CapsuleVOS: Semi-Supervised Video Object Segmentation Using Capsule Routing

Kevin Duarte, Yogesh S. Rawat, Mubarak Shah

ICCV 2019

Overview

- Introduction to Capsule Networks
- Video Capsule Networks
- Video Object Segmentation
- CapsuleVOS

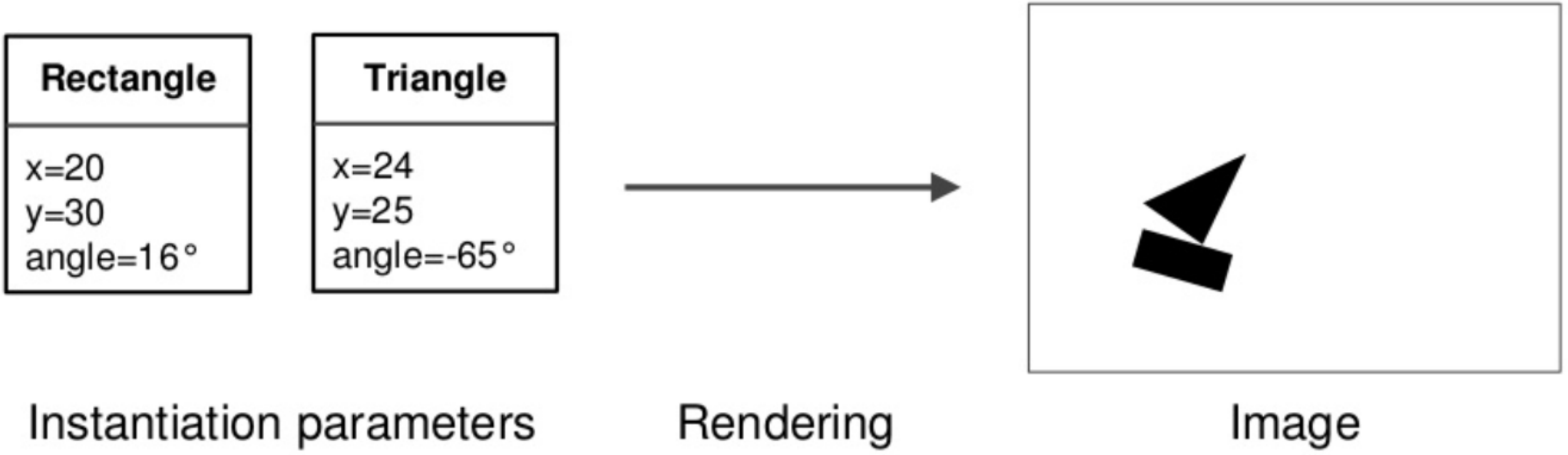
Introduction to Capsule Networks

Introduction to Capsule Networks

Motivation:

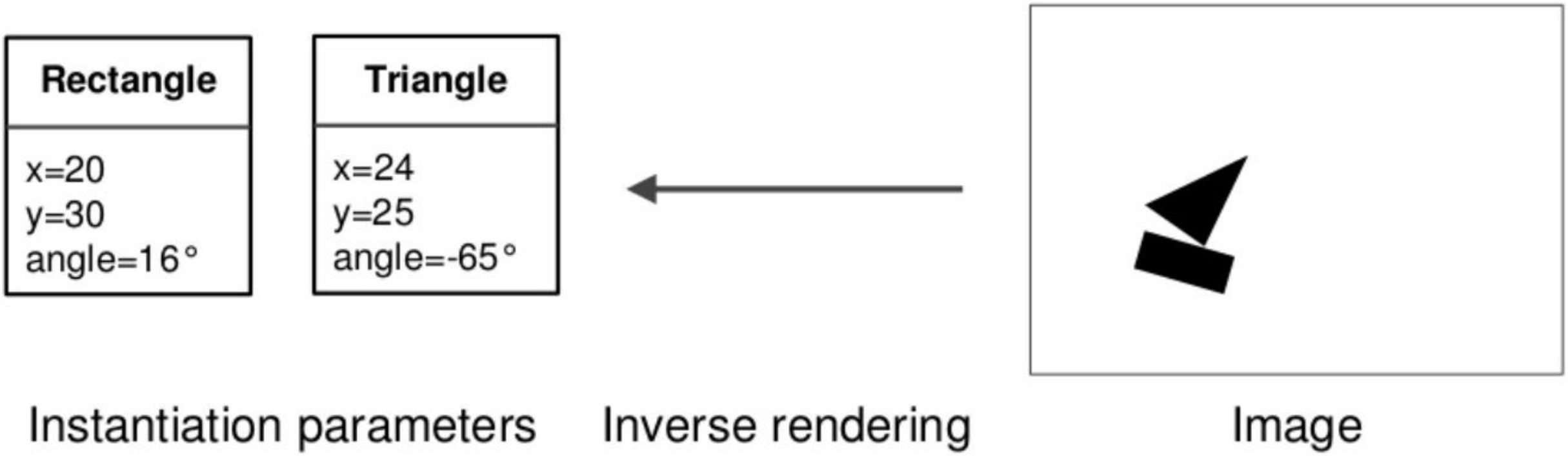
- CNNs do not explicitly model entities
- Add extra structure to CNNs to model entities
  - Entities modeled using a group of neurons
  - Routing-by-agreement to model part-to-whole relationships
- Capsules take inspiration from Inverse Graphics

Computer Graphics



Aurélien Géron (2017). Introduction to Capsule Networks (CapsNets).
https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

Inverse Graphics



Aurélien Géron (2017). Introduction to Capsule Networks (CapsNets).
https://www.slideshare.net/aureliengeron/introduction-to-capsule-networks-capsnets

Different capsule formulations

- *Dynamic Routing between Capsules* (NIPS 2017)
  - Each capsule is a vector
  - The length of the vector being its probability of existence
  - Values of the vector are the instantiation parameters of the object
  - Dynamic routing (dot product) finds similarity between capsule votes
- *Matrix Capsules with EM Routing* (ICLR 2018)
  - Each capsule is a 2d matrix with a separate activation neuron
  - The activation neuron represents the probability of existence
  - The 2d matrix contains the instantiation parameters of the object
  - EM routing (an EM clustering variant) finds similarity between capsule votes

What is capsule routing

- Routing “high-dimensional coincidence filtering” to model part-to-whole relationships
  - If multiple parts agree on the properties of a larger object, then it is likely to exists
- Given two capsule layers L and L+1,
  - The capsules in layer L vote on the properties of the capsules in L+1
  - The votes are compared, and clustered, to create the capsules in L+1

EM-Routing Example

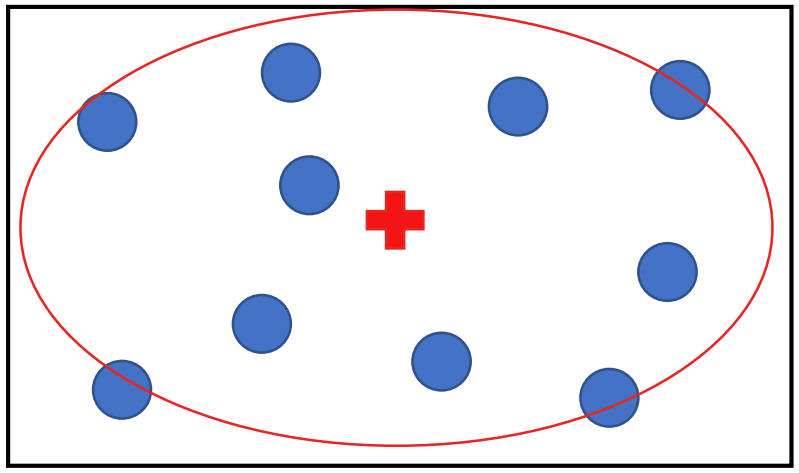
We have three higher level capsules: A, B, and C

= Vote from lower level capsule

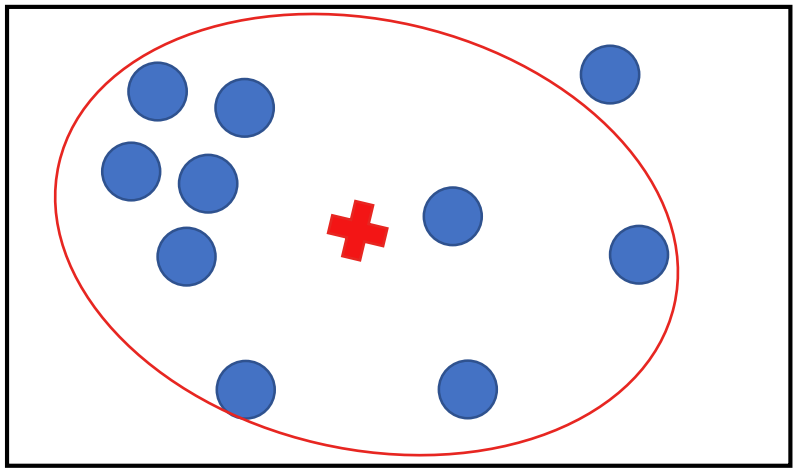
= Mean of the Gaussian

Iteration 1:

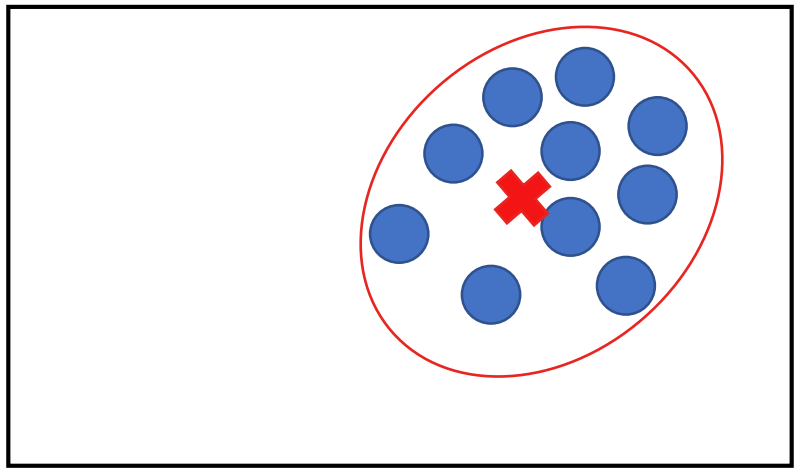
Capsule A



Capsule B



Capsule C



EM-Routing Example

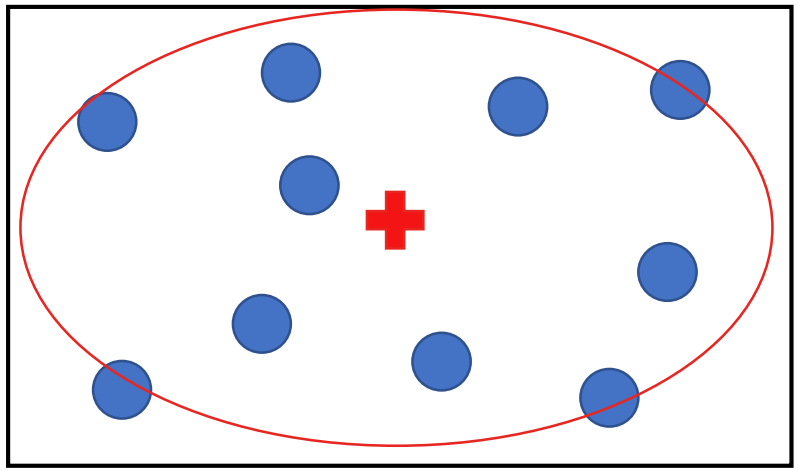
We have three higher level capsules: A, B, and C

= Vote from lower level capsule

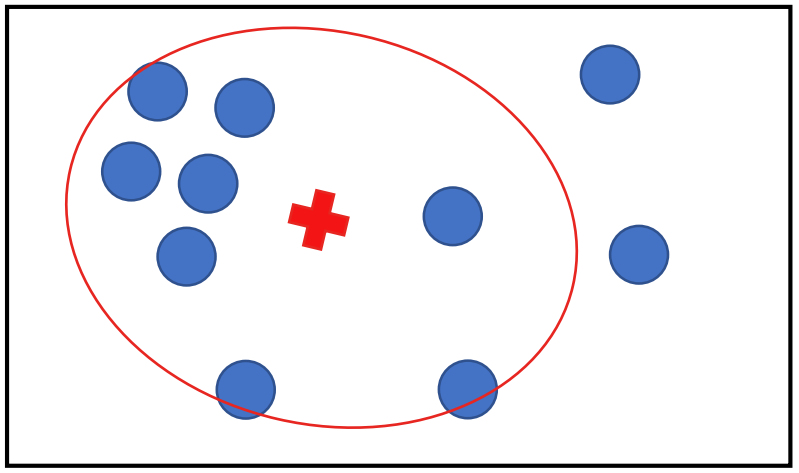
= Mean of the Gaussian

Iteration 2:

