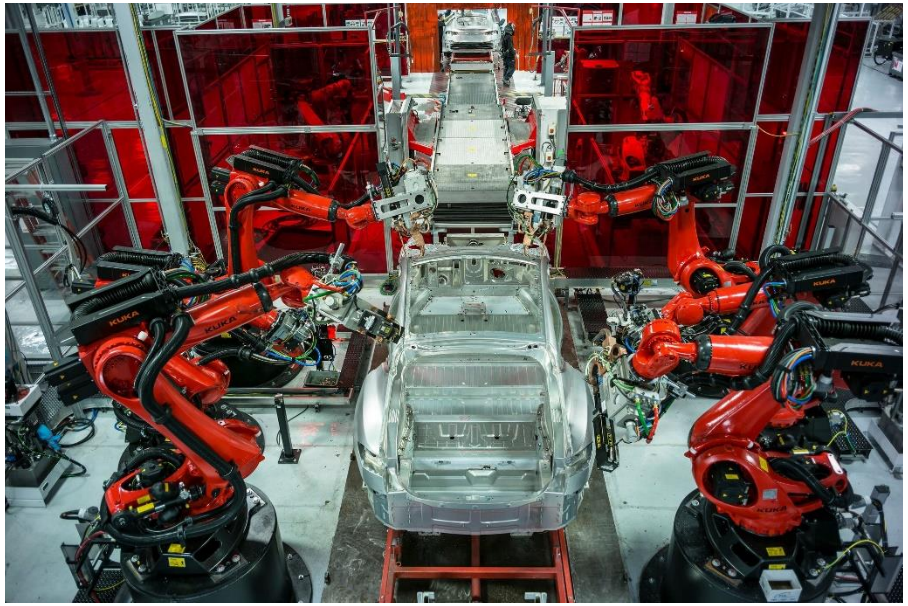
Segmenting Unseen Objects for Robotic Grasping

Yu Xiang
Assistant Professor
Computer Science
The University of Texas at Dallas

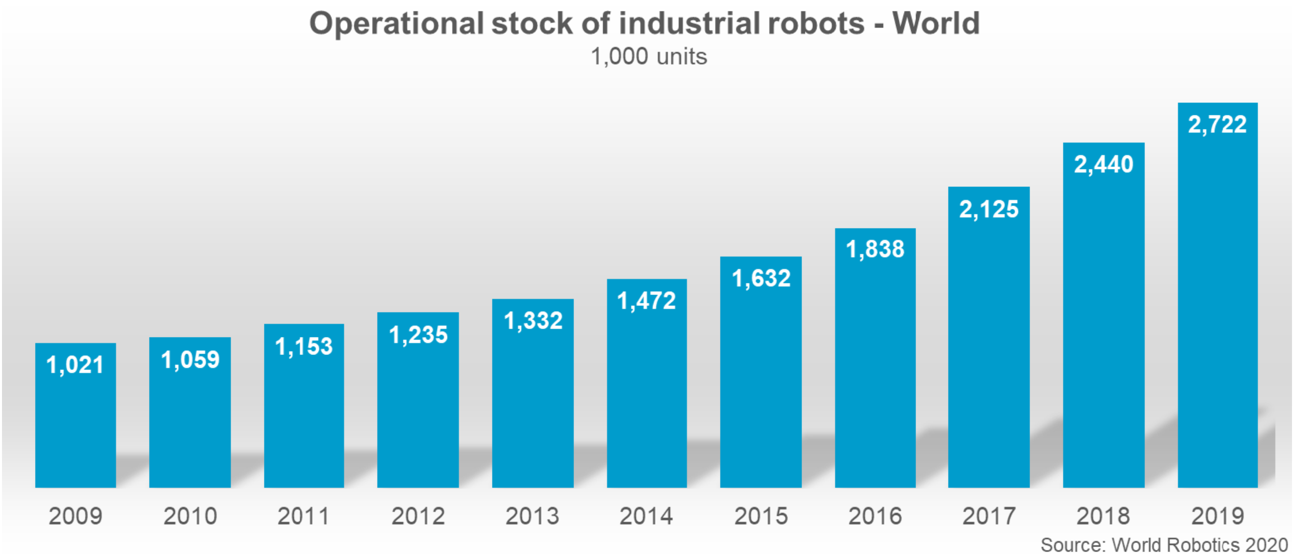
Robots in Factories and Warehouses



Welding and Assembling

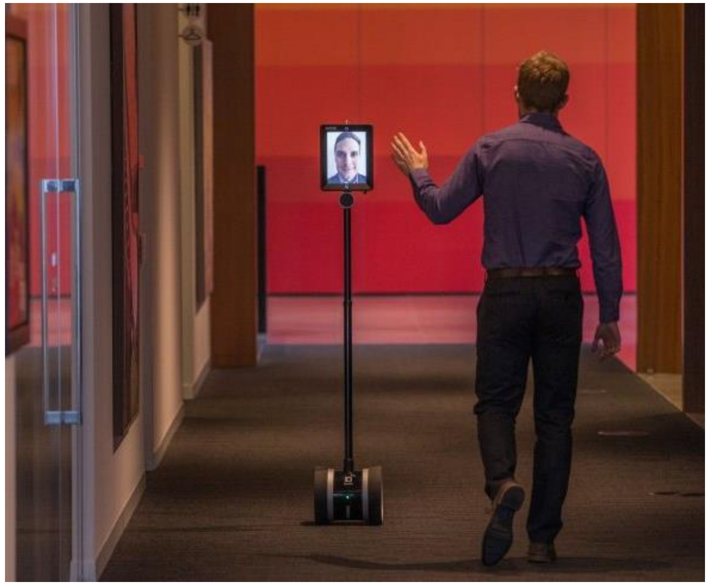
Material Handling

Delivering



Current Robots in Human Environments

Cleaning Robots



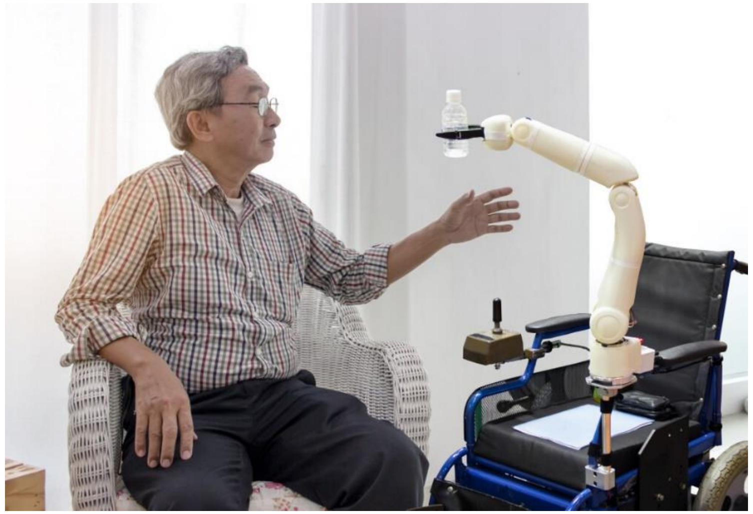
Telepresence Robots

Smart Speakers

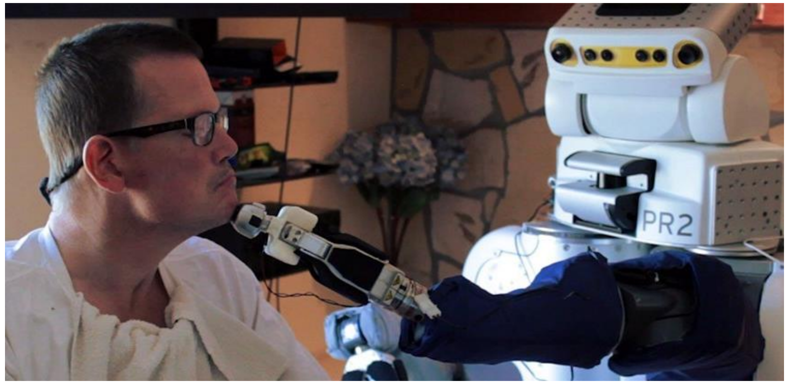
How can we have more powerful robots assisting people at homes or offices?

- Mobile manipulators
- Humanoids

Future Intelligent Robots in Human Environments



Senior Care



Assisting

Serving

Cooking

Cleaning

Dish washing

Robot Manipulation

Assembling

Cooking

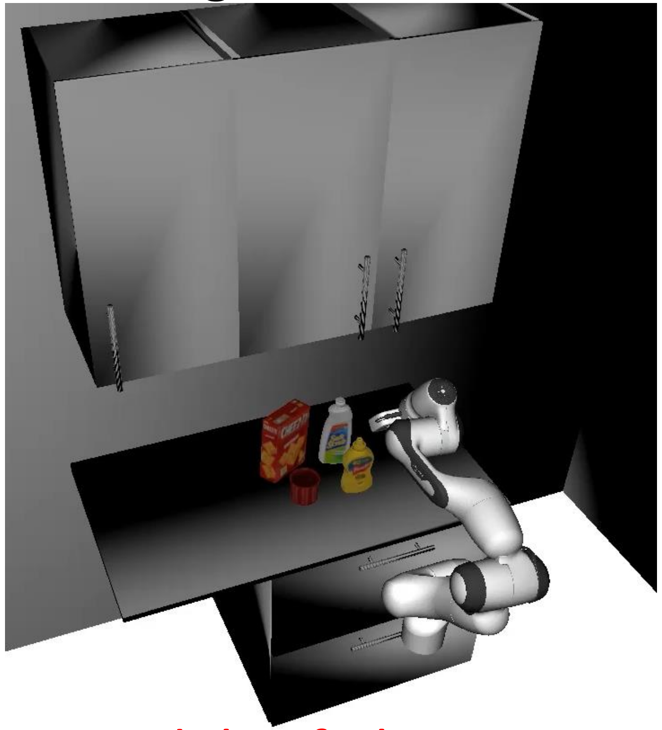
Model-based Robotic Grasping

Perception → **Planning** → **Control**

6D Object Pose Estimation | Motion and Grasp Planning | Position Control

Sensed image

Planning scene



Real world execution

We need to have 3D models of objects

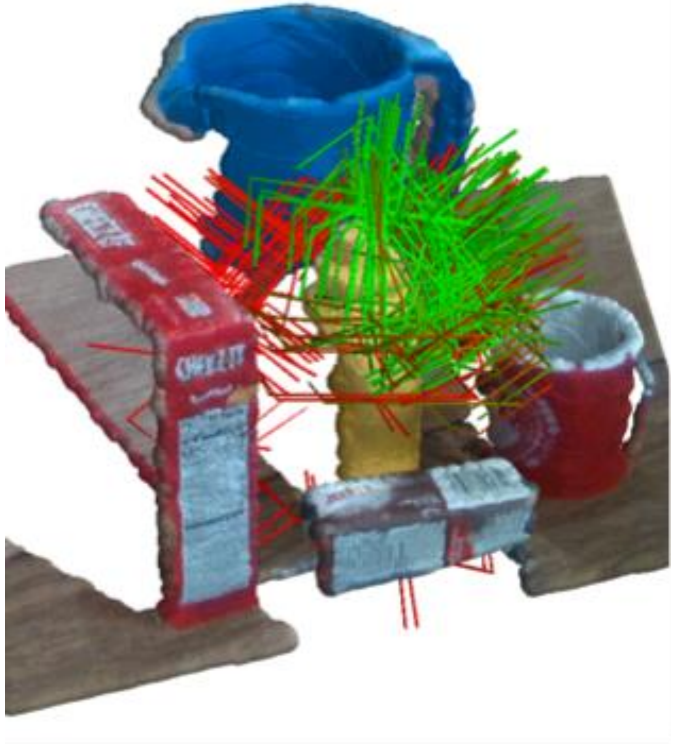
Robots in Unstructured Environments

How can a robot manipulate objects in this cluttered kitchen?

