamic objects (averaging across multiple scans).

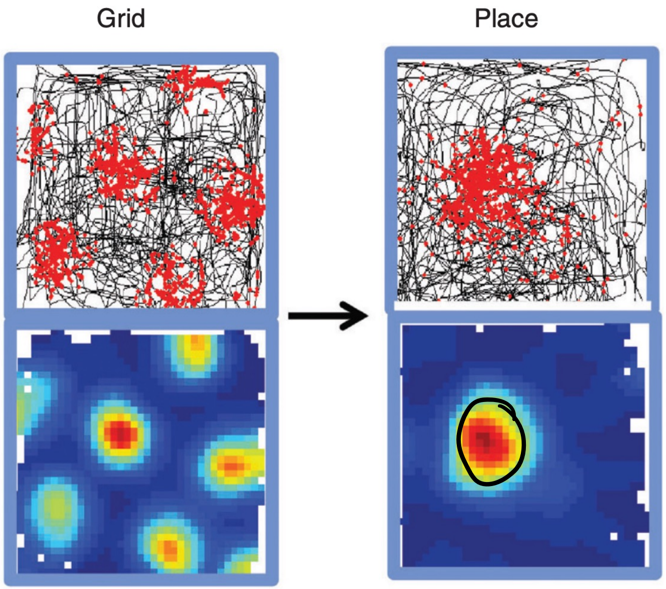
NYU

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?

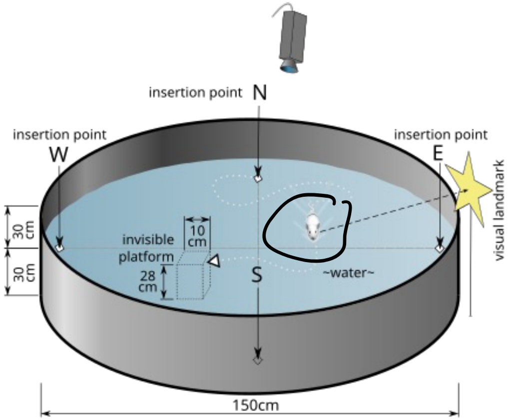
Mapping in the Brain: Grid and Place Cells



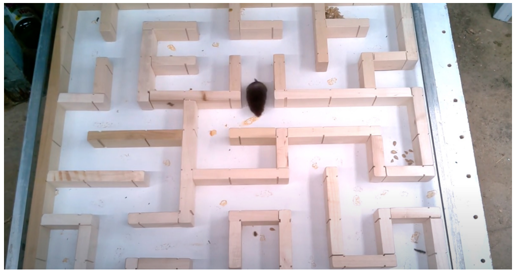


~~location.~~ localization

mapped location



Morris water maze



Matthias Wandel, 2018

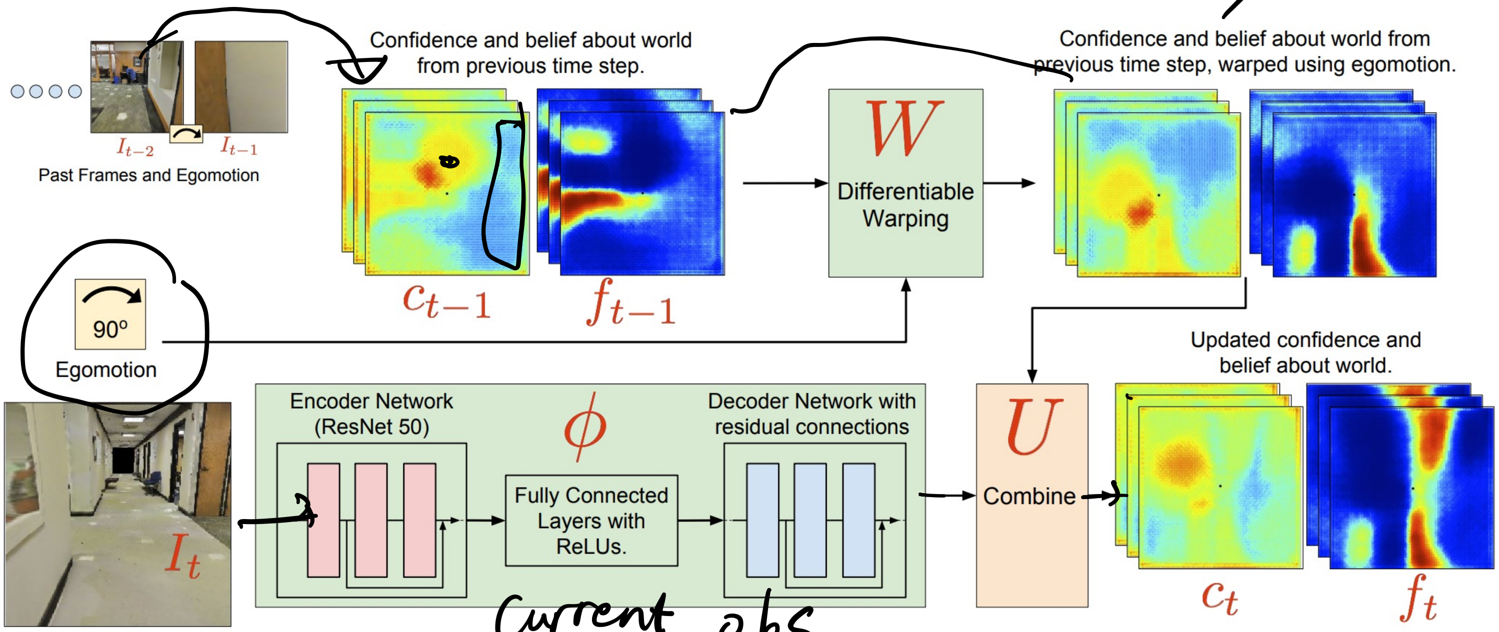
May-Britt Moser, David C. Rowland, and Edvard I. Moser. Place Cells, Grid Cells, and Memory.

NYU

Neural Mapping

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

past memory.



updated memory.

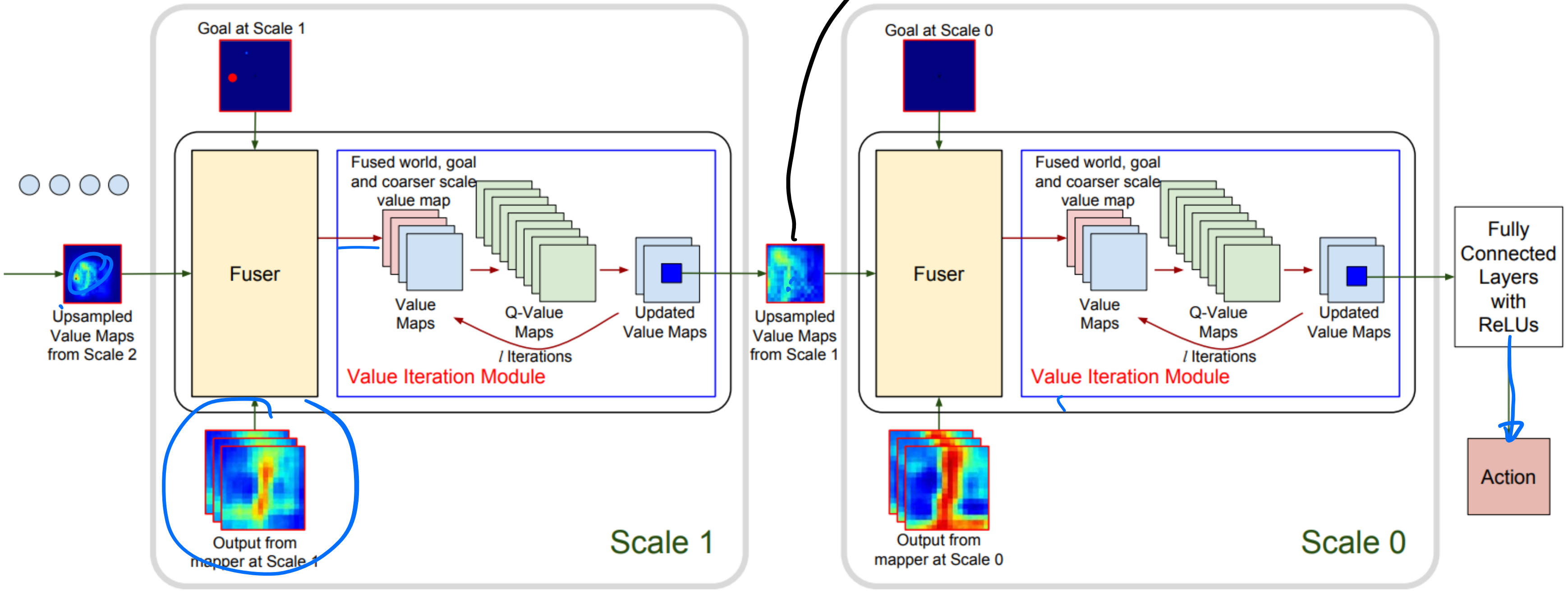
Current obs.

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

NYU

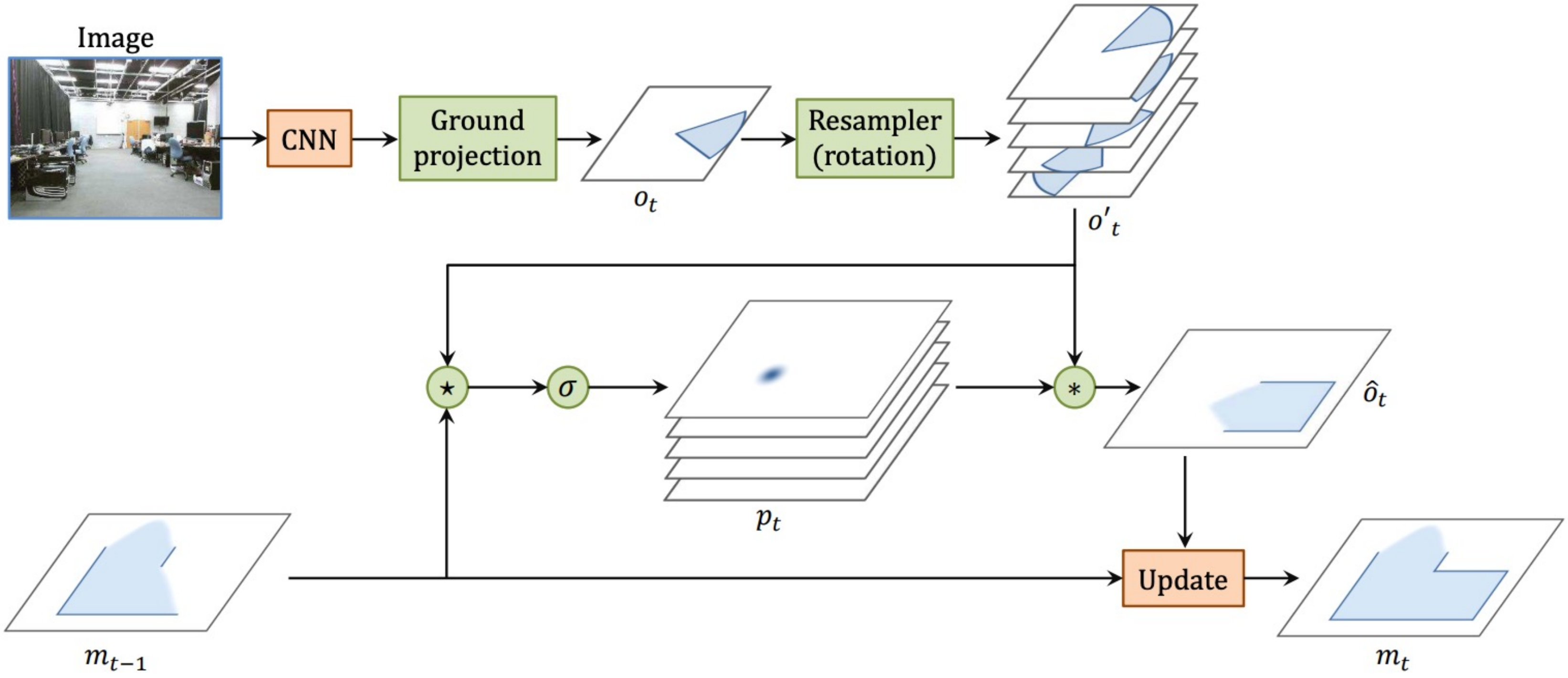
Hierarchical Planning

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



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

Simultaneous Localization and Registration



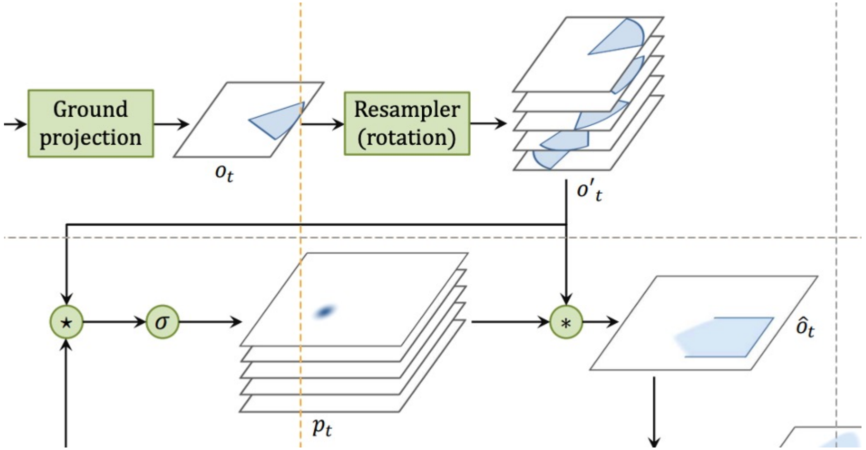
Henriques & Vedaldi. MapNet: An Allocentric Spatial Memory for Mapping Environments. CVPR 2018.

NYU

Simultaneous Localization and Registration

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

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



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

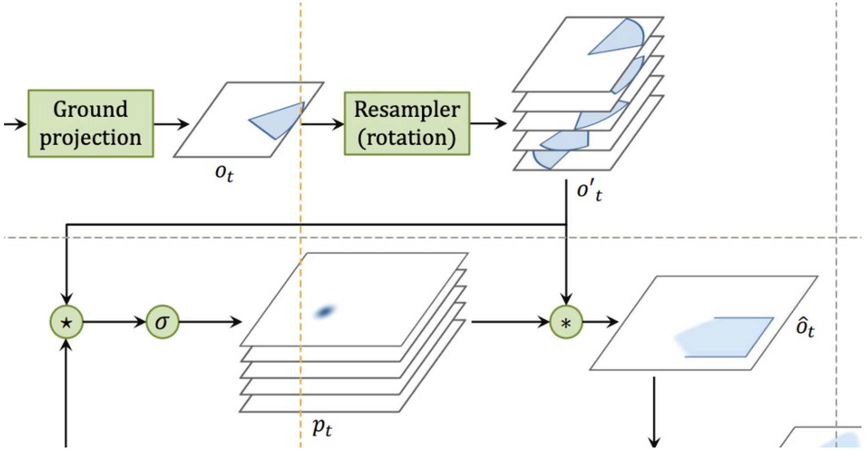
Simultaneous Localization and Registration

- 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.

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



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

NYU

Simultaneous Localization and Registration

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

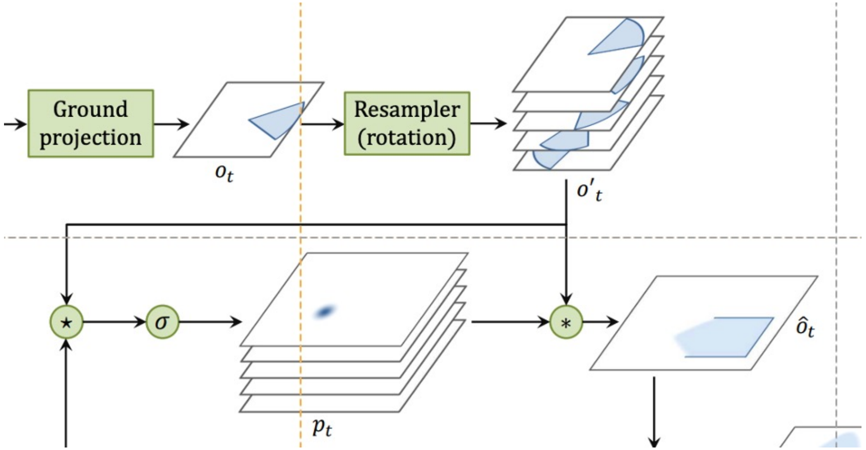
$$o'_{ijkl} = [R(o, 2\pi l/r)]_{ijk}.$$

- o'_t convolve with the base feature.

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

- Transform observations into allocentric

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



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.

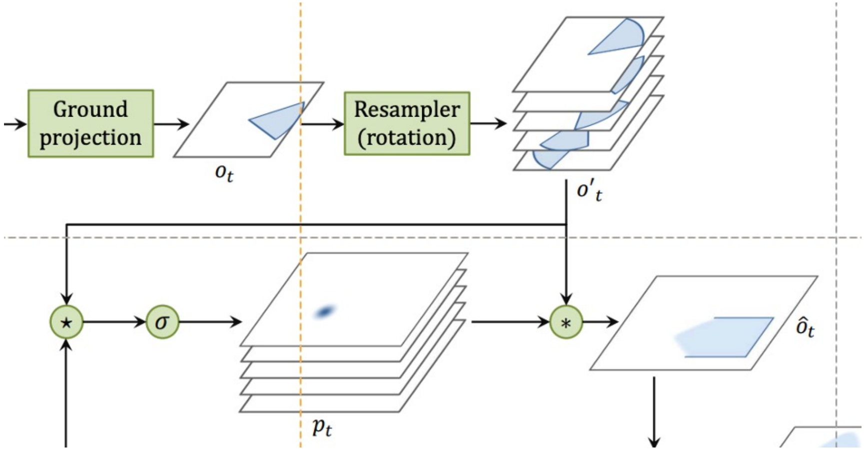
$$p_t = \text{Softmax}(m_{t-1} * o'_t).$$

- Transform observations into allocentric

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

- Update belief:

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



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.

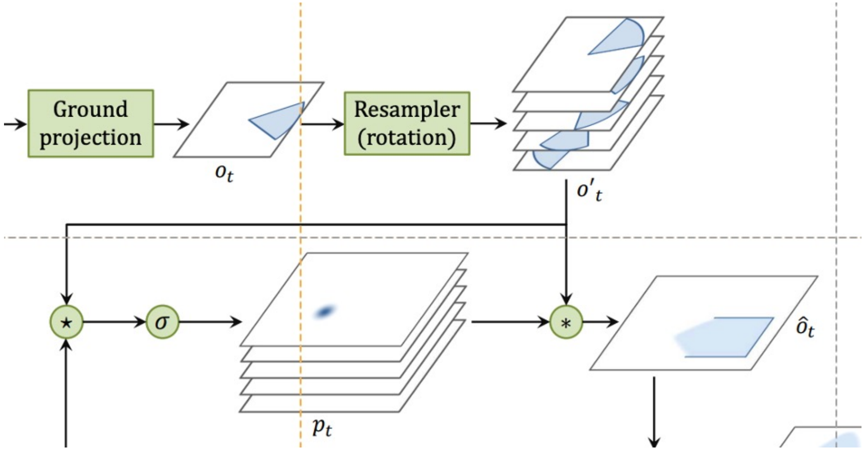
$$p_t = \mathrm{Softmax}(m_{t-1} * o'_t).$$

- Transform observations into allocentric

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

- Update belief:

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



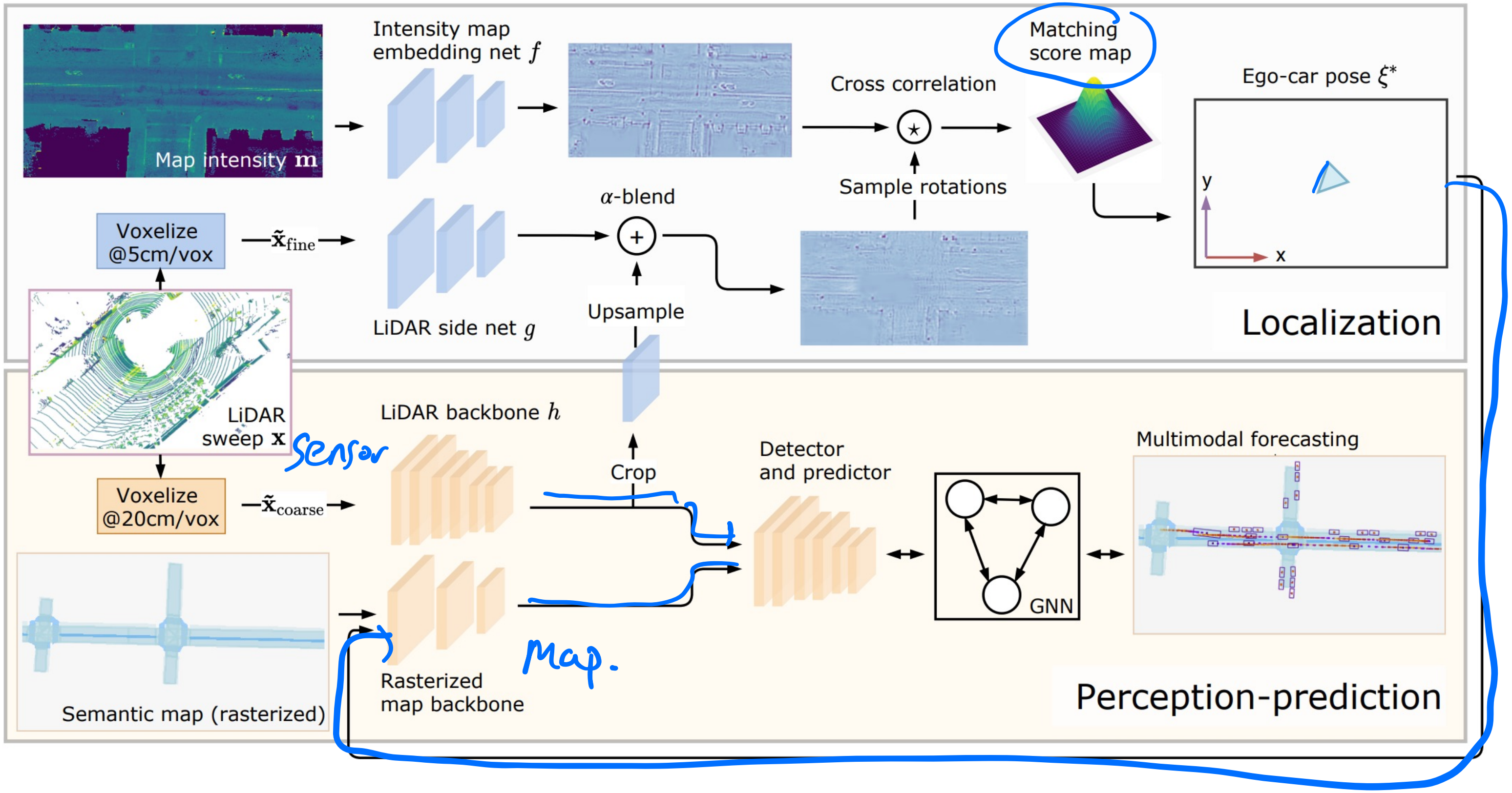
Loss:

$$\mathcal{L}(p) = -\log \sum_t p_{H_t W_t R_t t}.$$

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

NYU

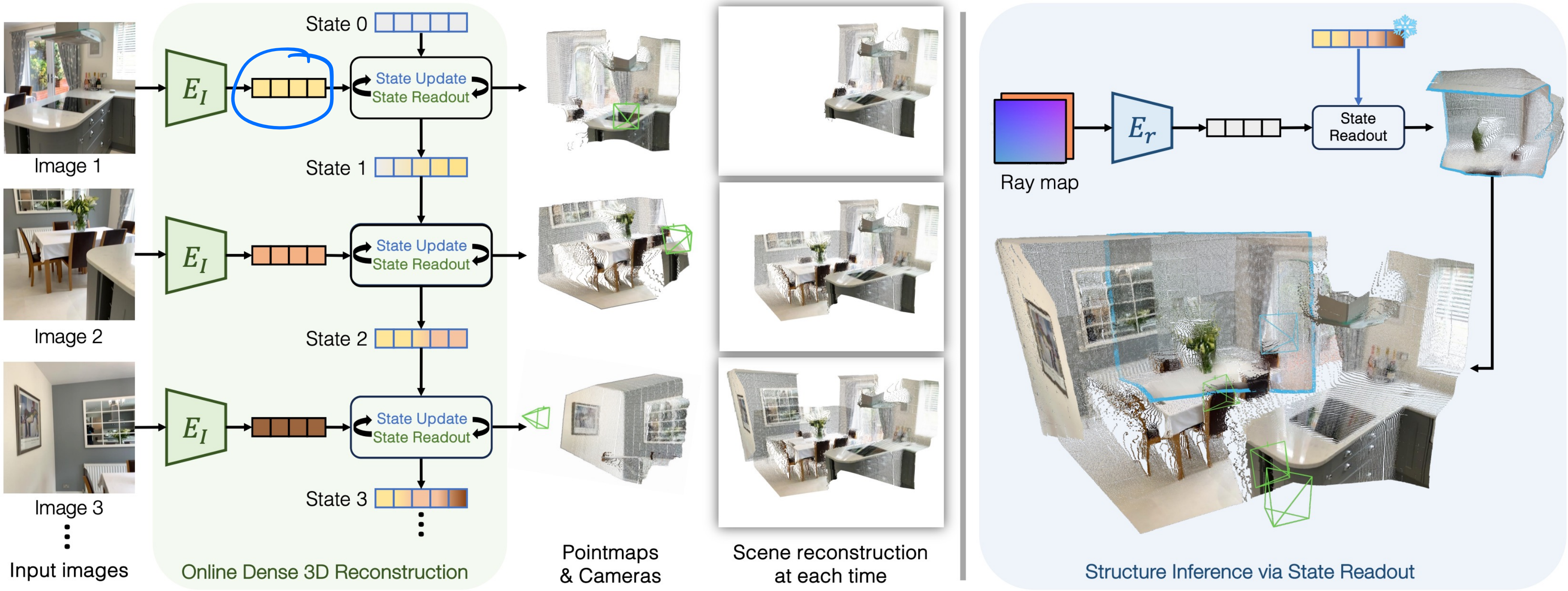
Joint Localization, Perception and Prediction