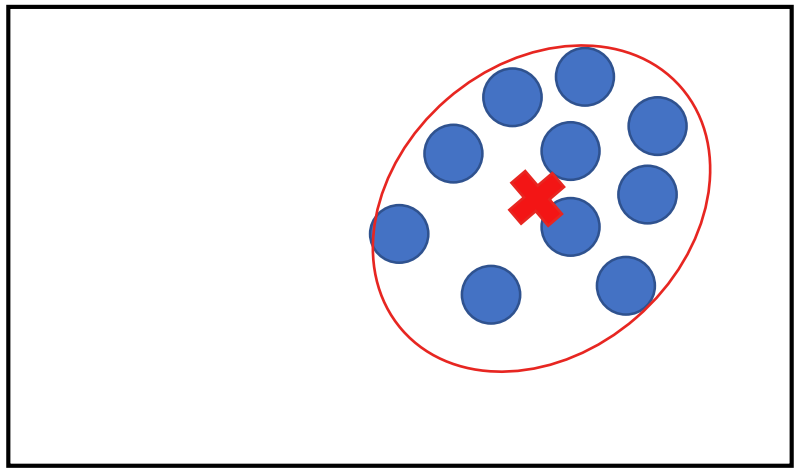
Capsule A



Capsule B



Capsule C



EM-Routing Example

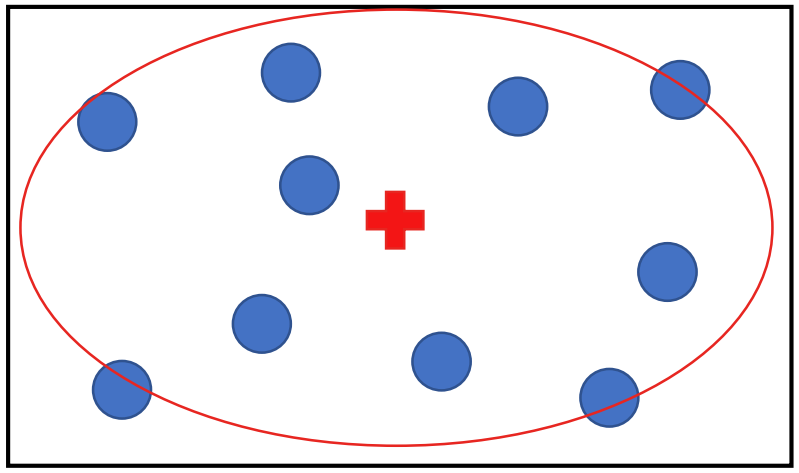
We have three higher level capsules: A, B, and C

= Vote from lower level capsule

= Mean of the Gaussian

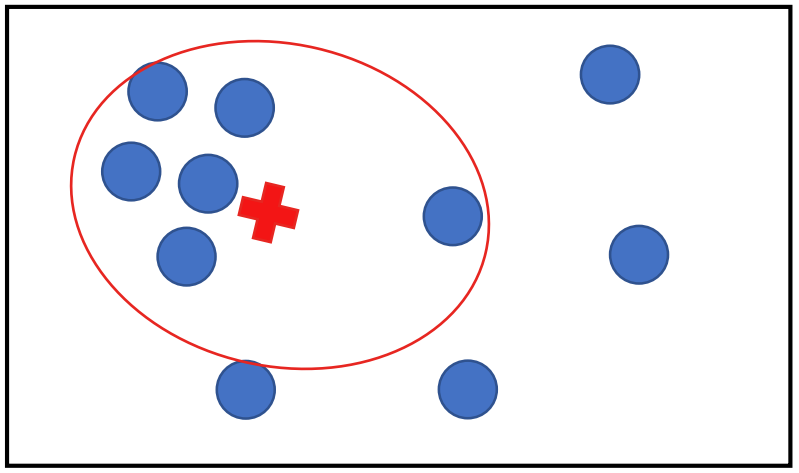
Iteration 3:

Capsule A



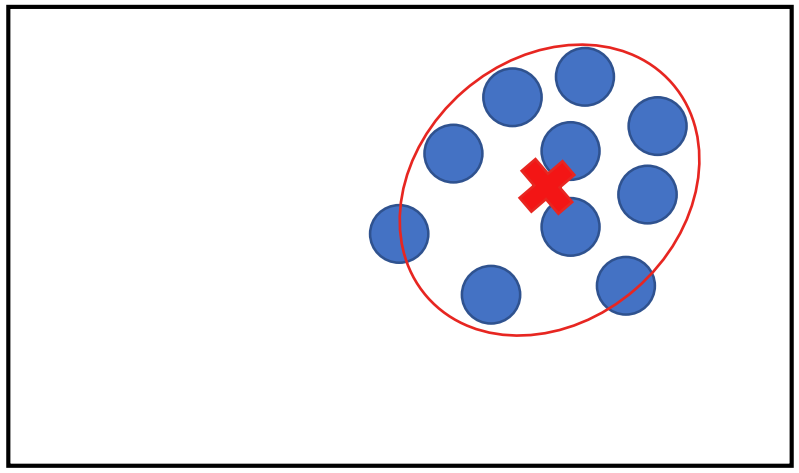
High Variance
Low Activation

Capsule B



Lower Variance
Higher Activation

Capsule C



Very Low Variance
Very High Activation

Video Capsule Networks

Capsule Networks

- Achieves good results classifying small images (MNIST and smallNorb)
- Has not been successfully applied on high dimensional data
  - Large images or videos
- Issues:
  - Computationally costly
  - Deeper networks cannot fit into memory

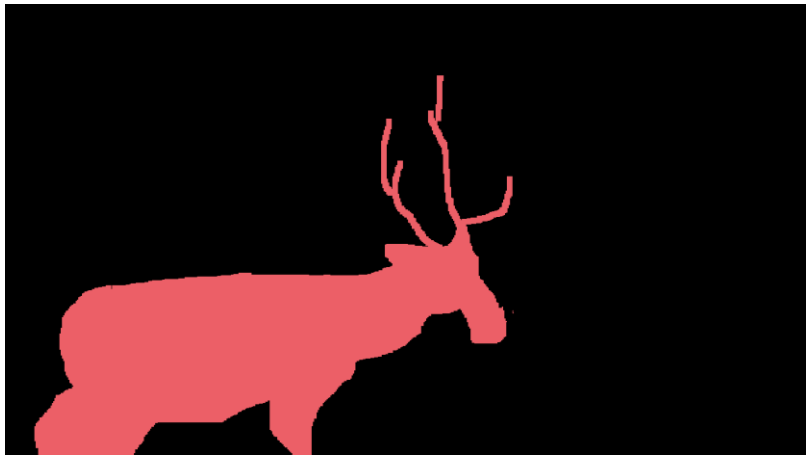
Video Capsule Networks

- Capsules learn very good representations with very few parameters
- This would be useful for videos
- *VideoCapsuleNet: A Simplified Network for Action Detection (NeurIPS 2018)*
  - Extends capsule networks to 3d videos
  - Presents an end-to-end method for action detection/segmentation
  - Achieves SOTA results on UCF-101 and JHMDB datasets

Semi-Supervised Video Object Segmentation

Semi-Supervised Video Object Segmentation

- Given the first frame’s segmentation and a video
- Segment the object/objects throughout the video







Semi-Supervised Video Object Segmentation

- **ALL training data is annotated**, so this is a fully supervised method
- Called semi-supervised because the first frame is given at test time
- Difficulties in problem:
  - Small objects
  - Fast motions – both camera motion and object motion
  - Multiple objects of interest in a single video
  - Similar objects to the object of interest (distractors)
  - Changes in illumination
  - Object deformations
  - Unseen objects (i.e. not seen in training, but found in testing)

Datasets

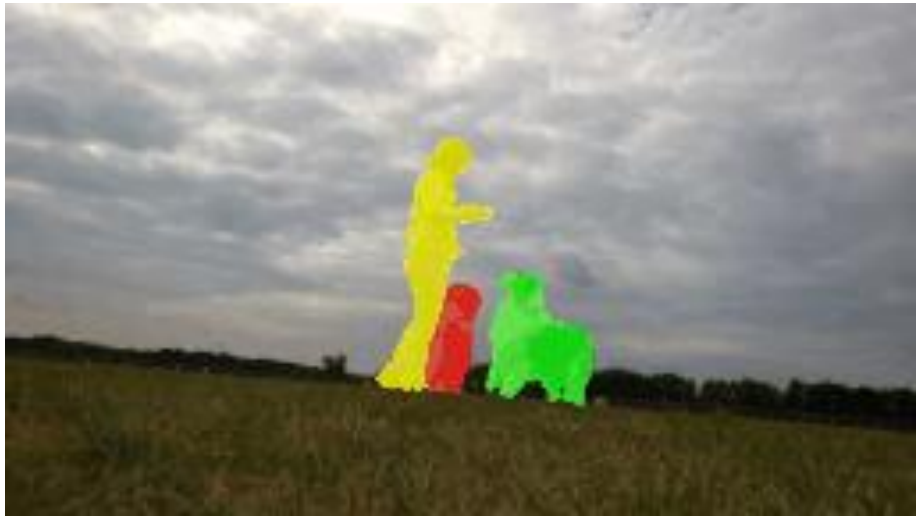
- DAVIS
  - 60 train, 30 validation, and 30 test videos
  - Annotated 30 fps
- YoutubeVOS
  - 3471 train, 474 validation, and 508 test videos
  - Annotated 6 fps

Example videos from DAVIS









CapsuleVOS: Semi-Supervised Video Object Segmentation Using Capsule Routing

Kevin Duarte, Yogesh S. Rawat, Mubarak Shah

ICCV 2019

VOS using Capsules

- Capsules model entities/objects
- Routing finds agreement, or similarity, between these entities/objects
- We leverage these 2 ideas for Video Object Segmentation (VOS):
  - We extract capsules from the video and the segmented first frame
  - The video capsules model objects within the video
  - The frame capsules model the object of interest
  - Routing can be used to find agreement between these two sets of capsules