Model-free Robotic Grasping

Perception → Planning → Control

Unseen object instance segmentation



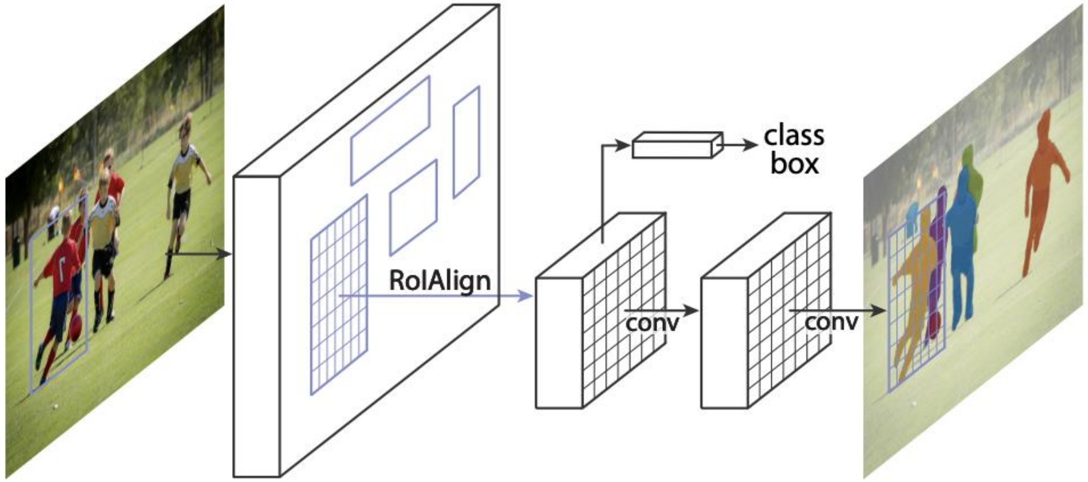
Grasp planning from point clouds

Position control to reach grasp

Figure Credit: Murali-Mousavian-Eppner-Paxton-Fox, ICRA'20

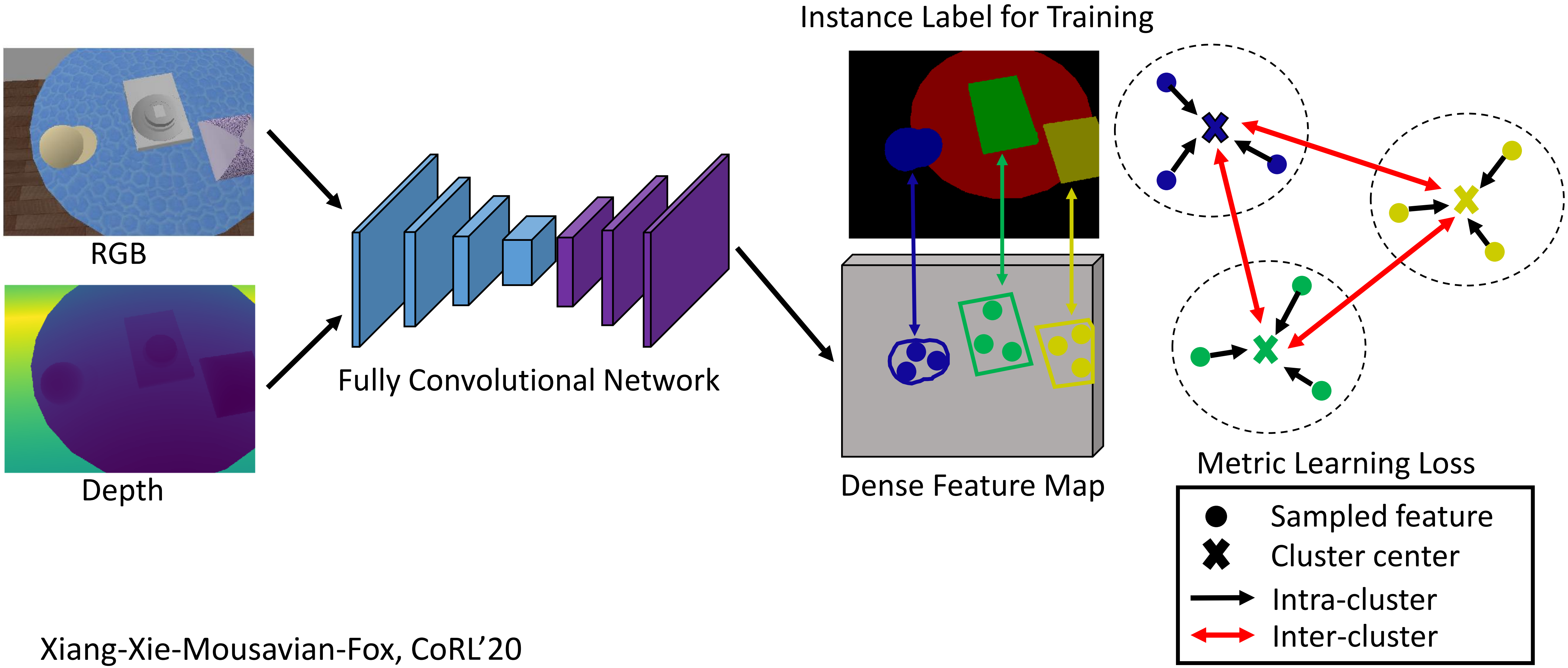
Unseen Object Instance Segmentation

- Top-down approaches
  - Mask R-CNN (objects vs. background)
  - UOAIS-Net (Back et al. ICRA’22)



- Bottom-up approaches
  - UOIS-Net (predicting object centers) Xie et al. CoRL’19, T-RO’21
  - UCN (feature learning + mean shift clustering) Xiang et al. CoRL’20
  - Fully Test-time RGBD Embeddings Adaptation (FTEA) Zhang et al. arXiv’23

Unseen Object Instance Segmentation: Learning RGB-D Feature Embeddings



Xiang-Xie-Mousavian-Fox, CoRL'20

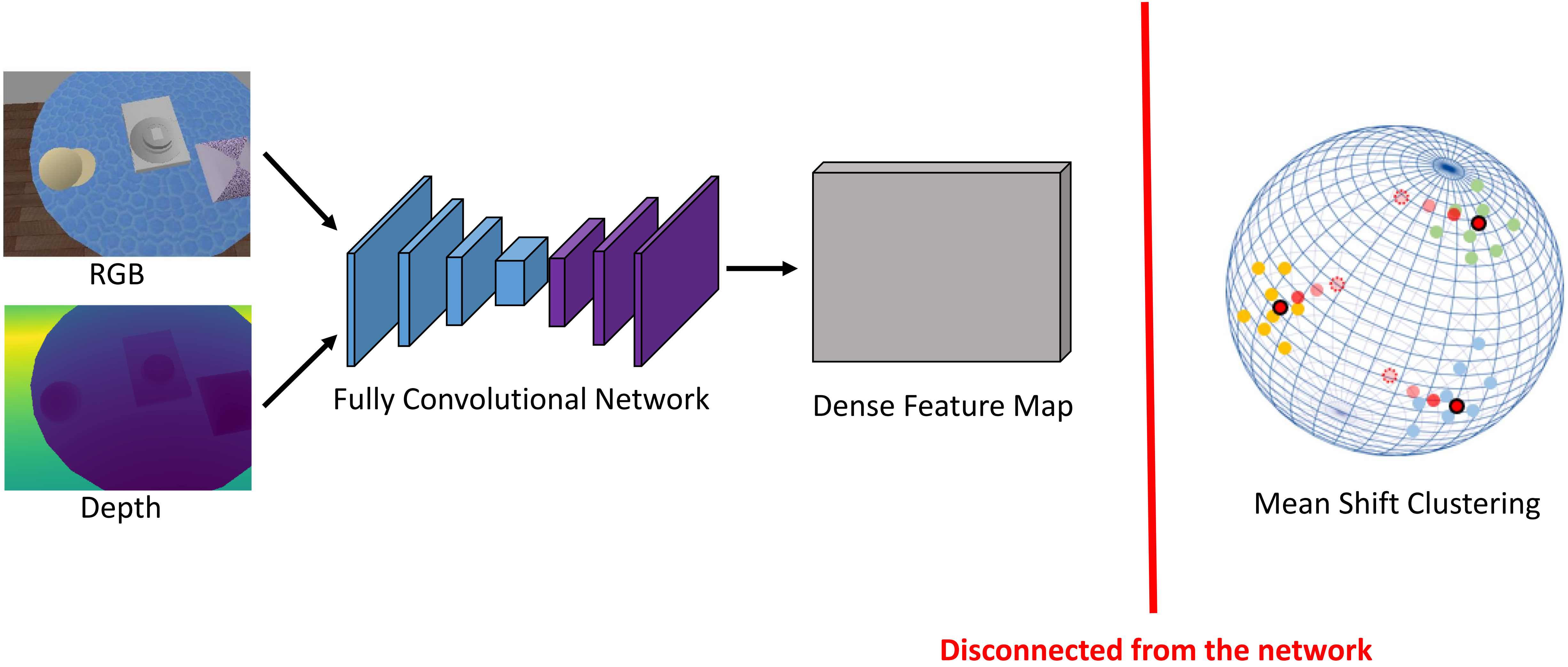
von Mises-Fisher (vMF) Mean Shift Clustering

- Input data points $\mathbf{X} \in \mathbb{R}^{n \times C}$ Unit length vectors
- Sample m initial clustering centers using furthest point sampling

$$\mu^{(0)} \in \mathbb{R}^{m \times C}$$

- For each of the T iterations
  - Compute weight matrix $\underset{m \times n}{\mathbf{W}} \leftarrow \exp(\kappa \mu^{(t-1)} \mathbf{X}^T)$
  - Update clustering centers $\underset{m \times C}{\mu^{(t)}} \leftarrow \mathbf{W}\mathbf{X}$ Normalize each row
- Merge clustering centers with cosine distance smaller than ϵ

Mean Shift Clustering is Non-Differentiable



Can we learn a differentiable clustering module jointly with the image feature embeddings?

