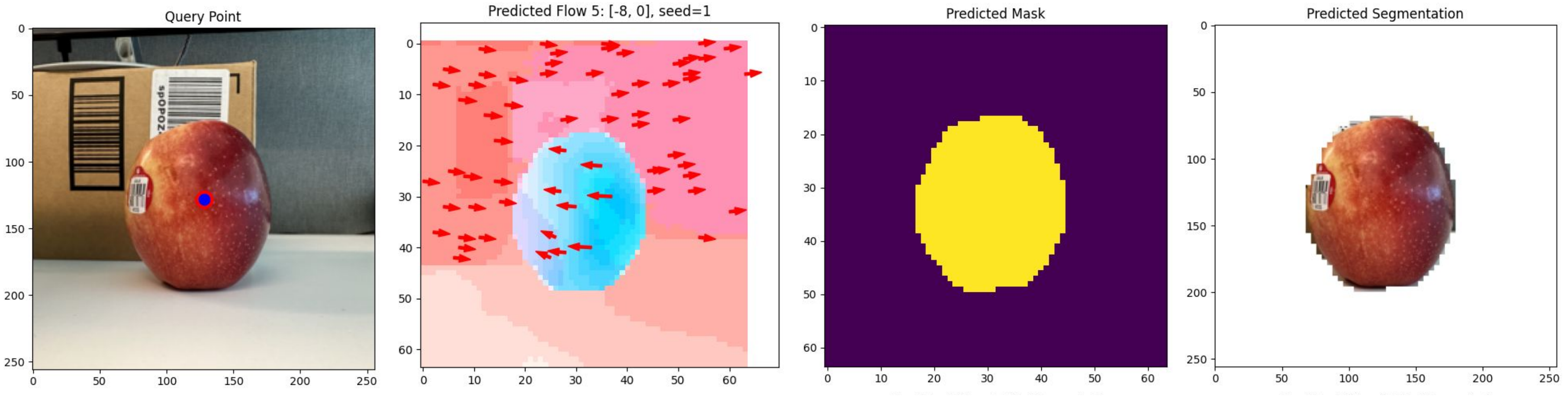
Spelke Object Segmentation with Counterfactual World Modeling



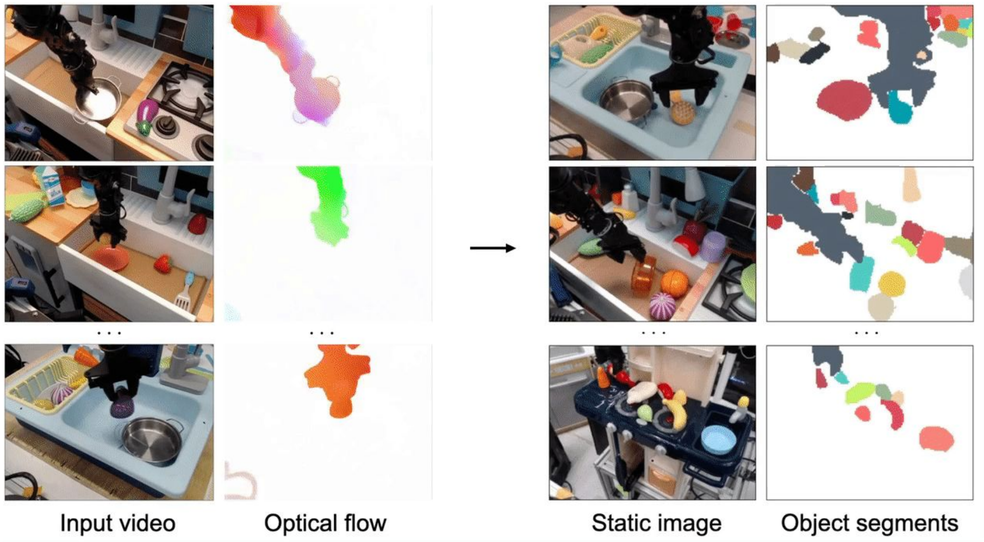
Jared Watrous

Stanford NeuroAILab - Klemen Kotar, Honglin Chen, Wanhee Lee, Rahul Vankatesh, Daniel Yamins

Spelke Objects

A Spelke object is *a collection of physical stuff that moves together during commonplace physical interactions* [1,3]

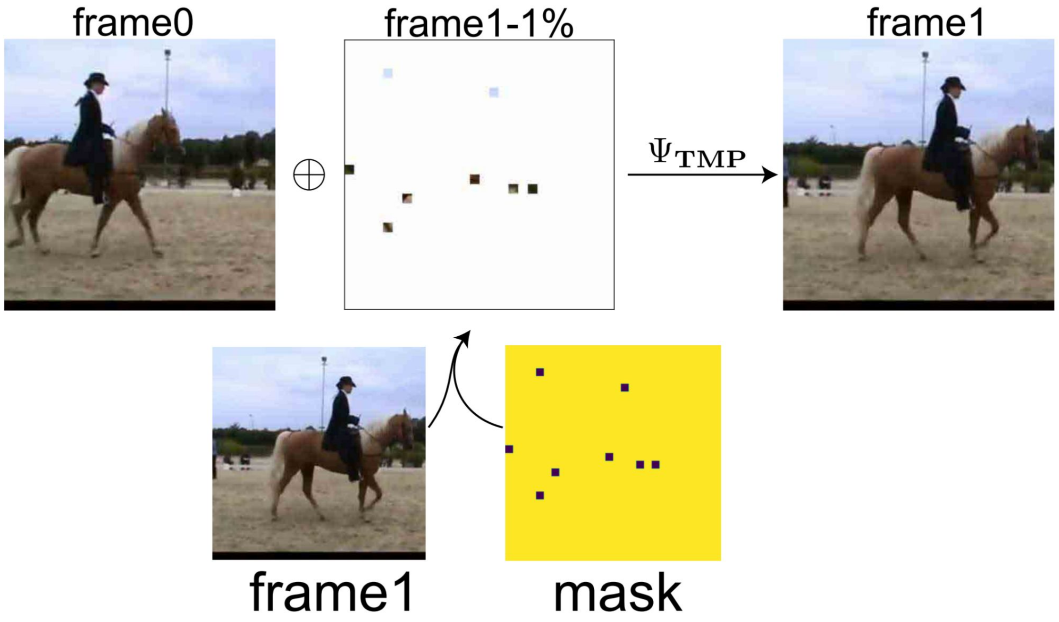
- Named after cognitive scientist Elizabeth Spelke