VOS using Capsules

Video



Reference Frame with Segmentation



Video Encoder

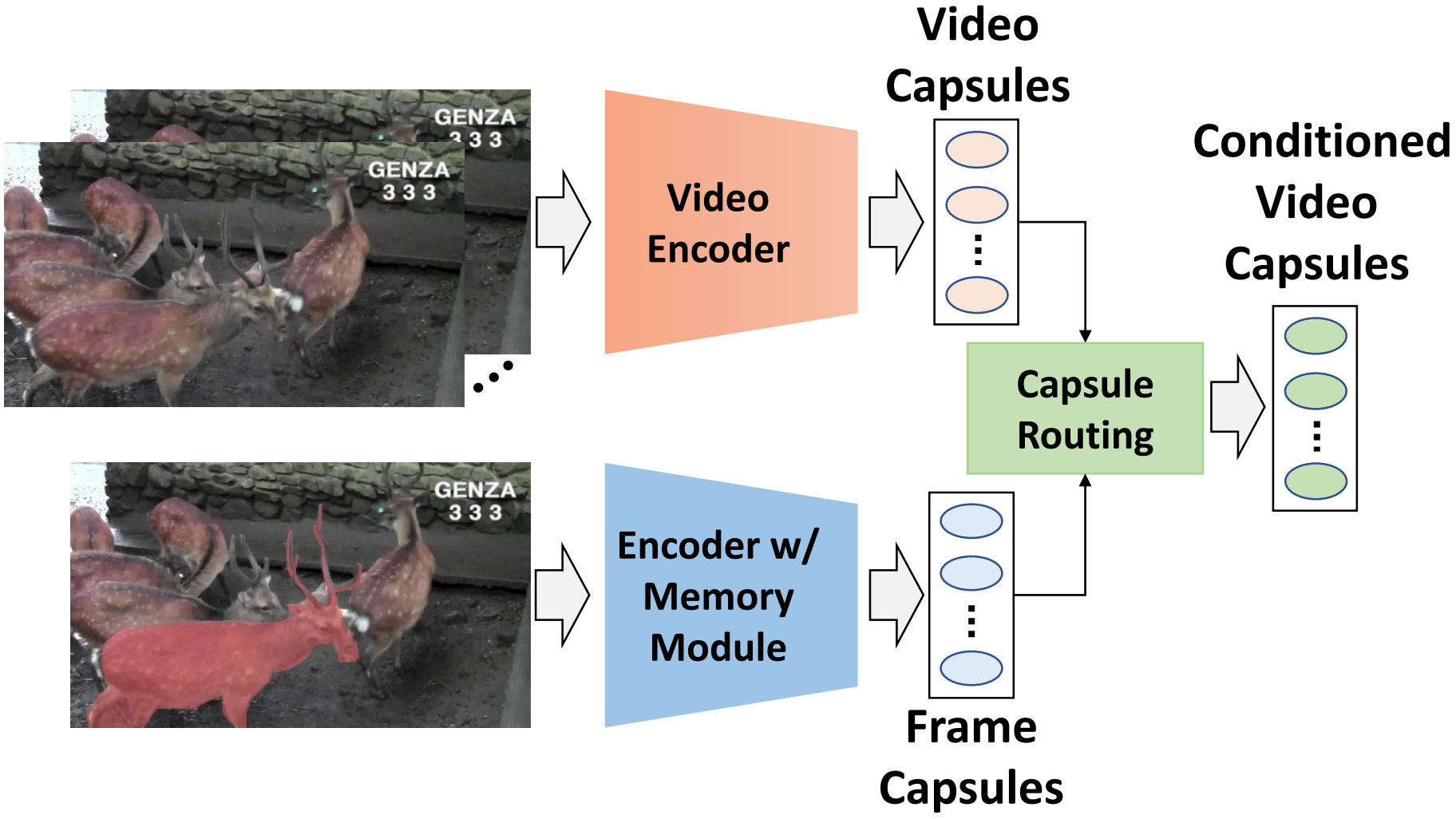
Frame Encoder

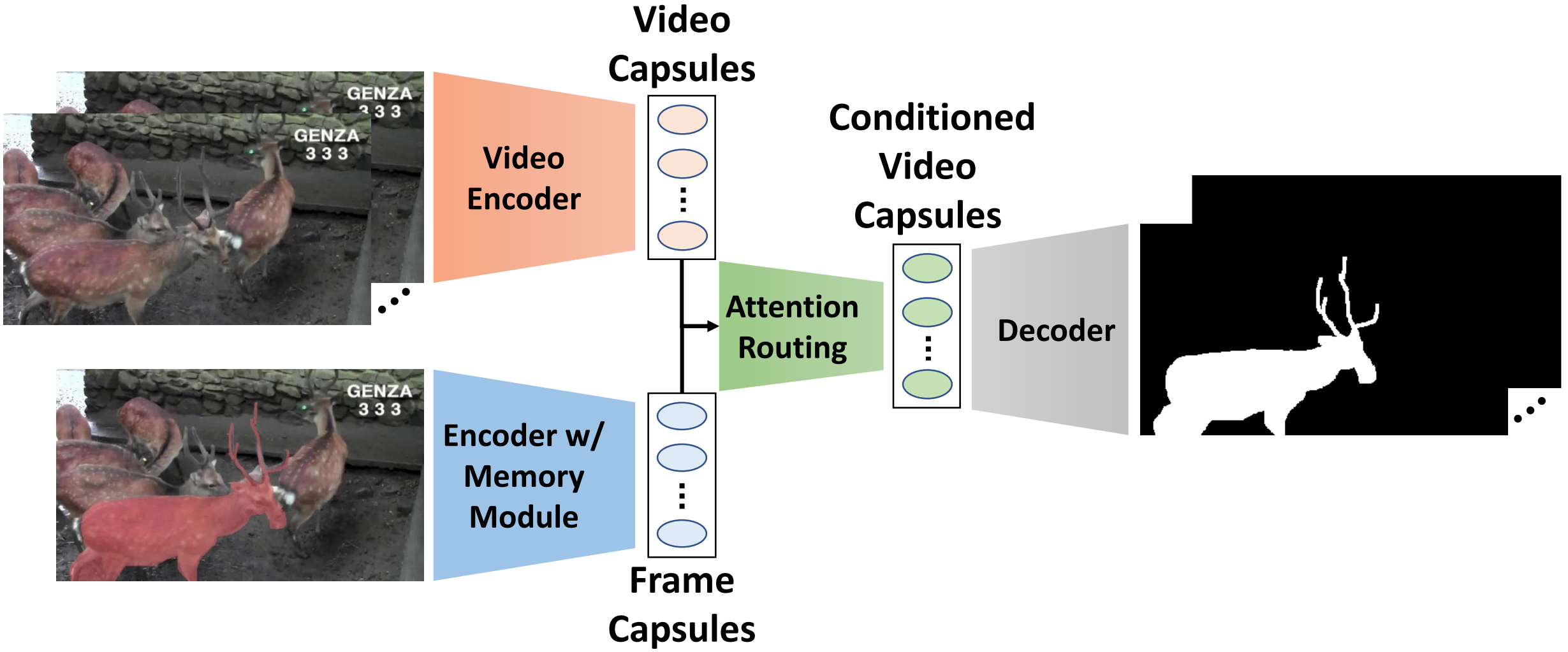
Video Capsules

Frame Capsules

Capsule Routing

Conditioned Video Capsules

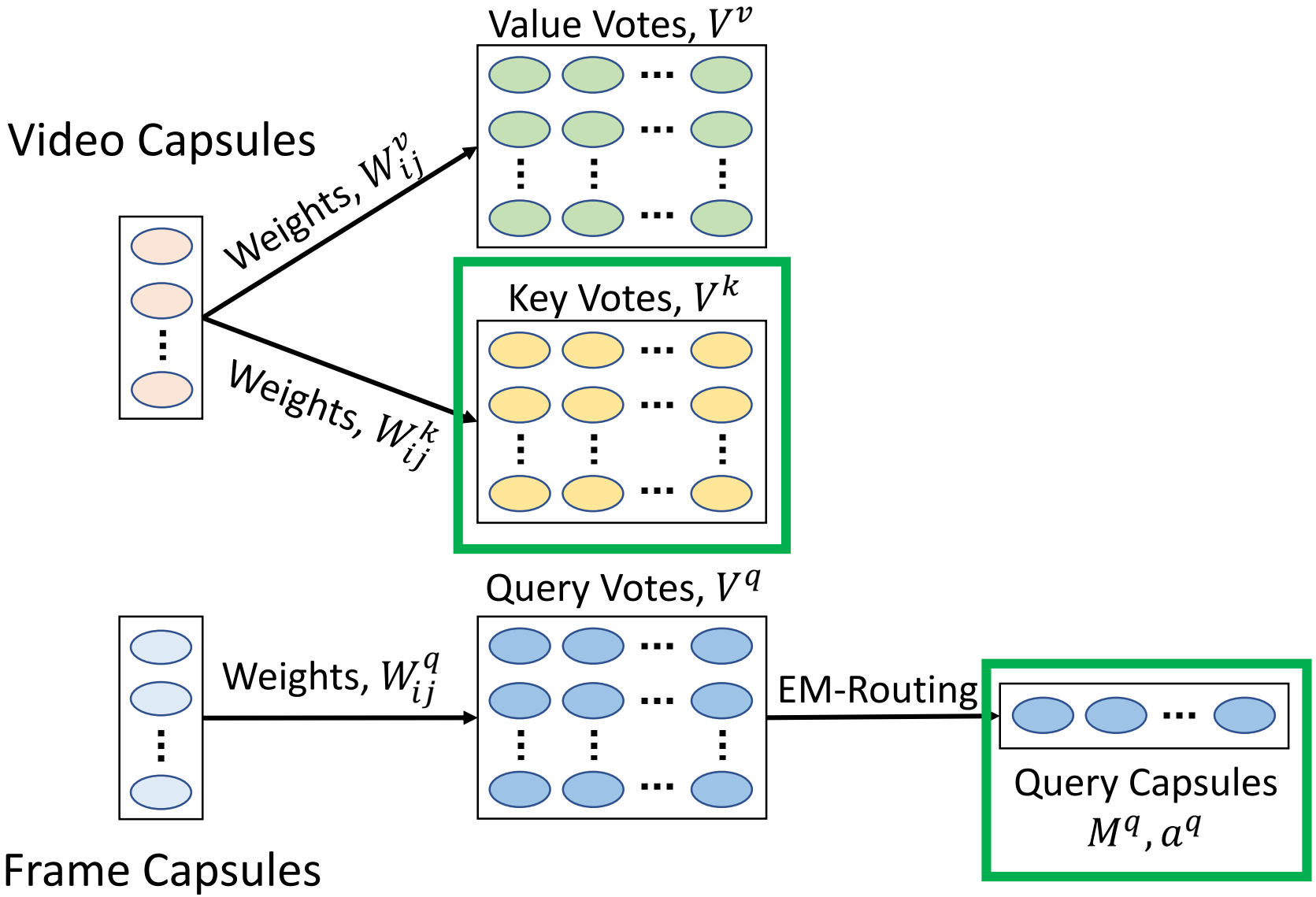

Video Capsules
Conditioned Video Capsules
Video Encoder
Capsule Routing
Encoder w/ Memory Module
Frame Capsules


Video Capsules
Video Encoder
Conditioned Video Capsules
Attention Routing
Decoder
Encoder w/ Memory Module
Frame Capsules

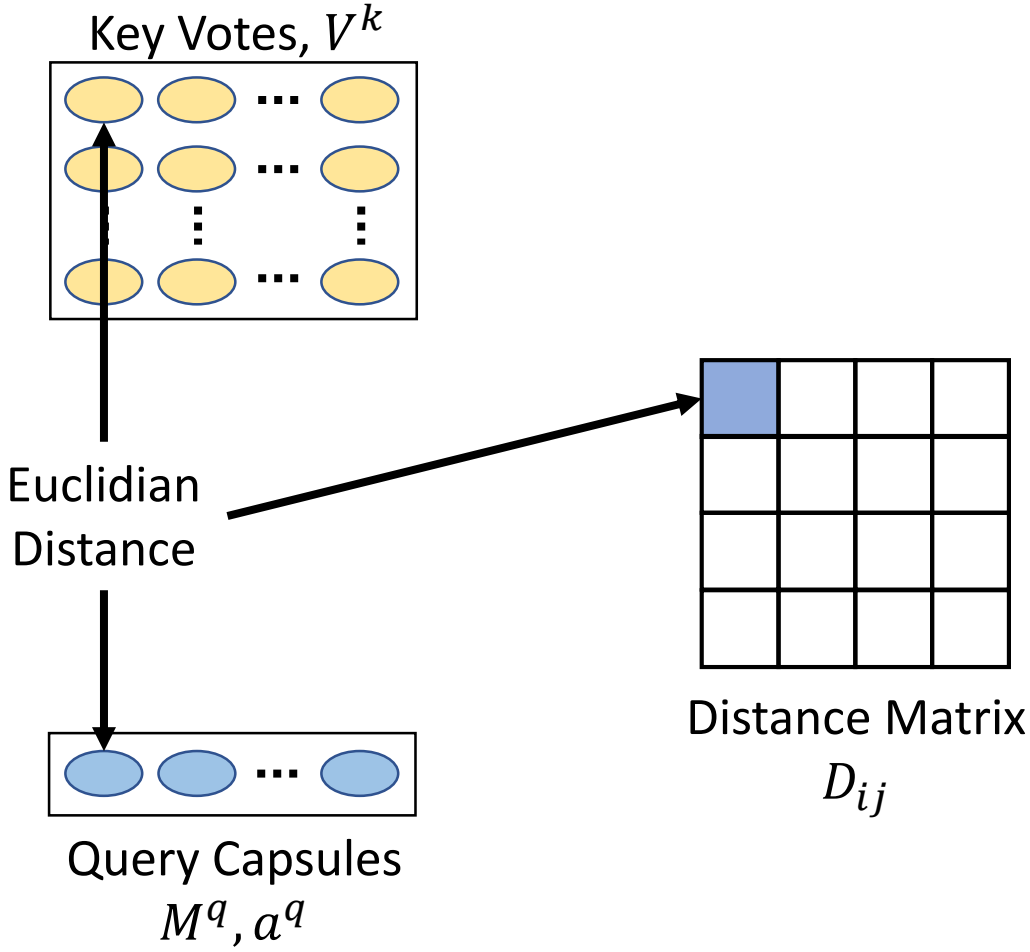
Attention through Routing

- We can use the multi-modal capsule routing discussed earlier
  - This does achieve good results, but more can be done
- An adjustment to the EM-routing algorithm should be made
  - This adjustment should find agreement between two **sets** of capsules
  - Routing should condition the video capsules based on the frame capsules