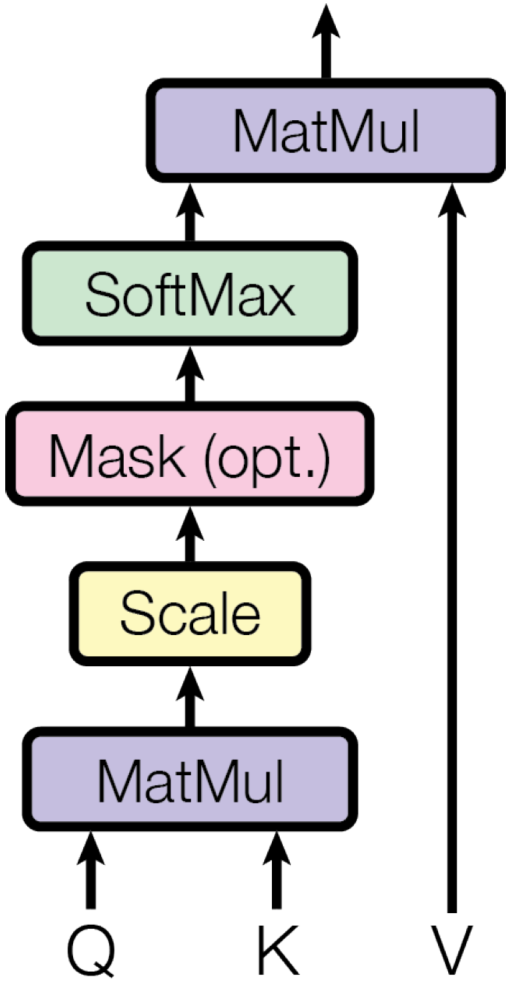
Transformer: Attention

- Scaled Dot-Product Attention
  - Keys $K : m \times d_k$
  - Values $V : m \times d_v$
  - n queries $Q : n \times d_k$

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$n \times d_v$$

weights



Attention is all you need. Vaswani et al., NeurIPS'17

vMF Mean Shift vs. Scaled Dot-Product Attention

- vMF mean shift updating rule

$$\mu^{(t)} \leftarrow \exp(\kappa\mu^{(t-1)}\mathbf{X}^T)\mathbf{X}$$

- Scaled Dot-Product Attention

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Query Q as clustering centers $\mu^{(t)} \in \mathbb{R}^{m \times C}$

Keys and values as data points $\mathbf{X} \in \mathbb{R}^{n \times C}$

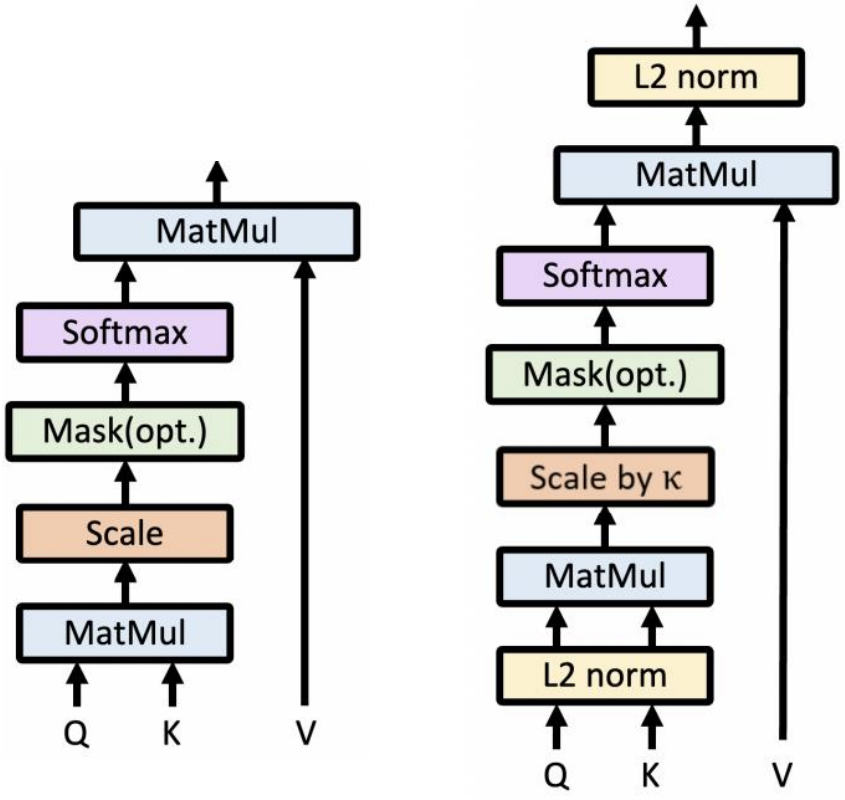
Our Proposed Hypersphere Attention

- Hypersphere Attention

$$\mathrm{HSAtten}(\mathbf{Q},\mathbf{K},\mathbf{V}) = g(\mathrm{softmax}(\kappa g(\mathbf{Q}) g(\mathbf{K})^T)\mathbf{V}) \qquad g(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

scaled dot-product attention

$$\mathrm{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V}$$



Our Masked Mean Shift Cross-Attention

$$\mu_l = \mu_{l-1} + g(\text{softmax}(\mathcal{M}_{l-1} + \kappa g(\mathbf{Q}_l) g(\mathbf{K}_l)^T)\mathbf{V}_l)$$

$\mu_l \in \mathbb{R}^{m \times C}$ Clustering centers at layer l

$$g(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|}$$

Query $\mathbf{Q}_l = f_Q(\mu_{l-1}) \in \mathbb{R}^{m \times C}$

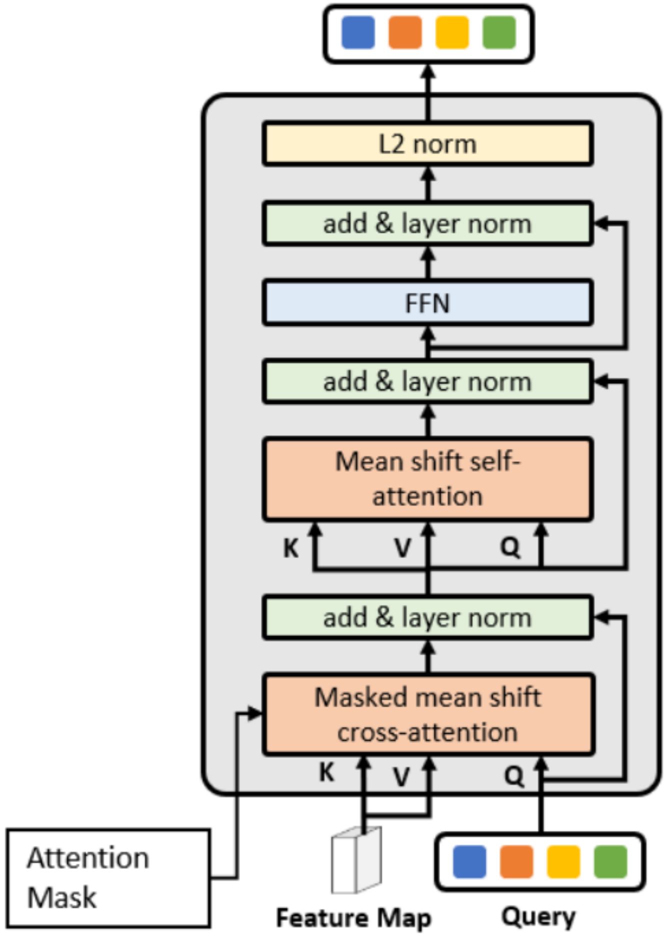
Key, Value $\mathbf{K}_l, \mathbf{V}_l \in \mathbb{R}^{H_l W_l \times C}$ Pixel embeddings

Attention mask $\mathcal{M}_{l-1}(x, y) = \begin{cases} 0 & \text{if } \mathrm{M}_{l-1}(x, y) = 1 \\ -\infty & \text{otherwise} \end{cases}$