[1] Daniel M. Bear, Kevin Feigelis, Honglin Chen, Wanhee Lee, Rahul Venkatesh, Klemen Kotar, Alex Durango, and Daniel L. K. Yamins. Unifying (machine) vision via counterfactual world modeling, 2023.
[3] Honglin Chen, Rahul Venkatesh, Yoni Friedman, Jiajun Wu, Joshua B Tenenbaum, Daniel LK Yamins, and Daniel M Bear. Unsupervised segmentation in real-world images via spelke object inference. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX, pages 719–735. Springer, 2022.

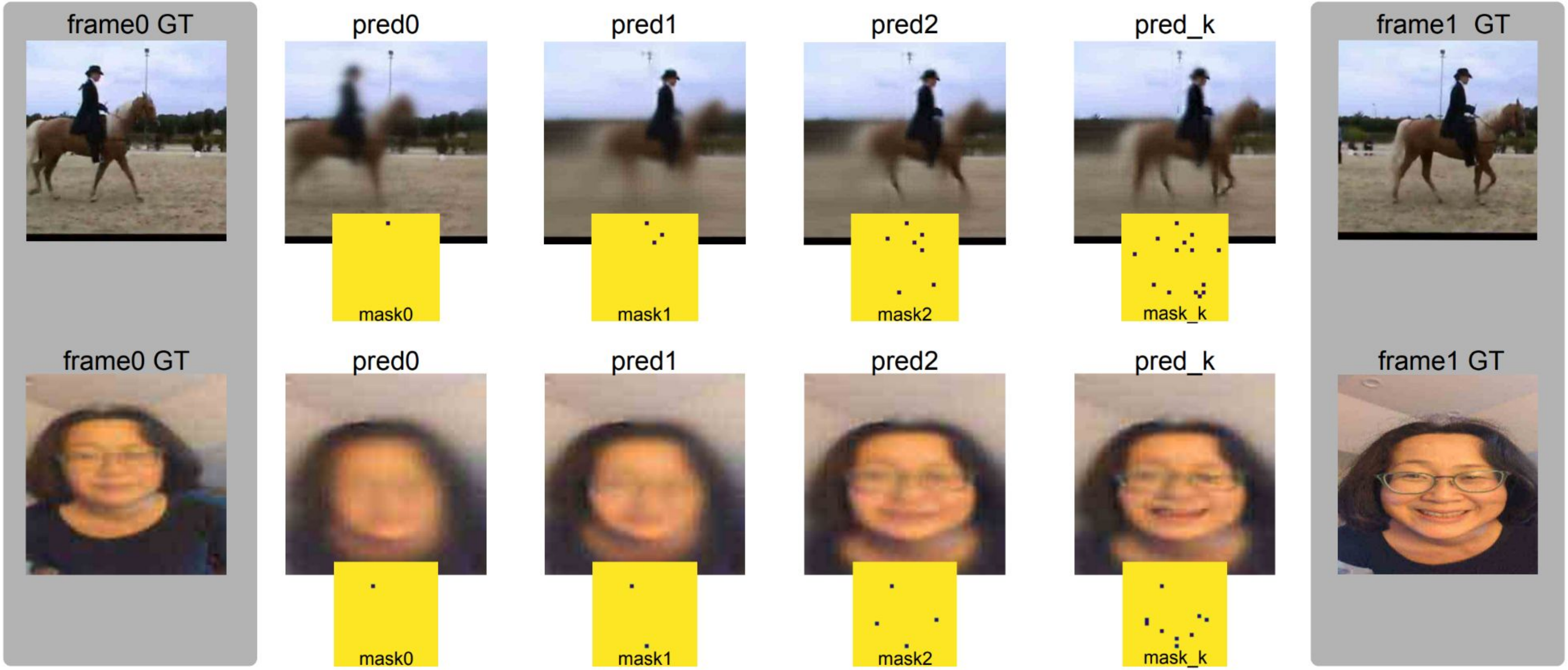
Counterfactual World Modeling

- “Next frame prediction” framework for videos
- Given frame0 and *part* of frame1, predict the rest of frame1
- Unsupervised training
- Referred to as a “temporally factored masked autoencoder”