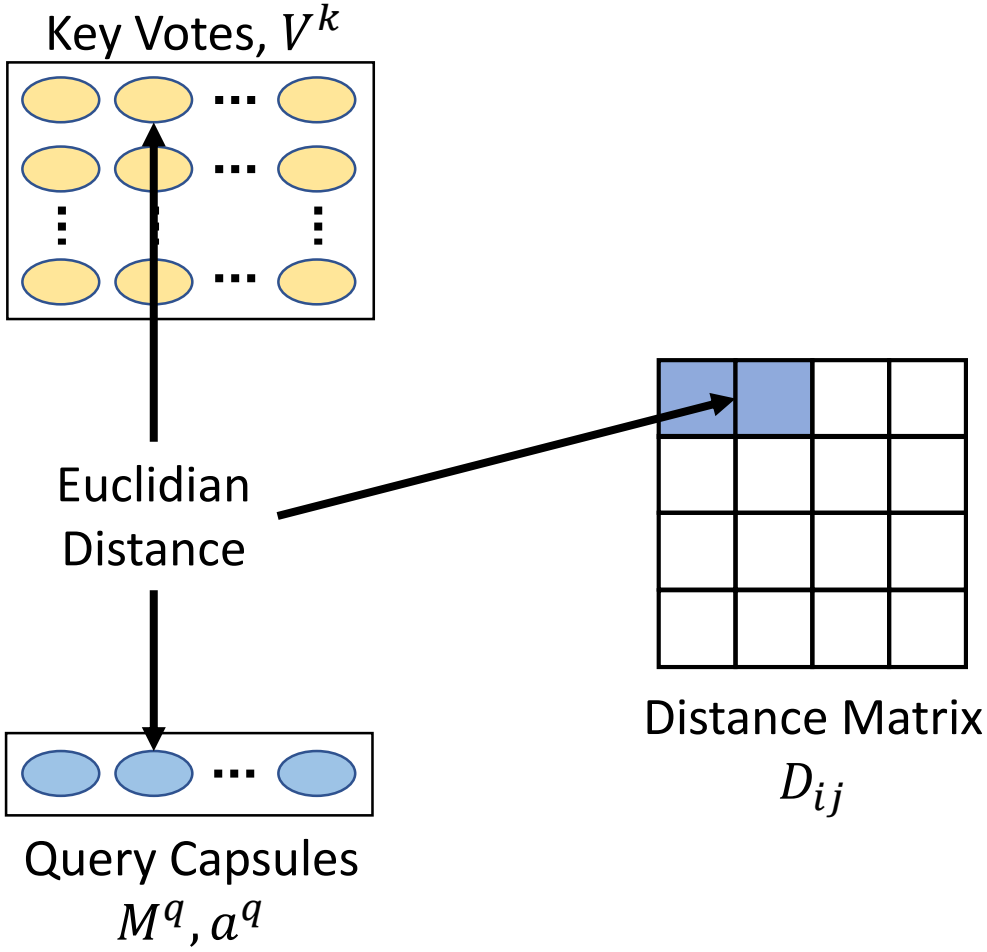
Attention Routing



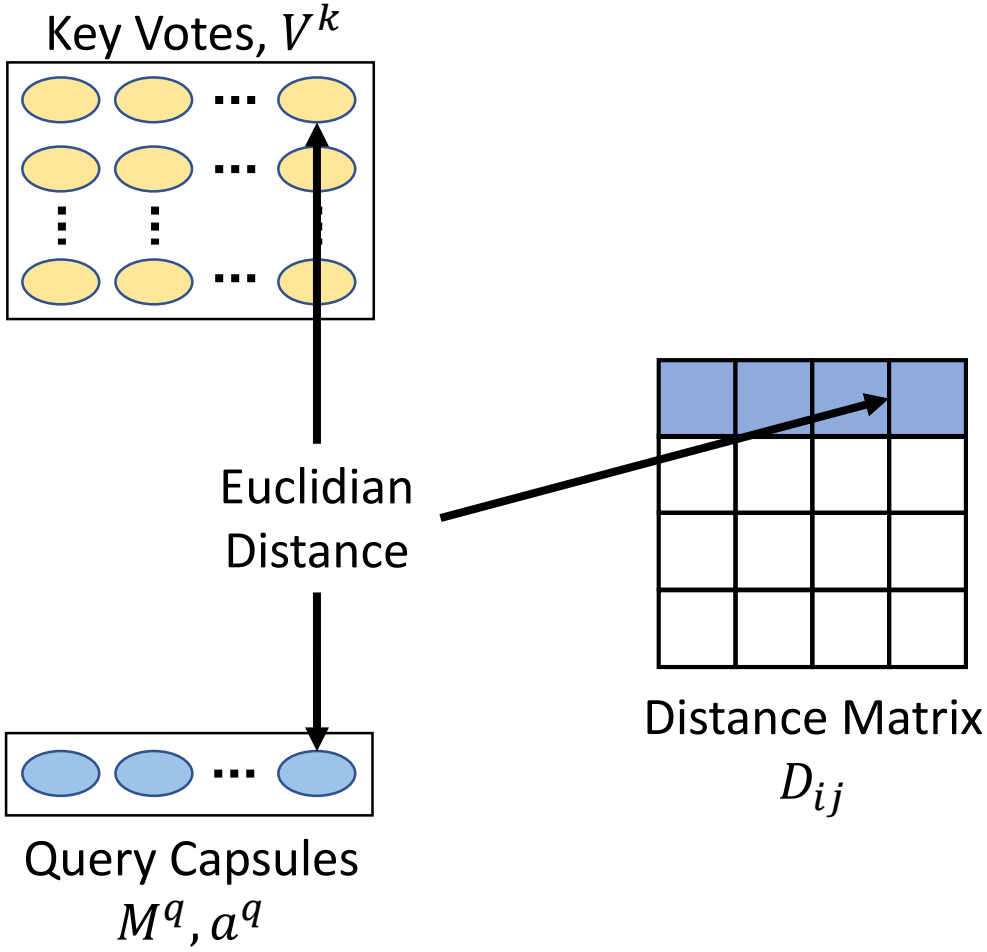
Attention Routing



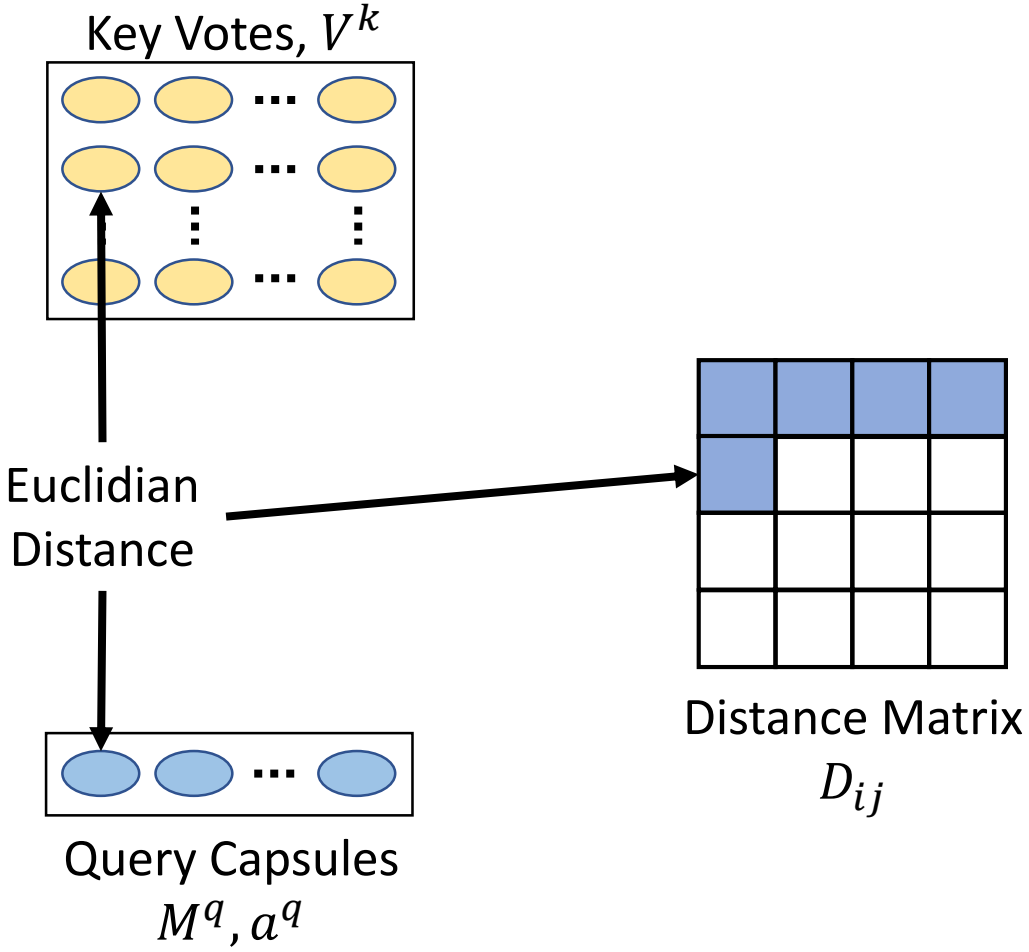
Attention Routing



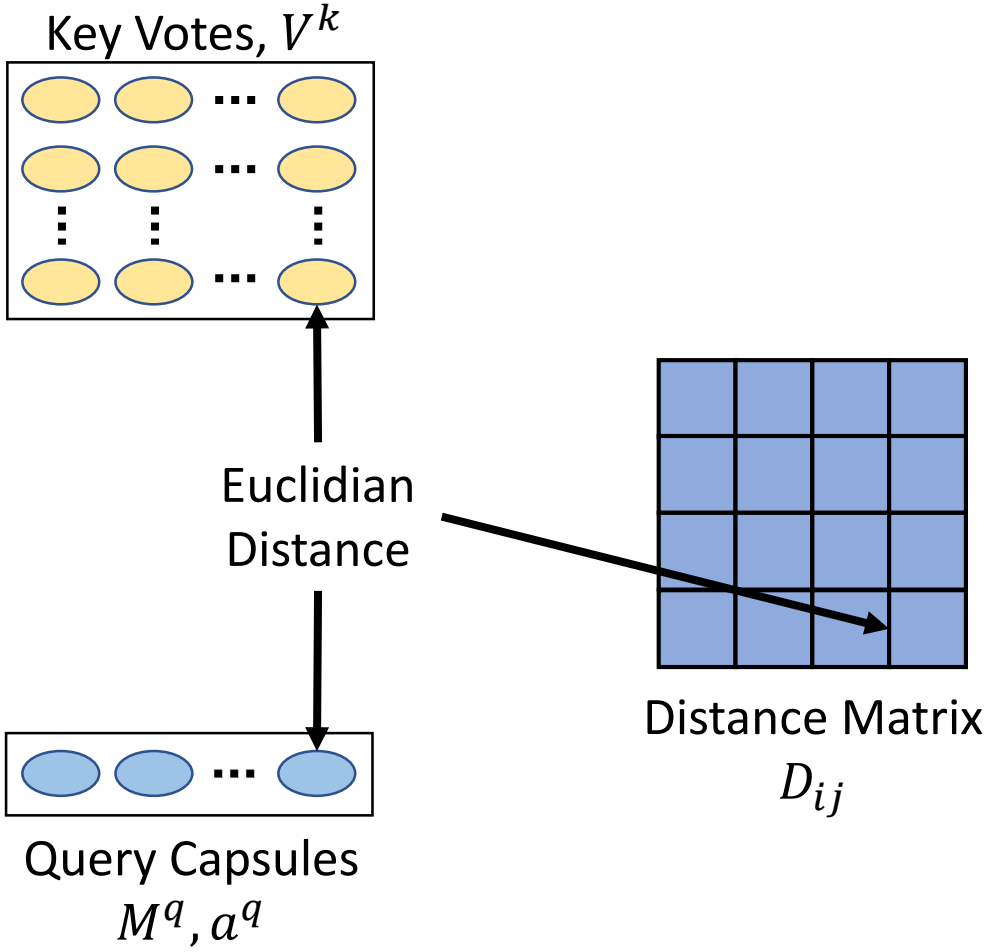
Attention Routing



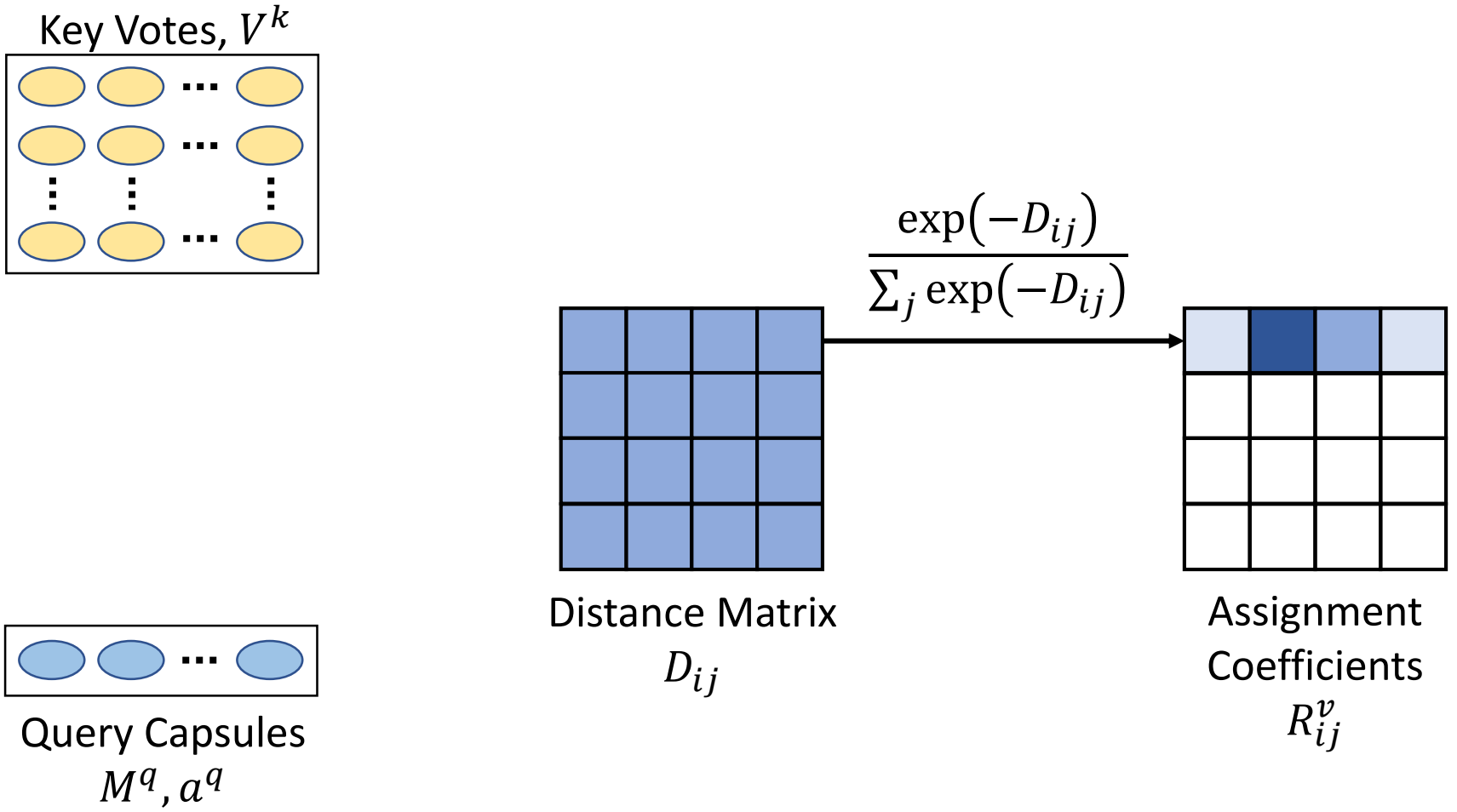
Attention Routing



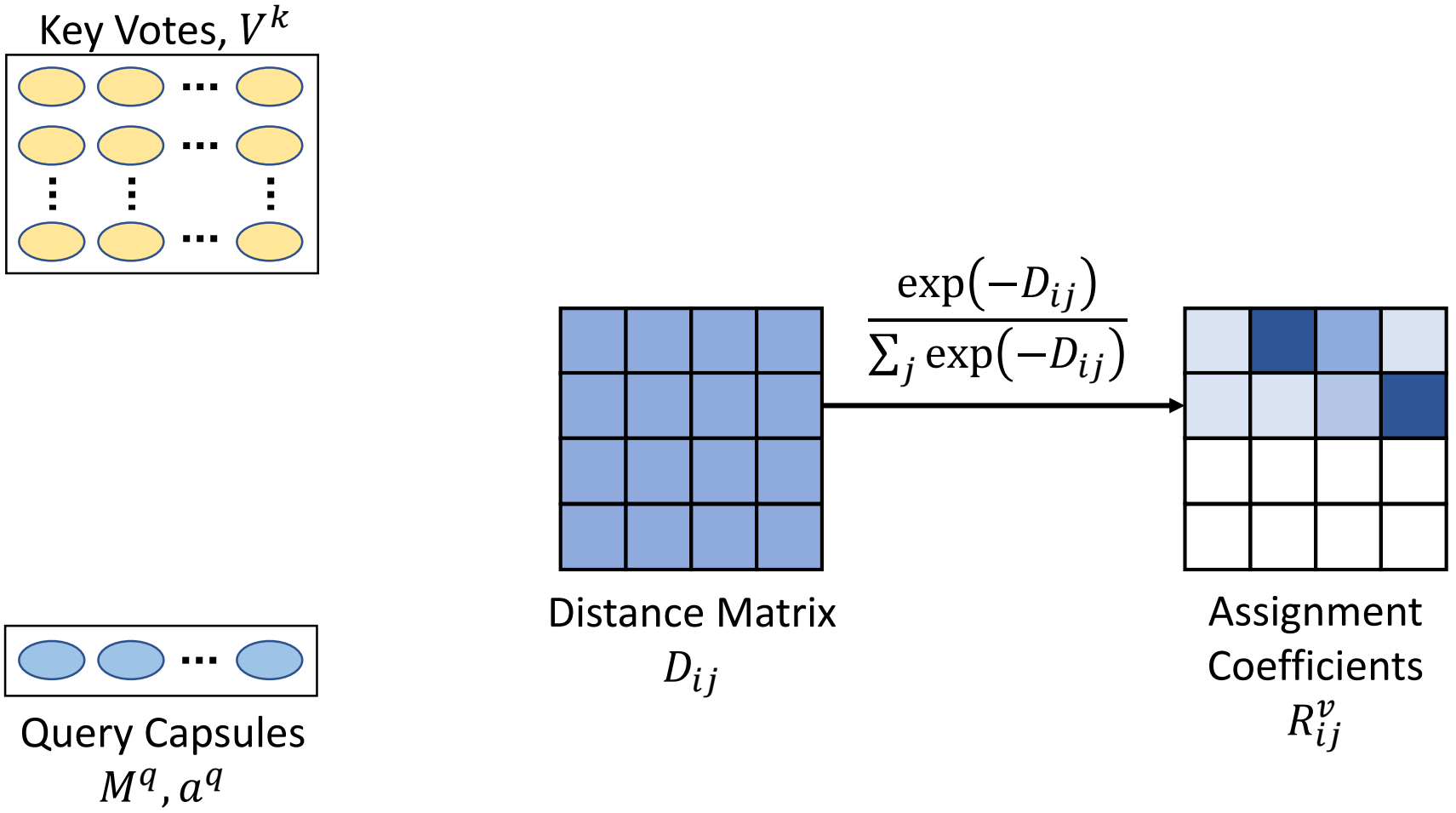
Attention Routing



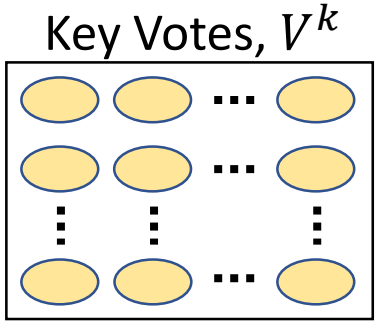
Attention Routing

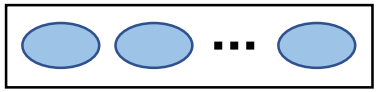


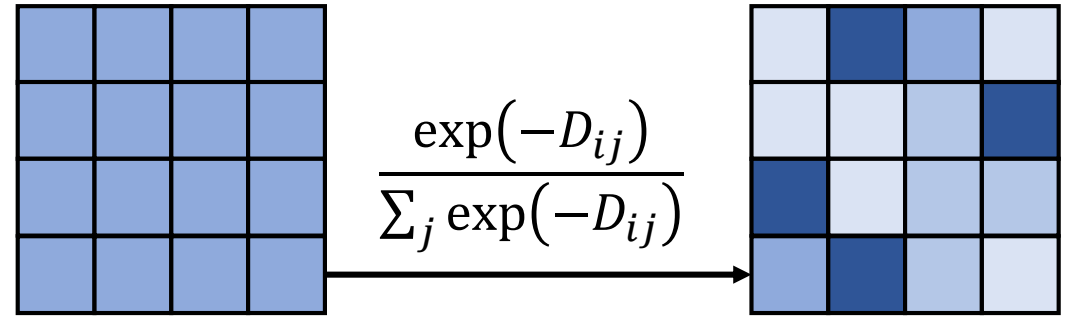
Attention Routing



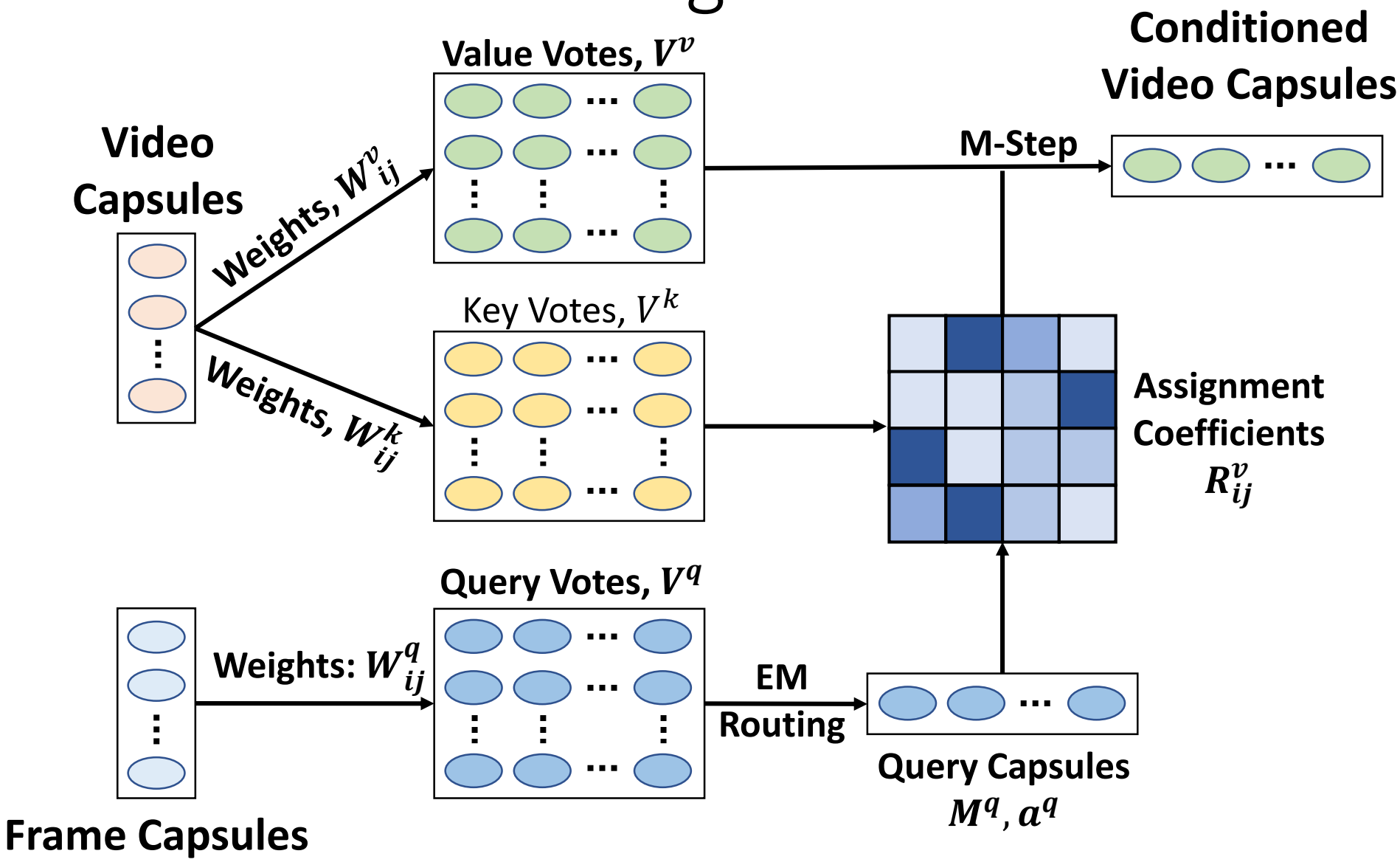
Attention Routing

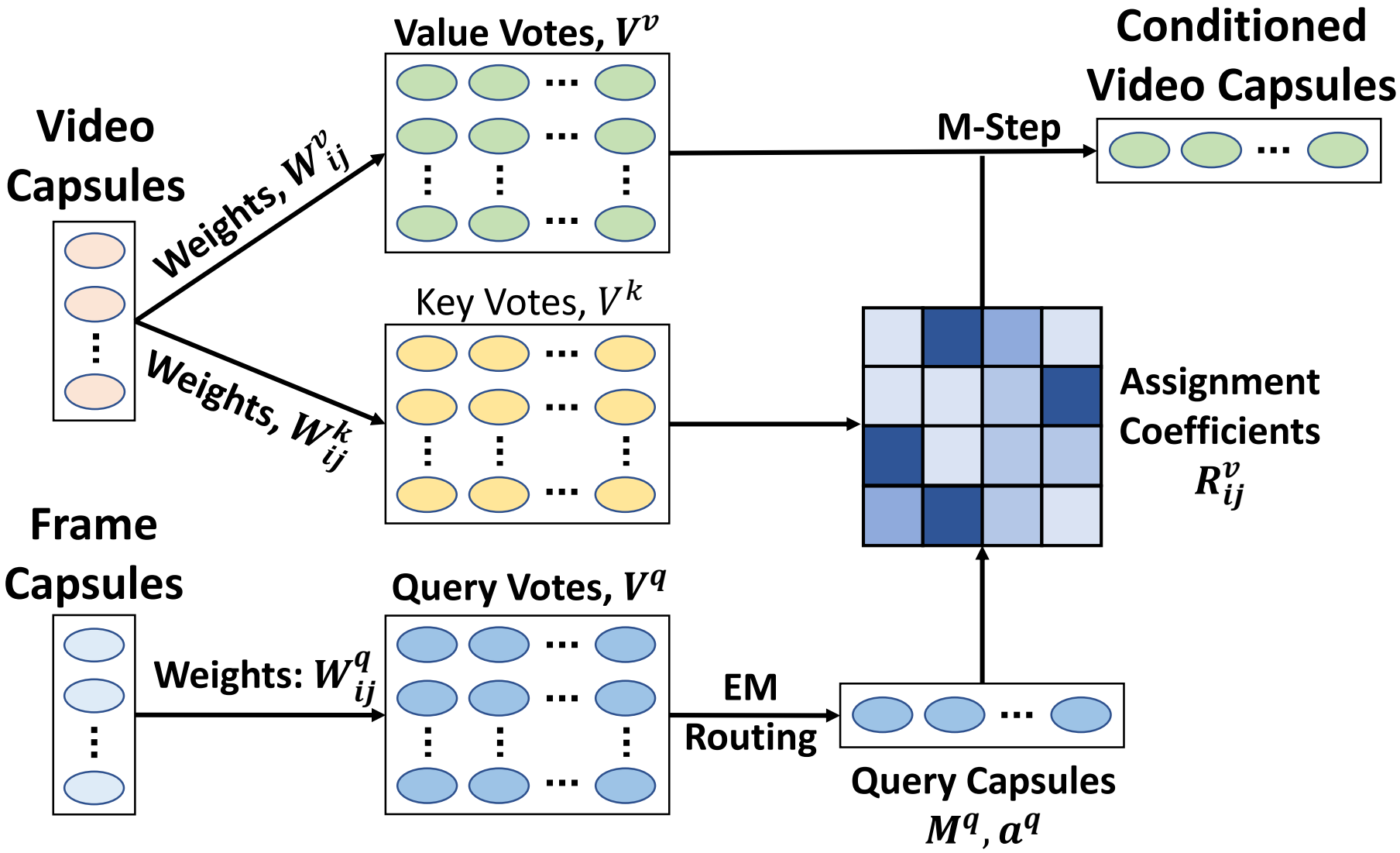






Attention Routing




Video Capsules
Weights, W_{ij}^{v}
Value Votes, V^{v}
Conditioned Video Capsules
M-Step
Weights, W_{ij}^{k}
Key Votes, V^{k}
Assignment Coefficients R_{ij}^{v}
Frame Capsules
Weights: W_{ij}^{q}
Query Votes, V^{q}
EM Routing
Query Capsules M^{q}, a^{q}

Attention Routing

- $M^{\mathcal{V}}, a^{\mathcal{V}}$ are the video capsules' poses and activations
- $M^{\mathcal{F}}, a^{\mathcal{F}}$ are the frame capsules' poses and activations
- W^v, W^k, W^q are the value, key, and query transformation matrices

procedure ATTROUTING($M^{\mathcal{V}}$, $a^{\mathcal{V}}$, $M^{\mathcal{F}}$, $a^{\mathcal{F}}$)