Mask prediction $M_{l-1} \in \{0, 1\}^{m \times H_l W_l}$

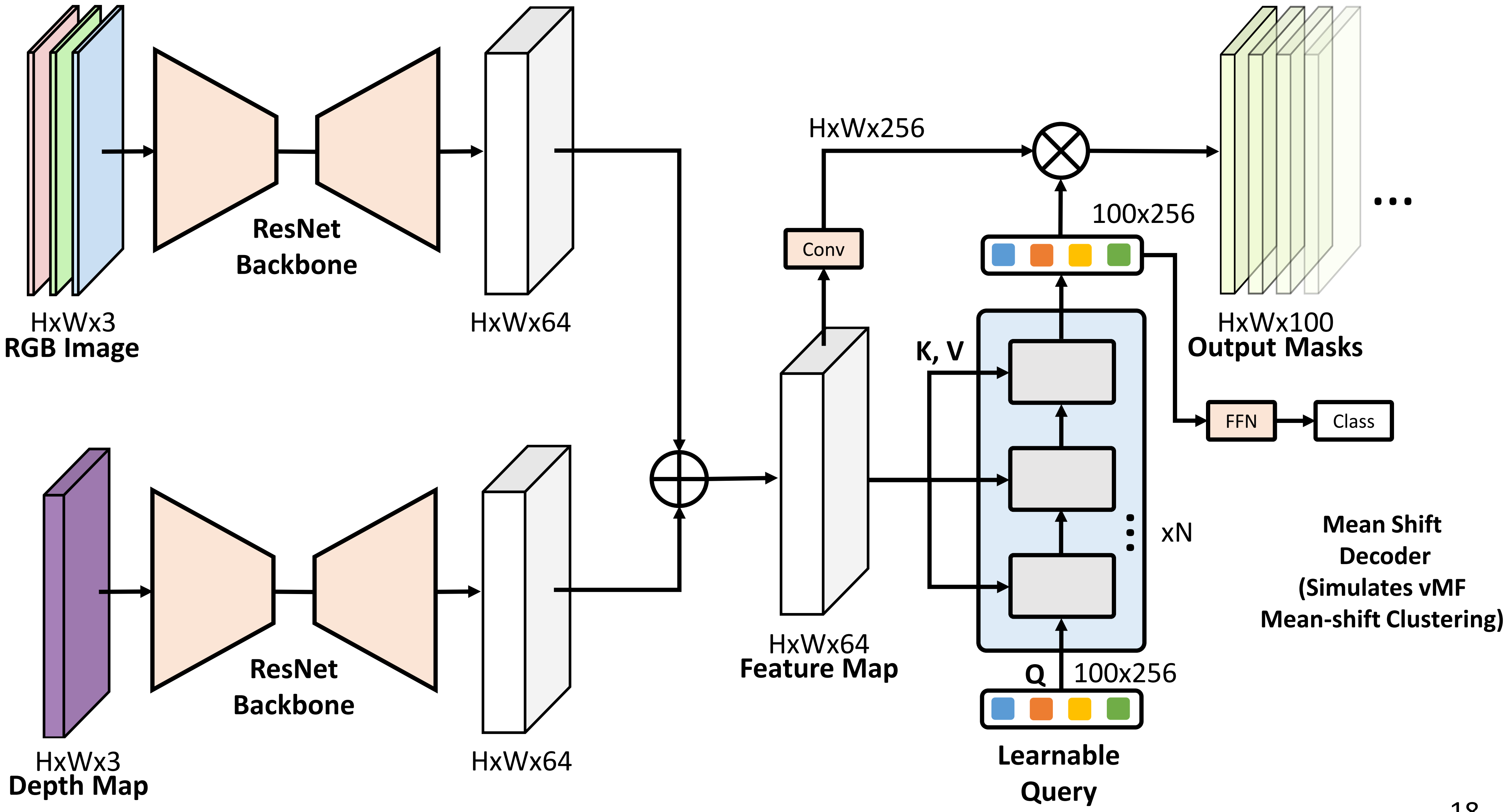
Our Mean Shift Decoder Layer

$$\mu_l = \mu_{l-1} + g(\text{softmax}(\mathcal{M}_{l-1} + \kappa g(\mathbf{Q}_l)g(\mathbf{K}_l)^T)\mathbf{V}_l)$$