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

NYU

Continuous 3D Perception and Mapping

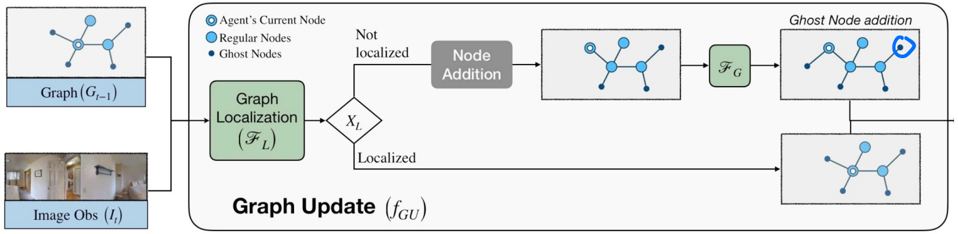


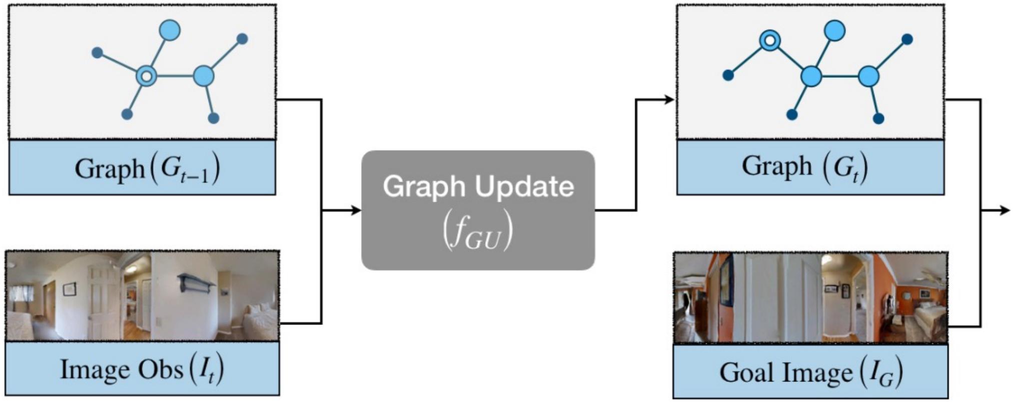
Wang et al. Continuous 3D Perception Model with Persistent State. arXiv 2025.

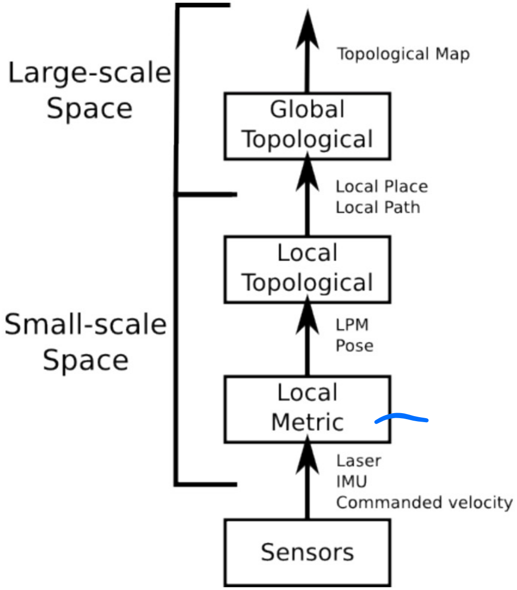
NYU

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.

NYU

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.

NYU

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.

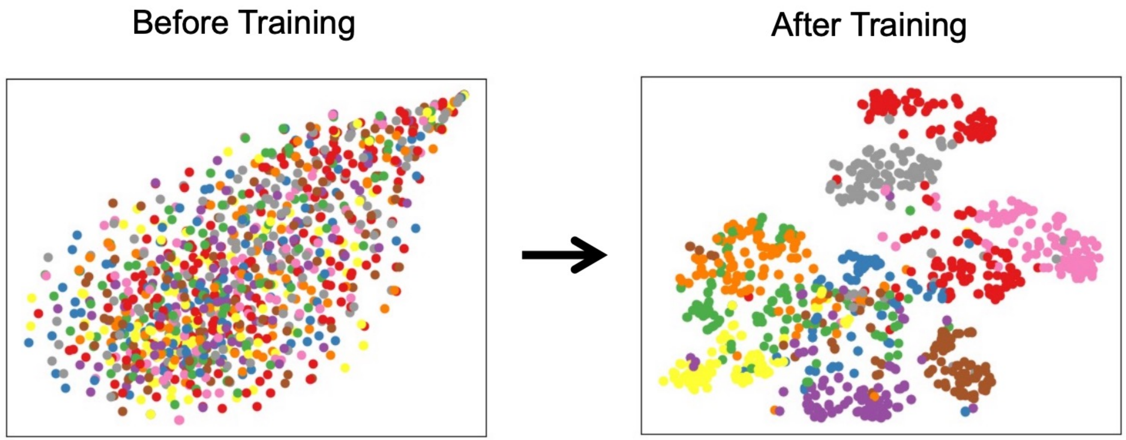
Summary

- 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).

NYU

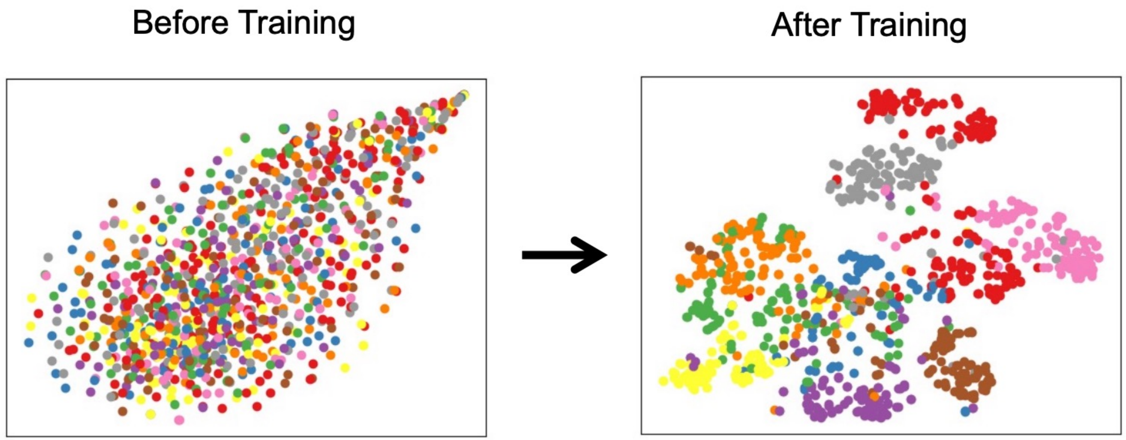
Representation Learning

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



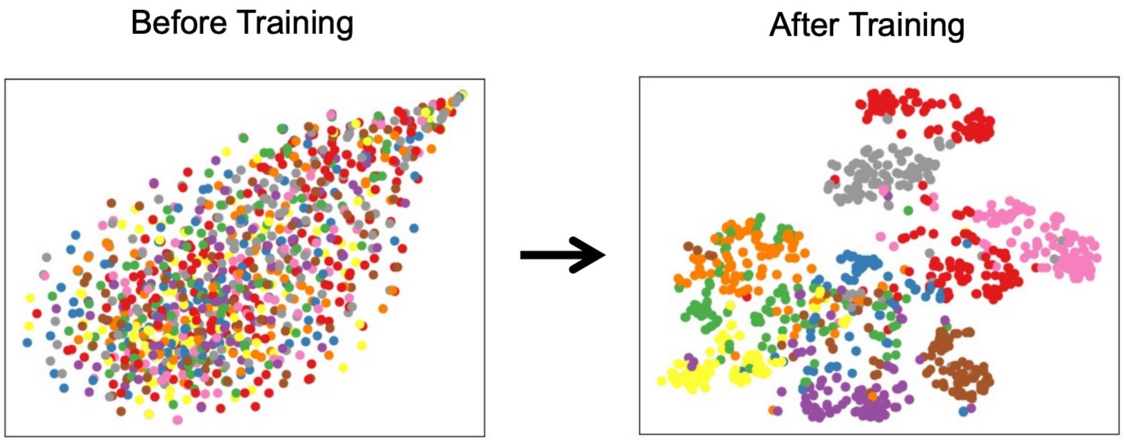
Representation Learning

- 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

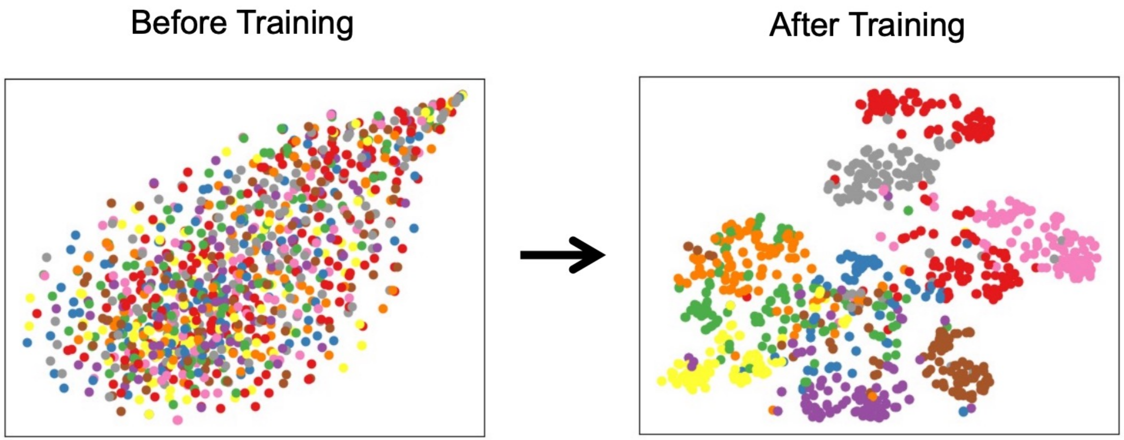
- Efficient encoding of the world that can help us recognize semantic concepts (high-level cognition).
- Efficient learning of visual data without extra supervision.
- Recognition of motion also requires global matching.



NYU

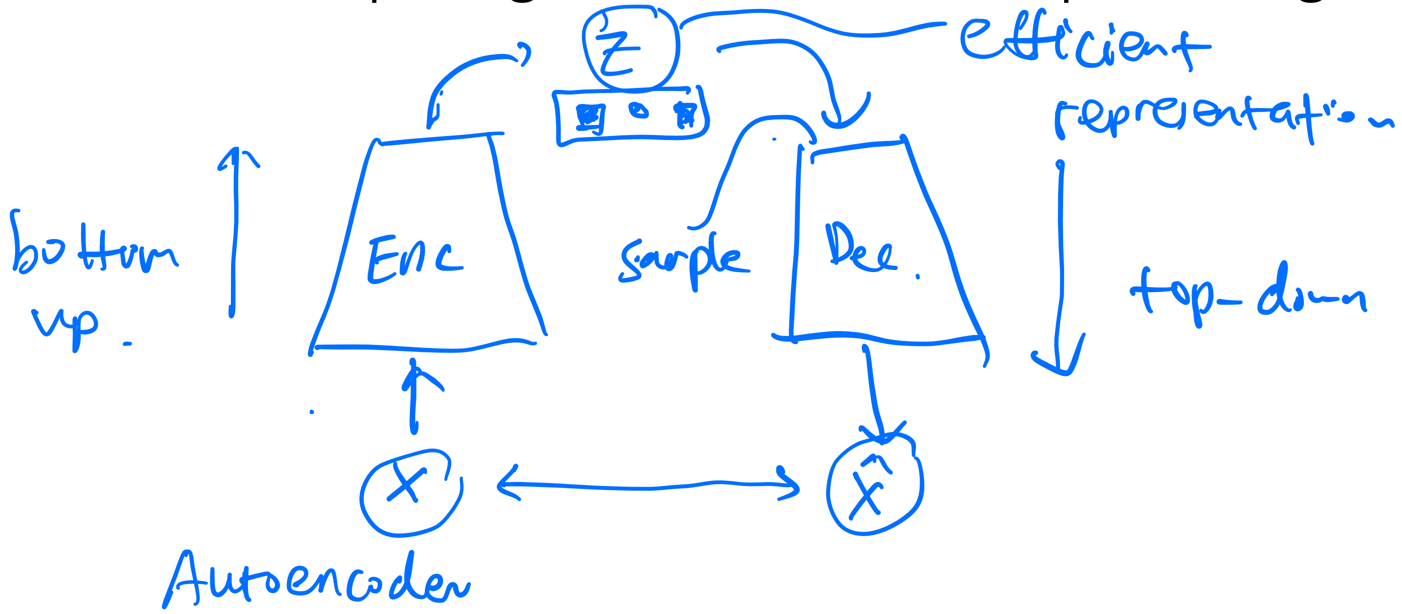
Representation Learning

- 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.



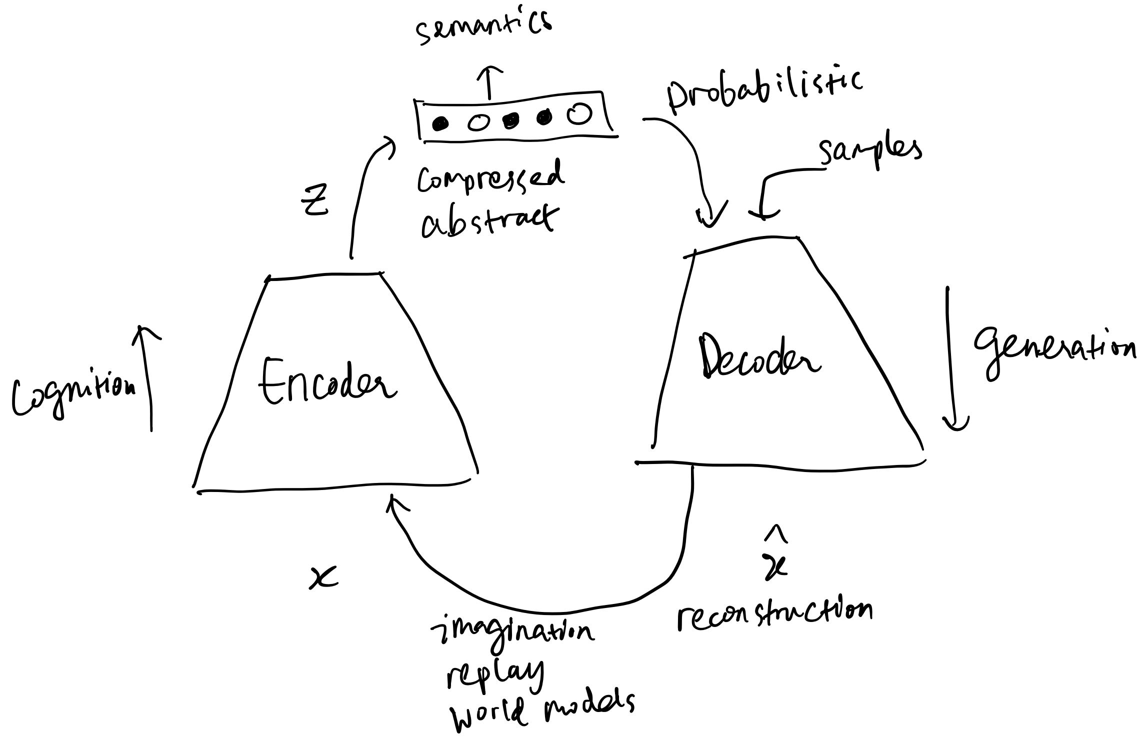
Unsupervised Learning

- 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

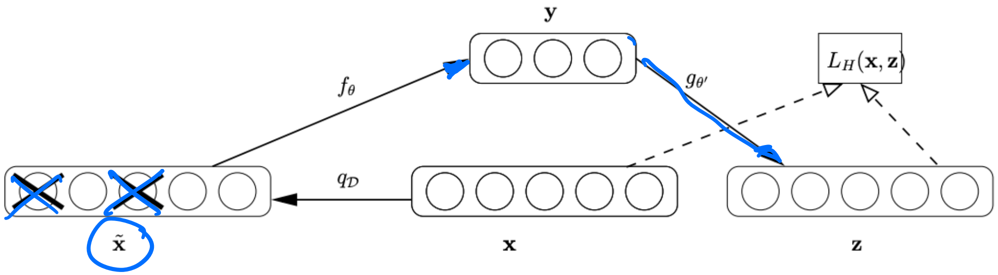
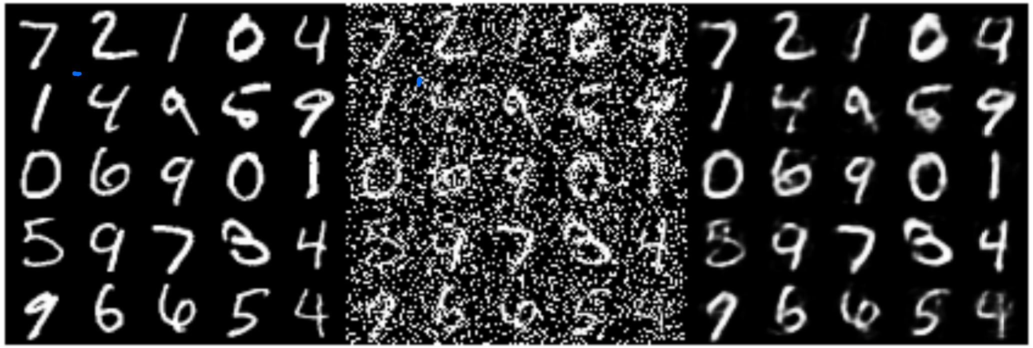


Figure 1. An example $\mathbf{x}$ is corrupted to $\tilde{\mathbf{x}}$. The autoencoder then maps it to $\mathbf{y}$ and attempts to reconstruct $\mathbf{x}$.



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: $p(x|\tilde{x}) = B_{g_{\theta'}(f_\theta)}(x)$.

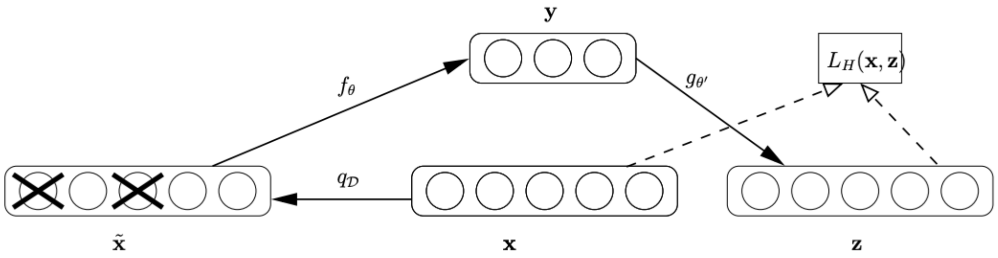
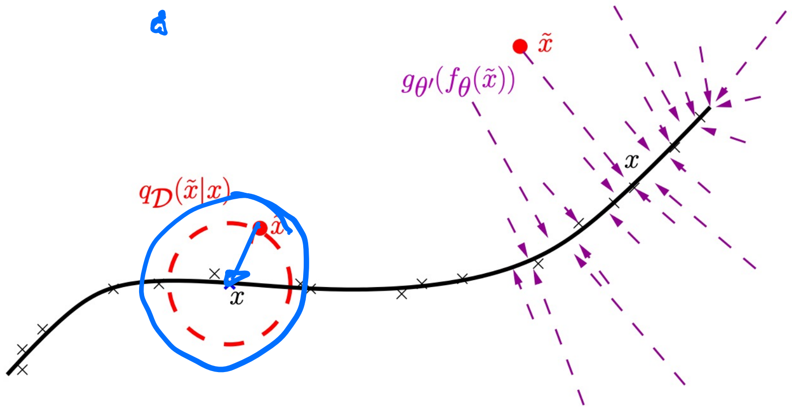
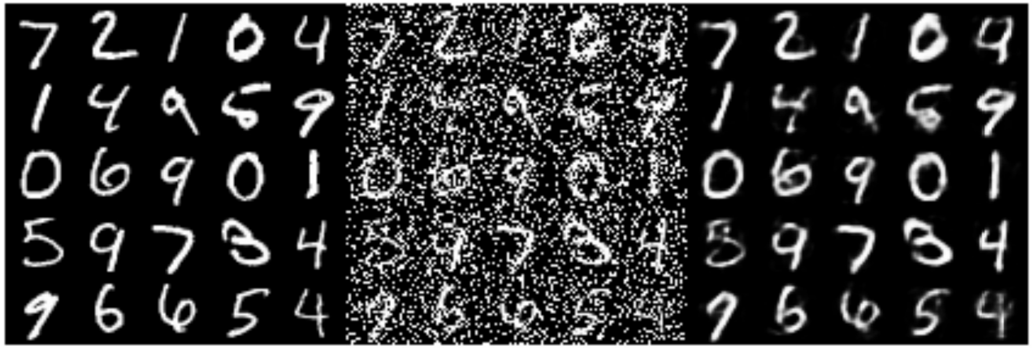


Figure 1. An example $\mathbf{x}$ is corrupted to $\tilde{\mathbf{x}}$. The autoencoder then maps it to $\mathbf{y}$ and attempts to reconstruct $\mathbf{x}$.