$V^v \leftarrow M^{\mathcal{V}} W^v$ — Get value votes from the video capsules

$V^k \leftarrow M^{\mathcal{V}} W^k$ — Get key votes from the video capsules

$V^q \leftarrow M^{\mathcal{F}} W^q$ — Get query votes from the frame capsules

$a^q, M^q \leftarrow$ EM ROUTING$(a^{\mathcal{F}}, V^q)$ — Get query capsules using EM-Routing

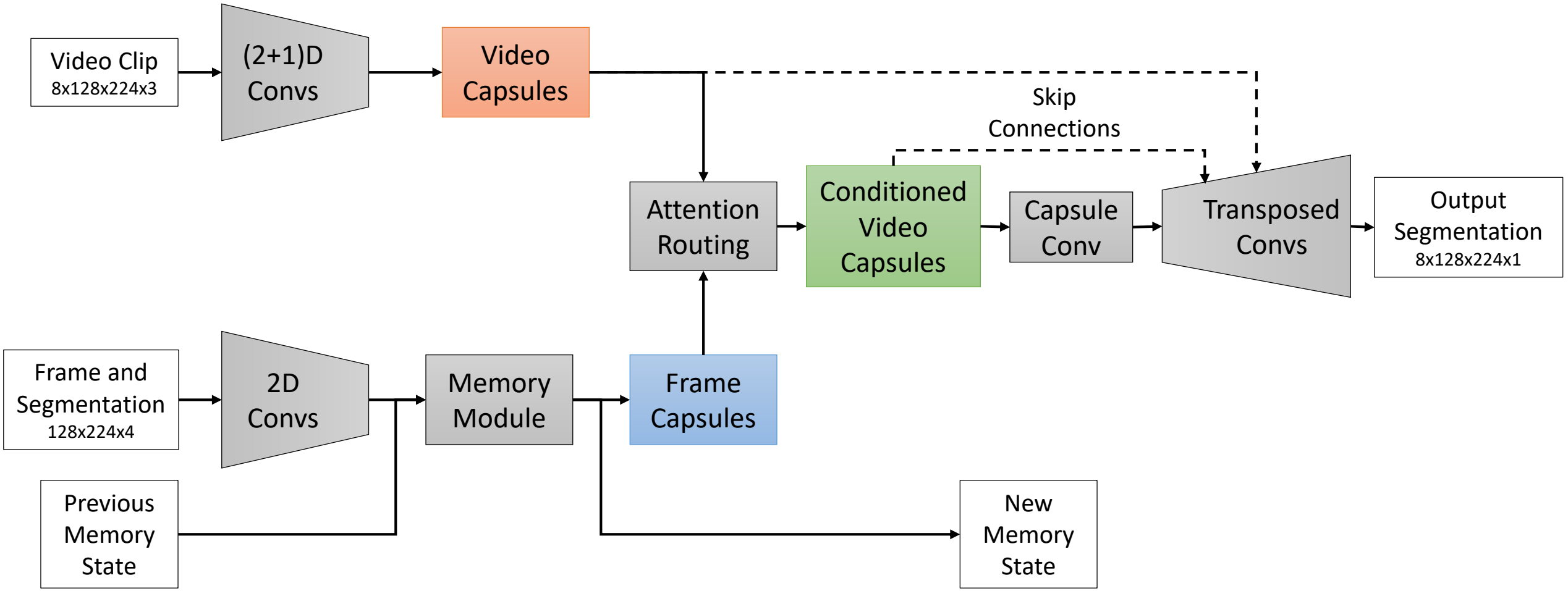
$D_{ij} \leftarrow \sum_h \left[\left(M_i^q - V_{ij}^k \right)^2 \right]^h$ ▷ For each i and j — Distance between query poses and key votes

$R_{ij}^v \leftarrow \frac{e^{-D_{ij}}}{\sum_j e^{-D_{ij}}}$ ▷ For each i — Obtain assignment coefficients

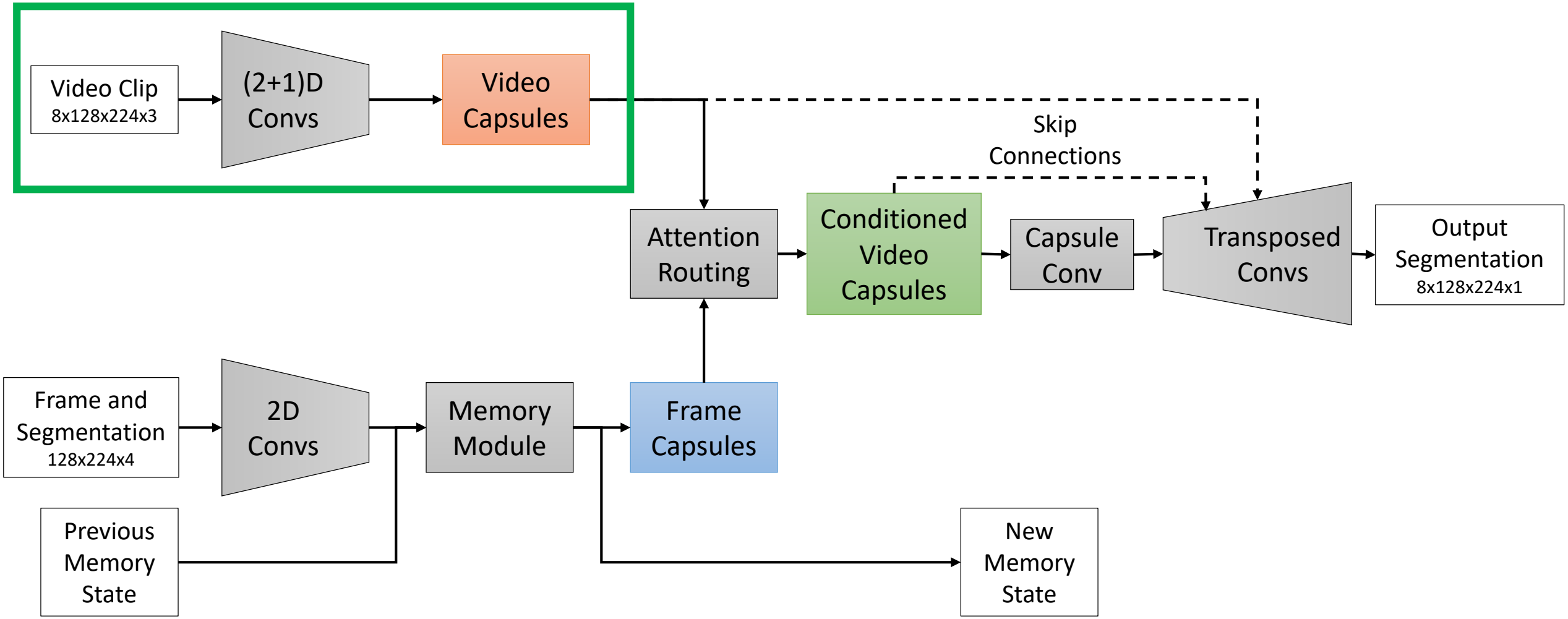
$a_j^v, M_j^v \leftarrow$ M-STEP$(a^{\mathcal{V}}, R^v, V^v, j)$ ▷ For each j — Get conditioned capsules through M-Step of EM-Routing algorithm

return a^v, M^v

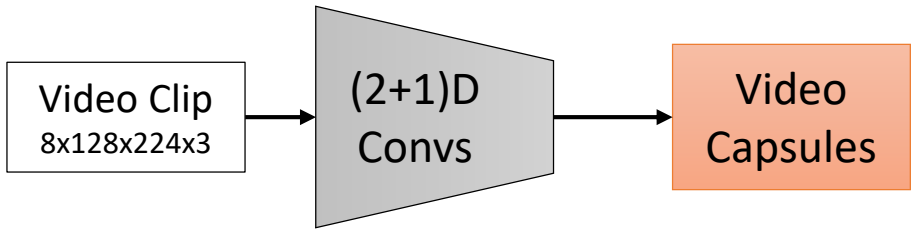
CapsuleVOS Architecture



Video Encoder

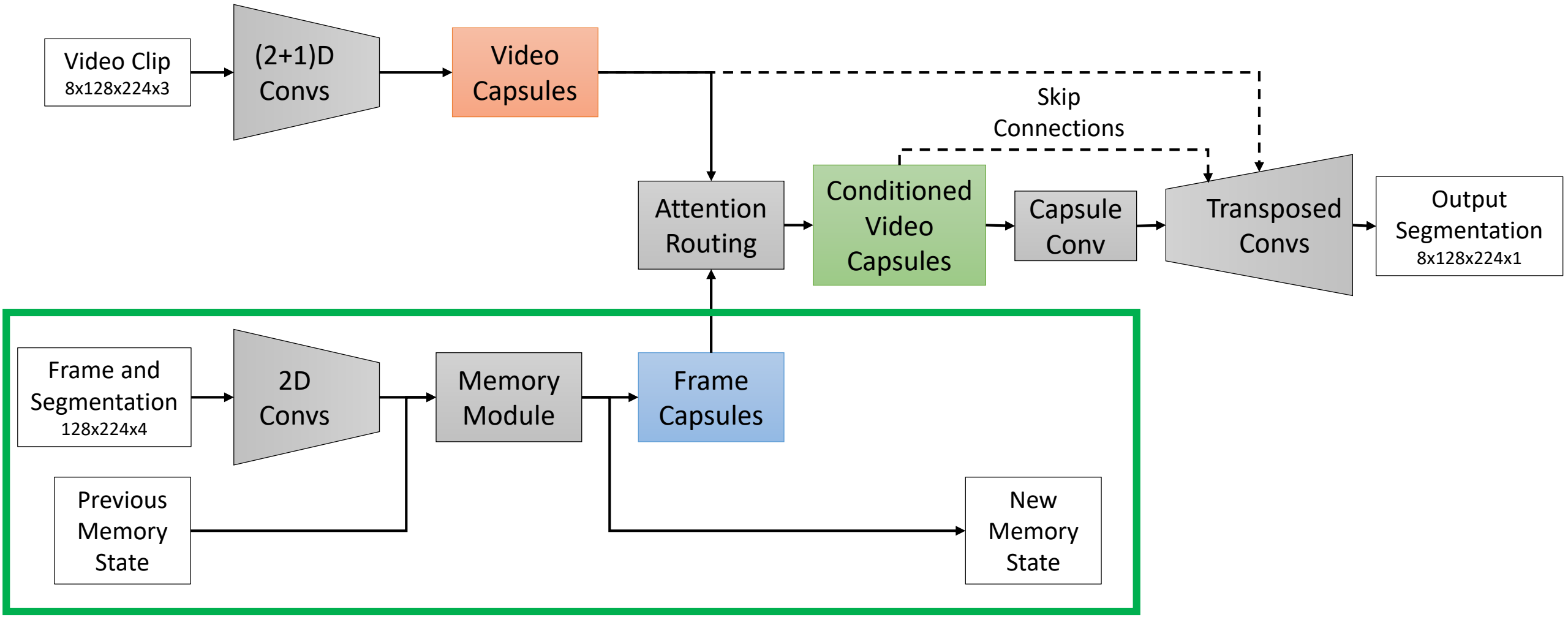


Video Encoder



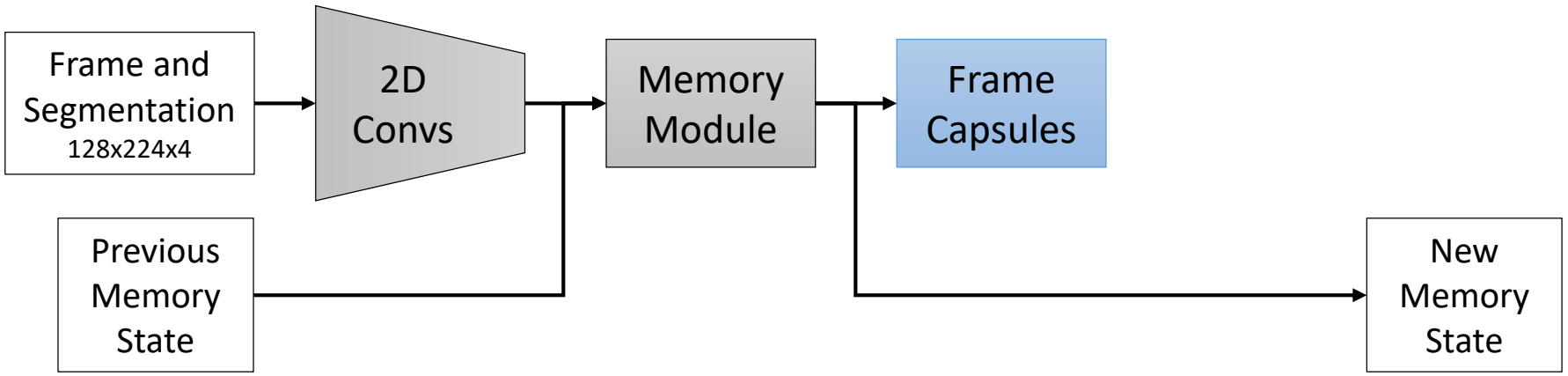
- Input video consists of 8 frames with a 128x224 resolution
- Six (2+1)D convolutions create 512 - 8x32x56 feature maps
- Video Capsules are obtained from a strided 3x3x3 convolution
  - The result is an 8x16x28 capsule layer with 12 capsule types