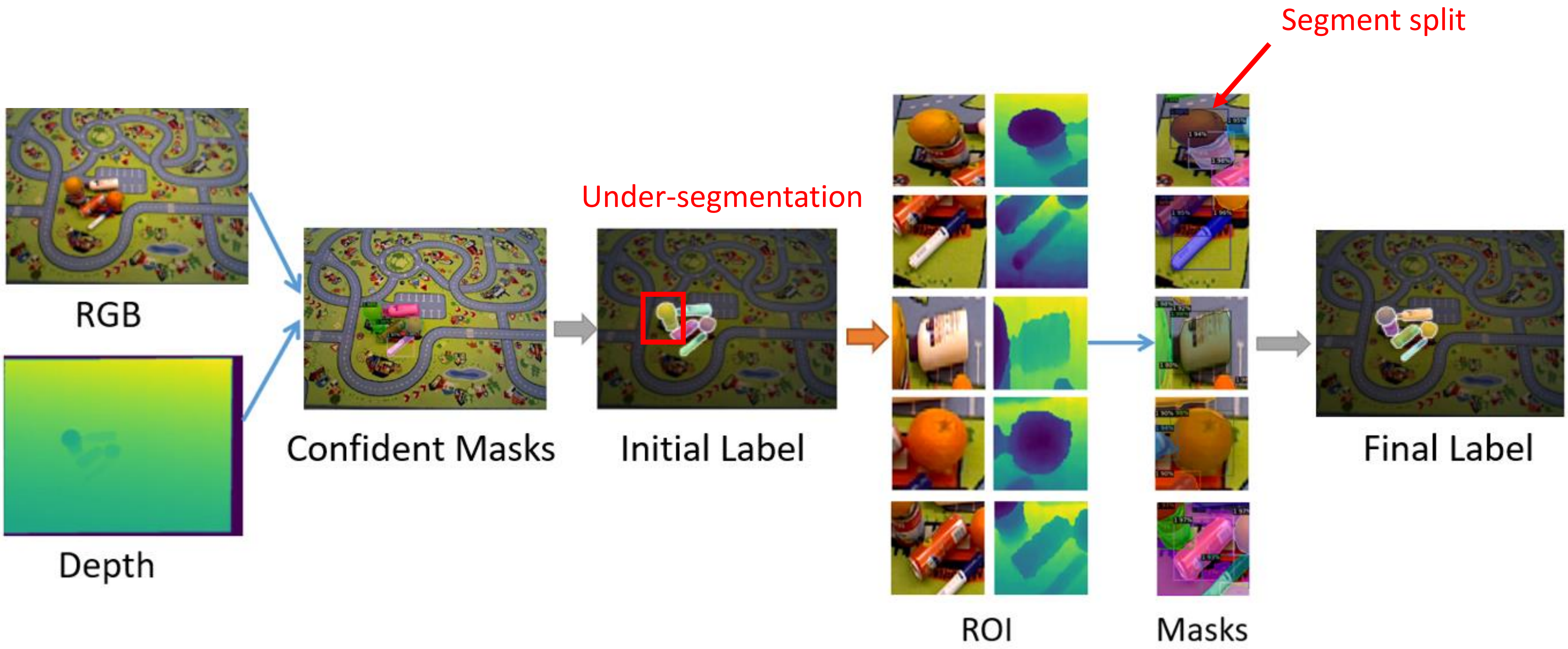


Our Mean Shift Mask Transformer

Can be trained end-to-end



Two-stage Segmentation

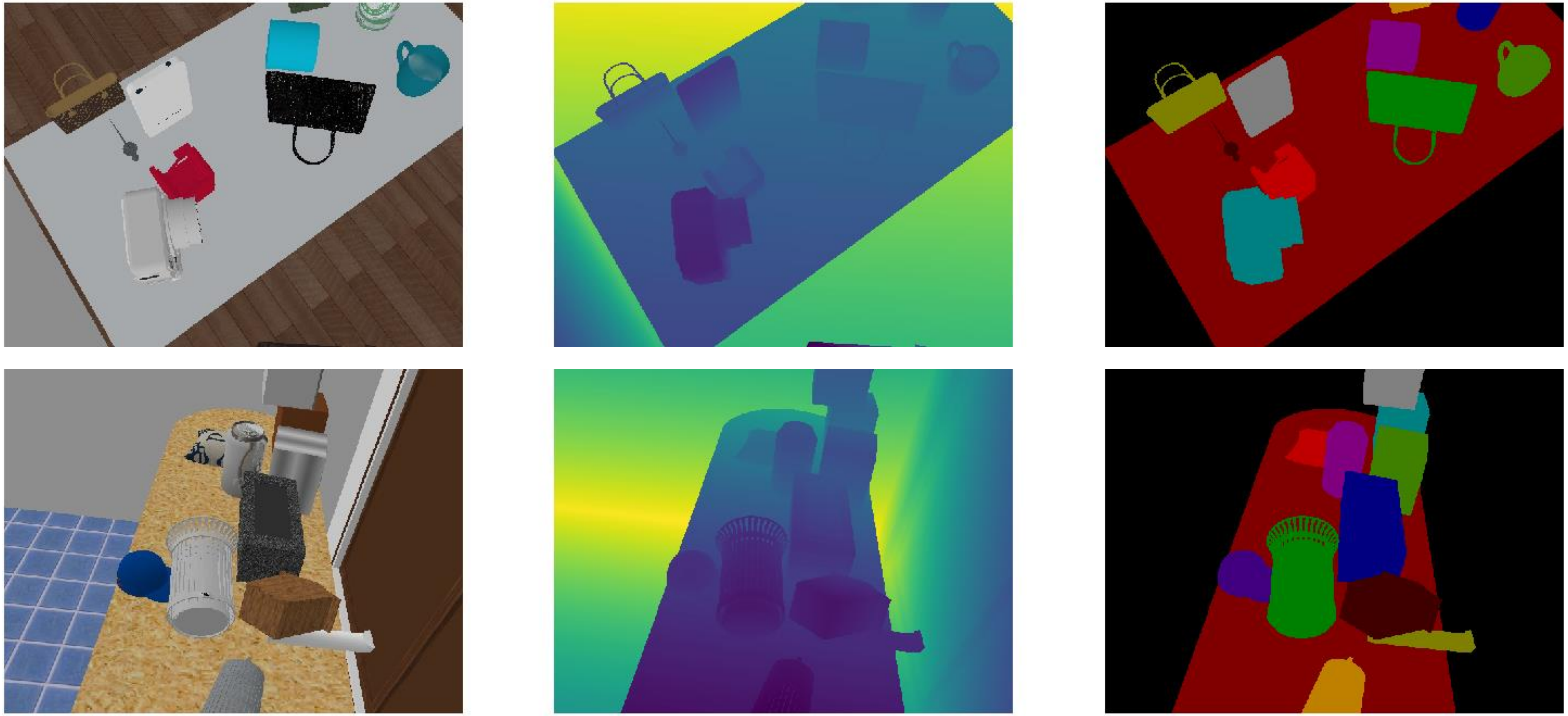


Experiments: Testing Datasets

- Object Cluster Indoor Dataste (OCID), 2,390 RGB-D images Sushi et al. ICRA'19

- Object Segmentation Database (OSD), 111 RGB-D images Richtsfeld et al. IROS'12

Experiments: Learning from Synthetic Data



RGB　　Depth　　Instance Label

40,000 scenes
7 RGB-D images per scene

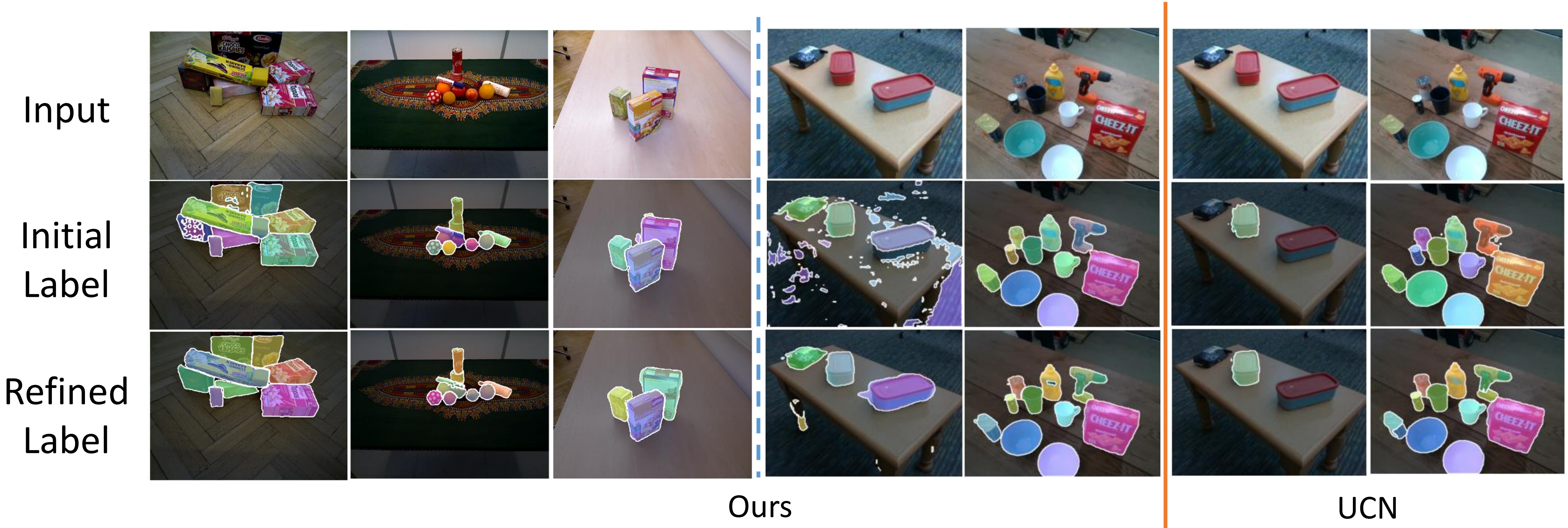
ShapeNet objects in the PyBullet simulator

Xie et al. CoRL'19

Experimental Results

Method	Input	OCID (2390 images)							OSD (111 images)						
		Overlap			Boundary				Overlap			Boundary			
		P	R	F	P	R	F	%75	P	R	F	P	R	F	%75
MRCNN [14]	RGB	**77.6**	67.0	67.2	**65.5**	53.9	54.6	55.8	**64.2**	61.3	62.5	50.2	40.2	44.0	31.9
UCN [40]	RGB	54.8	**76.0**	59.4	34.5	45.0	36.5	48.0	57.2	**73.8**	63.3	34.7	50.0	39.1	52.5
UCN+ [40]	RGB	59.1	74.0	61.1	40.8	55.0	43.8	**58.2**	59.1	71.7	**63.8**	34.3	**53.3**	39.5	**52.6**
Mask2Former [5]	RGB	67.2	73.1	67.1	55.9	**58.1**	54.5	54.3	60.6	60.2	59.5	48.2	41.7	43.3	32.4
MSMFormer (Ours)	RGB	72.9	68.3	**67.7**	60.5	56.3	**55.8**	52.9	63.4	64.7	63.6	48.6	47.4	**47.0**	40.2
MSMFormer+ (Ours)	RGB	73.9	67.1	66.3	64.6	52.9	54.8	52.8	63.9	63.7	62.7	**51.6**	45.3	**47.0**	41.1
MRCNN [14]	Depth	85.3	85.6	84.7	83.2	76.6	78.8	72.7	77.8	85.1	80.6	52.5	57.9	54.6	77.6
UOIS-Net-2D [42]	Depth	88.3	78.9	81.7	82.0	65.9	71.4	69.1	80.7	80.5	79.9	66.0	67.1	65.6	71.9
UOIS-Net-3D [43]	Depth	86.5	86.6	86.4	80.0	73.4	76.2	77.2	85.7	82.5	83.3	**75.7**	68.9	71.2	73.8
UCN [40]	RGBD	86.0	92.3	88.5	80.4	78.3	78.8	82.2	84.3	**88.3**	86.2	67.5	67.5	67.1	79.3
UCN+ [40]	RGBD	91.6	**92.5**	**91.6**	86.5	**87.1**	86.1	**89.3**	**87.4**	87.4	**87.4**	69.1	70.8	69.4	**83.2**
UOAIS-Net [1]*	RGBD	70.7	86.7	71.9	68.2	78.5	68.8	78.7	85.3	85.4	85.2	72.7	**74.3**	**73.1**	79.1
Mask2Former [5]	RGBD	78.6	82.8	79.5	69.3	76.2	71.1	69.3	75.6	79.2	77.3	54.1	64.0	58.0	65.2
MSMFormer (Ours)	RGBD	88.4	90.2	88.5	84.7	83.1	83.0	80.3	79.5	86.4	82.8	53.5	71.0	60.6	79.4
MSMFormer+ (Ours)	RGBD	**92.5**	91.0	91.5	**89.4**	85.9	**87.3**	86.0	87.1	86.1	86.4	69.0	68.6	68.4	80.4

Segmentation Examples



UCN: Xiang-Xie-Mousavian-Fox, CoRL’20

Failure Cases

Under-segmentation

Over-segmentation

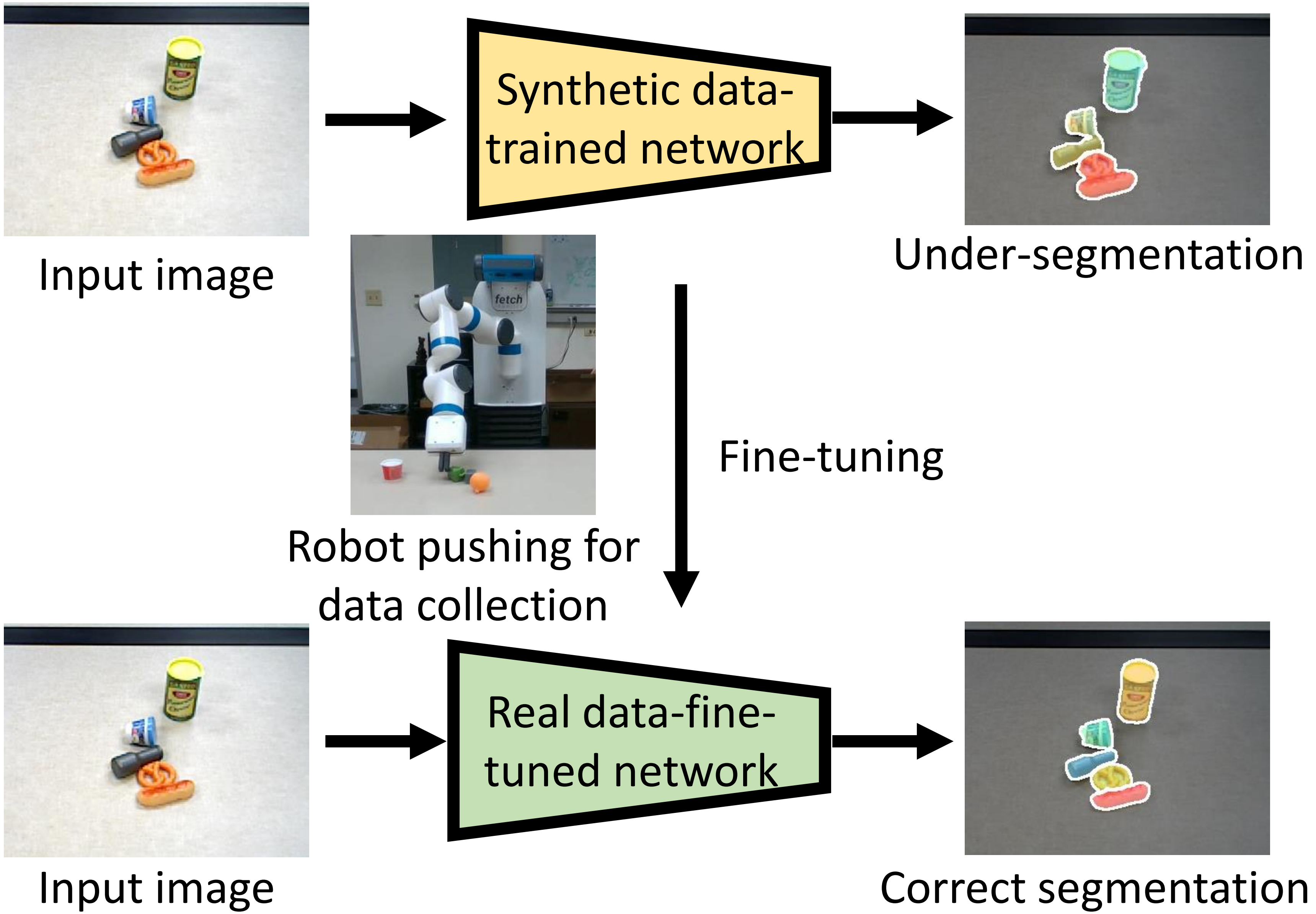
How Can We Fix These Failures?

- Better models
  - Swin Transformers
  - OpenAI CLIP
  - ?