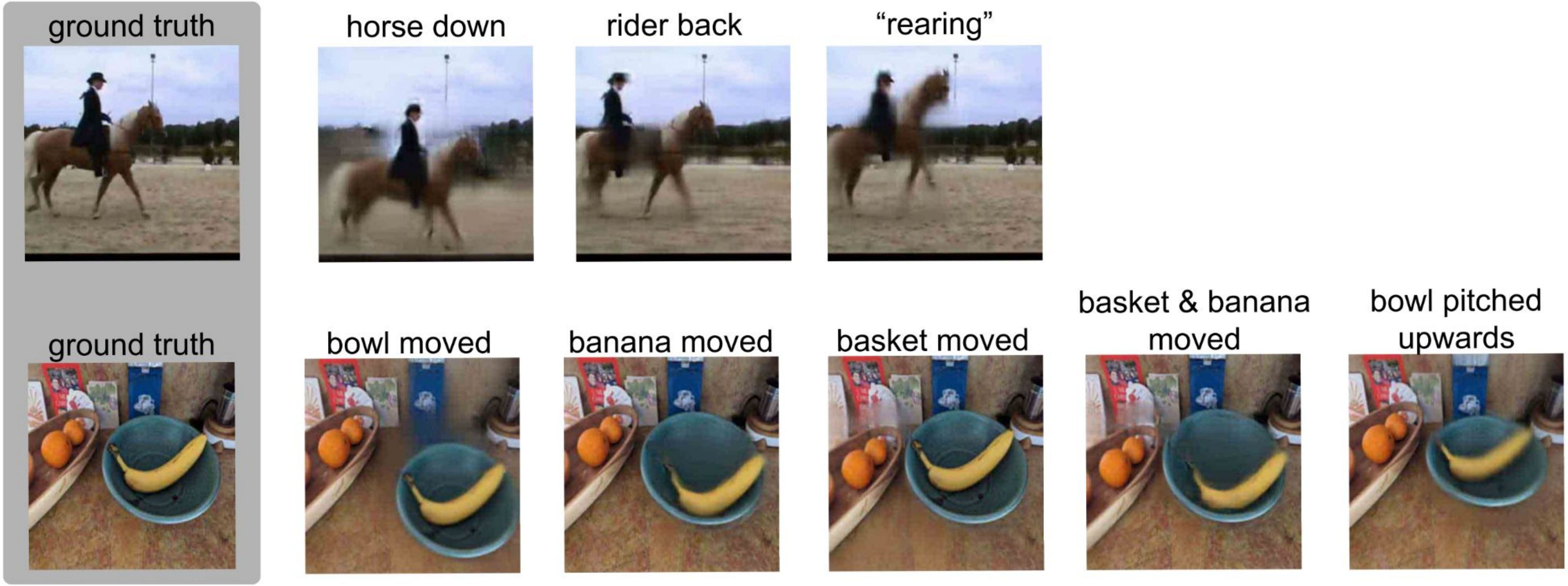
Counterfactual World Modeling

Factual prediction: Provide “ground truth” frame1 patches



Counterfactual World Modeling

Counterfactual prediction: Provide "counterfactual" frame1 patches



Counterfactual World Modeling

CWM offers many natural readouts:

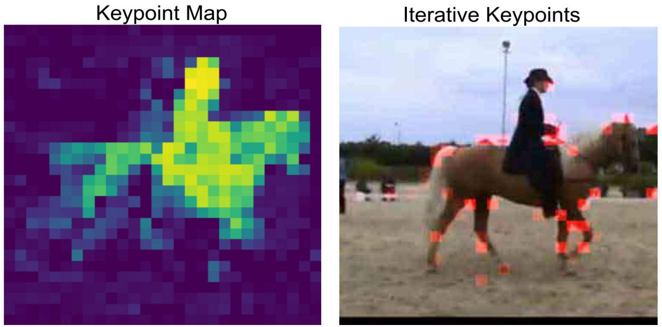
- RGB Prediction
- Keypoint Extraction

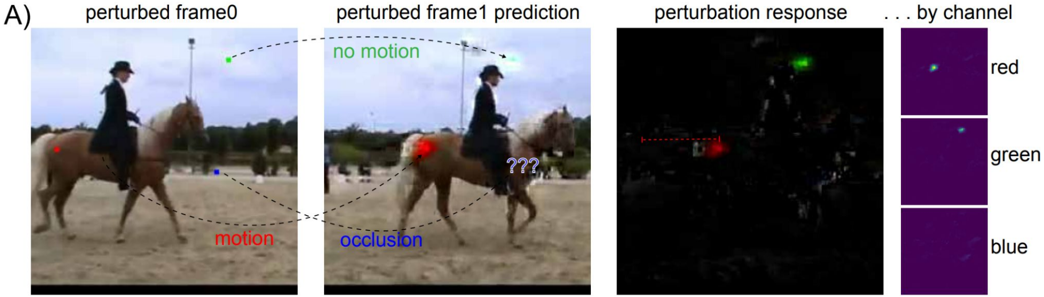
*If the model is **differentiable:***

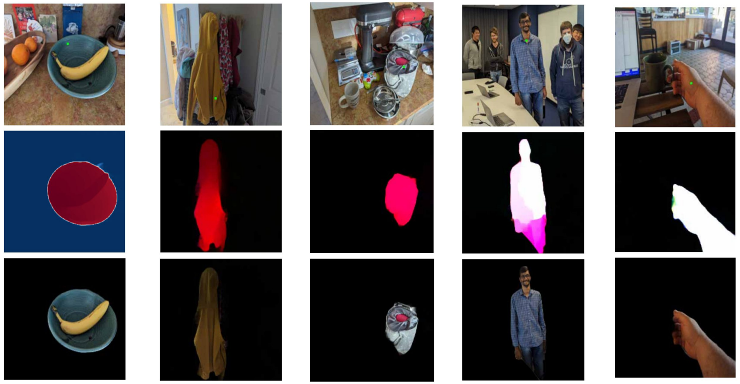
- Optical Flow
- Spelke Segmentation

New task: Control the camera

- Novel View Synthesis
- Relative Depth





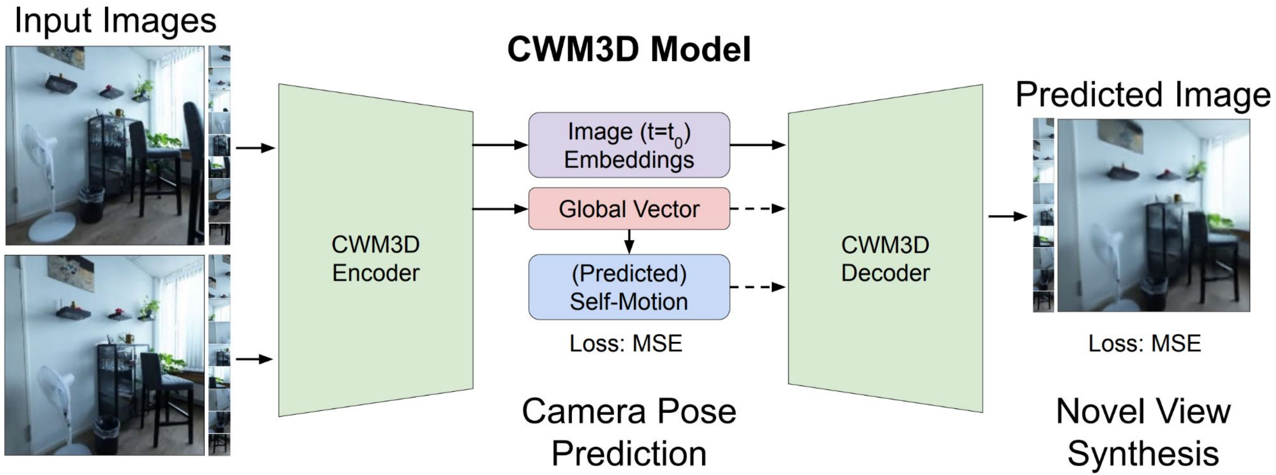


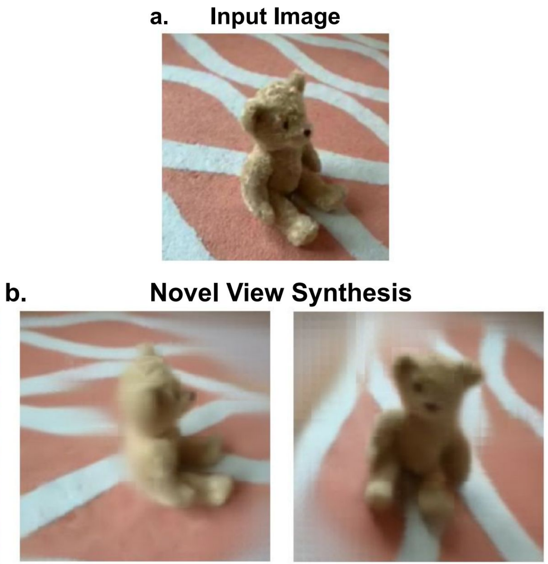
Counterfactual Camera Motion

Original model is a ViT-based regression model (L2 pixel loss)

First attempt: Add a new “patch” representing camera motion

- Linearly mapped from 6dof relative camera pose
- Somewhat works, but very blurry





Causal Counterfactual World Modeling

Why are the images blurry?

- Blurriness is a result of the model’s uncertainty
- Natural result of mean regression (L2 pixel loss)

How do we avoid mean regression?

- Choose a model architecture that samples predictions instead of regressing
- Diffusion models won’t work for CWM

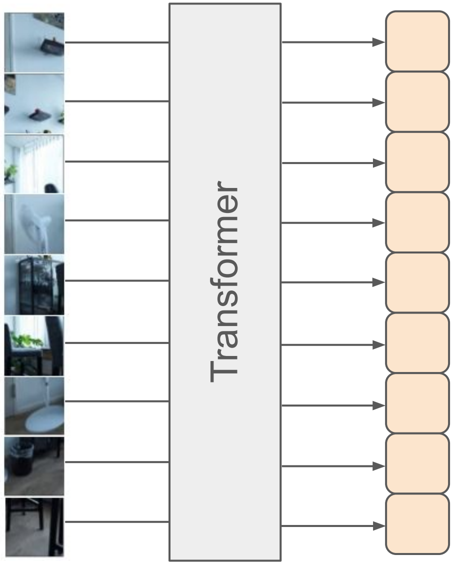
Idea: Use next token prediction (LLM/VLM architecture)