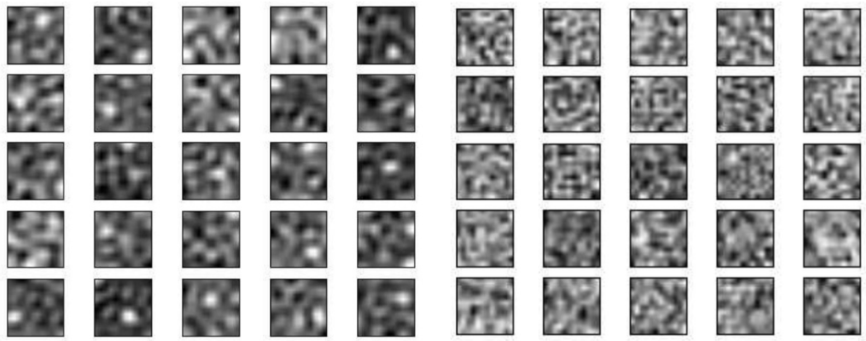


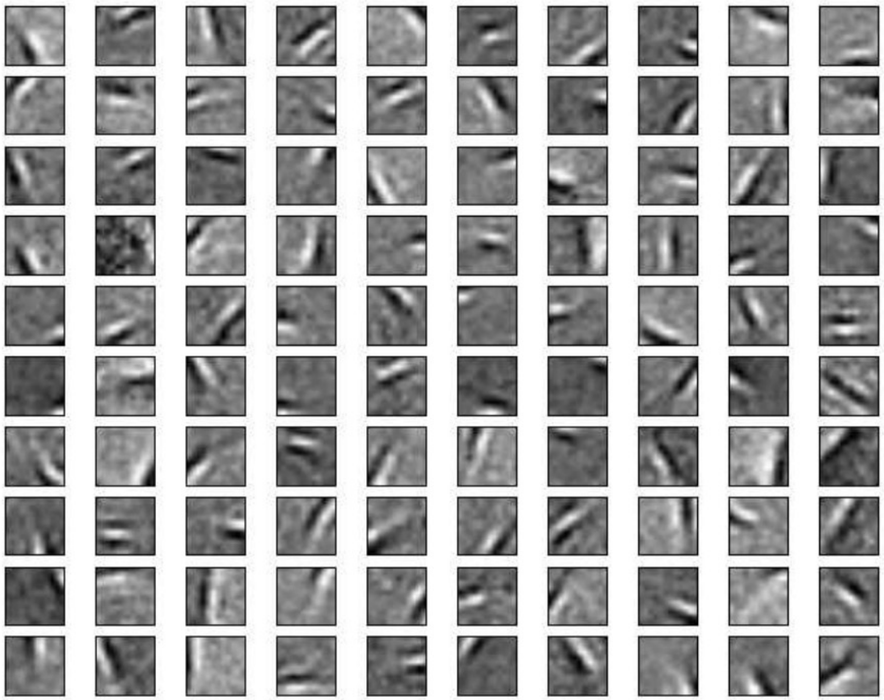


Vincent et al. Extracting and Composing Robust Features with Denoising Autoencoders. ICML 2008.

Denoising Autoencoder (DAE)

- Regular autoencoders do not learn good filters.





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.

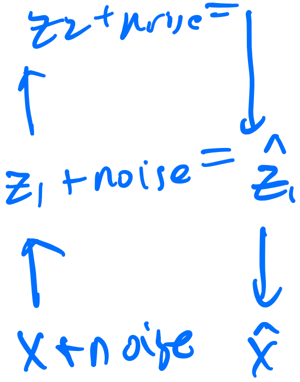
Connection to Diffusion Models

- 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.

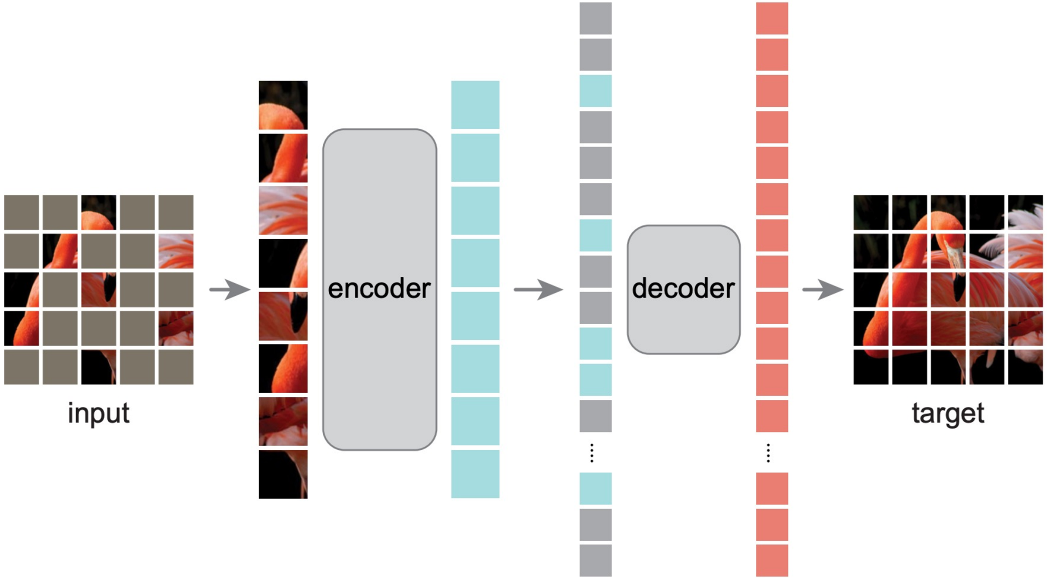
Connection to Diffusion Models



- 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.

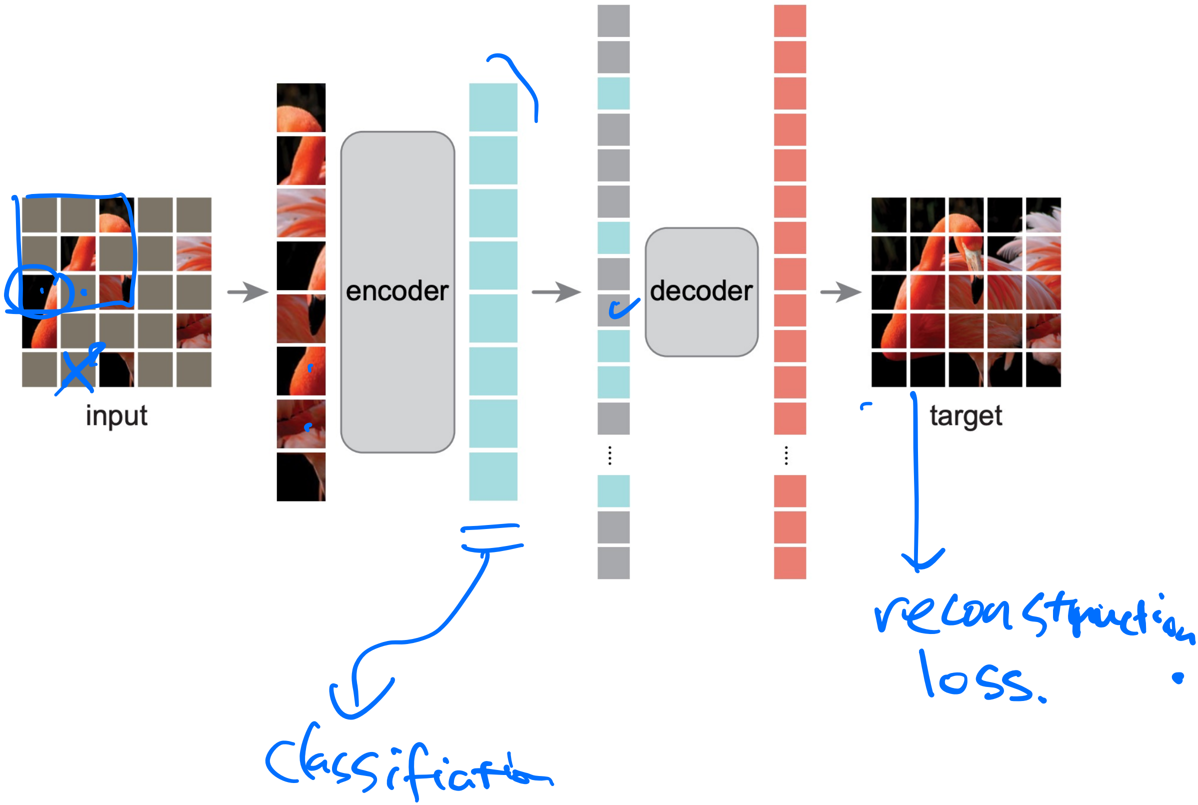
Masked Autoencoder (MAE)

- Modernized version of denoising autoencoder.



Masked Autoencoder (MAE)

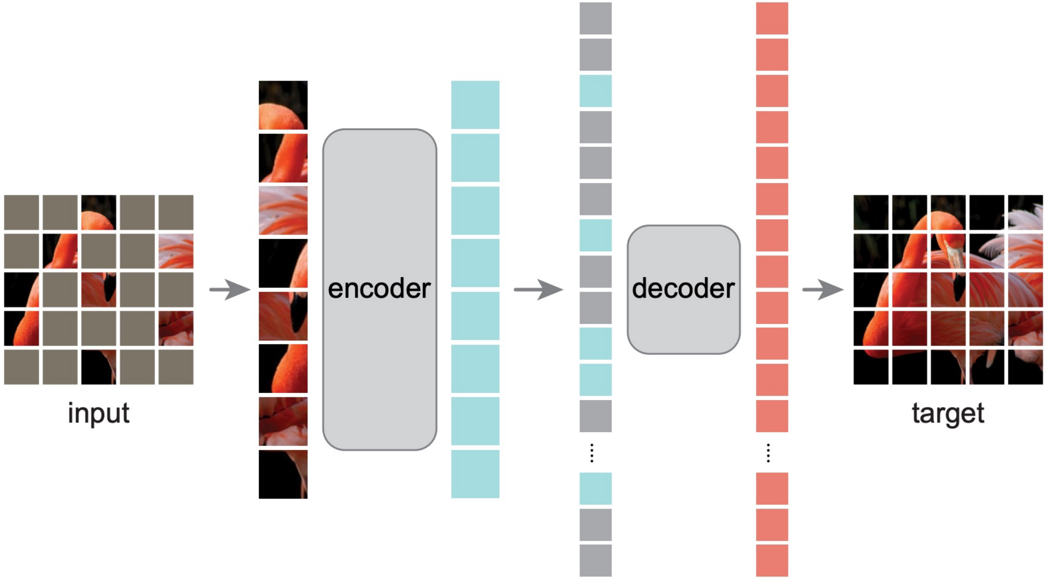
- Modernized version of denoising autoencoder.
- Mask noise: No artifacts
- ViT: No overlapping region, no empty space, no boundary.

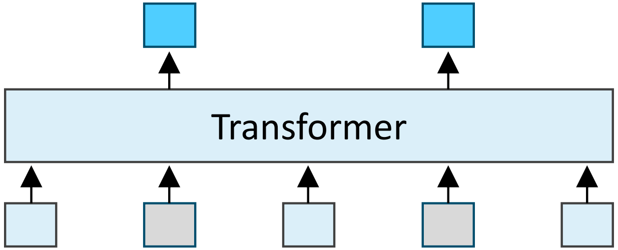


NYU

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.



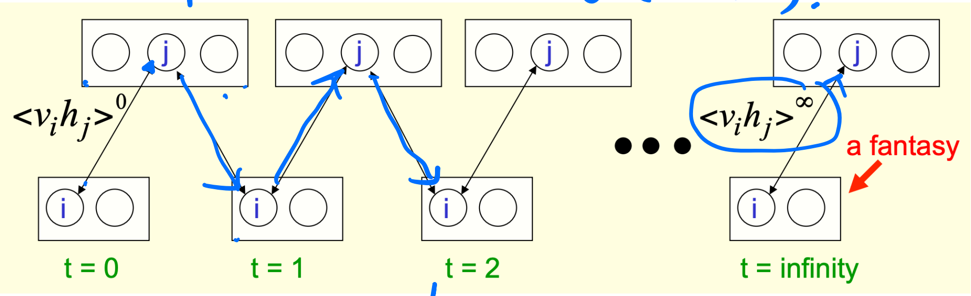


NYU

Energy-Based Learning

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

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



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

Hinton. Restricted Boltzmann Machines.

NYU

General EBMs

- Inference requires running gradient descent and MCMC samples.

General EBMs

- Inference requires running gradient descent and MCMC samples.

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

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

NYU

General EBMs

- Inference requires running gradient descent and MCMC samples.

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

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

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

NYU

General EBMs

- Inference requires running gradient descent and MCMC samples.

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

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

- 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.

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

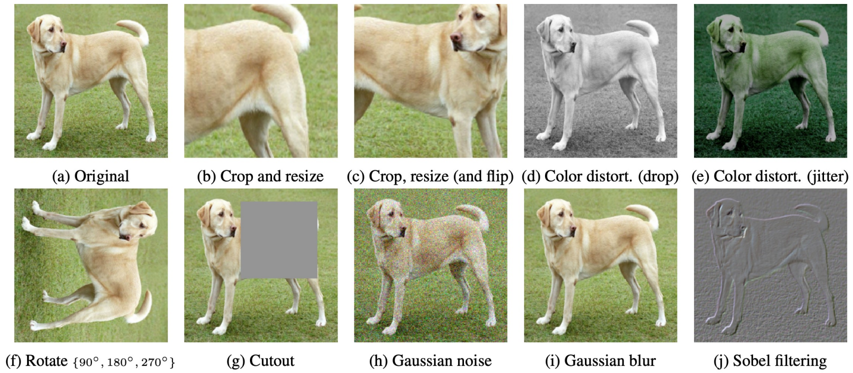
$$\nabla_\theta \mathcal{L}_{\mathrm{ML}} \approx \mathbb{E}_{\mathbf{x}^+ \sim p_D}\left[\nabla_\theta E_\theta(\mathbf{x}^+)\right] - \mathbb{E}_{\mathbf{x}^- \sim q_\theta}\left[\nabla_\theta E_\theta(\mathbf{x}^-)\right].$$

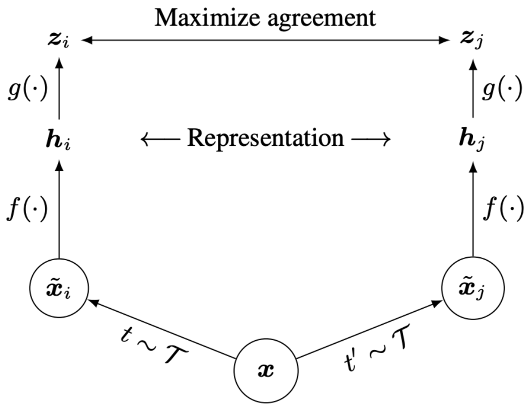
- Can be applied on hand manipulation trajectory generation.
- Good results in generation but still not a generalized representation learning algorithm.

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

Self-Supervised Visual Learning

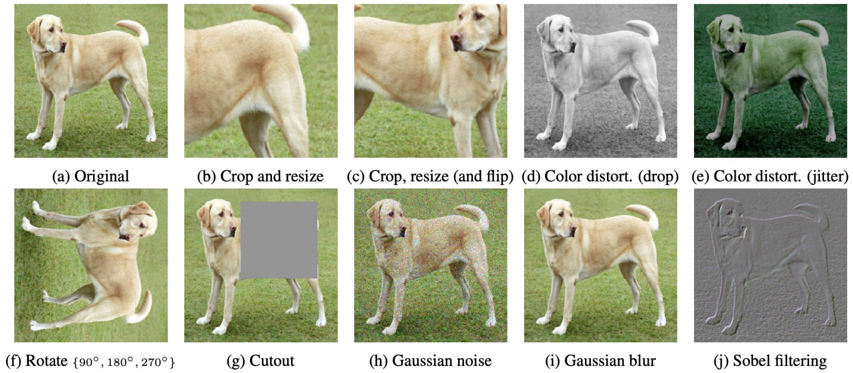
- Match the same image (with severe augmentation)

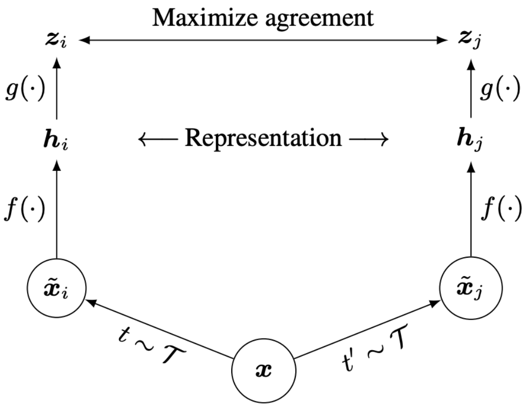




Self-Supervised Visual Learning

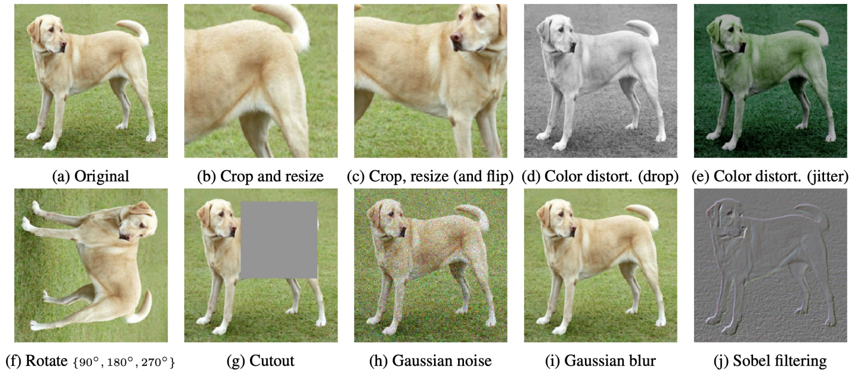
- Match the same image (with severe augmentation)
- Joint embedding approach: Apply loss on the embedding level.

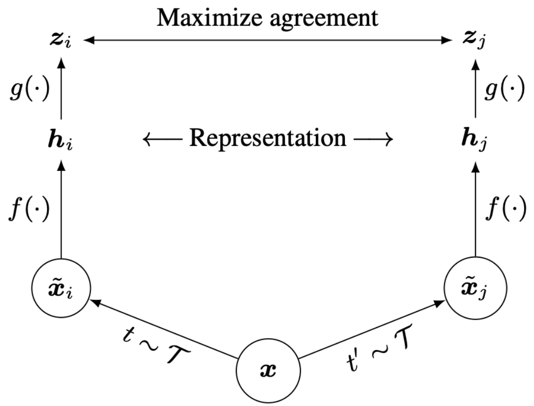




Self-Supervised Visual Learning

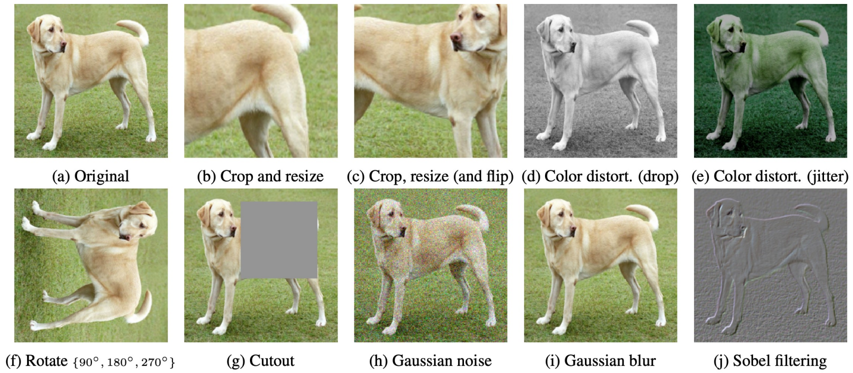
- Match the same image (with severe augmentation)
- Joint embedding approach: Apply loss on the embedding level.
- Use negative examples (contrastive) or not (non-contrastive).

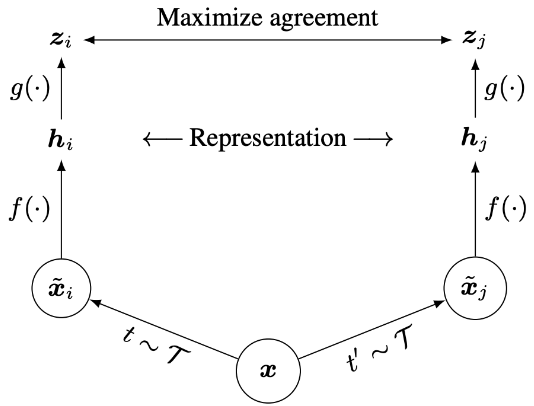




Self-Supervised Visual Learning

- 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.



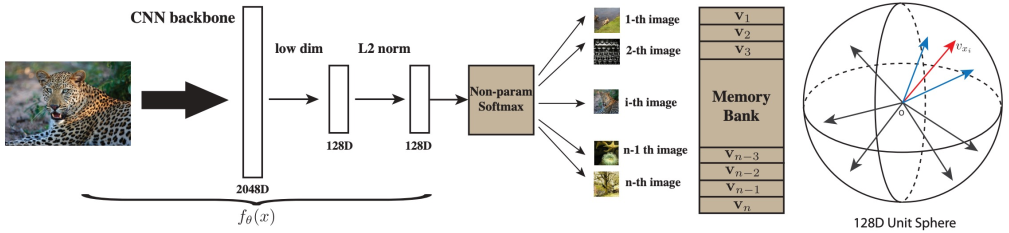


NYU

Several Embedding Loss Formulations

Wu et al., 2018

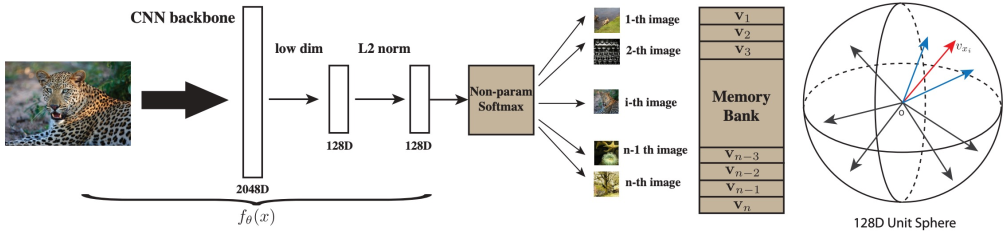
- Instance Classification:



Several Embedding Loss Formulations

Wu et al., 2018

- Instance Classification:



- Contrastive Learning: Cross entropy on pairs

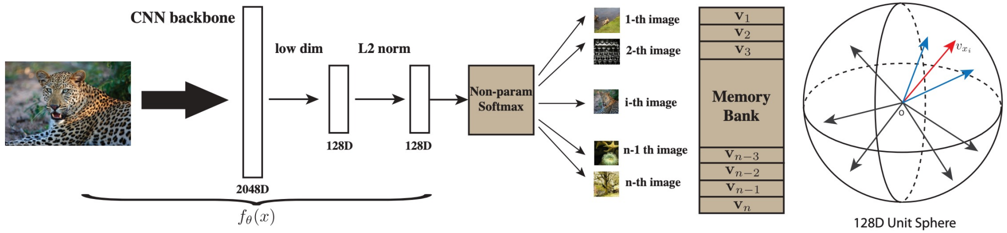
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Chen et al. 2020

Several Embedding Loss Formulations

Wu et al., 2018

- Instance Classification:



- Contrastive Learning: Cross entropy on pairs

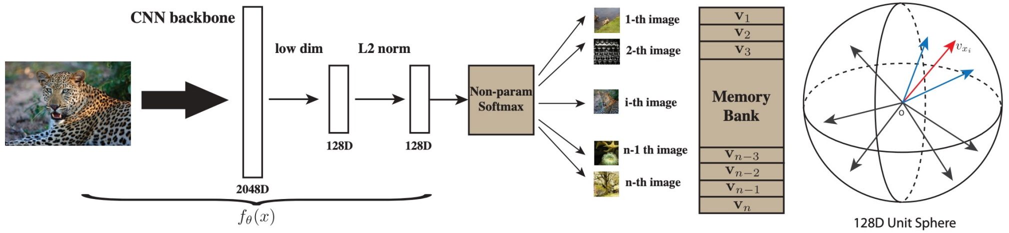
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Chen et al. 2020

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

Several Embedding Loss Formulations

Wu et al., 2018



- Instance Classification:
- Contrastive Learning: Cross entropy on pairs

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

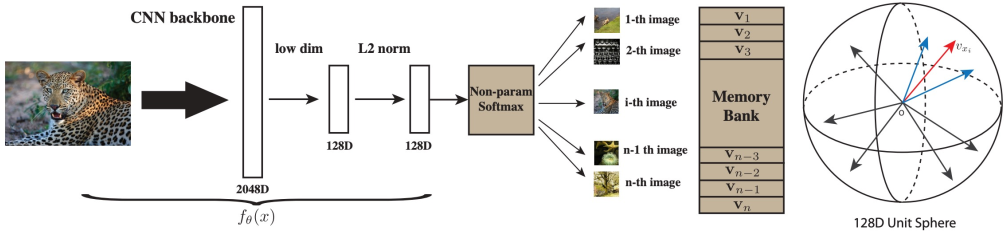
Chen et al. 2020

- 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

Wu et al., 2018

- Instance Classification:

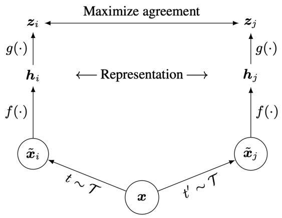


- Contrastive Learning: Cross entropy on pairs

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

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

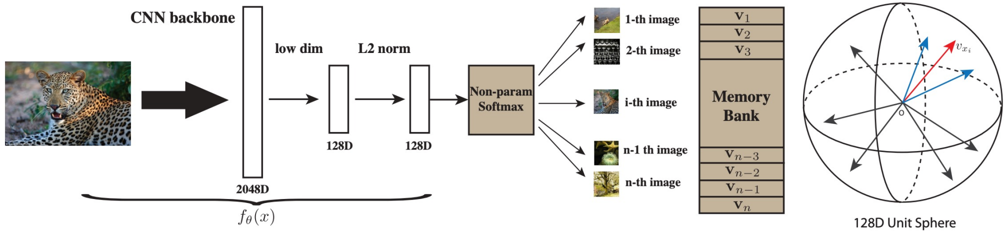


Chen et al. 2020

Several Embedding Loss Formulations

- Instance Classification:

Wu et al., 2018

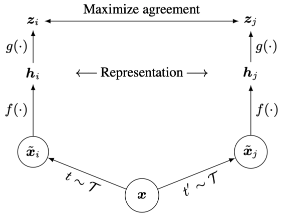


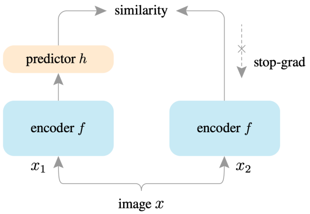
- Contrastive Learning: Cross entropy on pairs

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

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



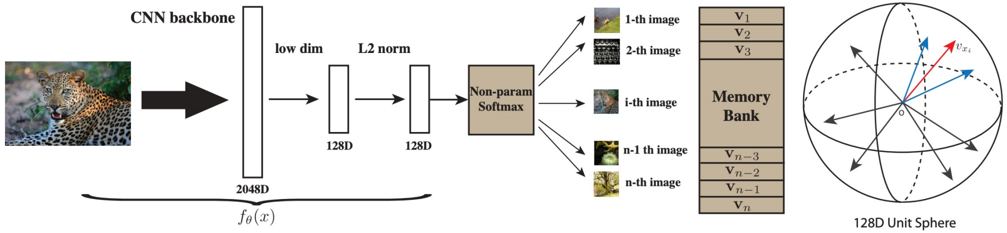


Chen et al. 2020 Chen & He, 2021

Several Embedding Loss Formulations

- Instance Classification:

Wu et al., 2018

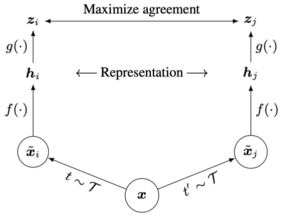


- Contrastive Learning: Cross entropy on pairs

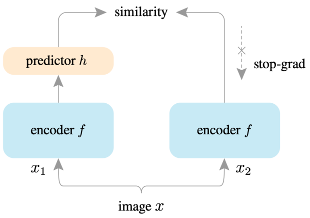
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

Chen et al. 2020

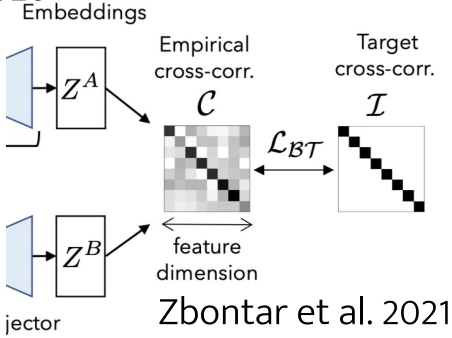
- Non-contrastive Learning (Positive Only)
  - Moving Average [Grill et al., 2020]
  - Stop Gradient [Chen & He, 2020]
- Use of projectors and predictors
- Use of co-variance regularization



Chen et al. 2020



Chen & He, 2021

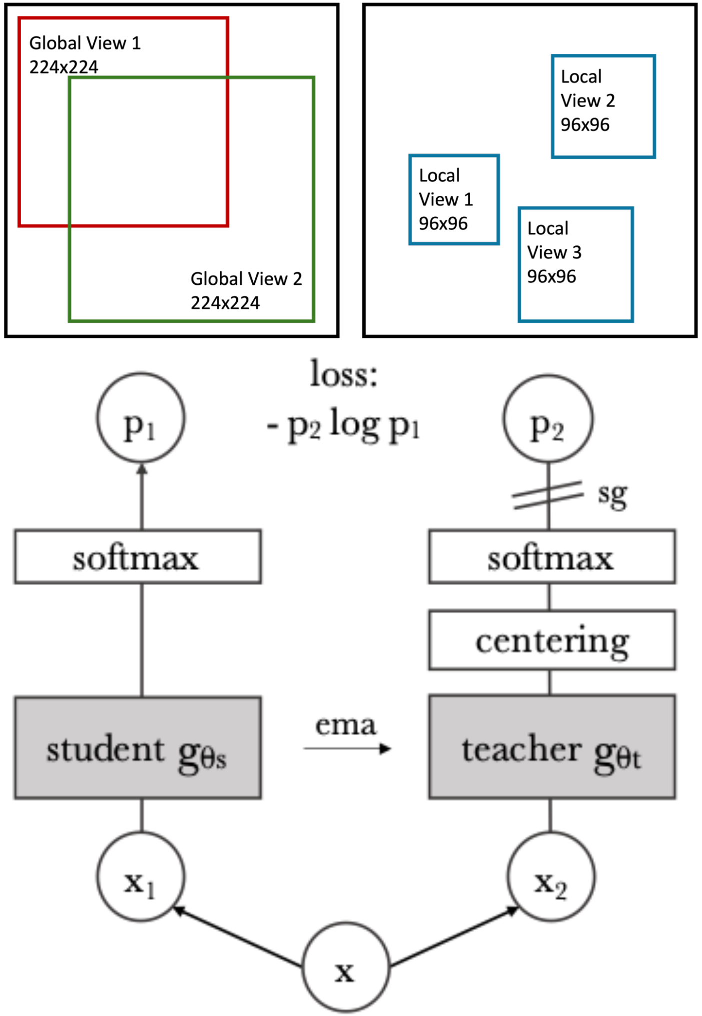


Zbontar et al. 2021
Bardes et al. 2021

NYU

DINO

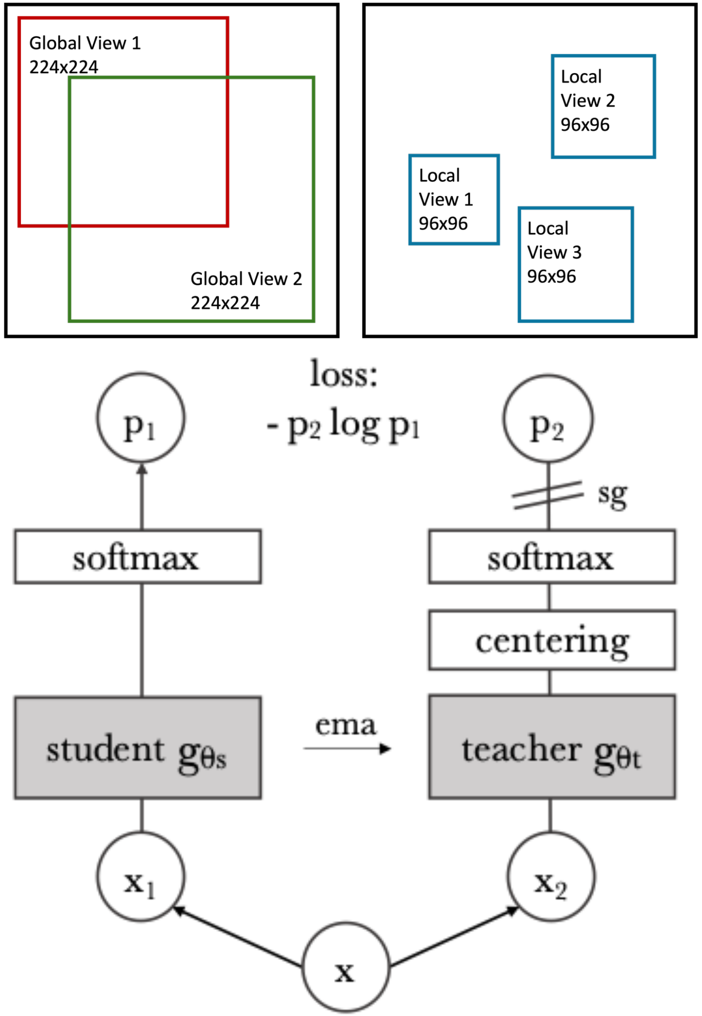
- Knowledge distillation between a student and a teacher network.



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

DINO

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

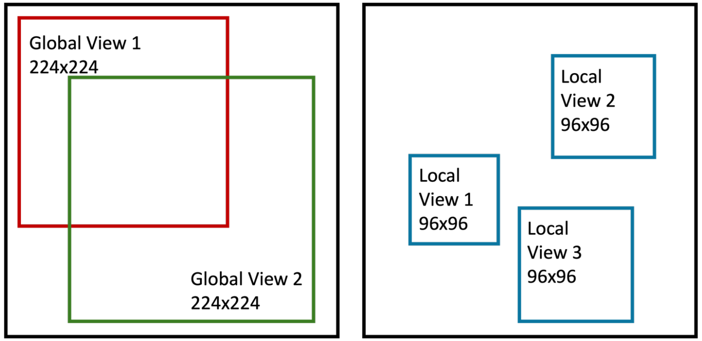


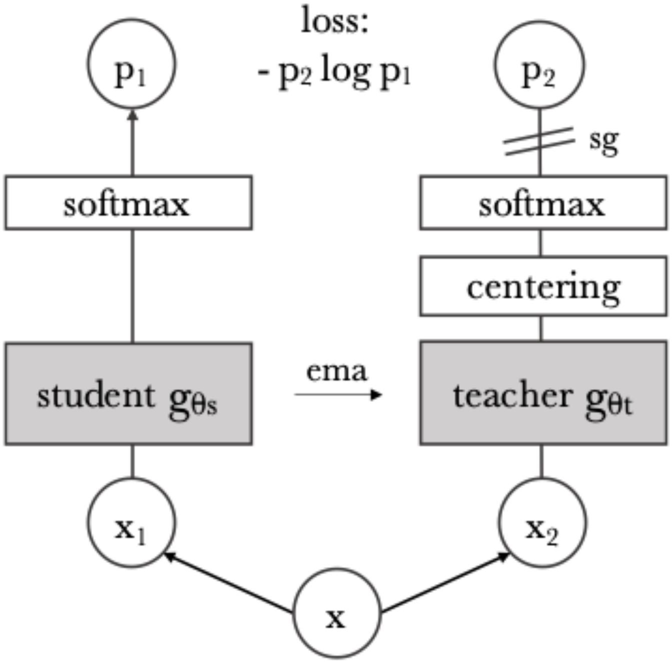
Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.

NYU

DINO

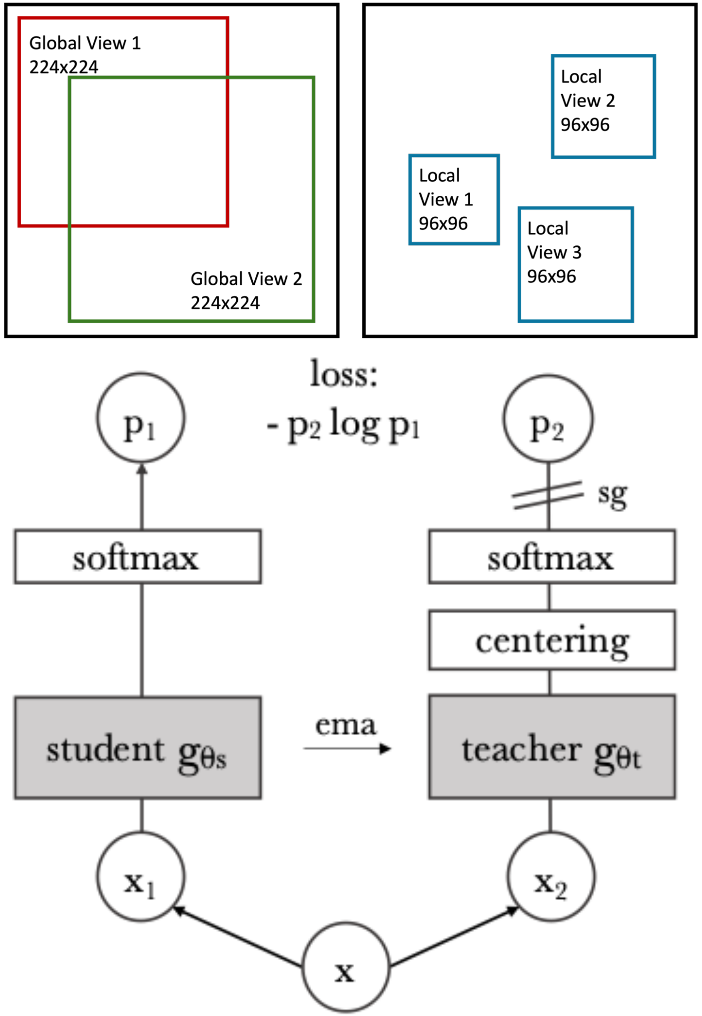
- Knowledge distillation between a student and a teacher network.
- Student: $p_s(x) = \frac{\exp(g_{\theta_s}(x)_i/\tau_s)}{\sum_k \exp(g_{\theta_s}(x)_k/\tau_s)}.$
- Minimize CE: $\min_\theta H(p_t(x), p_s(x)).$





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

DINO

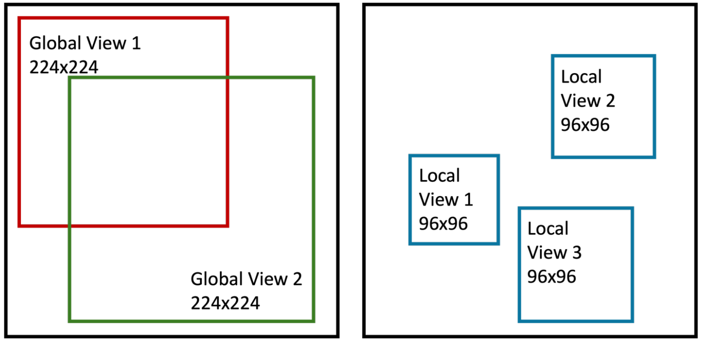


- Knowledge distillation between a student and a teacher network.
- Student: $p_s(x) = \frac{\exp(g_{\theta_s}(x)_i/\tau_s)}{\sum_k \exp(g_{\theta_s}(x)_k/\tau_s)}.$
- Minimize CE: $\min_\theta H(p_t(x), p_s(x)).$
- Stop gradient on the teacher (no true label).

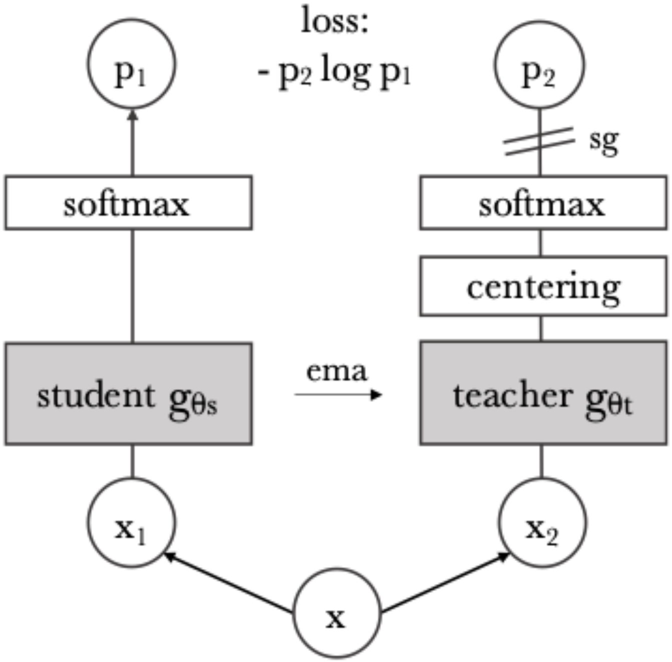
Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.

NYU

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).
- 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.

Preventing Collapse

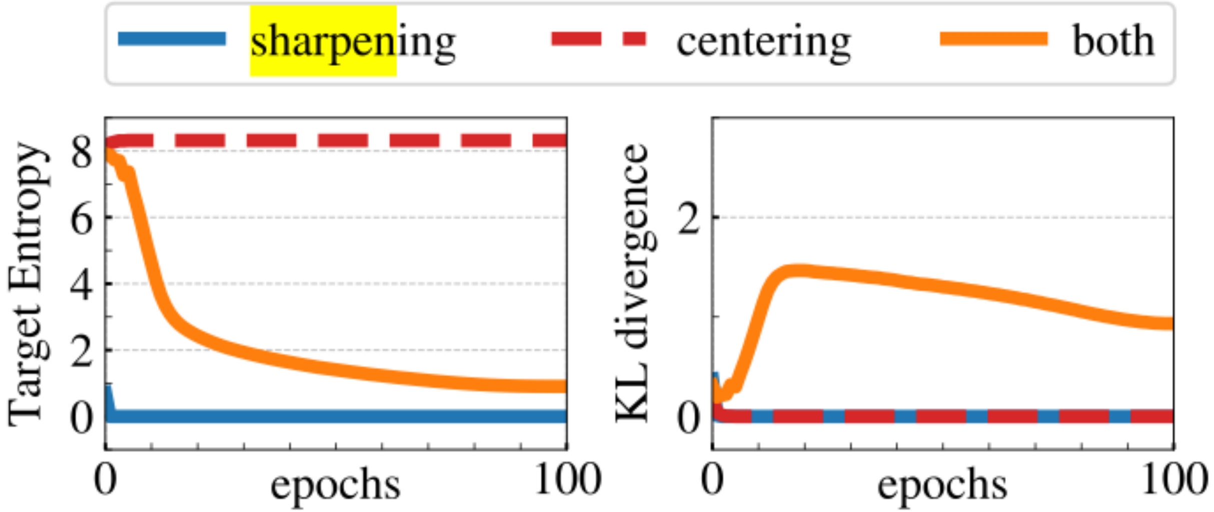
- Cross entropy objective can make both sides collapse to uniform distribution.
  - Apply sharpening, apply a temperature term on both teacher and student.
  - $\mathrm{softmax}(g/\tau)$ The higher the temperature, the more uniform.
- It can also collapse into always activating a single unit.
  - Mean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

NYU

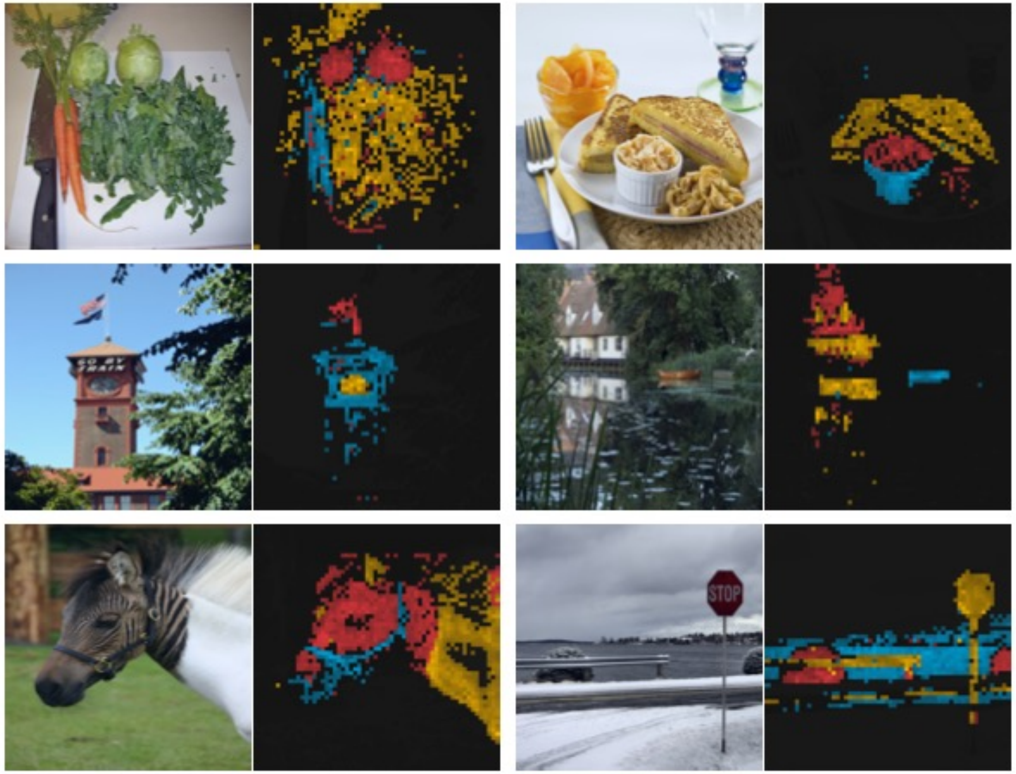
Centering and Sharpening

- 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.

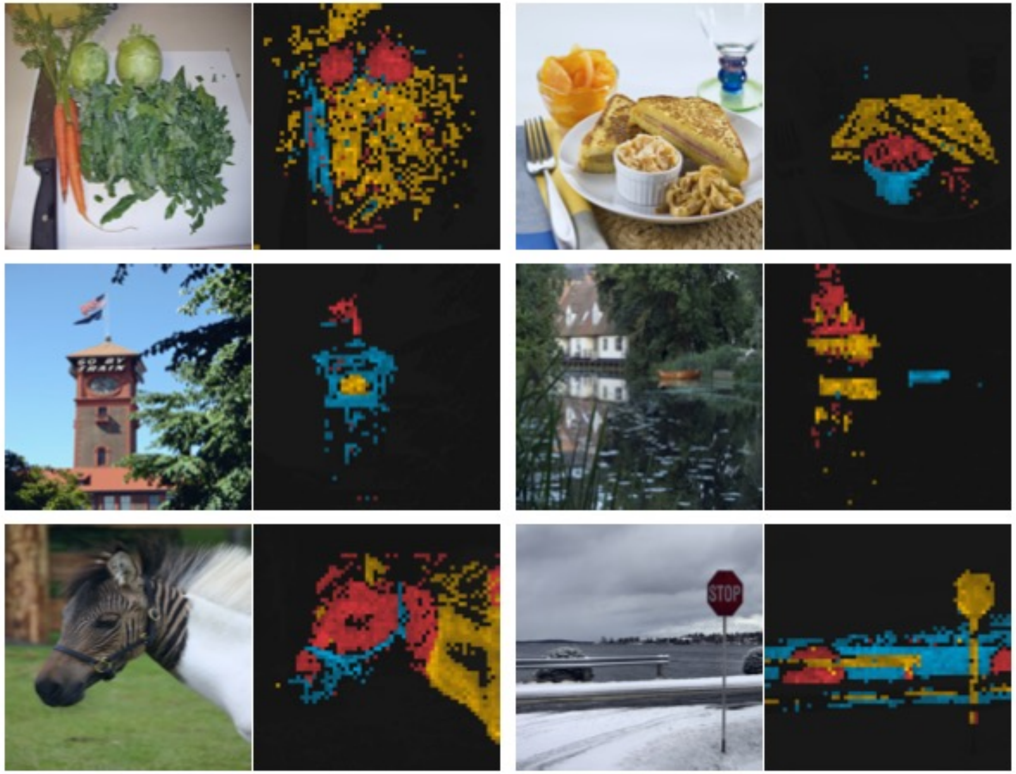


Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.

NYU

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.