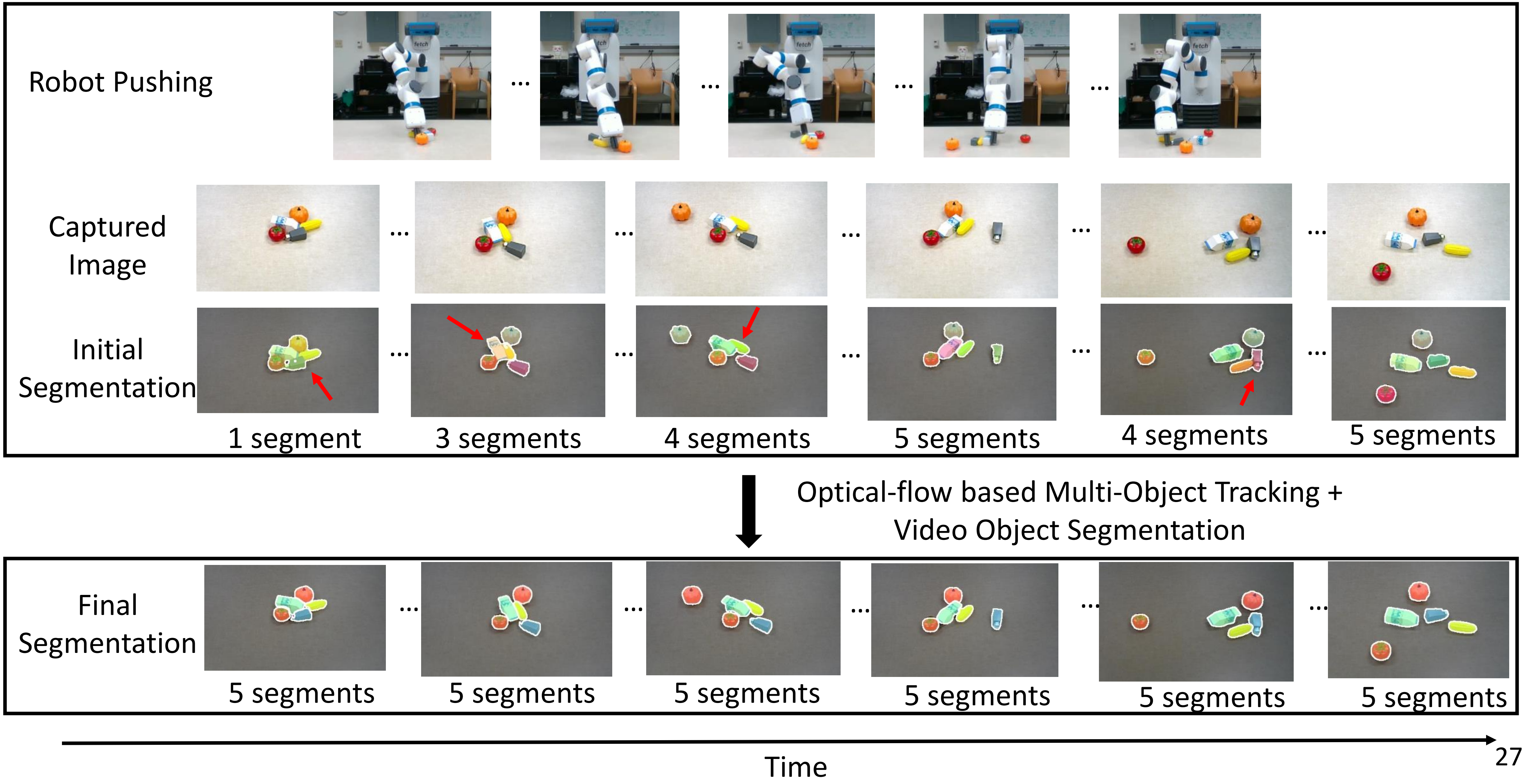
- Better training data
  - Photo-realistic synthetic data
  - Real-world data
    (How can we obtain real-world data for training?)

UOAIS-Net (Back et al. ICRA'22)

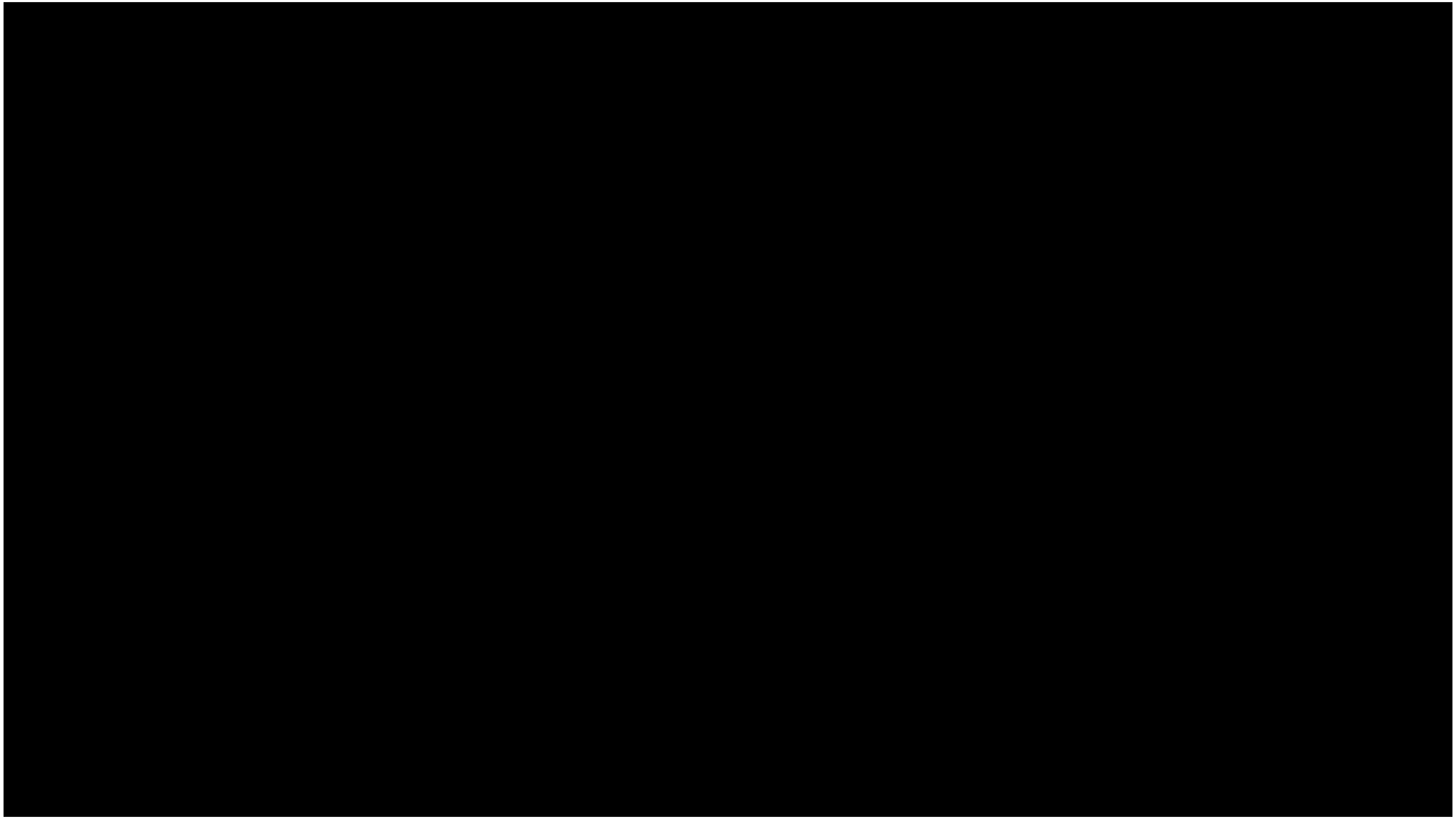
Self-supervised Segmentation with Robot Interaction