- **Causal Counterfactual World Modeling (CCWM)**

Causal CWM: Tokenization

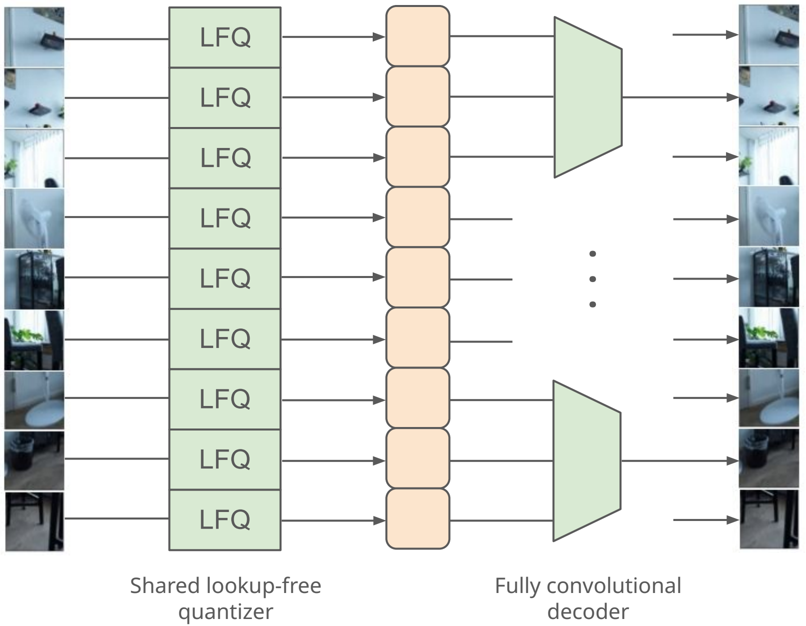
VLMs typically use a ViT-based tokenizer

- Often loses spatial locality

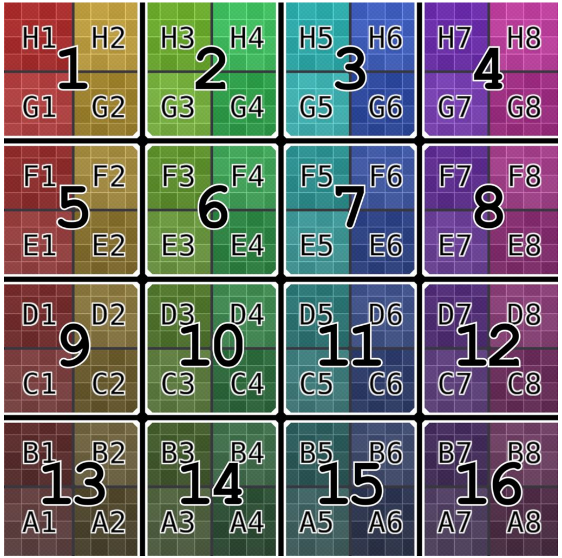


CCWM uses a **local patch quantizer**

- Guarantees spatial locality

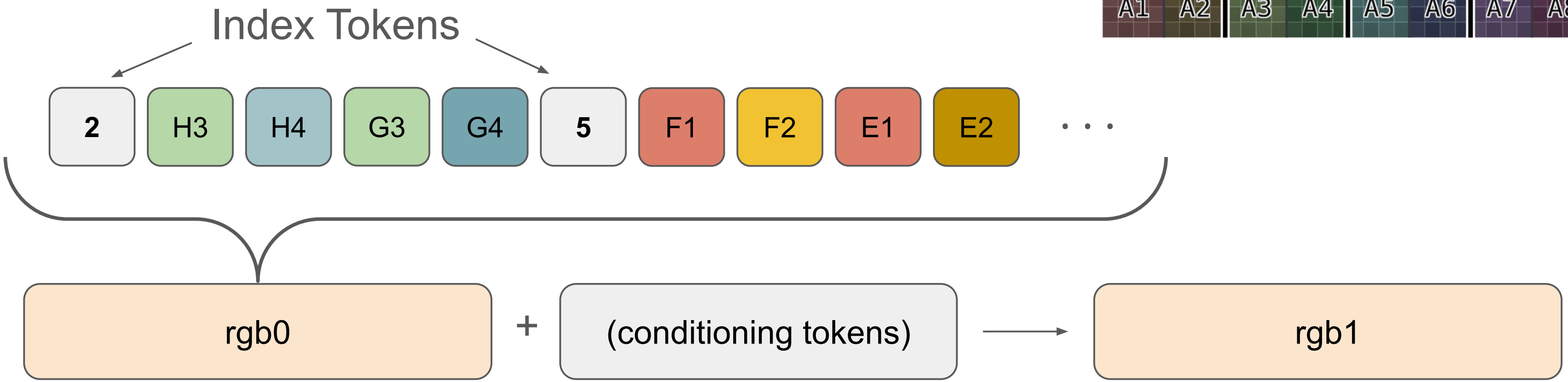


Causal CWM: Sequence Construction



Instead of raster order, use **index tokens** to indicate which part of the image the next tokens represent