Frame Encoder with Memory Module

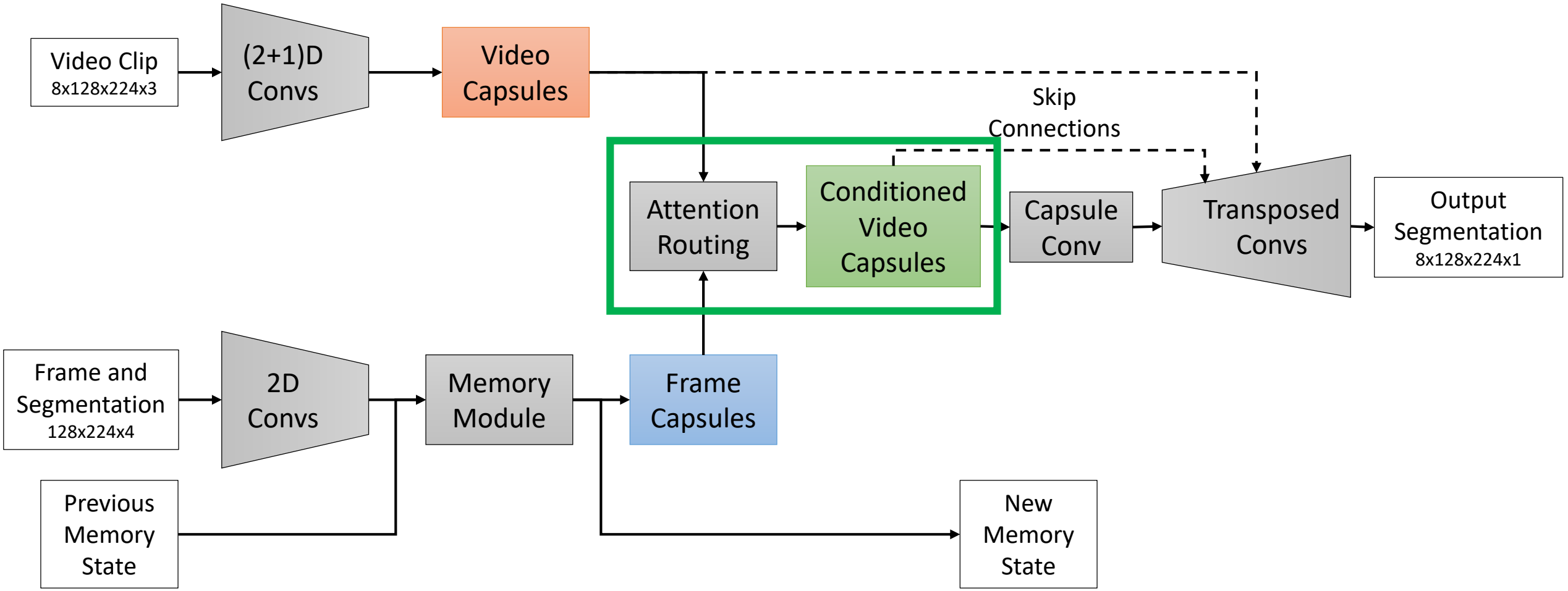


Frame Encoder with Memory Module

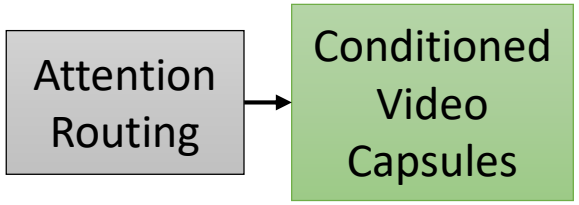
- Input consists of the first frame and segmentation mask
  - The input dimension is 128x224x4
- Four 2D convolutions create 128 - 32x56 feature maps
- The memory module consists of a ConvLSTM
  - This helps with objects that leave the scene or are occluded
- The frame capsule layer is 16x28, with 8 capsule types



Attention Routing

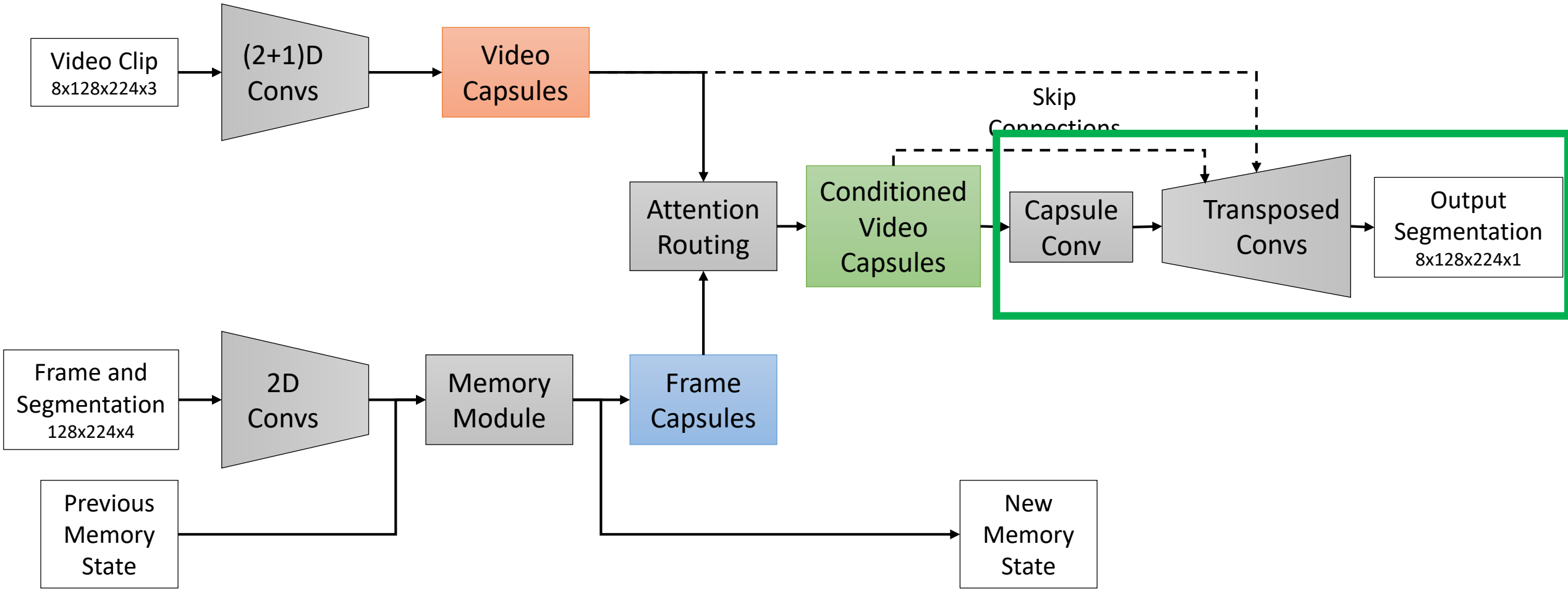


Attention Routing

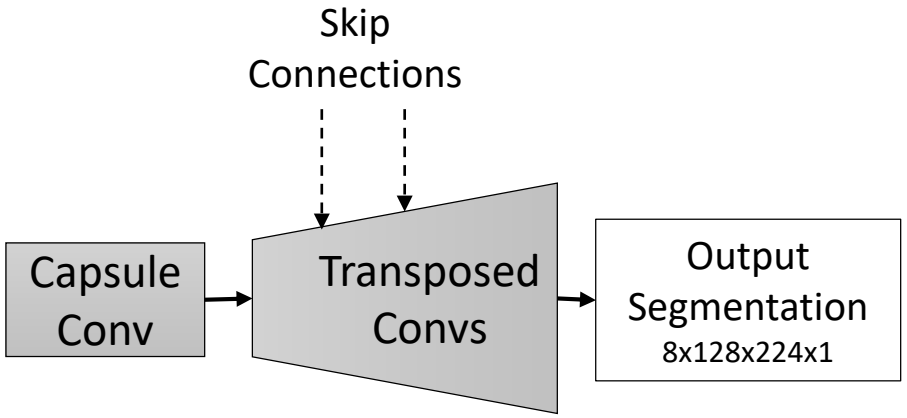


- Attention routing conditions the video capsules using frame capsules
- The conditioned capsule layer contains 16 capsule types
  - The operation is strided, so the dimension is 4x8x14

Conv Capsule Layer and Decoder Network

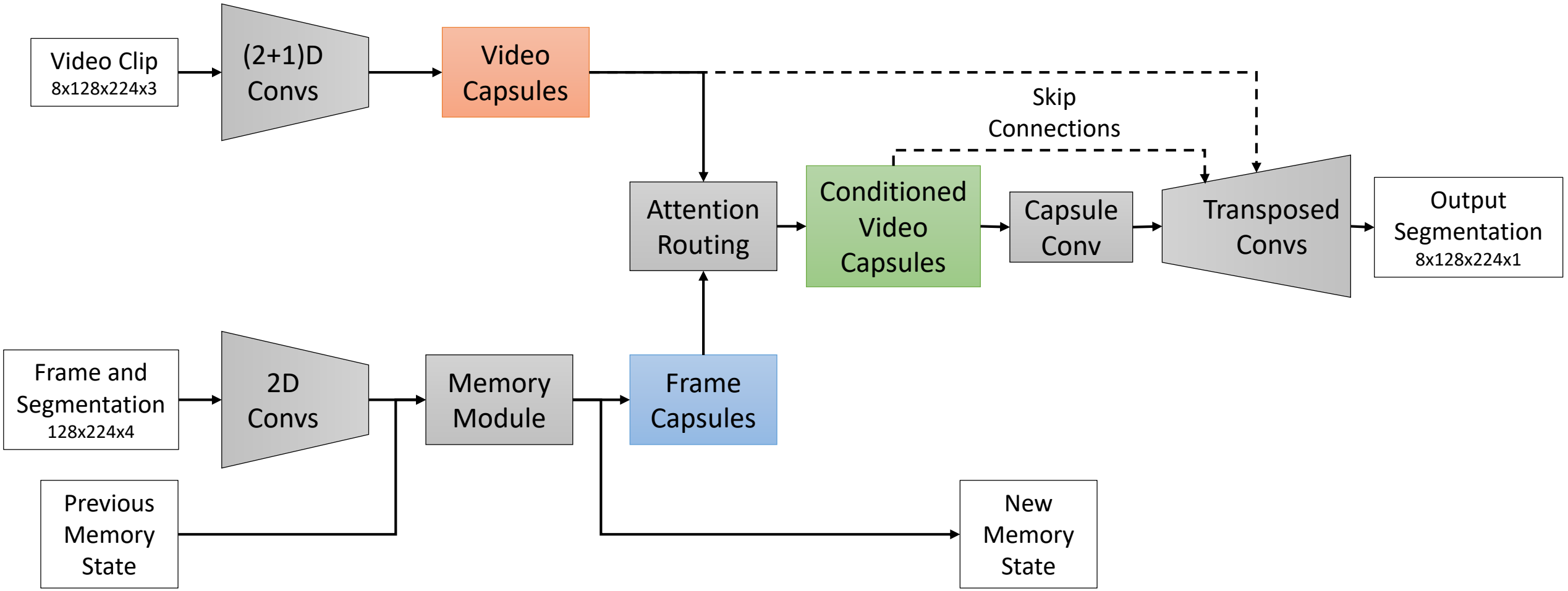


Conv Capsule Layer and Decoder Network



- A convolutional capsule layer follows the conditioned capsules
  - It has 16 capsule types and a dimension of 2x5x7
- The decoder network consists of 5 transposed convolutions
  - Has parameterized skip connections from previous capsule layers
- The output is 8 frames of binary segmentations with dimension 128x224

CapsuleVOS Architecture



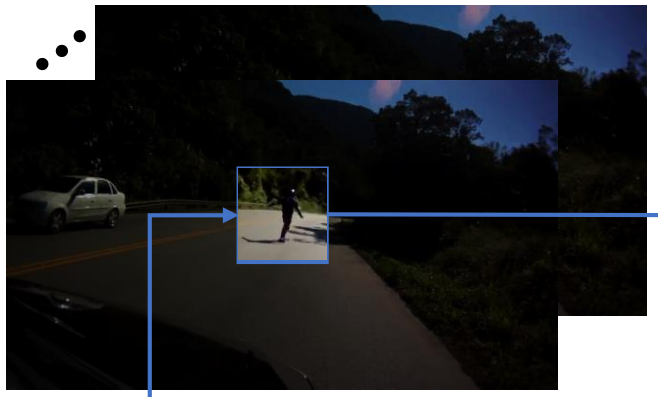
Zooming Module

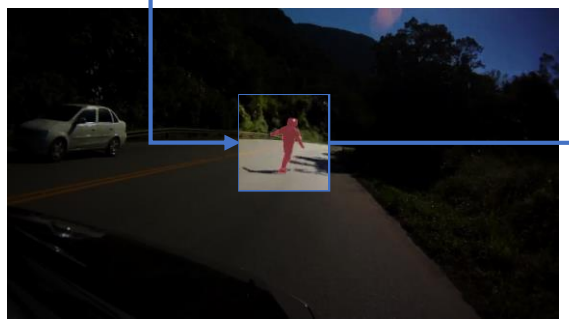


First Frame and Segmentation

Zooming Module

RGB Video Frames





First Frame and Segmentation

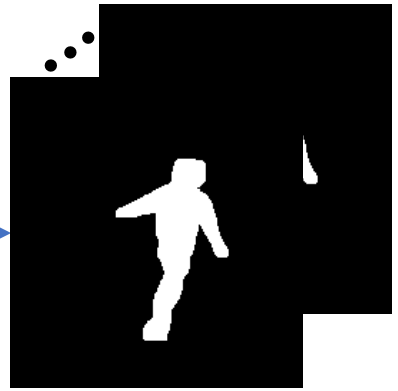
Zoomed in RGB Video Frames





Zoomed in First Frame and Segmentation

CapsuleVOS



Output Segmentations

Zooming Module

- Allows the method to segment a smaller objects successfully
  - Reduces the spatial region needed to be processed by CapsuleVOS
- Consists of a 2D ConvNet with an LSTM layer
  - The input is the concatenated reference frame and segmentation mask
- Outputs bounding box dimensions centered on the object of interest
  - These dimensions should encompass the object in the future 7 frames