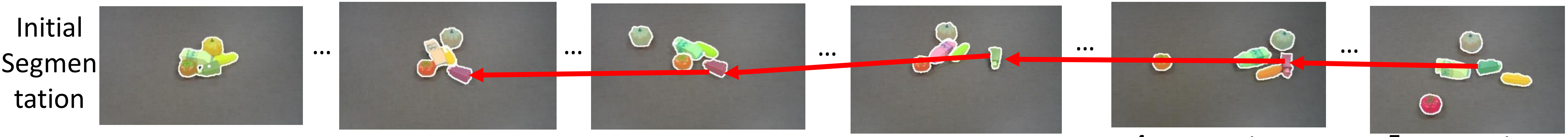
Leveraging Long-term Robot Interaction



Data Collection in the Real World

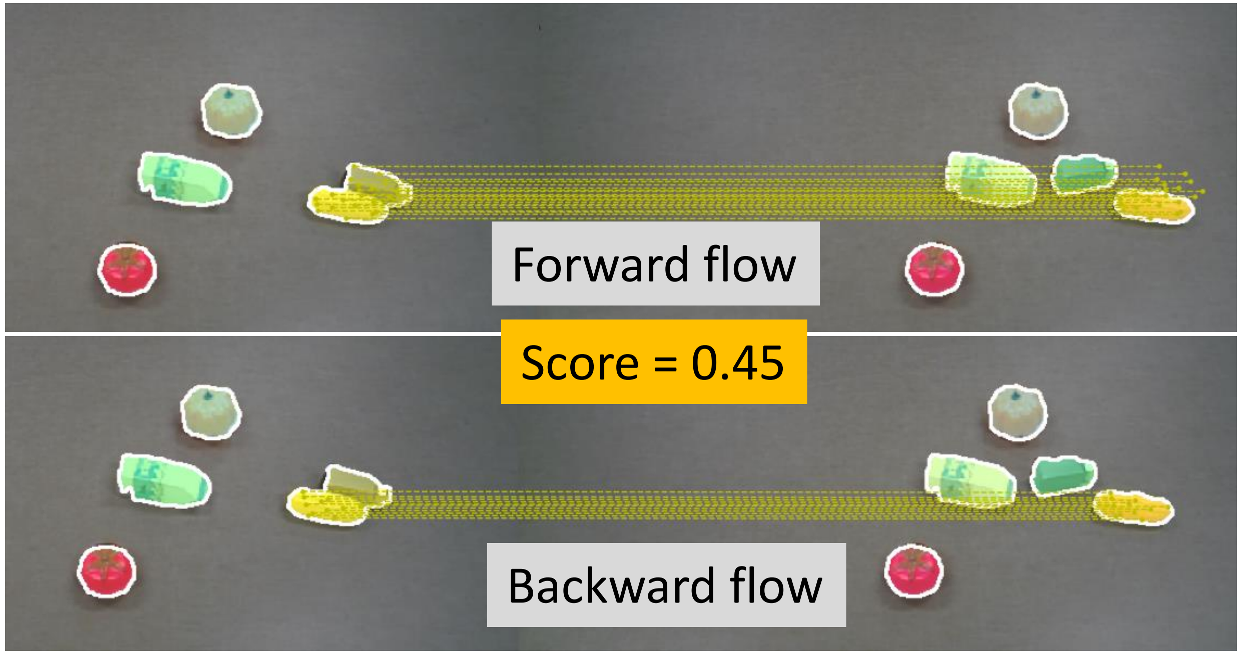


Tracking by Segmentation with Optical Flow



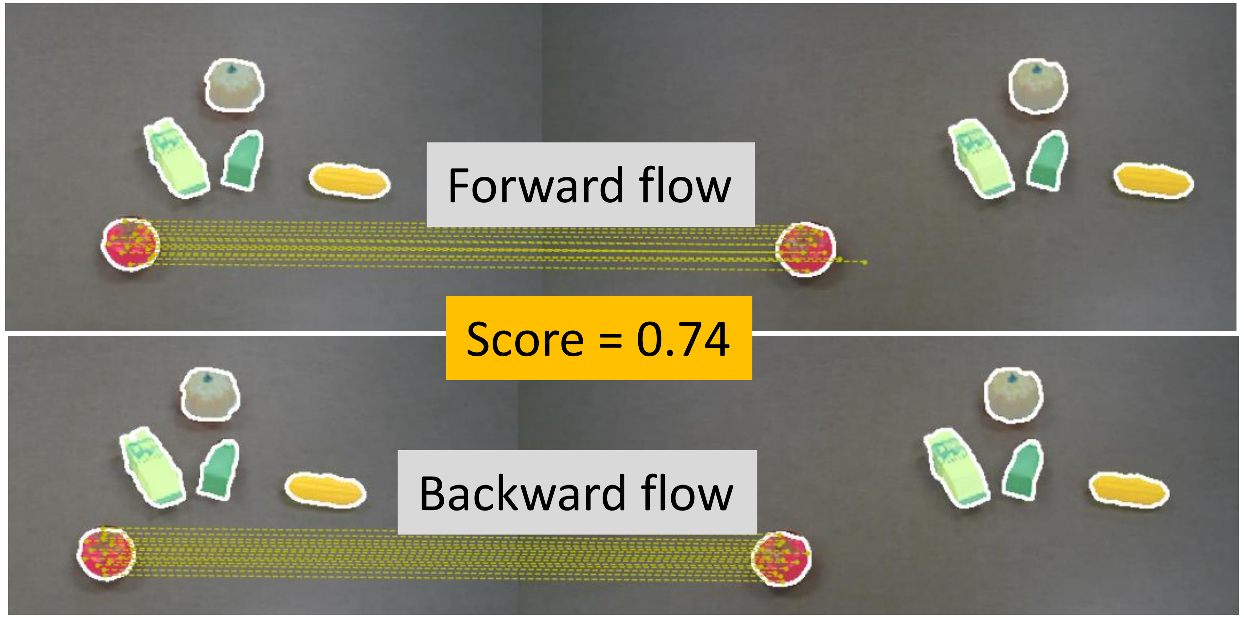
1 segment | 3 segments | 4 segments | 5 segments | 4 segments | 5 segments

Tracklet



t_1 t_2

(a)

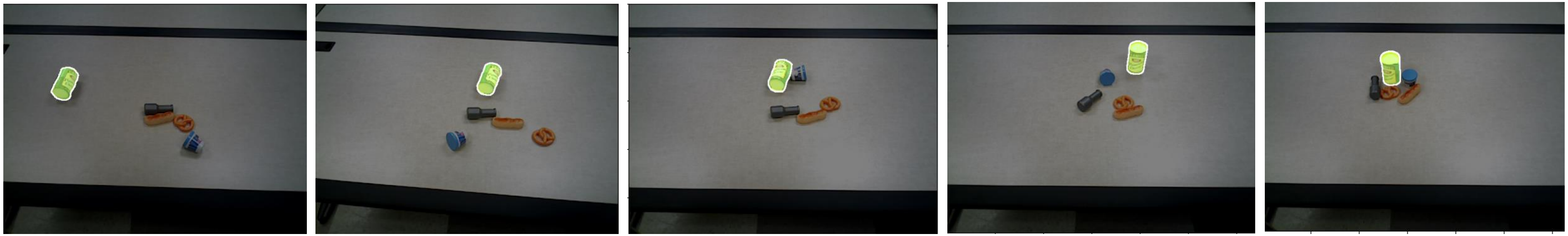


t_1 t_2

(b)

Score based on IoU of propagated pixels using flow

Mask Propagation via Video Object Segmentation



Initial mask: frame 20 → frame 10 → frame 7 → frame 4 → frame 0

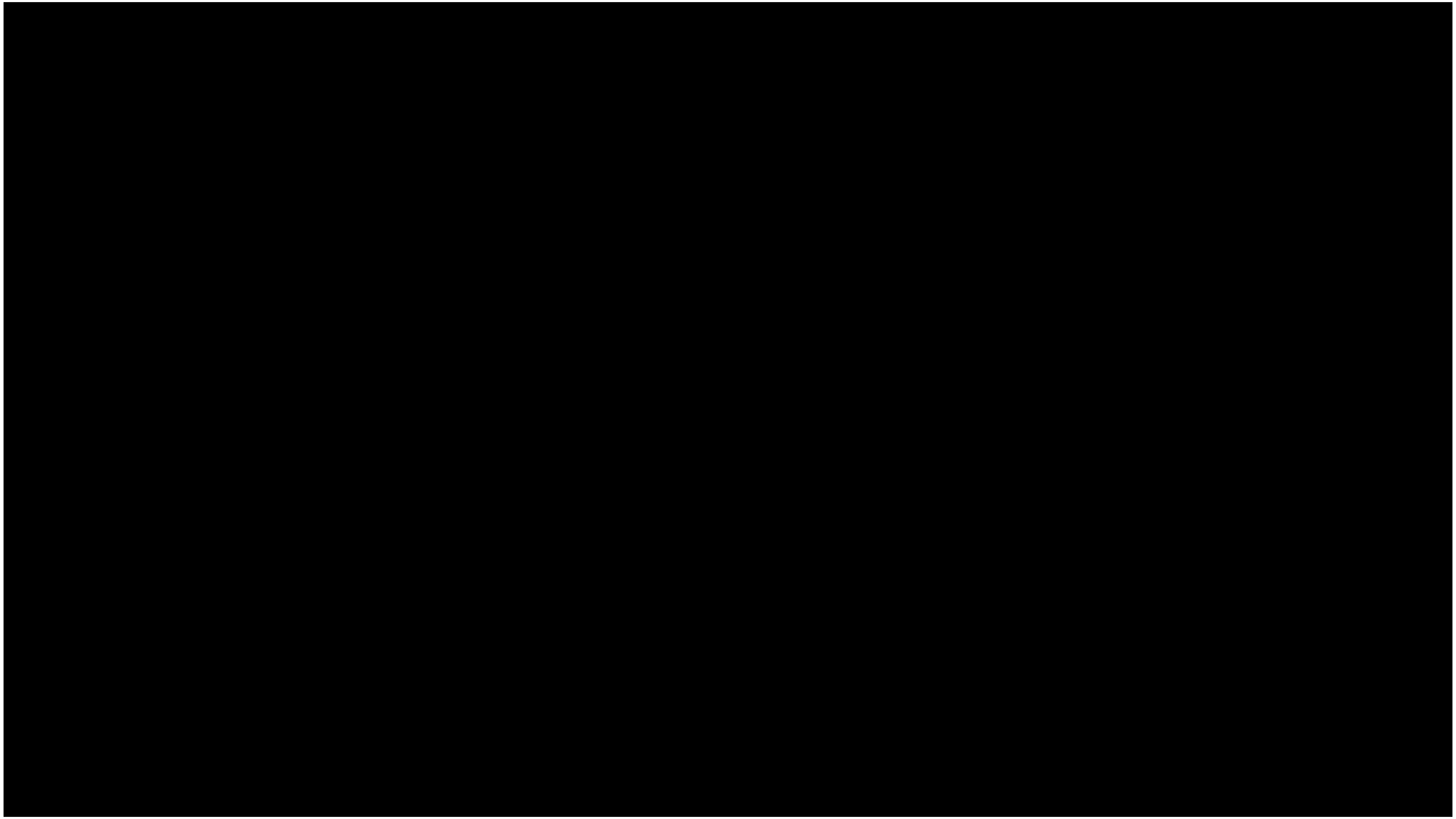
Select the highest score mask in a tracklet

Propagation to other frames

Initial mask: frame 21 → frame 19 → frame 9 → frame 3 → frame 0

Long-Term Video Object Segmentation with an Atkinson-Shiffrin Memory Model.
[Ho Kei Cheng](), [Alexander Schwing](), ECCV, 2022. https://github.com/hkchengrex/XMem

Data Collected by the Robot



Fine-tuning MSMFormer for Unseen Object Segmentation

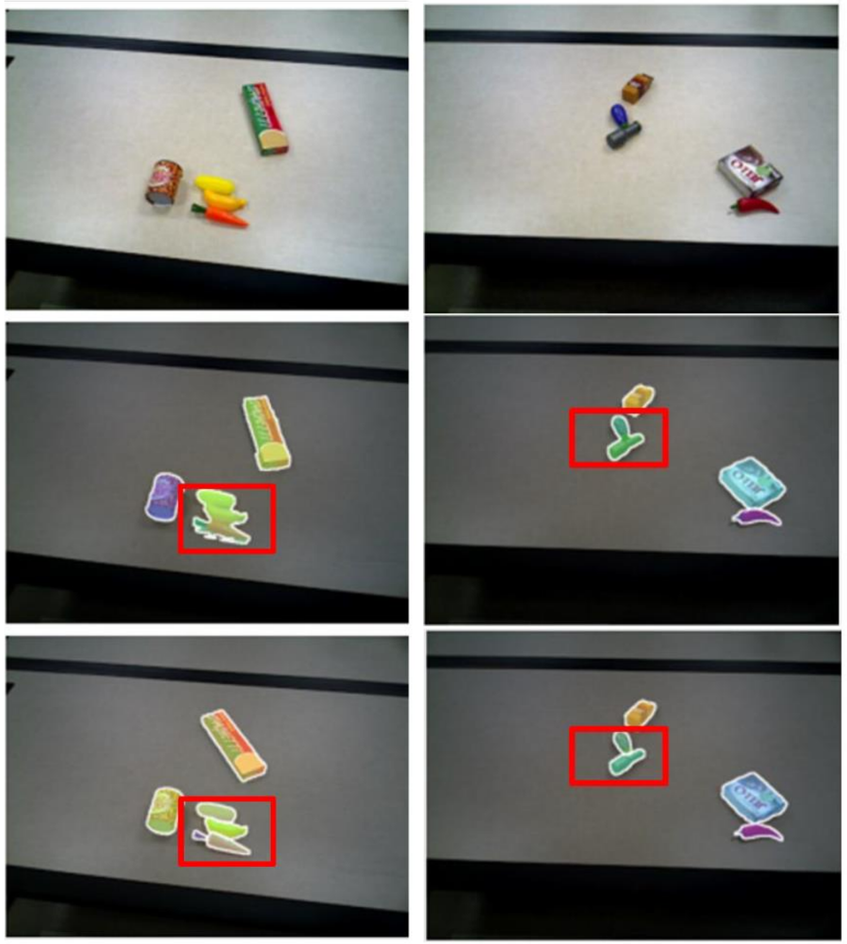
Method	Same Domain Dataset (107 images)						
	Overlap			Boundary			
	P	R	F	P	R	F	%75
RGB Input with ResNet-50 backbone							
MF [19]	81.7	81.7	81.6	75.7	73.1	73.7	66.2
MF*	**90.6**	**92.7**	**91.6**	**87.3**	**88.6**	**87.6**	**90.7**
MF+Zoom-in	75.9	81.0	78.1	68.0	63.7	65.1	61.6
MF+Zoom-in*	90.1	89.6	89.7	88.0	84.4	85.5	83.5
MF*+Zoom-in	83.2	90.9	86.7	74.4	78.2	75.8	85.5
MF*+Zoom-in*	**91.0**	**93.3**	**92.1**	**89.7**	**89.6**	**89.3**	**92.2**
RGB-D Input with ResNet-34 backbone							
MF [19]	85.8	88.9	87.2	81.7	78.7	79.9	75.1
MF*	**90.9**	**91.9**	**91.3**	**86.5**	**85.9**	**85.9**	**84.8**
MF+Zoom-in	88.9	89.8	89.3	86.6	84.4	85.3	80.7
MF+Zoom-in*	90.7	90.2	90.4	86.0	85.9	85.6	84.3
MF*+Zoom-in	91.0	**91.9**	91.3	**89.6**	87.2	88.2	87.0
MF*+Zoom-in*	**92.5**	**91.9**	**92.1**	89.3	**87.8**	**88.3**	**88.0**