Allows us to decode frame 1 in **any arbitrary order**

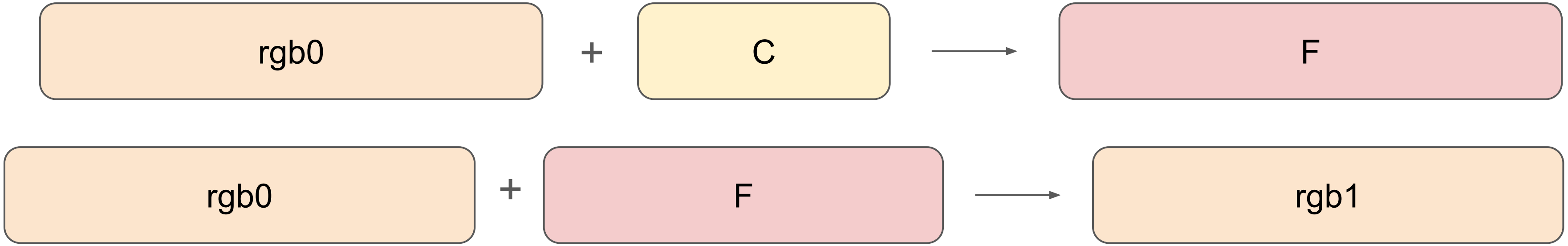


Causal CWM: Current Models

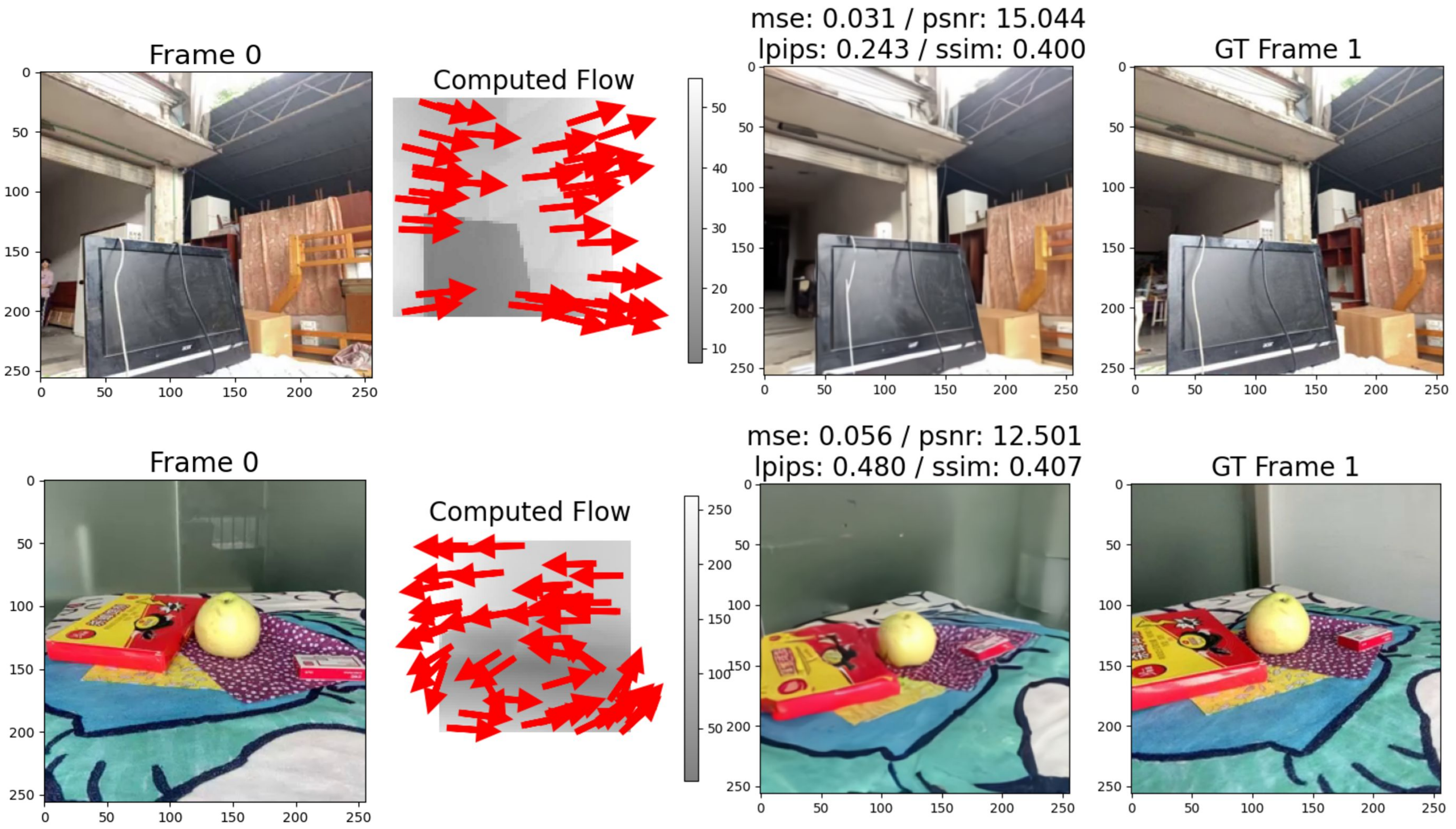
In the long term, we hope to include every feature in one model