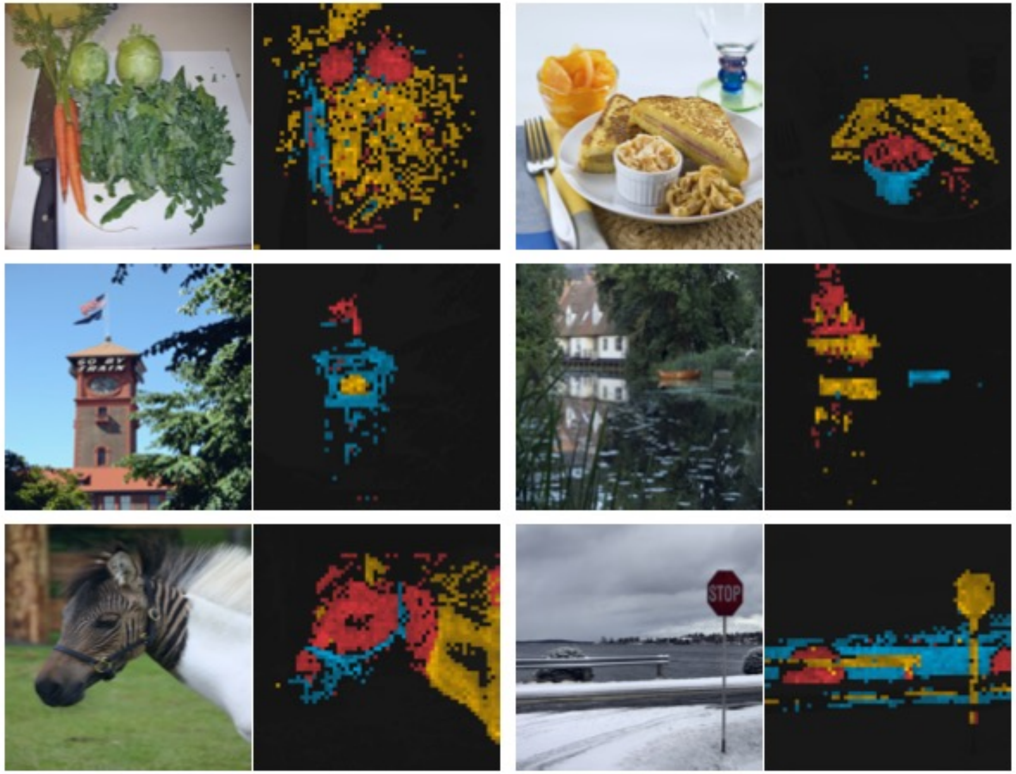


Caron et al. Emerging Properties in Self-Supervised Vision Transformers. ICCV 2021.

NYU

Visualizing Attention

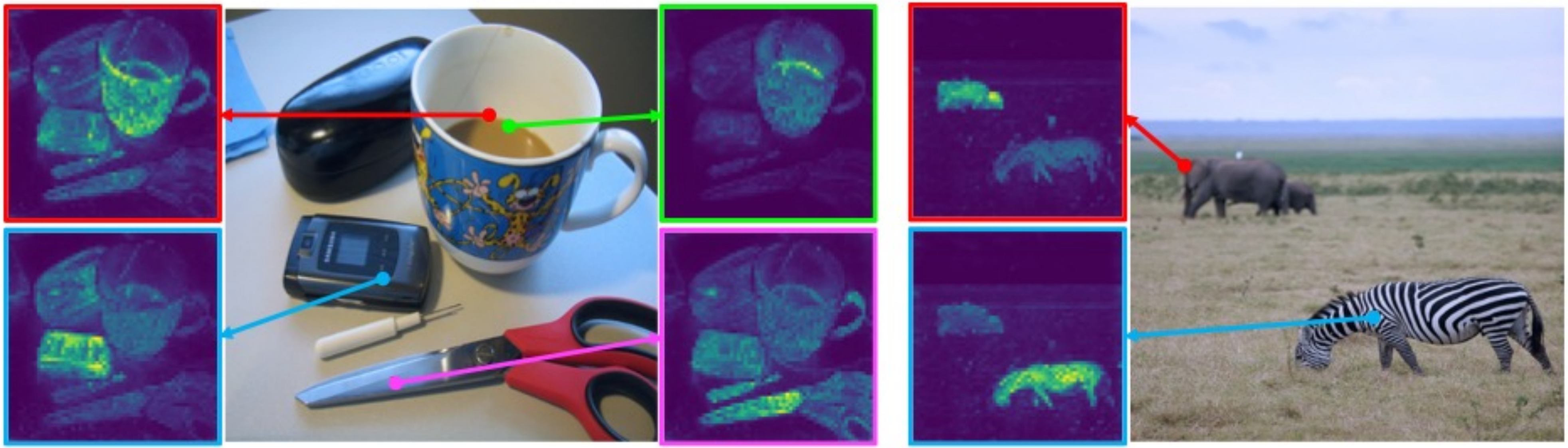
- The [CLS] token is an extra token added to summarize the whole image into a vector.
- Visualize the attention map of different attention heads using different colors.
- Showing understanding of different objects and parts.



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

Visualizing Attention

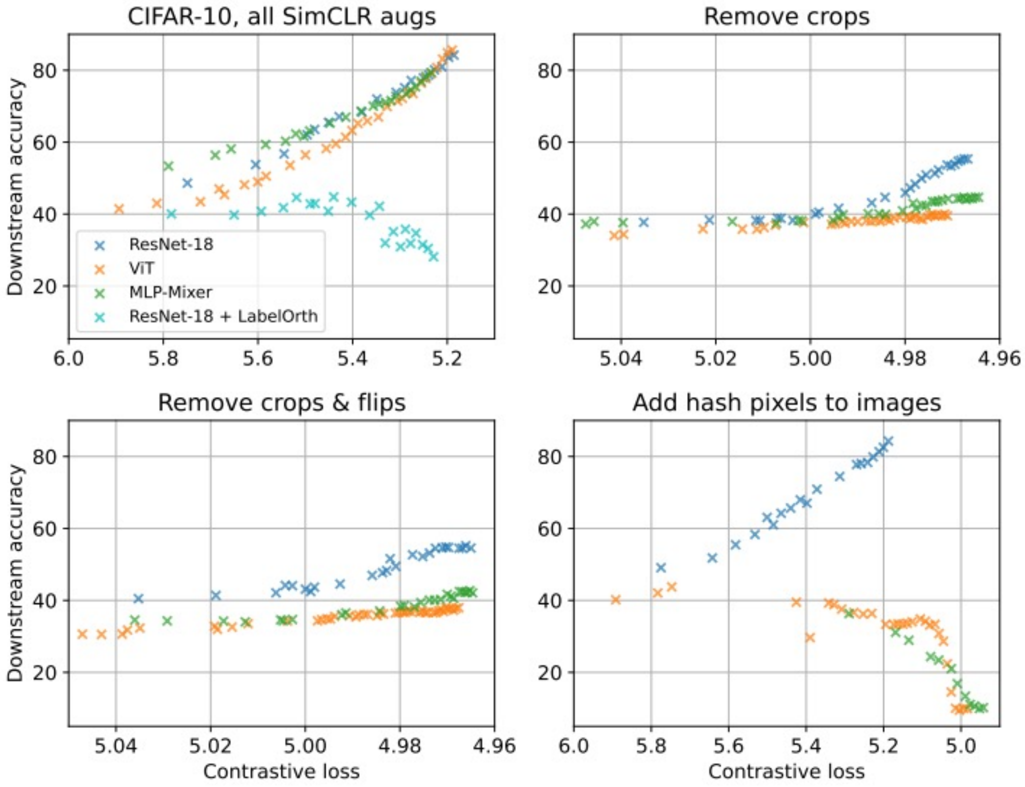
- We can also visualize the attention by querying from a location.
- Weak separation of objects.



NYU

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.

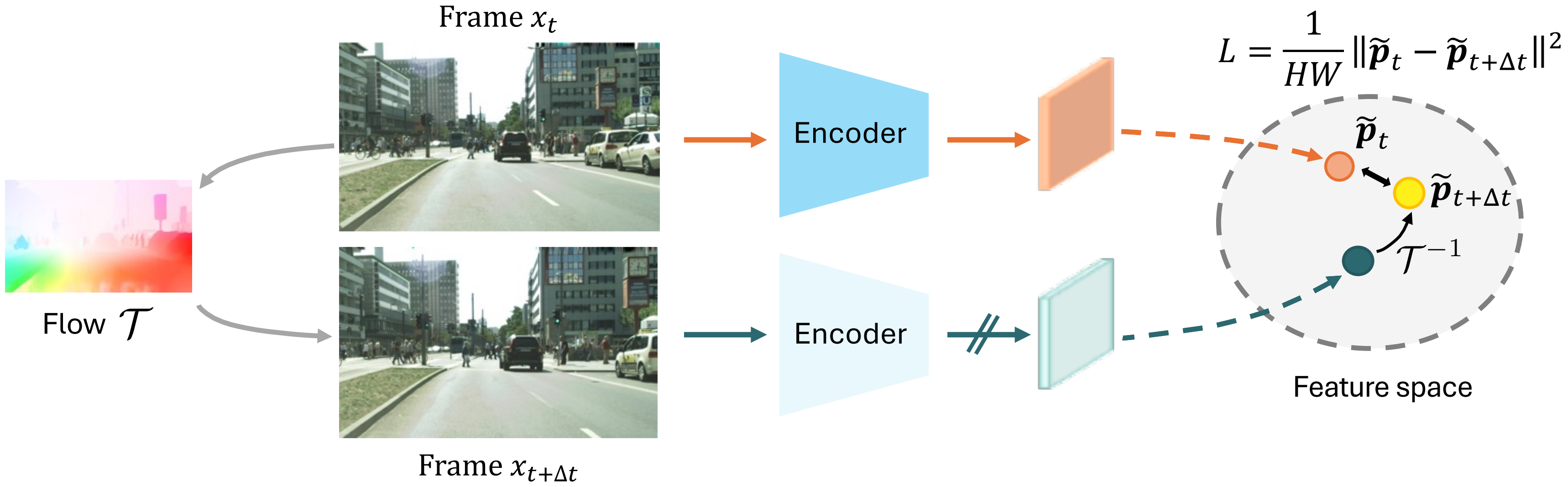


Aurora et al. A Theoretical Analysis of Contrastive Unsupervised Representation Learning.
Saunshi et al. Understanding Contrastive Learning Requires Incorporating Inductive Biases. 2022.

NYU

SSL with Motion

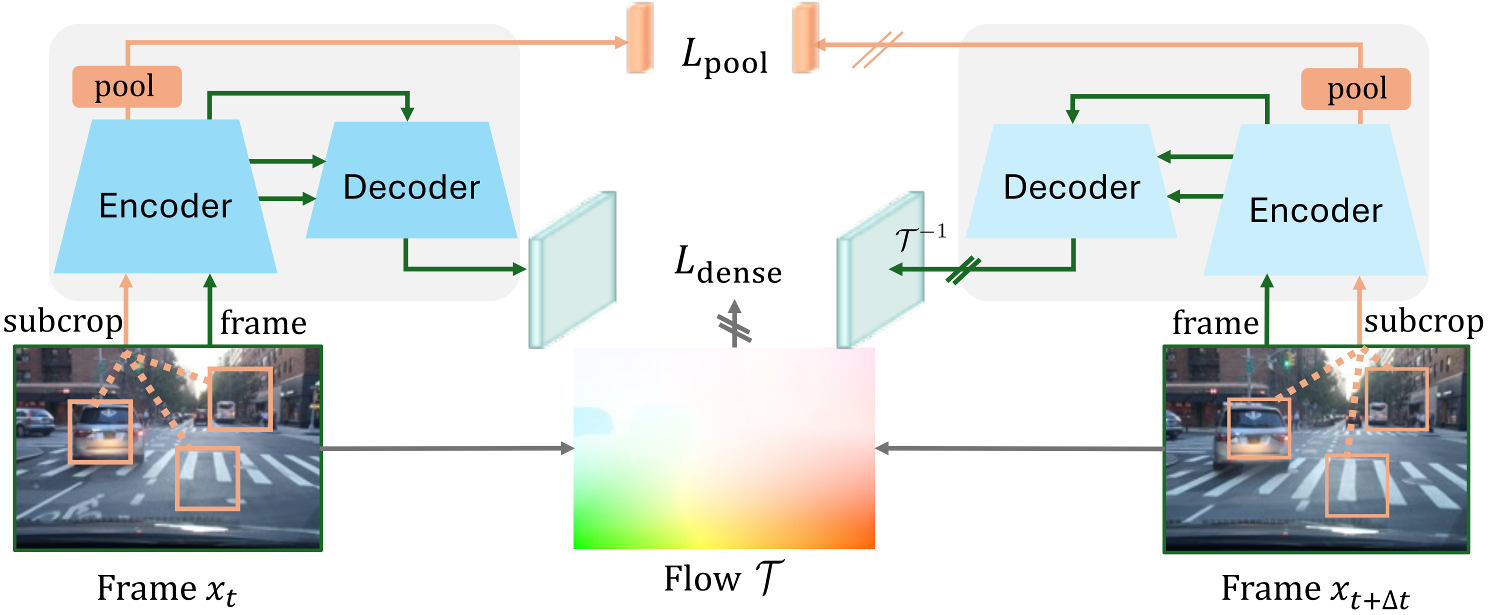
- Can we use adjacent frames as self-supervision?
- Objects move densely throughout the image.



NYU

SSL with Motion

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

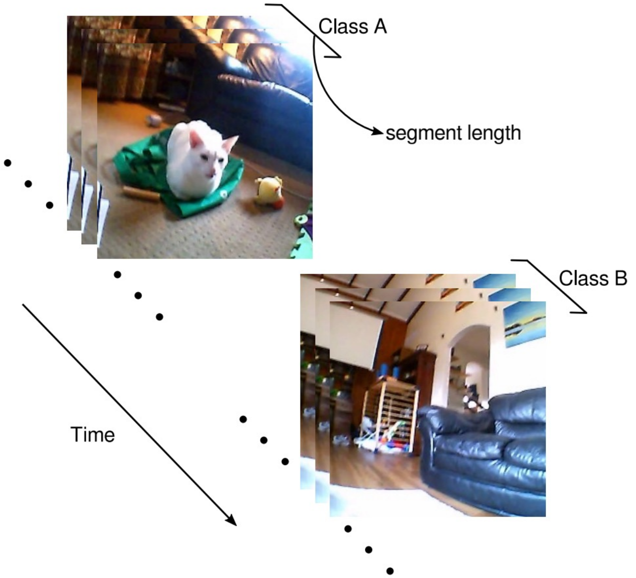


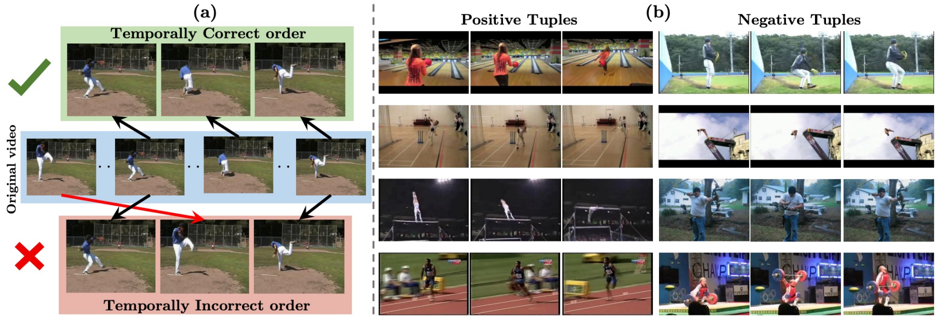
Wang et al. PooDLe: Pooled and Dense Self-Supervised Learning from Naturalistic Videos. ICLR 2025.

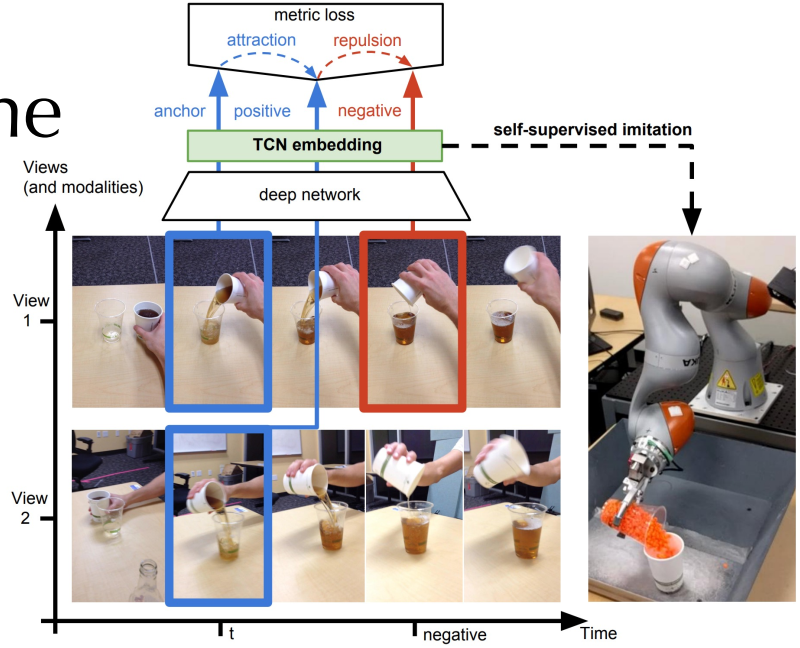
NYU

SSL with Time

- Use time as an additional source of supervision.



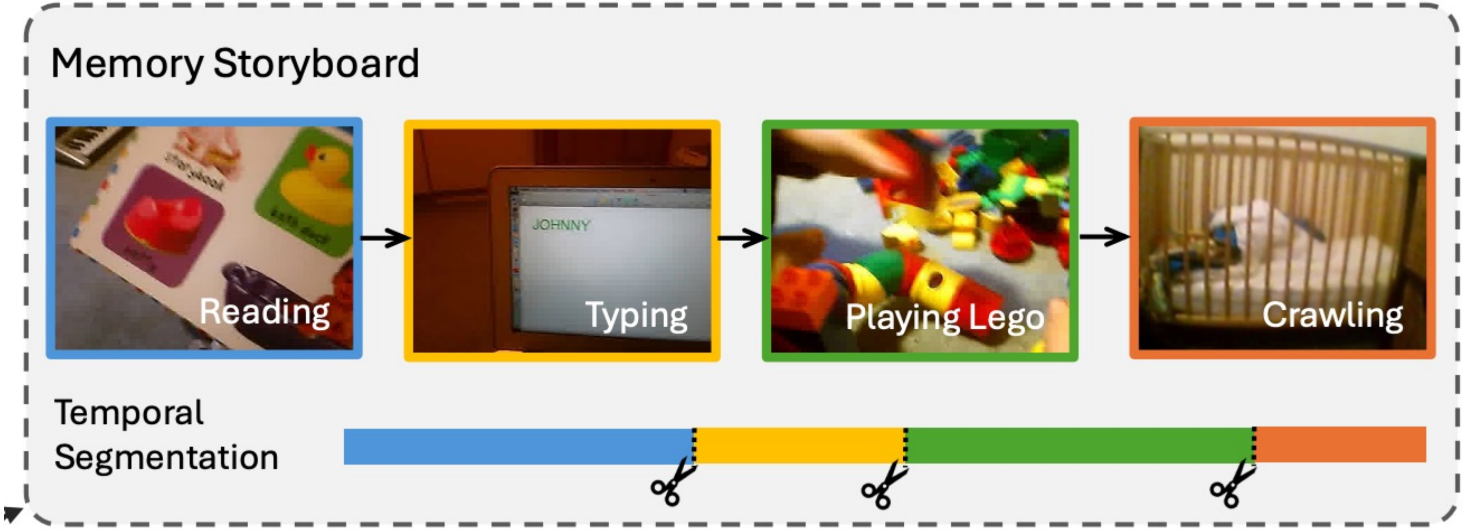




Misra et al. Shuffle and Learn: Unsupervised Learning using Temporal Order Verification. ECCV 2016.
Sermanet et al. Time-Contrastive Networks: Self-Supervised Learning from Video. ICRA 2018.
Orhan et al. Self-Supervised Learning through the Eyes of a Child. NeurIPS 2020.

SSL with Time

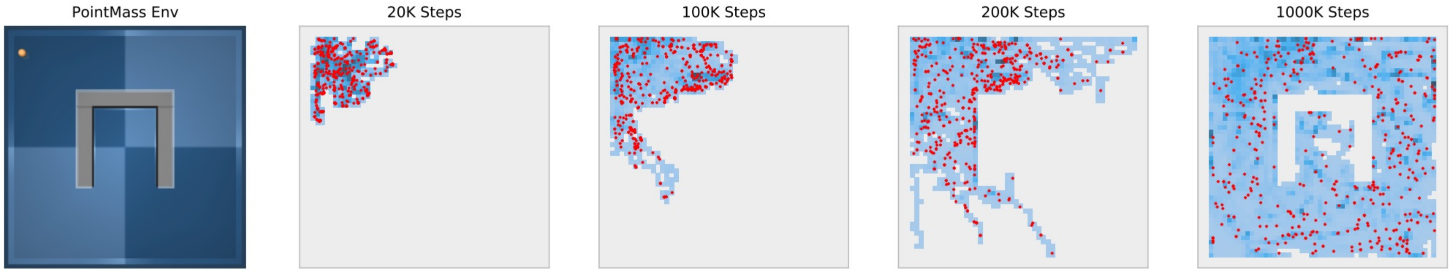
- We can segment videos into meaningful events.
- Leverage the spatiotemporal continuity structure.

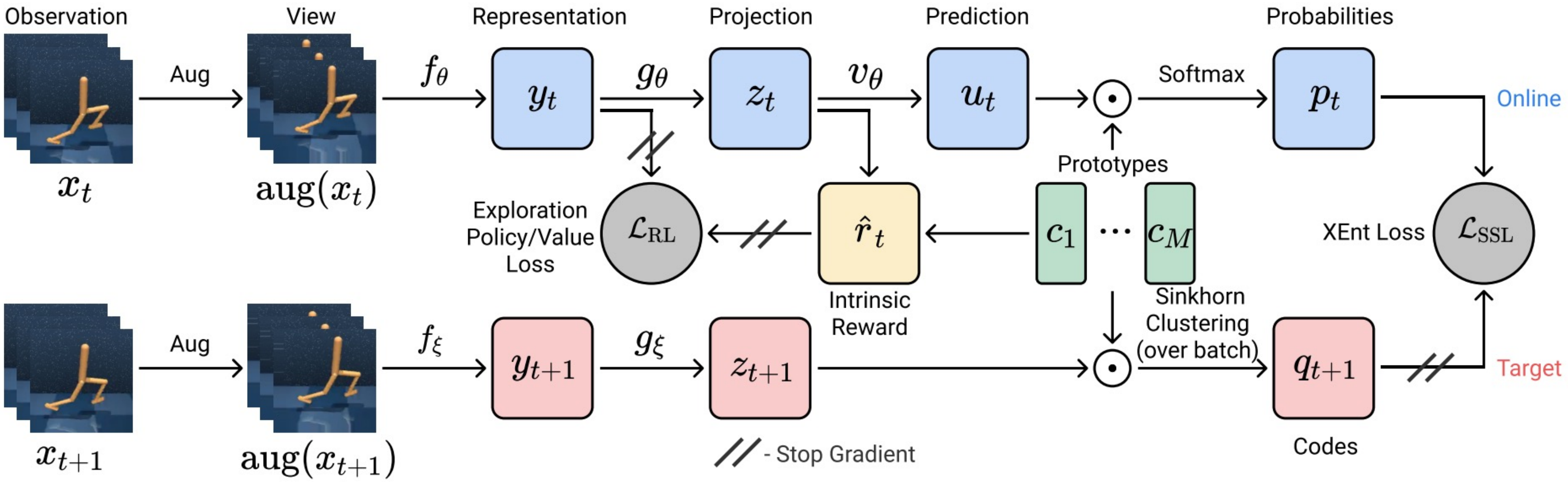


Yang & Ren. Memory Storyboard: Leveraging Temporal Segmentation for Streaming Self-Supervised Learning from Egocentric Videos. arXiv 2025.