*: model after fine-tuning

Image

Before Fine-tuning

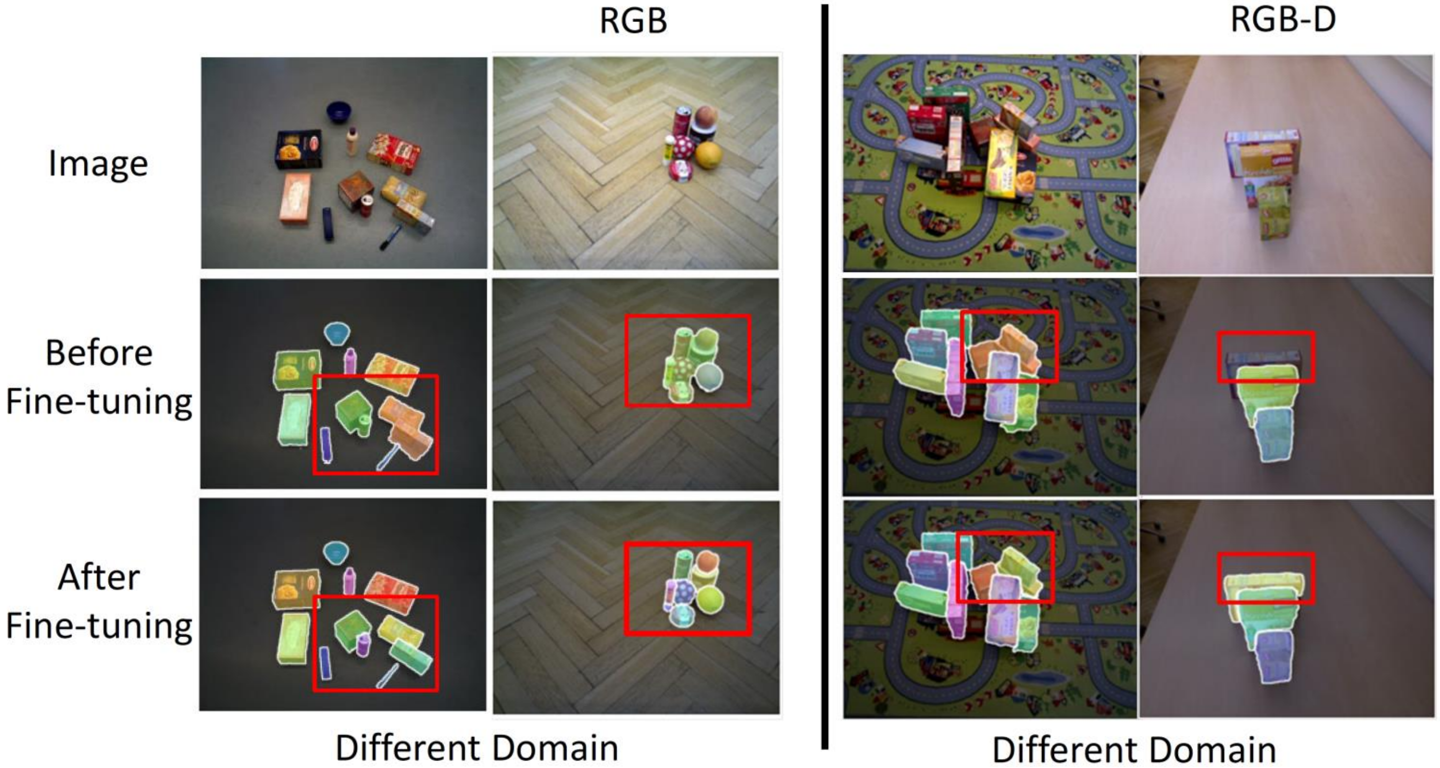
After Fine-tuning



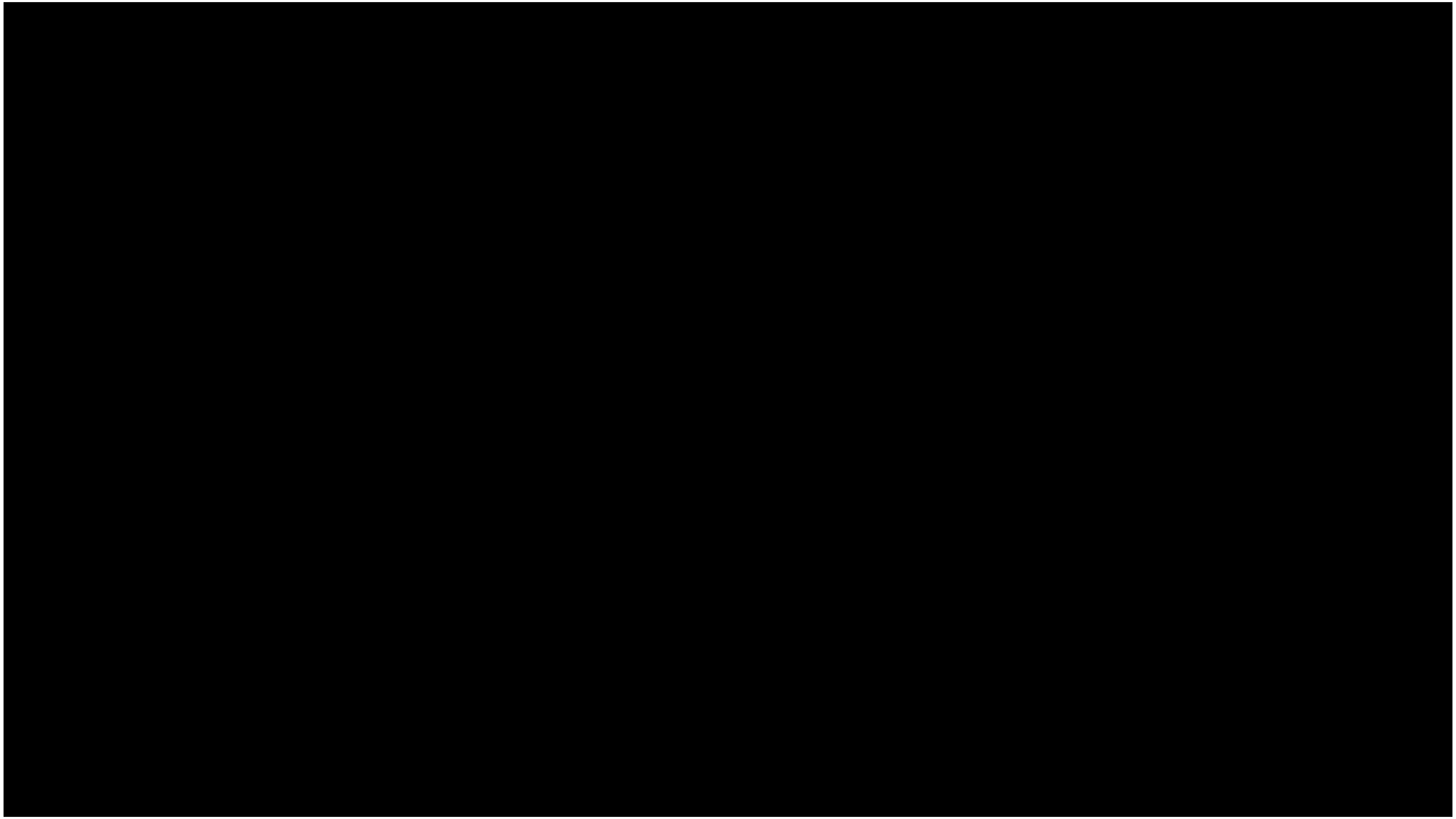
Same Domain

Fine-tuning MSMFormer for Unseen Object Segmentation

# of scenes	# of images	OCID (2390 images)							OSD (111 images)						
		Overlap			Boundary				Overlap			Boundary			
		P	R	F	P	R	F	%75	P	R	F	P	R	F	%75
MSMFormer [19]	0	88.4	**90.2**	88.5	84.7	83.1	83.0	80.3	79.5	86.4	82.8	53.5	71.0	60.6	79.4
3	62	89.7	89.8	88.7	82.8	85.5	83.0	85.3	83.6	85.8	84.6	58.7	75.4	65.5	80.6
6	124	91.0	89.1	89.5	80.7	85.0	82.0	**87.0**	83.7	85.1	84.3	59.1	74.6	65.3	78.0
9	190	91.4	89.6	90.0	83.7	**85.6**	84.0	86.0	83.9	86.4	85.1	58.6	76.4	65.8	81.0
12	256	**92.1**	89.7	**90.3**	86.2	84.9	84.9	86.3	**87.6**	**86.6**	**87.0**	64.6	**77.5**	**69.7**	**85.6**
15 (All)	321	91.2	90.1	90.1	**87.2**	85.5	**85.7**	83.9	85.1	84.4	84.6	**67.8**	71.4	69.0	76.2



Top-Down Grasping



Conclusion

- Mean Shift Mask Transformer for Unseen Object Instance Segmentation https://arxiv.org/abs/2211.11679
  - Convert vMF mean shift clustering into decoder layer in transformer
  - An end-to-end differentiable segmentation model

- Self-supervised unseen object instance segmentation https://arxiv.org/abs/2302.03793
  - Leverage long-term robot interaction with objects
  - Combine multi-object tracking and video object segmentation to obtain ground truth segmentation labels
  - Fine-tune segmentation networks with the collected real-world data

yu.xiang@utdallas.edu

Thank you!