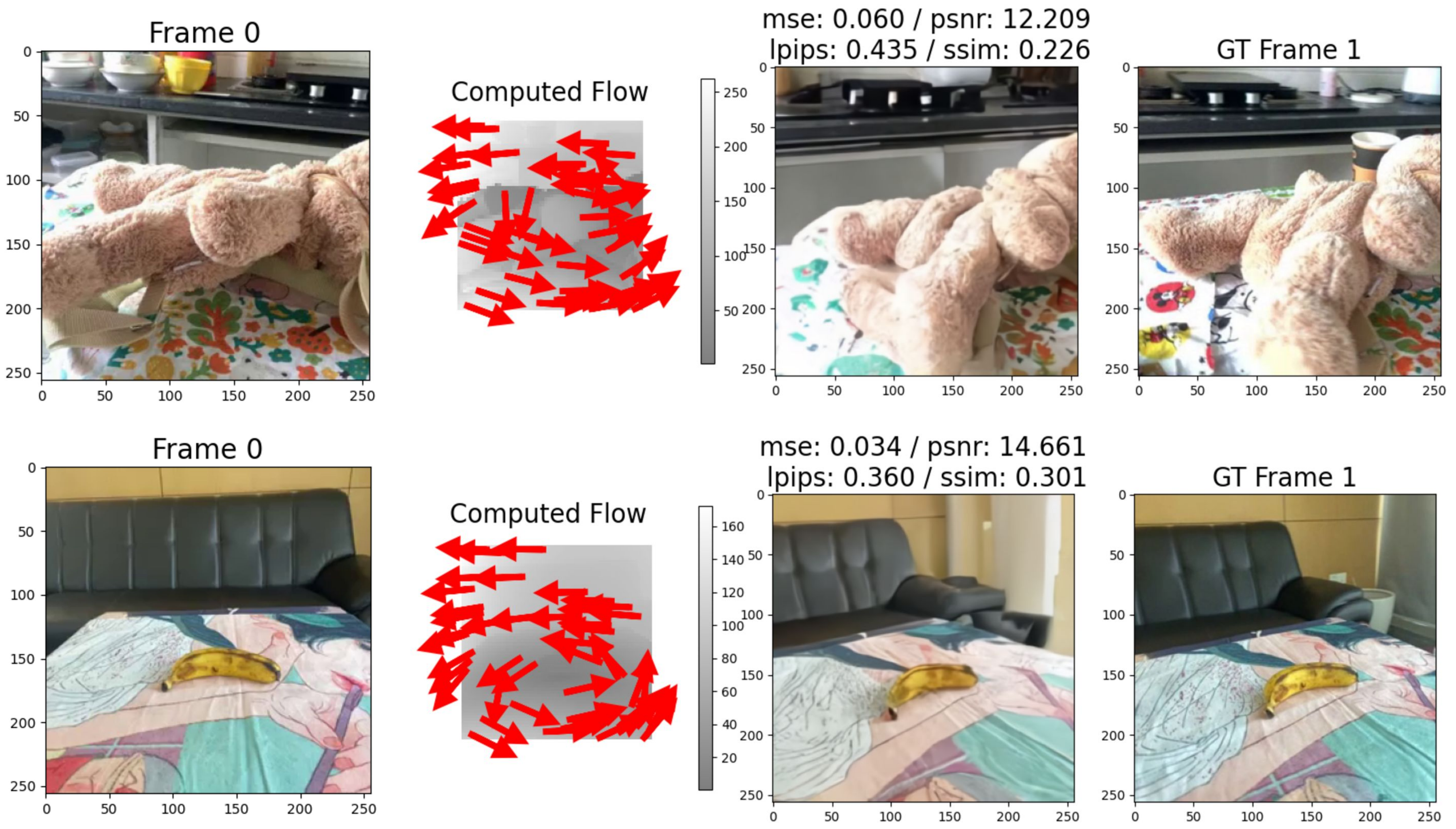
Currently, we have two separate models:

1. Frame 0 (rgb0) + Camera Pose (C) → Optical Flow (F)
2. Frame 0 (rgb0) + Optical Flow (F) → Frame 1 (rgb1)



Flow images are constructed the same way as rgb images, with a separate quantizer.
Each flow token corresponds to one rgb patch (4 flow tokens per index token)


Frame 0
Computed Flow
mse: 0.031 / psnr: 15.044
lpips: 0.243 / ssim: 0.400
GT Frame 1
Frame 0
Computed Flow
mse: 0.056 / psnr: 12.501
lpips: 0.480 / ssim: 0.407
GT Frame 1


Frame 0
Computed Flow
mse: 0.060 / psnr: 12.209
lpips: 0.435 / ssim: 0.226
GT Frame 1
Frame 0
Computed Flow
mse: 0.034 / psnr: 14.661
lpips: 0.360 / ssim: 0.301
GT Frame 1

CWM vs. CCWM

Compared to ViT-based CWM, Causal CWM **keeps:**

- RGB Prediction
- Keypoint Extraction
- Optical Flow (by adding flow tokens)

CCWM **adds:**

- Novel View Synthesis
- Uncertainty Management (no more blur)

CCWM **loses:**

- Segmentation (the model is no longer differentiable)

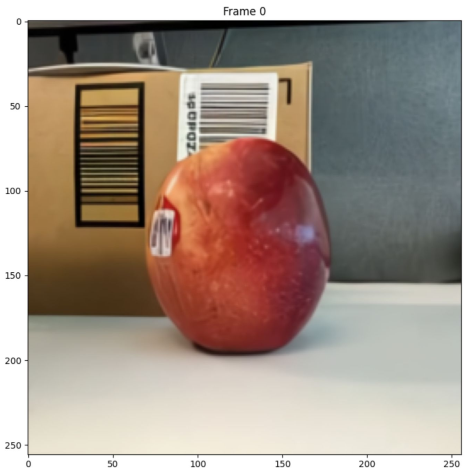
<u>**How do we reintroduce Spelke object segmentation into Causal CWM?**</u>

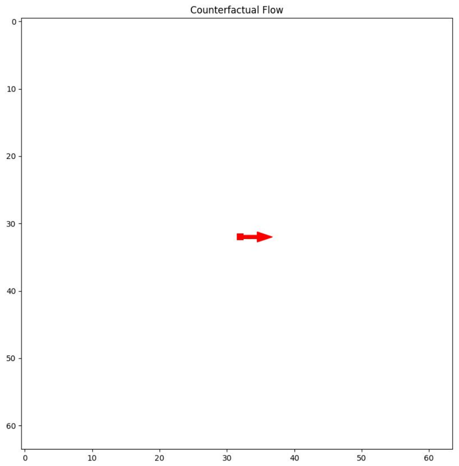
Method: Sparse to Dense Flow

Observation: Just as we can condition on *some* rgb1, we can condition on *some* flow