NYU

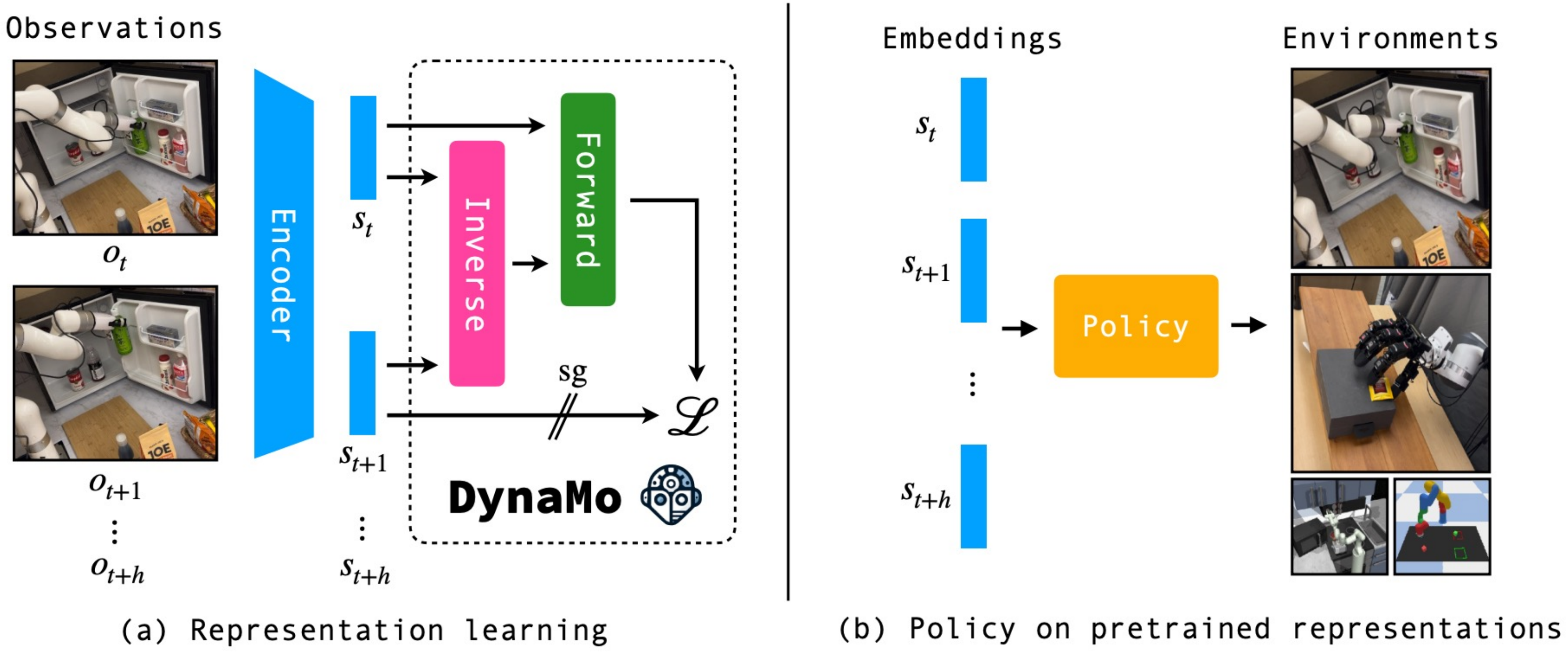
SSL for Visual Control





Yarats. Reinforcement Learning with Prototypical Representations. ICML 2021.

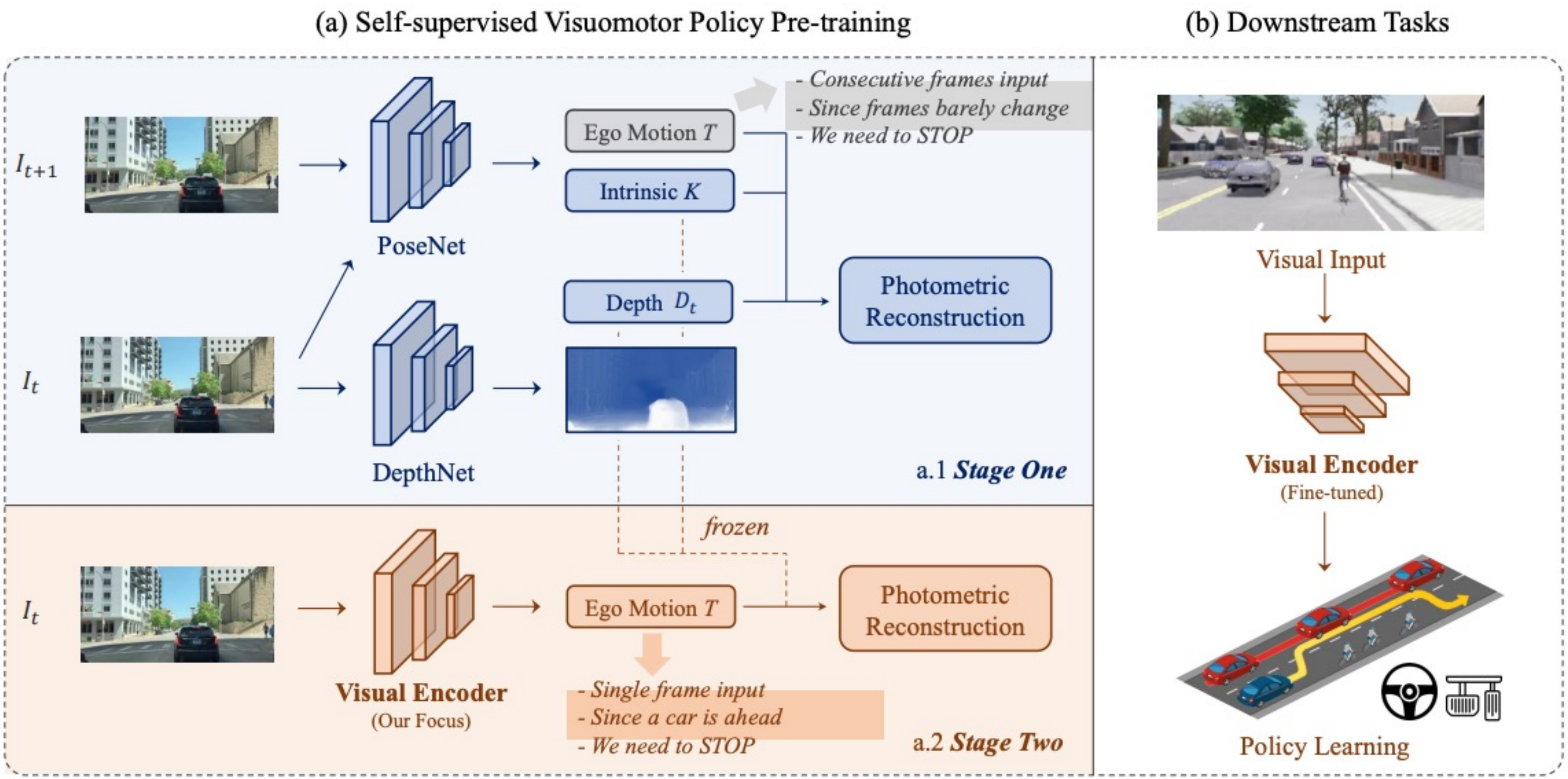
SSL for Visual Control



Cui et al. DynaMo: In-Domain Dynamics Pretraining for Visuo-Motor Control. NeurIPS 2024.

NYU

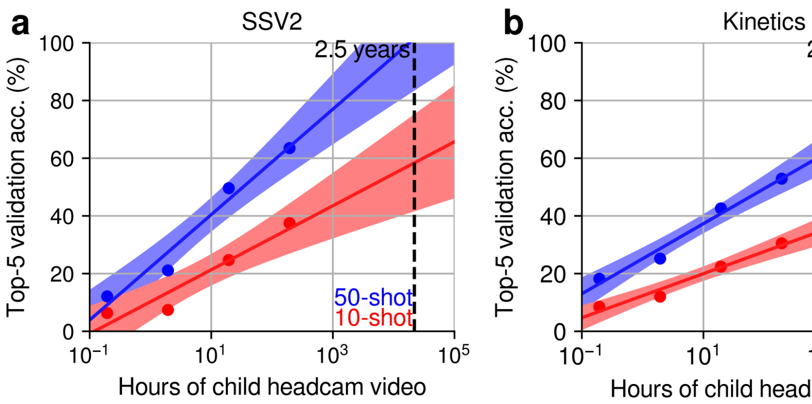
SSL for Visual Control

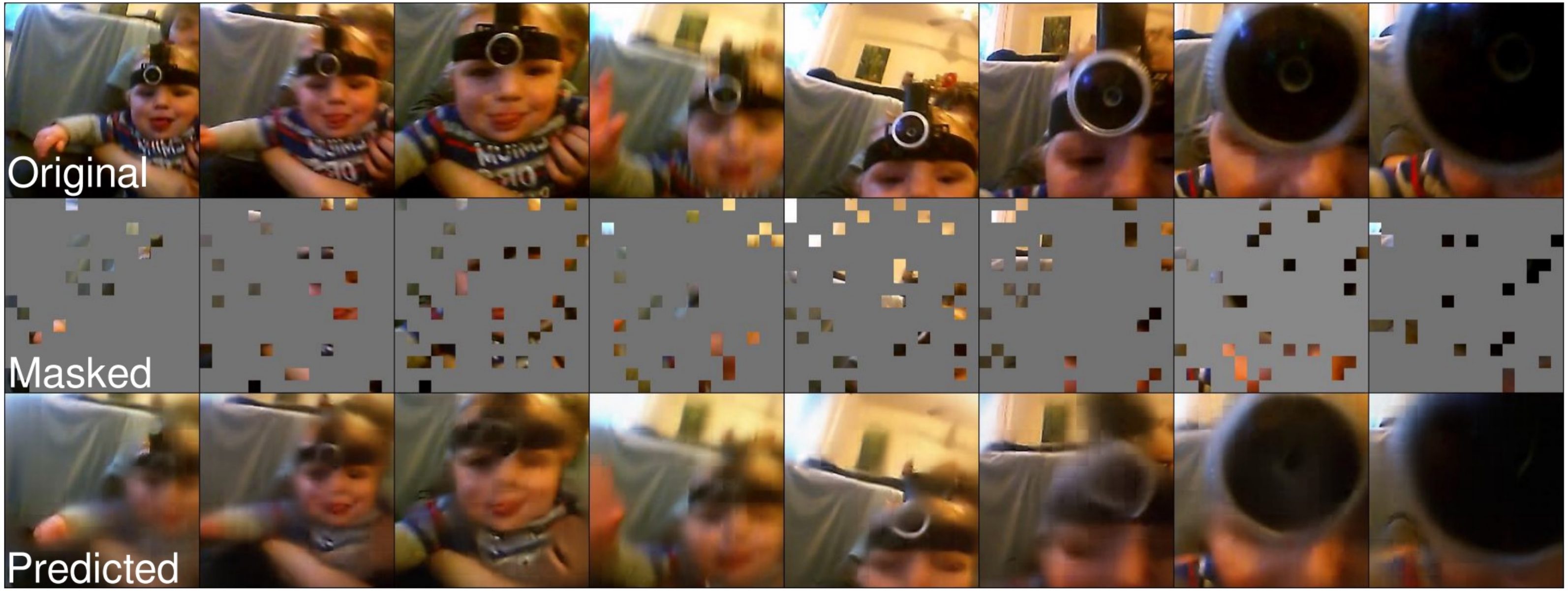


Wu et al. Policy Pre-training for Autonomous Driving via Self-supervised Geometric Modeling. ICLR 2023.

BabyCam

- Run visual learning algorithms on baby headcam videos.





NYU

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

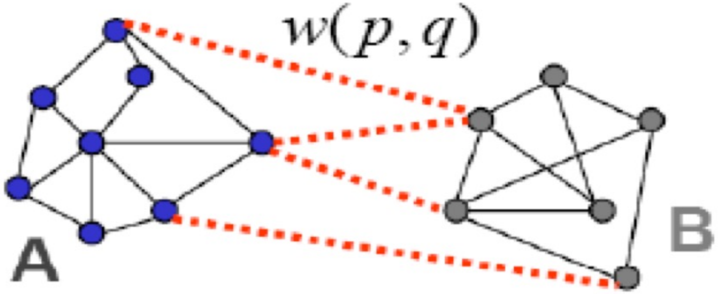
- General idea: Use self-supervised learning to learn good features, which allow us to generate low-quality masks.
- Then use these masks as pseudo labels and supervise the network to predict these low-quality masks.

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.

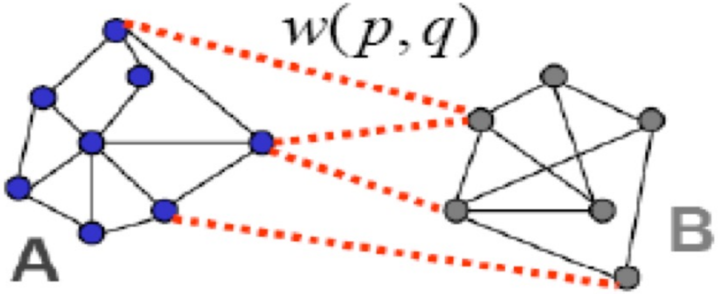


$$cut(A,B) = \sum_{p \in A, q \in B} w(p,q)$$

NYU

Graph Cut

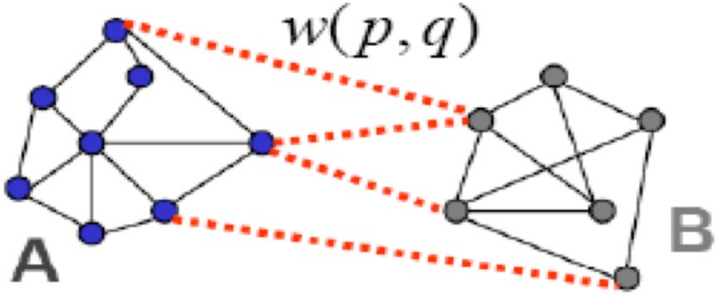
- Segmentation is essentially a clustering problem.
- We can transform the clustering problem with the graph cut problem.



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

Graph Cut

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

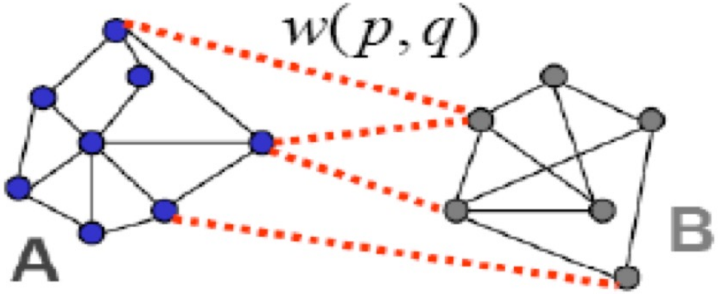


$$cut(A,B) = \sum_{p \in A, q \in B} w(p,q)$$

NYU

Graph Cut

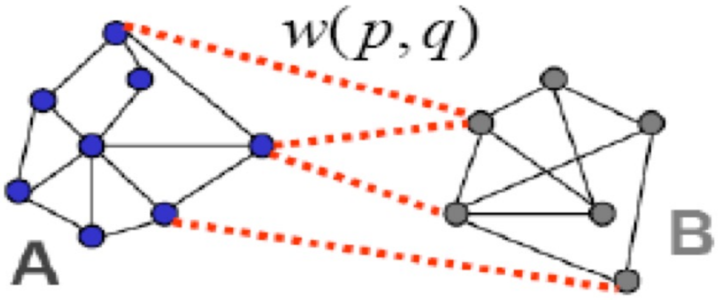
- Segmentation is essentially a clustering problem.
- We can transform the clustering problem with the graph cut problem.
- Pixel = node.
- Affinity between the two pixels = edge value (flow).



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

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.

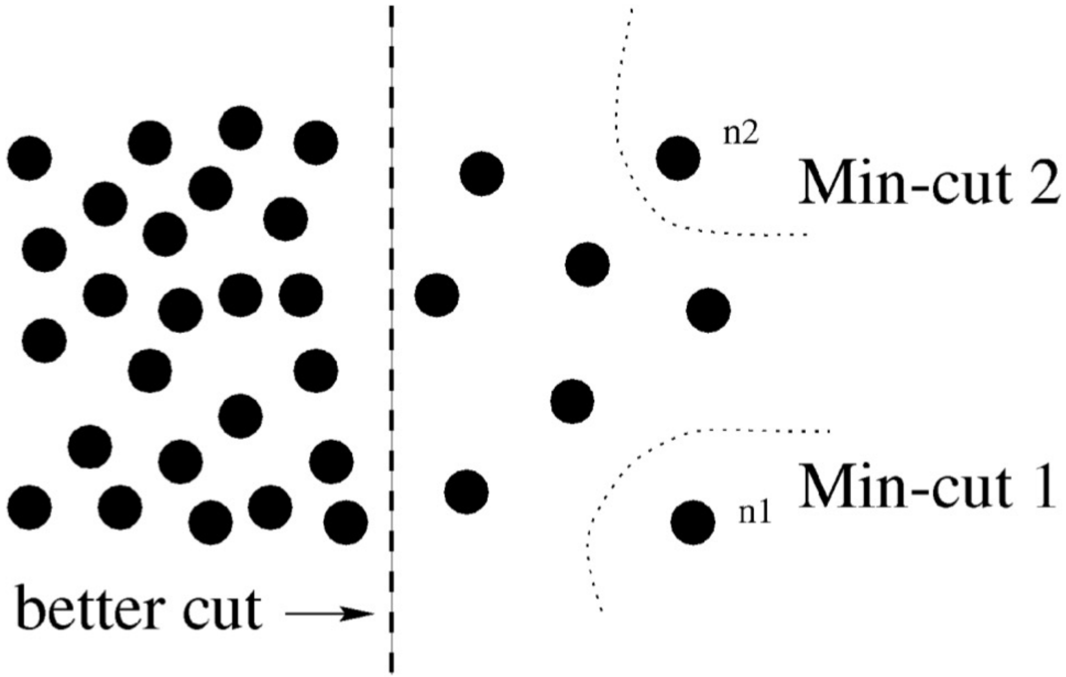


$$cut(A,B) = \sum_{p \in A, q \in B} w(p,q)$$

NYU

Normalized Graph Cut (NCut)

- How to prevent cutting small isolated nodes?



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

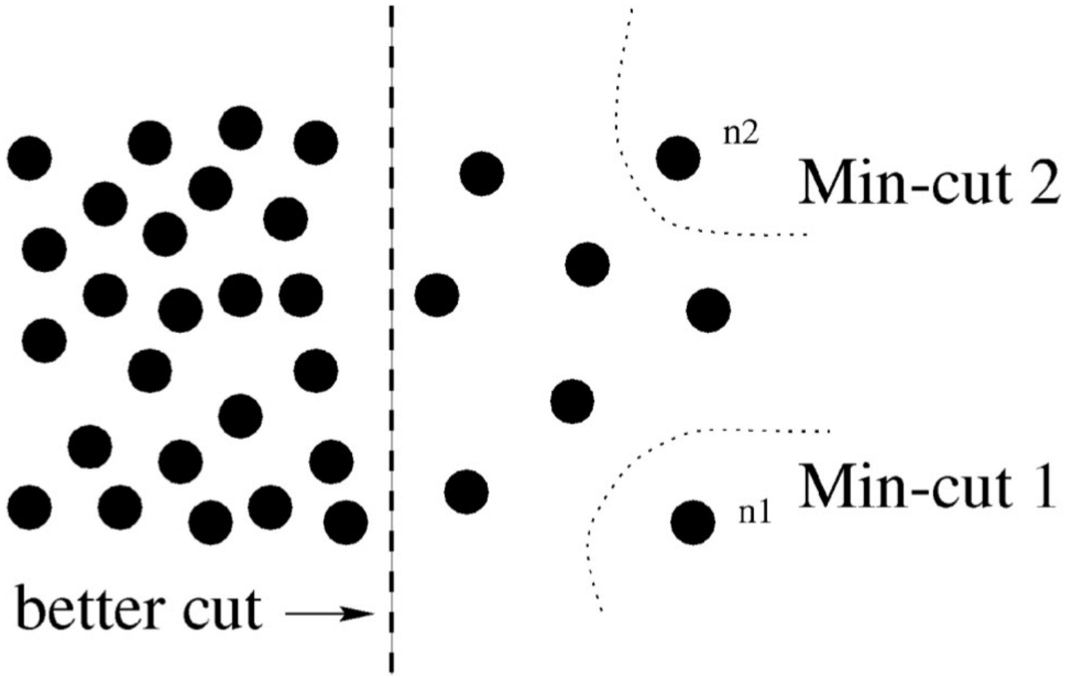
Shi and Malik. Normalized Cuts and Image Segmentation. TPAMI 2000.

NYU

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.

NYU

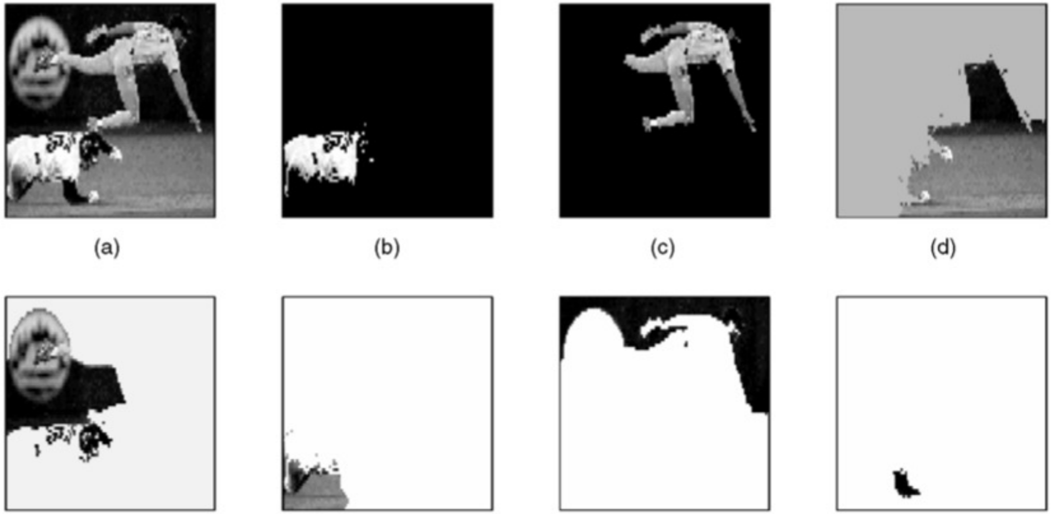
NCut Details (Optional)

- 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.



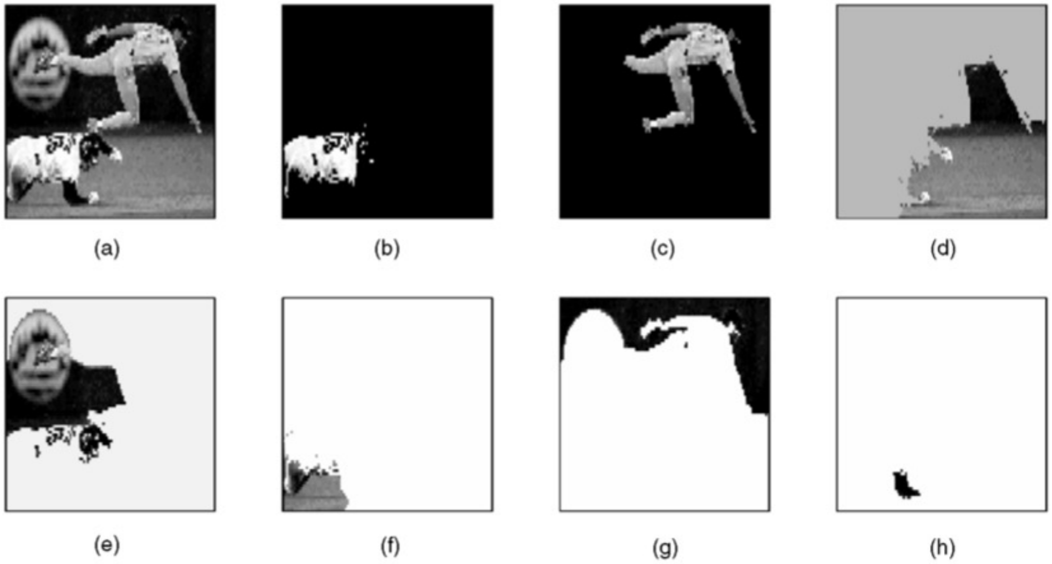
(a) (b) (c) (d)
(e) (f) (g) (h)

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

NYU

NCut

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

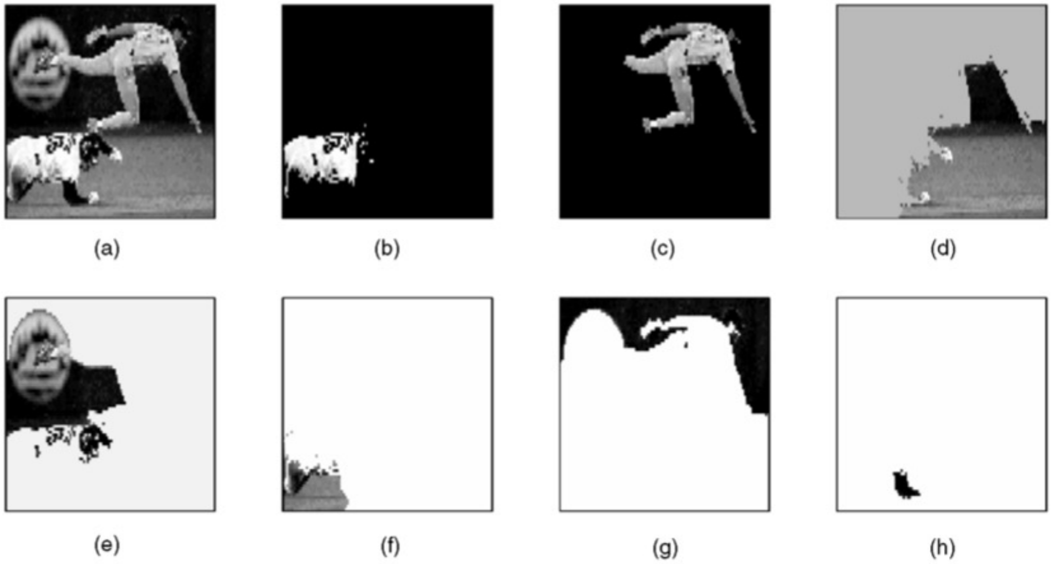


Shi and Malik. Normalized Cuts and Image Segmentation. TPAMI 2000.