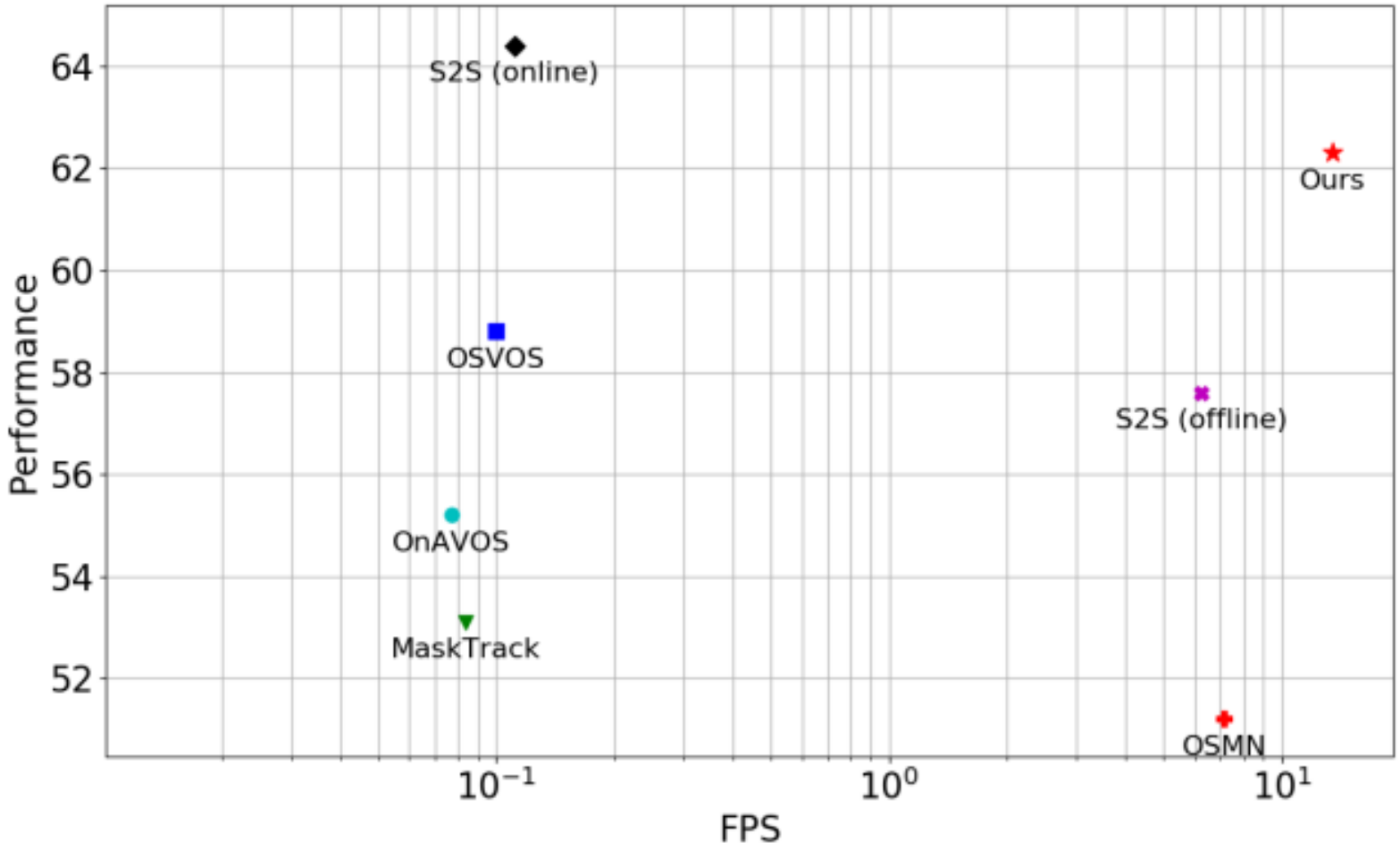
Objective Function

- CapsuleVOS is trained with two segmentation losses:
  - Binary cross-entropy loss: $L_S = -\frac{1}{N}\sum_{j=1}^{N} p_j \log(\hat{p}_j) - (1-p_j)\log(1-\hat{p}_j)$
  - Dice loss: $L_D = 1 - \frac{\sum_{i=1}^{N} \hat{y}_i y_i + \epsilon}{\sum_{i=1}^{N} \hat{y}_i + y_i + \epsilon} - \frac{\sum_{i=1}^{N} (1-\hat{y}_i)(1-y_i) + \epsilon}{\sum_{i=1}^{N} 2 - \hat{y}_i - y_i + \epsilon}$
- The zooming module uses an L2 loss:
  - $L_r = (b_h - \hat{b}_h)^2 + (b_w - \hat{b}_w)^2$
- The entire pipeline is trained end-to-end using a sum of these losses
  - $L = L_S + L_D + L_r$

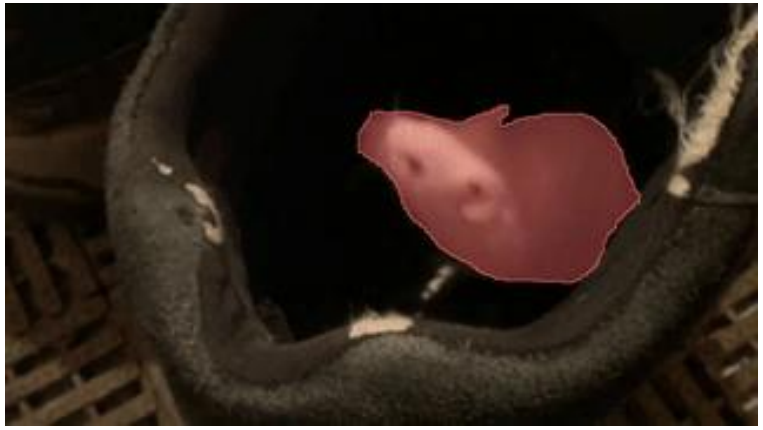
Quantitative Results – YoutubeVOS Dataset

Method	$\mathcal{J}$ seen	$\mathcal{J}$ unseen	$\mathcal{F}$ seen	$\mathcal{F}$ unseen	Overall	Speed (frames/s)
OSVOS [2]	59.8	**54.2**	60.5	**60.7**	58.8	0.10
OSMN [37]	60.0	40.6	60.1	44.0	51.2	7.14
S2S (offline) [35]	66.7	48.2	65.5	50.3	57.6	6.25
Our Method	**67.3**	53.7	**68.1**	59.9	**62.3**	**13.5**

Quantitative Results – Speed Analysis



Qualitative Results – Single Object








Qualitative Results – Multiple Objects













Effect of Memory Module

Network without Memory Module



Object Successfully Segmented

Object Successfully Segmented

Object Successfully Segmented

Object leaves the scene

Object reenters the scene but is lost

Object is lost

Object is lost



Network with Memory Module

Effect of Memory Module

Network without Memory Module



Object Successfully Segmented

Object Successfully Segmented

Object Successfully Segmented

Object leaves the scene

Object reenters the scene and is successfully segmented

Object Successfully Segmented

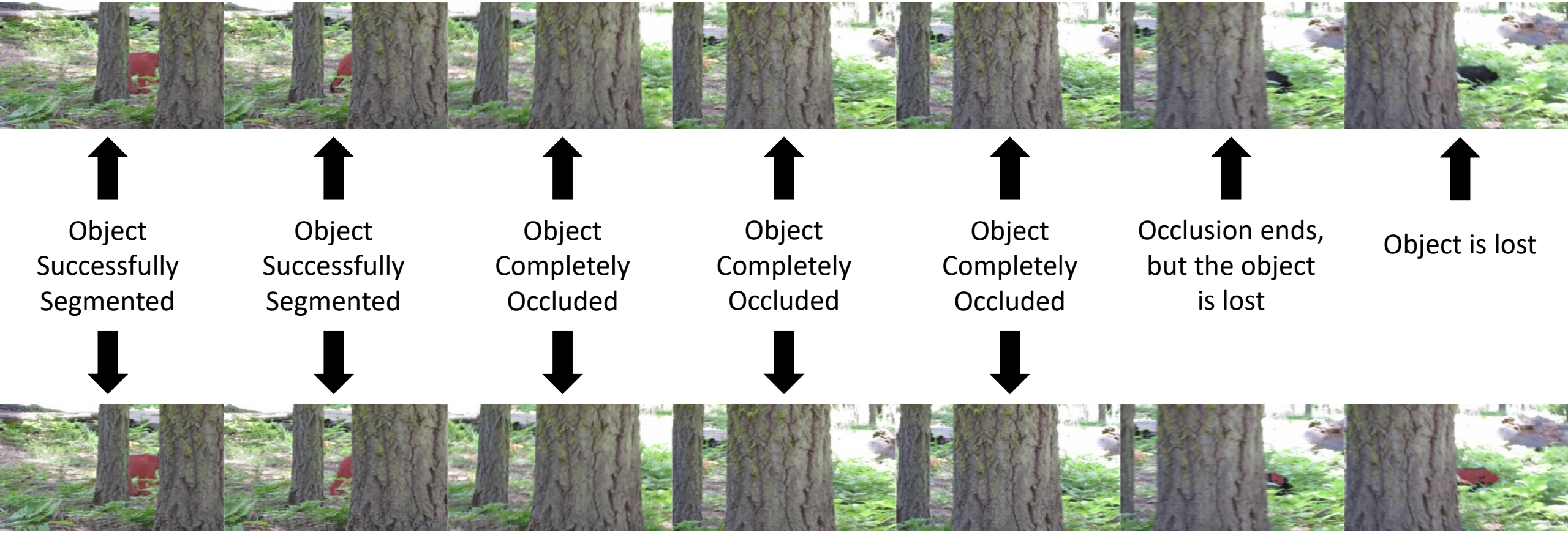
Object Successfully Segmented



Network with Memory Module

Effect of Memory Module

Network without Memory Module



Object Successfully Segmented

Object Successfully Segmented

Object Completely Occluded

Object Completely Occluded

Object Completely Occluded

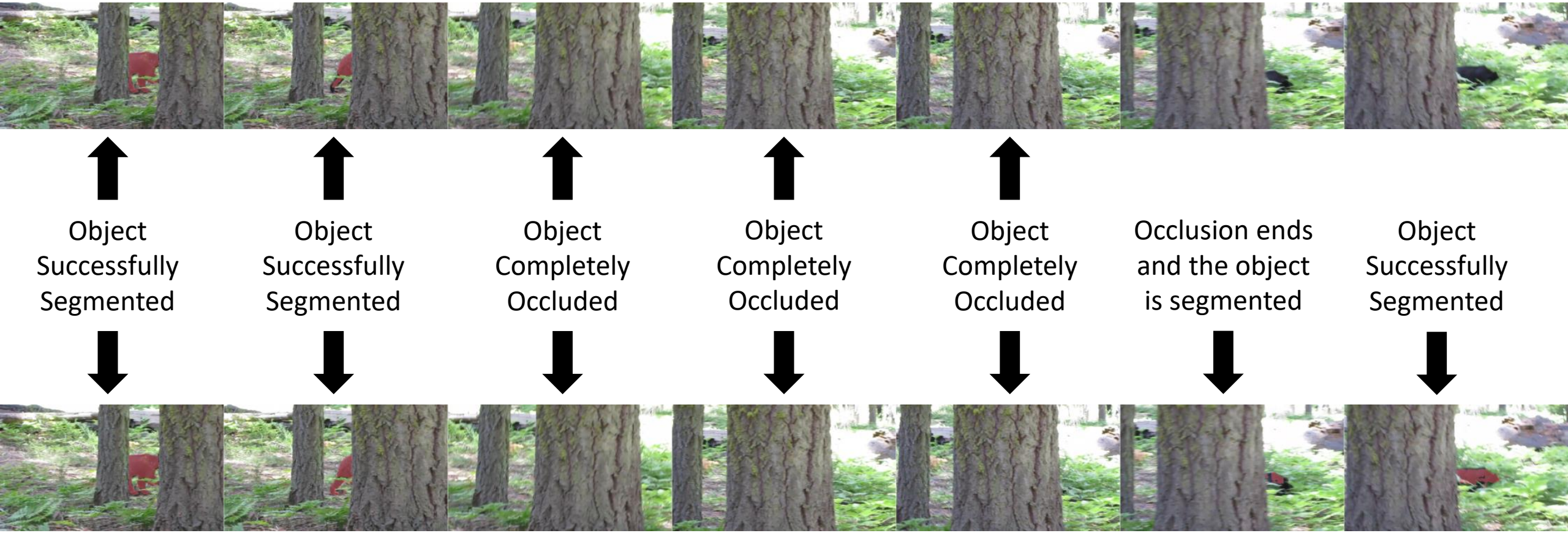
Occlusion ends, but the object is lost

Object is lost

Network with Memory Module

Effect of Memory Module

Network without Memory Module



Network with Memory Module

Effect of Memory Module





Effect of the Zooming Module



Effect of the Zooming Module

Network without Zooming Module:





Frame #20

Frame #90





Network with Zooming Module:

Effect of the Zooming Module



Effect of the Zooming Module

Network without Zooming Module:





Frame #30

Frame #95





Network with Zooming Module:

Effect of Zooming Module