NYU

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.

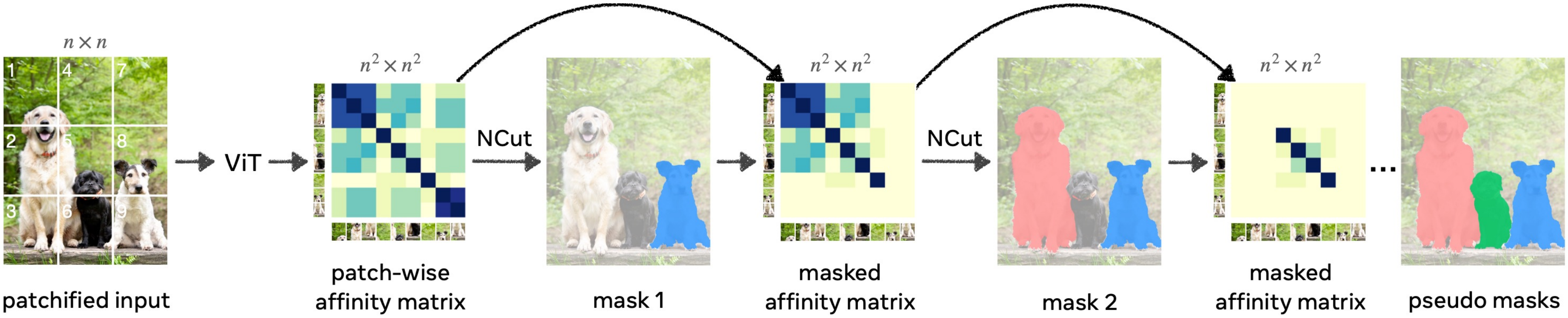


(a) (b) (c) (d) (e) (f) (g) (h)

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

MaskCut

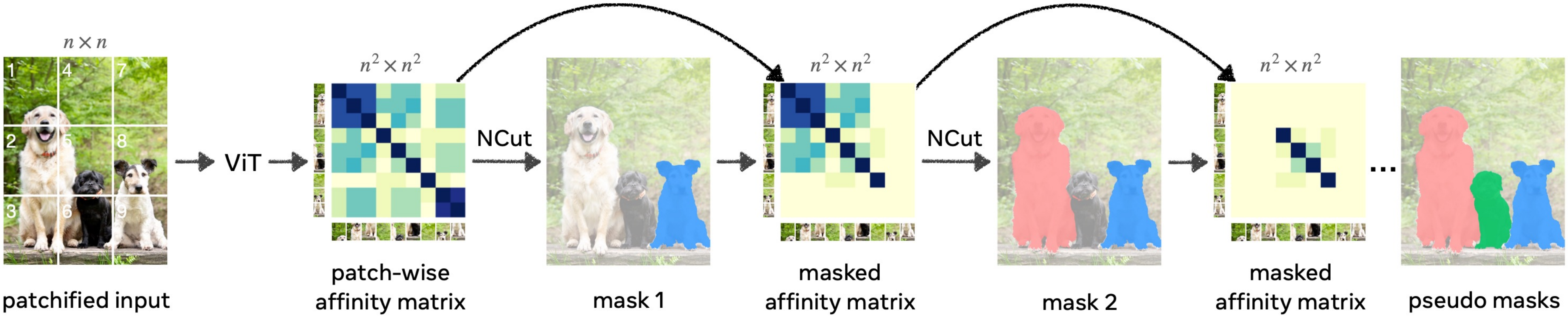
- Use a pretrained DINO ViT network.



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: $W_{ij} = \frac{K_i K_j}{\|K_i\|_2 \|K_j\|_2}$

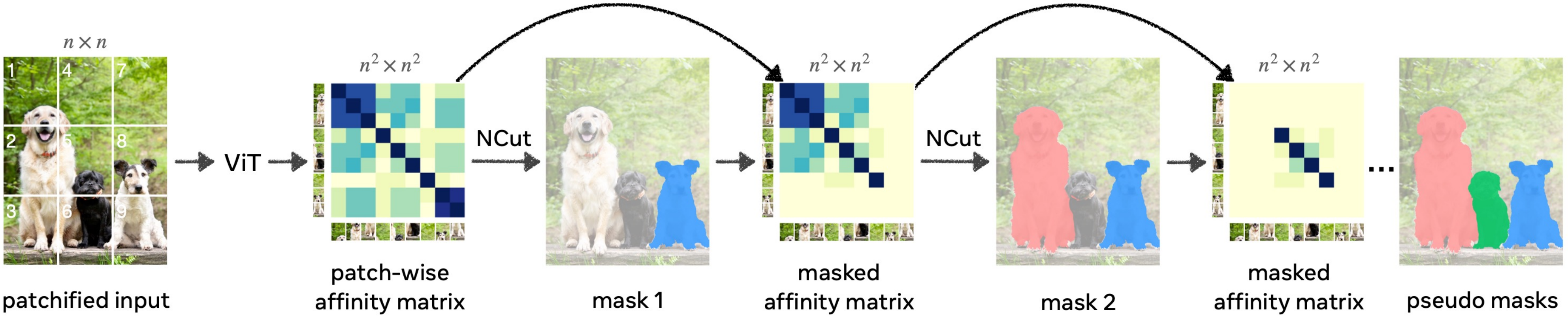


Wang et al. Self-supervised transformers for unsupervised object discovery using normalized cut. CVPR 2022.
Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.

NYU

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.

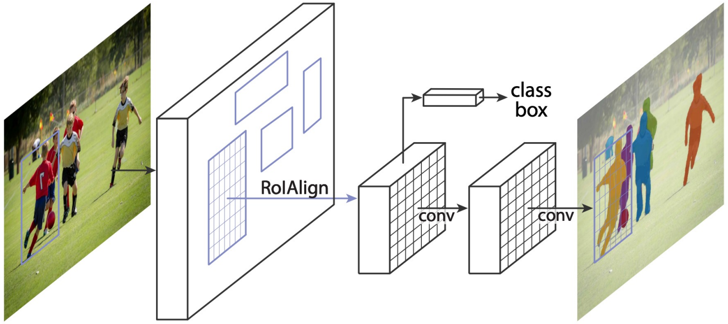


Wang et al. Self-supervised transformers for unsupervised object discovery using normalized cut. CVPR 2022.
Wang et al. Cut and Learn for Unsupervised Object Detection and Instance Segmentation. CVPR 2023.

NYU

Iterative Self-Training

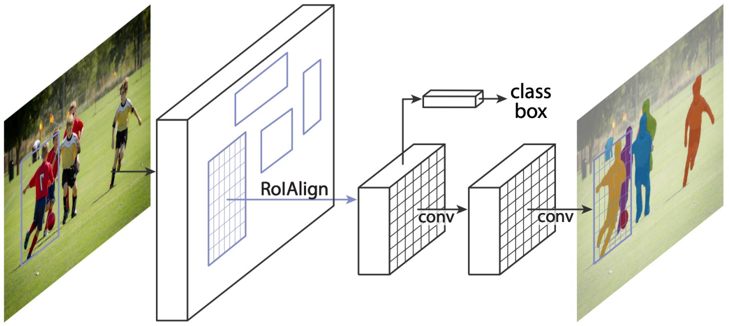
- Now add a MaskRCNN structure on top of the pretrained network.



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

Iterative Self-Training

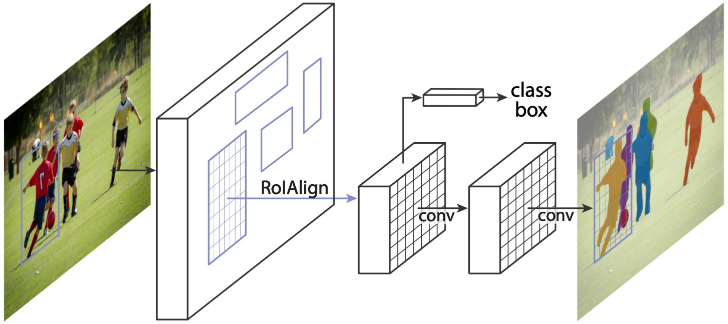
- Now add a MaskRCNN structure on top of the pretrained network.
- Select the predictions with the highest confidence score and use them as labels.



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

Iterative Self-Training

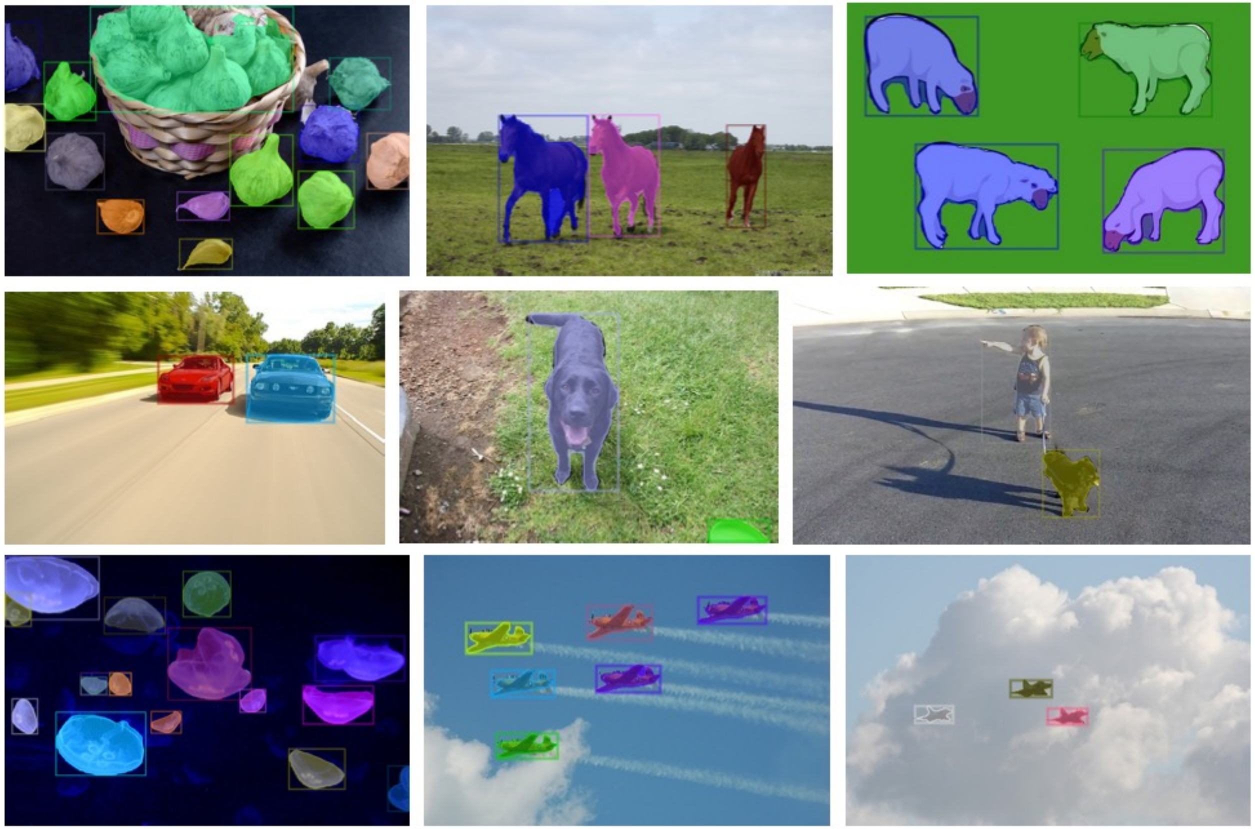
- Now add a MaskRCNN structure on top of the pretrained network.
- Select the predictions with the highest confidence score and use them as labels.
- Neural networks can learn from the noisy labels and output smoother predictions.



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

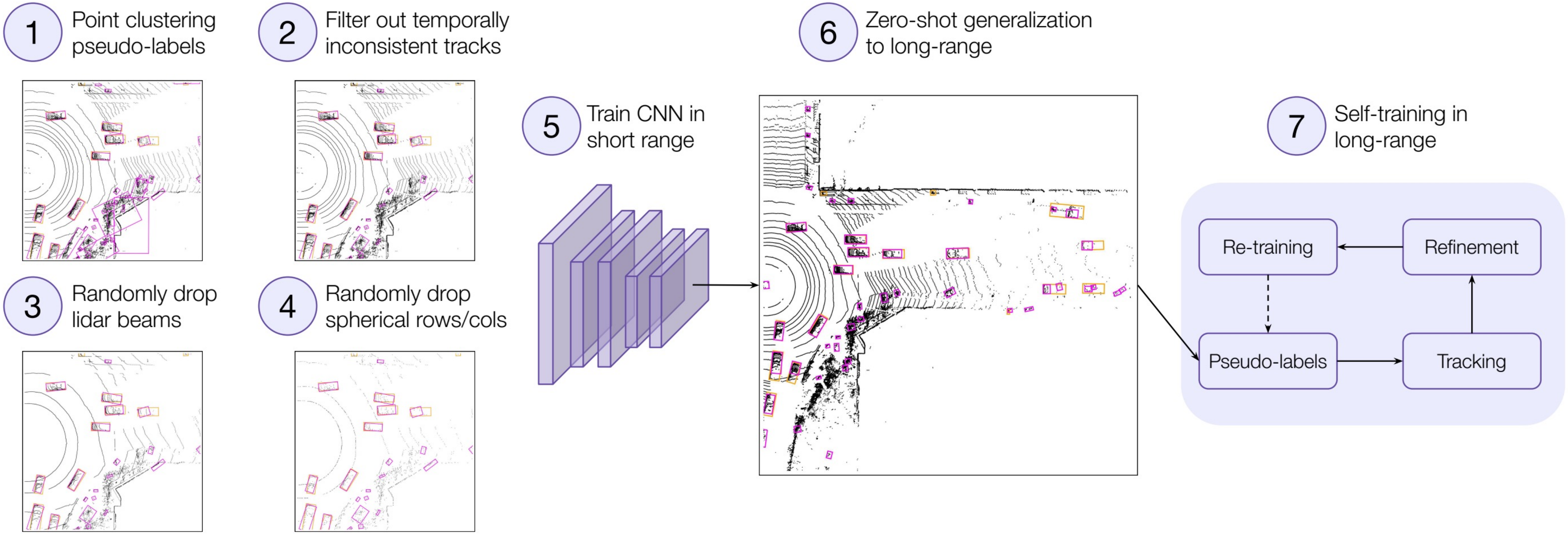
NYU

More Visualization



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

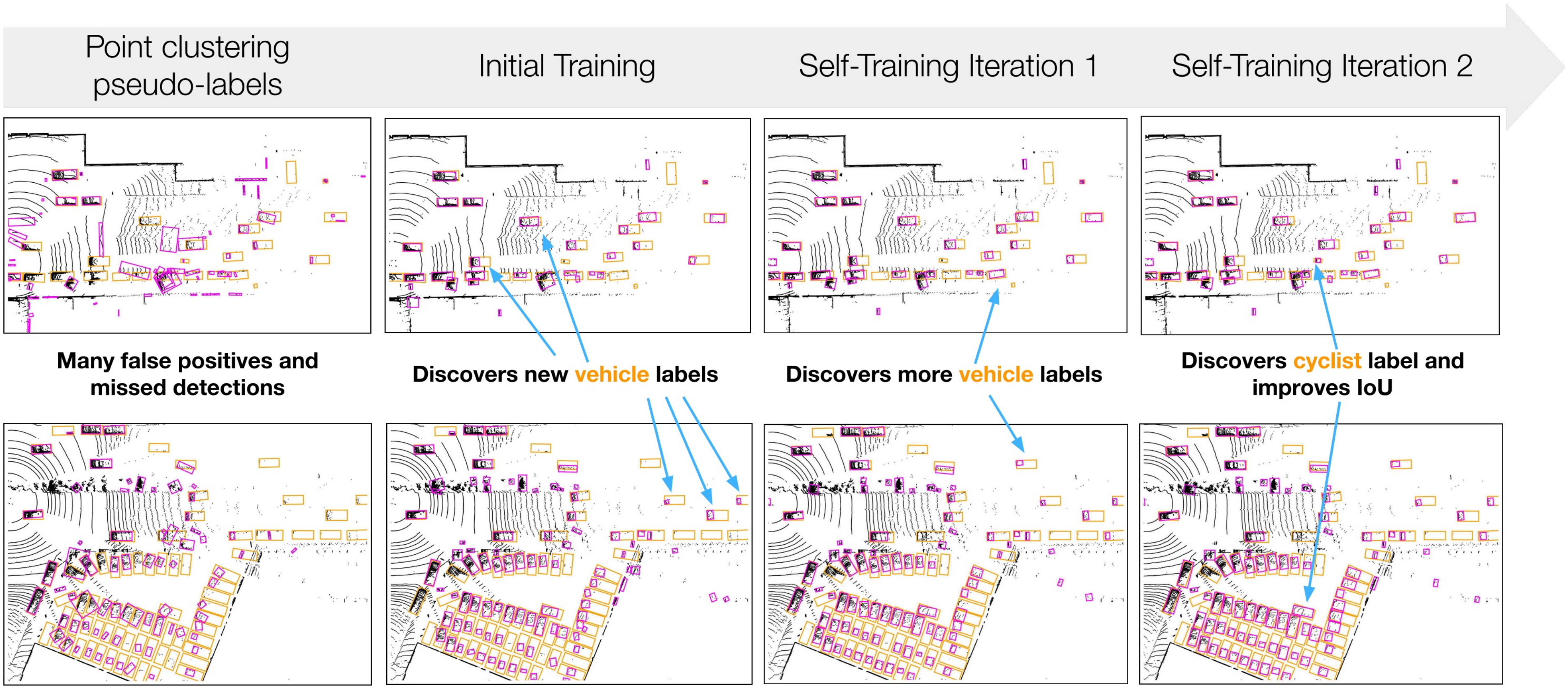
Pseudo Labels in 3D



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

NYU

Iterative Refinement of Pseudo Labels

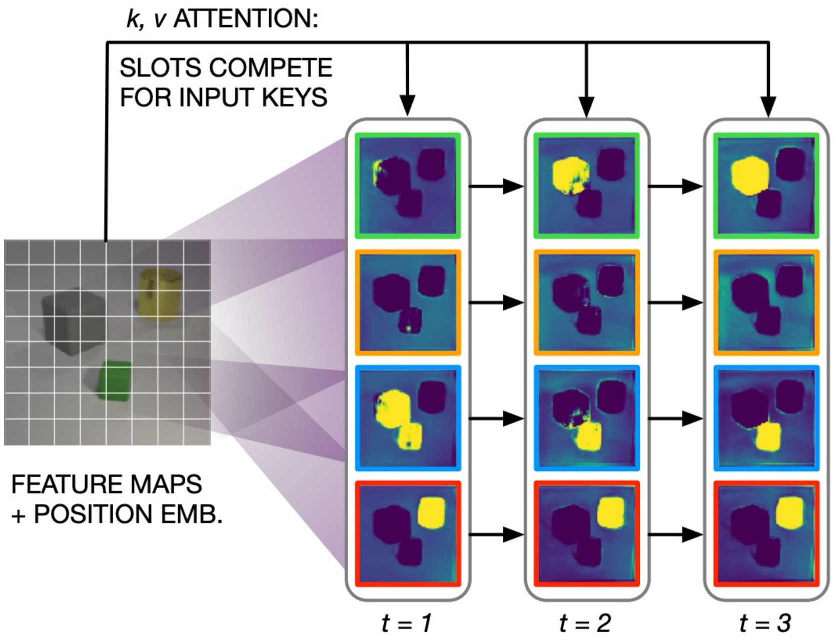


Zhang et al. Towards Unsupervised Object Detection from LiDAR Point Clouds. CVPR 2023.

NYU

Slot Attention Networks

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

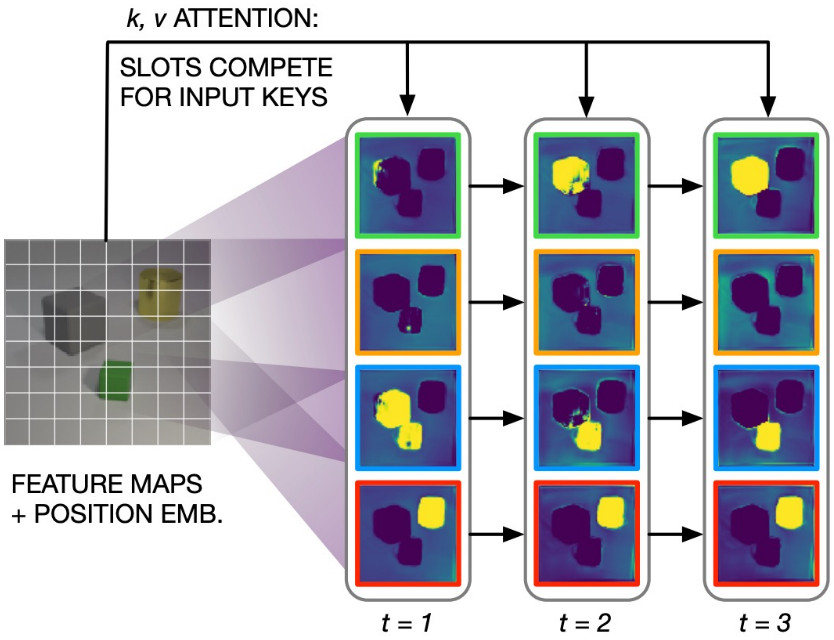


(a) Slot Attention module.

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

Slot Attention Networks

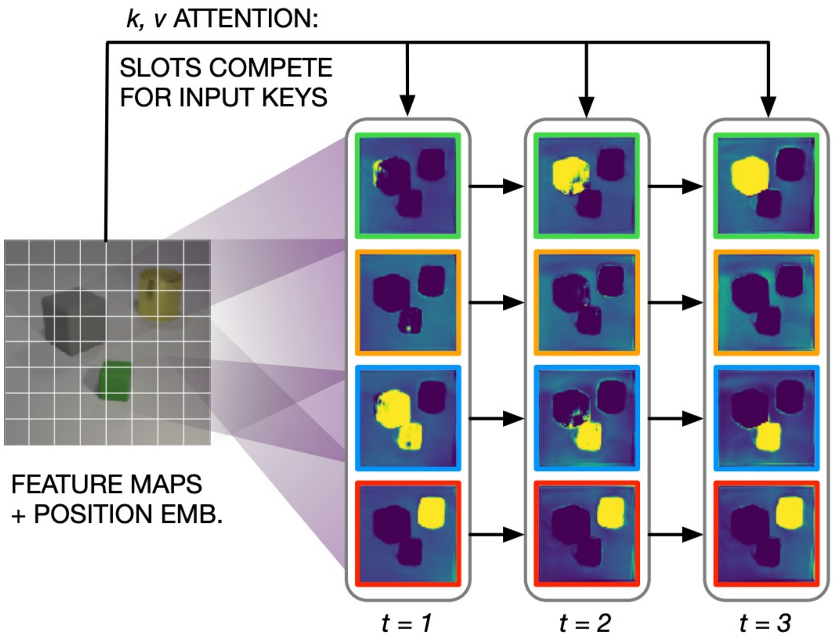
- Can we learn clustering as an end-to-end operation?
- Slot attention is inspired by the success of the attention mechanism.



(a) Slot Attention module.

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.



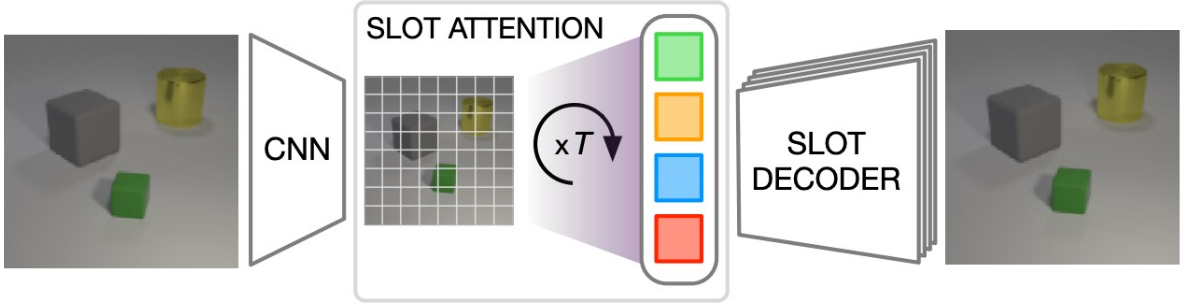
(a) Slot Attention module.

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

NYU

Slot Attention Networks

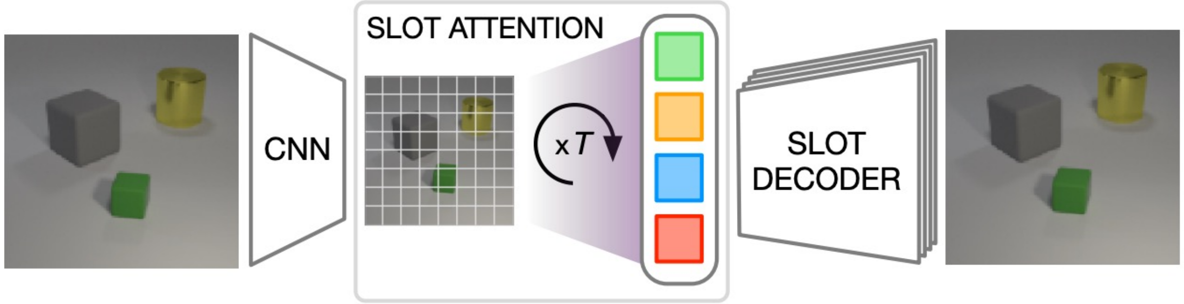
- Goal: Reconstruct the image with a concise slot-based representation.



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: $\tilde{m}_{t-1} = LN(m_{t-1})$.

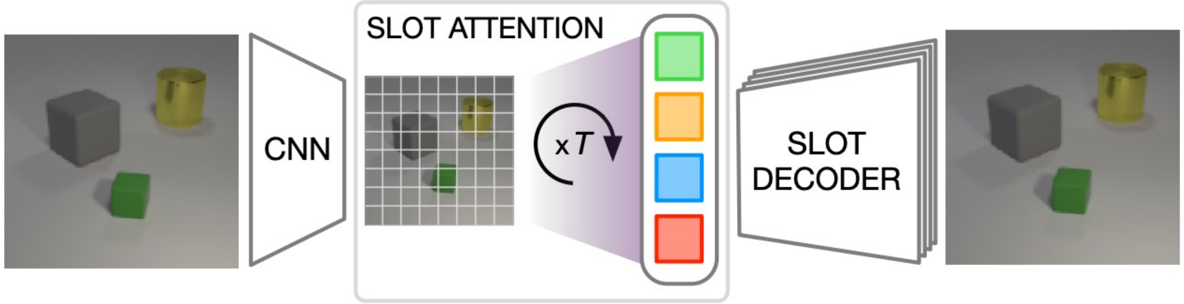


Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.

NYU

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: $a_{t,i,j} = \frac{\frac{1}{\sqrt{D}} k(x_i) \cdot q(\tilde{m}_j)^\top}{\sum_j \frac{1}{\sqrt{D}} k(x_i) \cdot q(\tilde{m}_j)^\top}$.

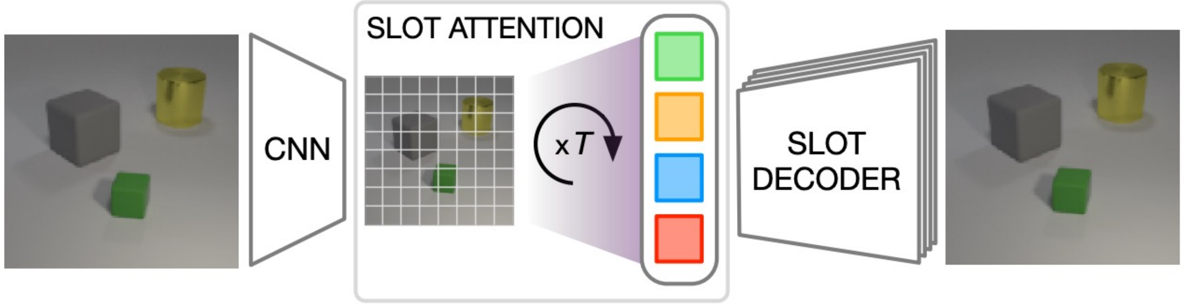


Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.

NYU

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: $u_{tj} = \sum_i a_{tij} v(x_i)$.

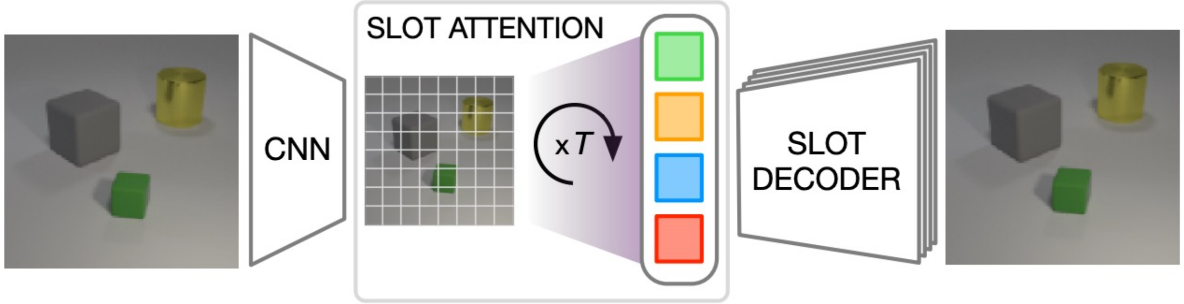


Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.

NYU

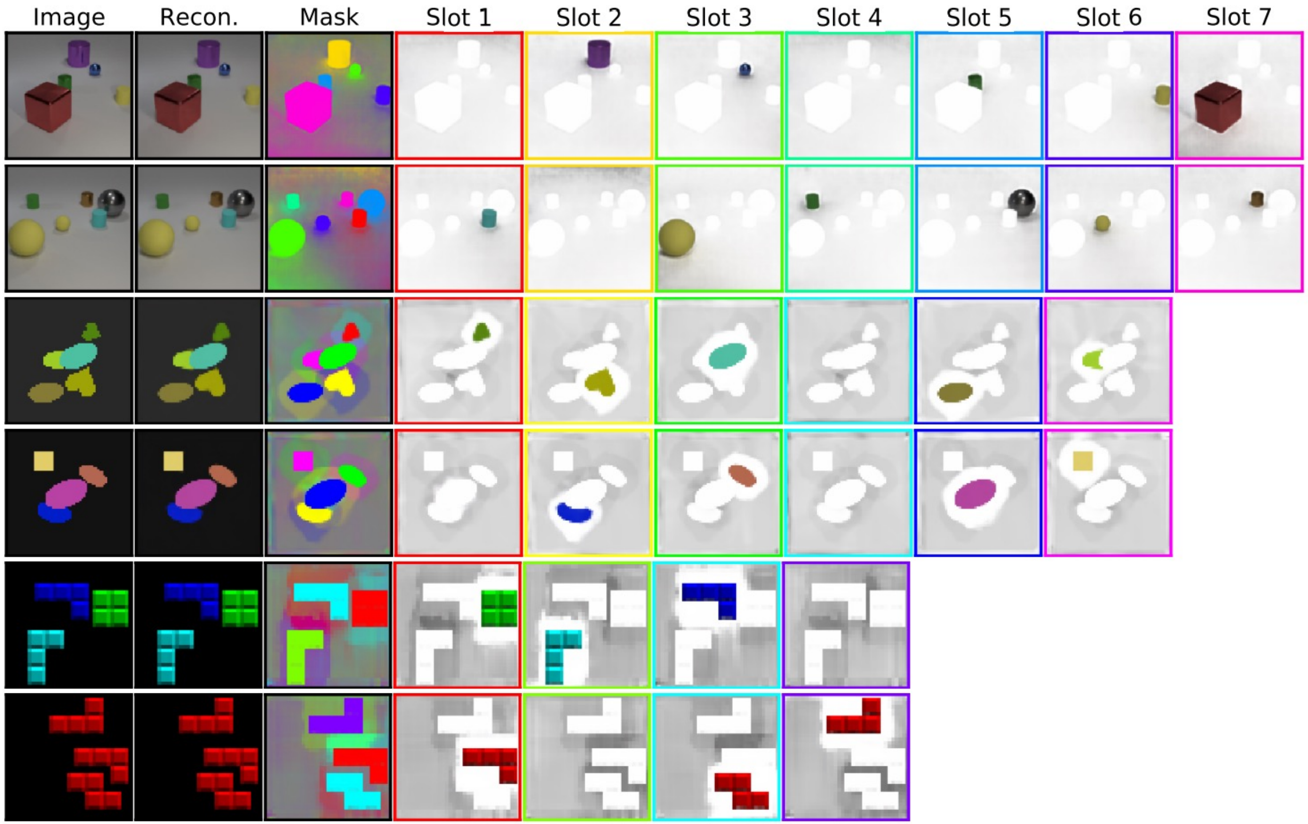
Slot Attention Networks

- 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: $u_{tj} = \sum_i a_{tij} v(x_i)$.
- 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

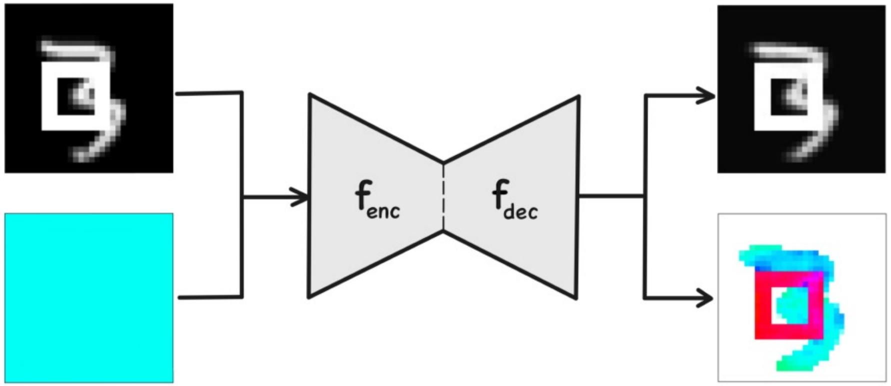


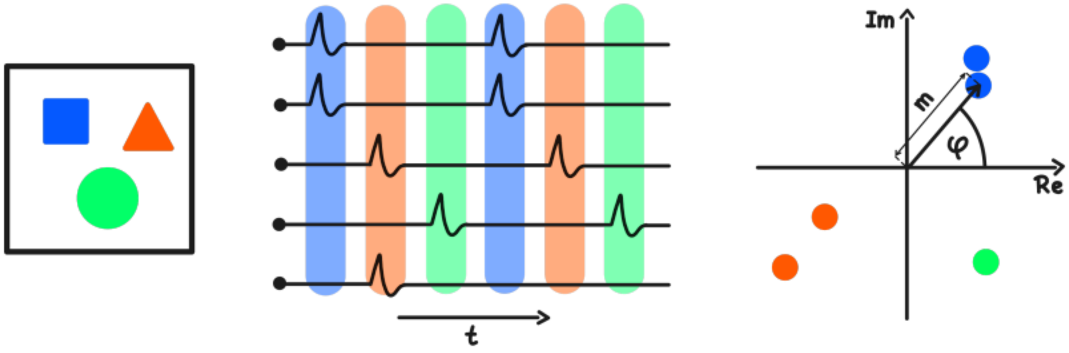
Locatello et al. Object-Centric Learning with Slot Attention. NeurIPS 2020.

NYU

Complex-Valued Autoencoders (CAEs)

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

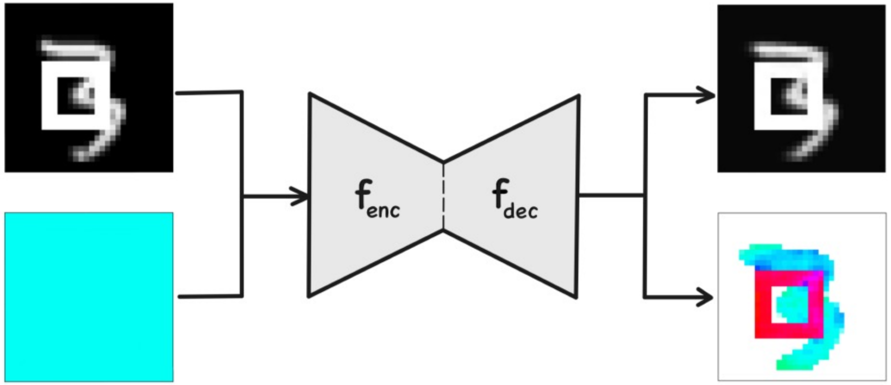


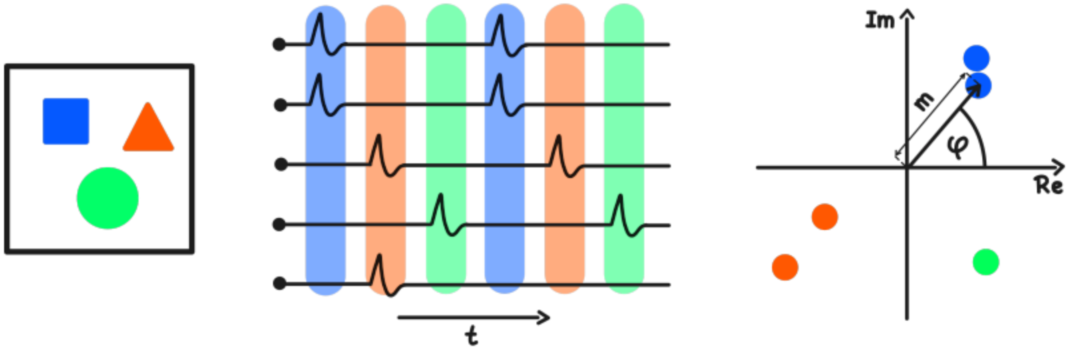


Löwe et al. *Complex-Valued Autoencoders for Object Discovery*. TMLR 2022.

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.



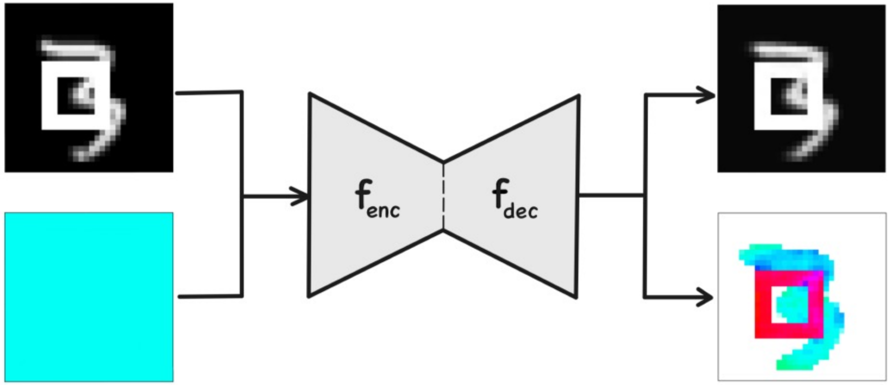


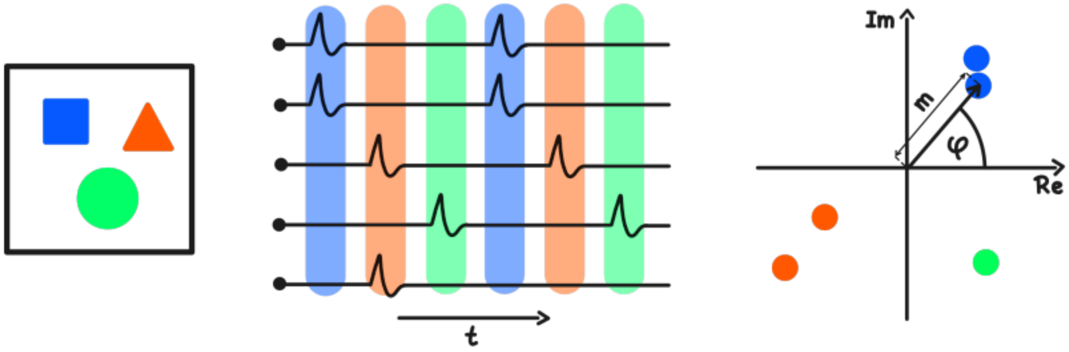
Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.

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.

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





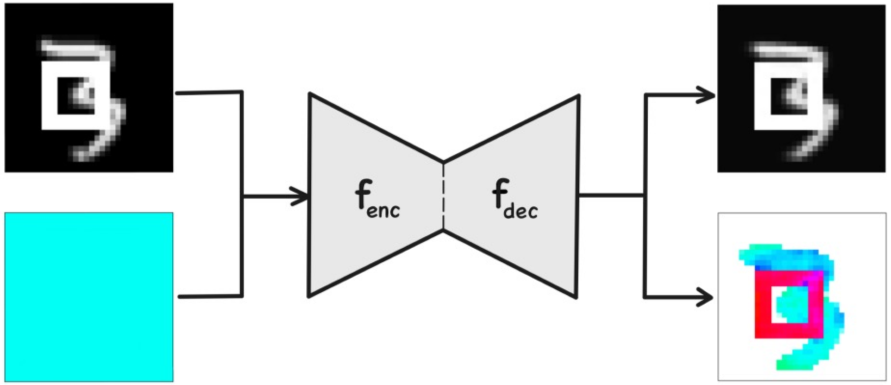
Löwe et al. Complex-Valued Autoencoders for Object Discovery. TMLR 2022.

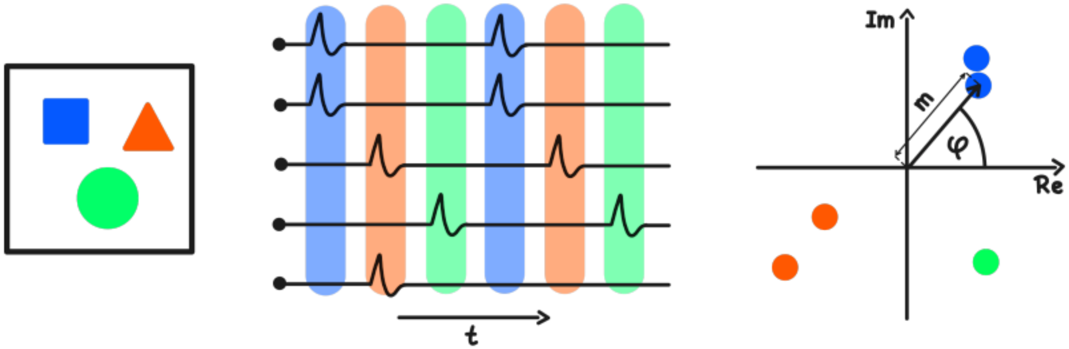
Complex-Valued Autoencoders (CAEs)

- 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.

NYU

CAE: More Details

- Apply weights separately to real and imaginary:

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

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

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

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

NYU

CAE: More Details

- Apply weights separately to real and imaginary:

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

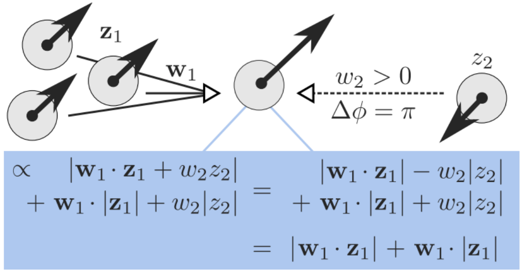
- Bias on magnitude and phase:

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

- Gating:

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

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



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

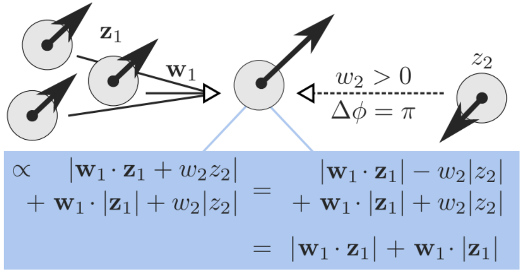
CAE: More Details

- 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:

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

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

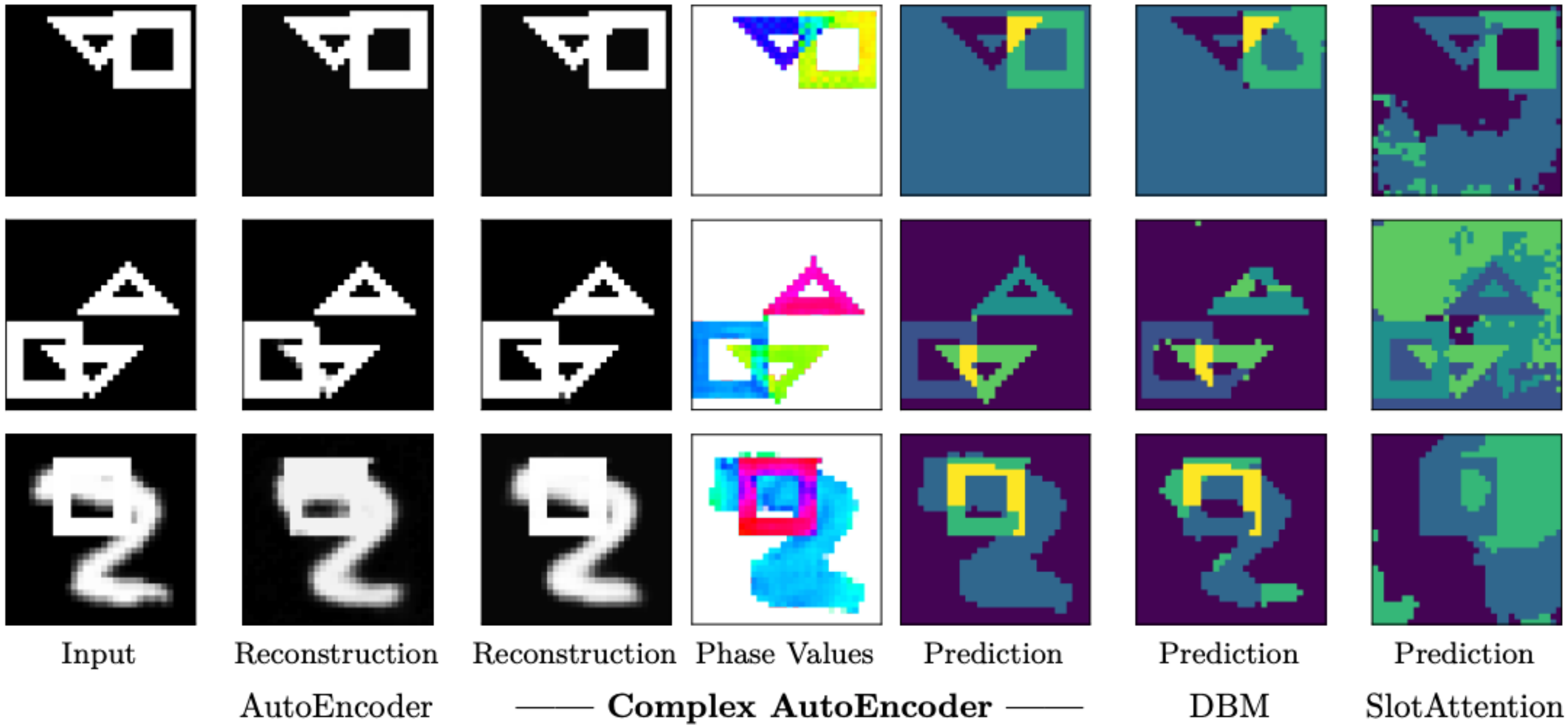
- Activation

$$\mathbf{z}' = \mathrm{ReLU}(\mathrm{BatchNorm}(\boldsymbol{m}_{\mathbf{z}})) \circ e^{i\boldsymbol{\varphi}_{\boldsymbol{\psi}}} \quad \in \mathbb{C}^{d_{\text{out}}}$$

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

NYU

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.

NYU

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!

NYU

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?

NYU