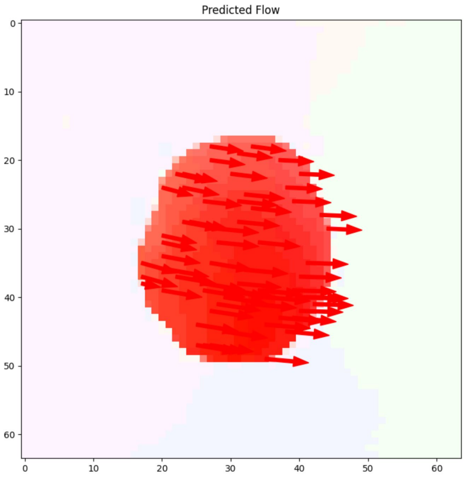
- This creates a **sparse to dense flow** predictor

rgb0 + C + 1% F → 99% F





→



Method: Sparse to Dense Flow

Procedure:

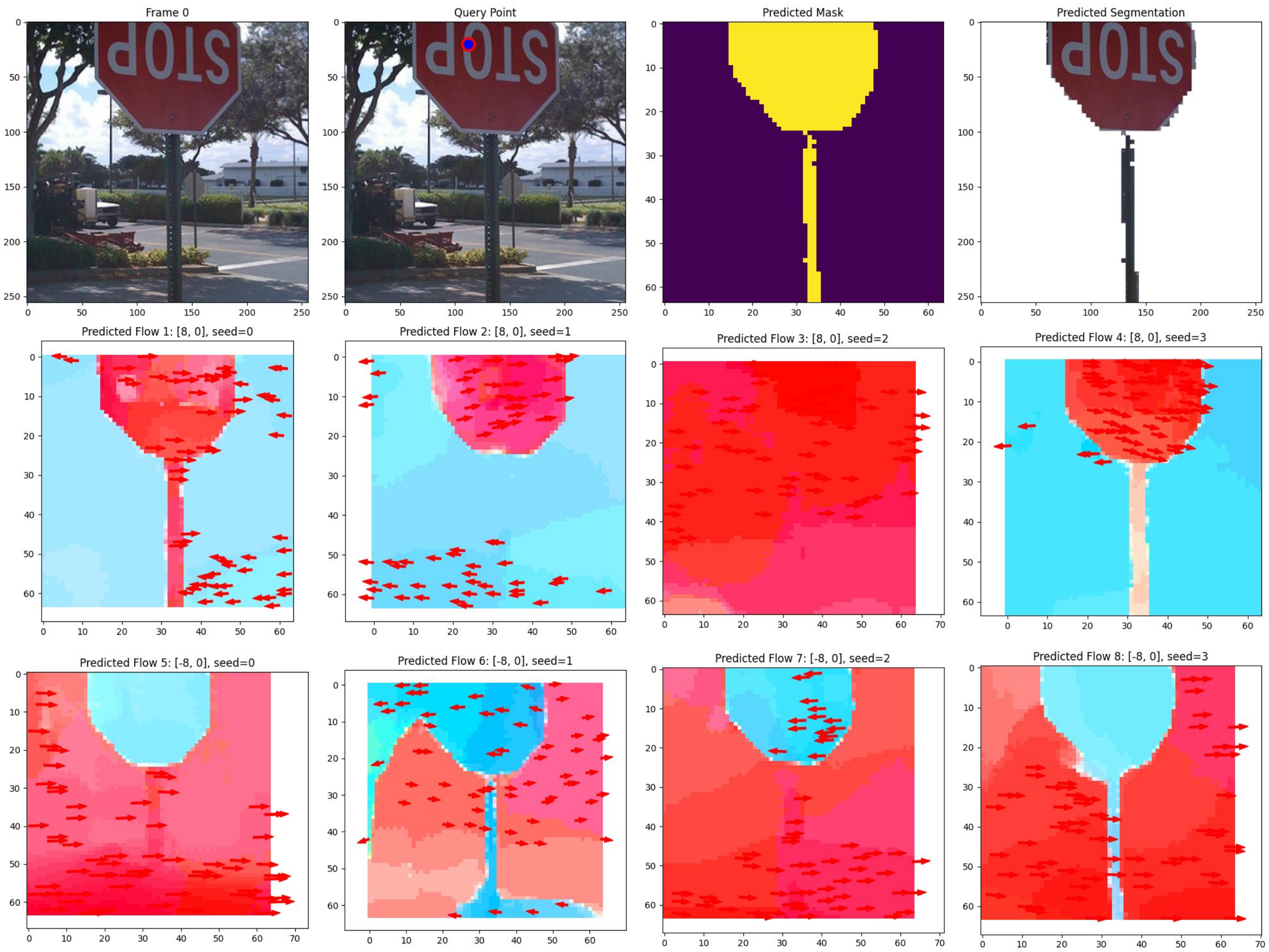
1. Give counterfactual camera motion to the right
2. Give counterfactual flow to the right at the query point
   - The object “moves right”, and everything else “moves left”
3. Compute sparse to dense flow
4. Repeat with different motion directions/seeds

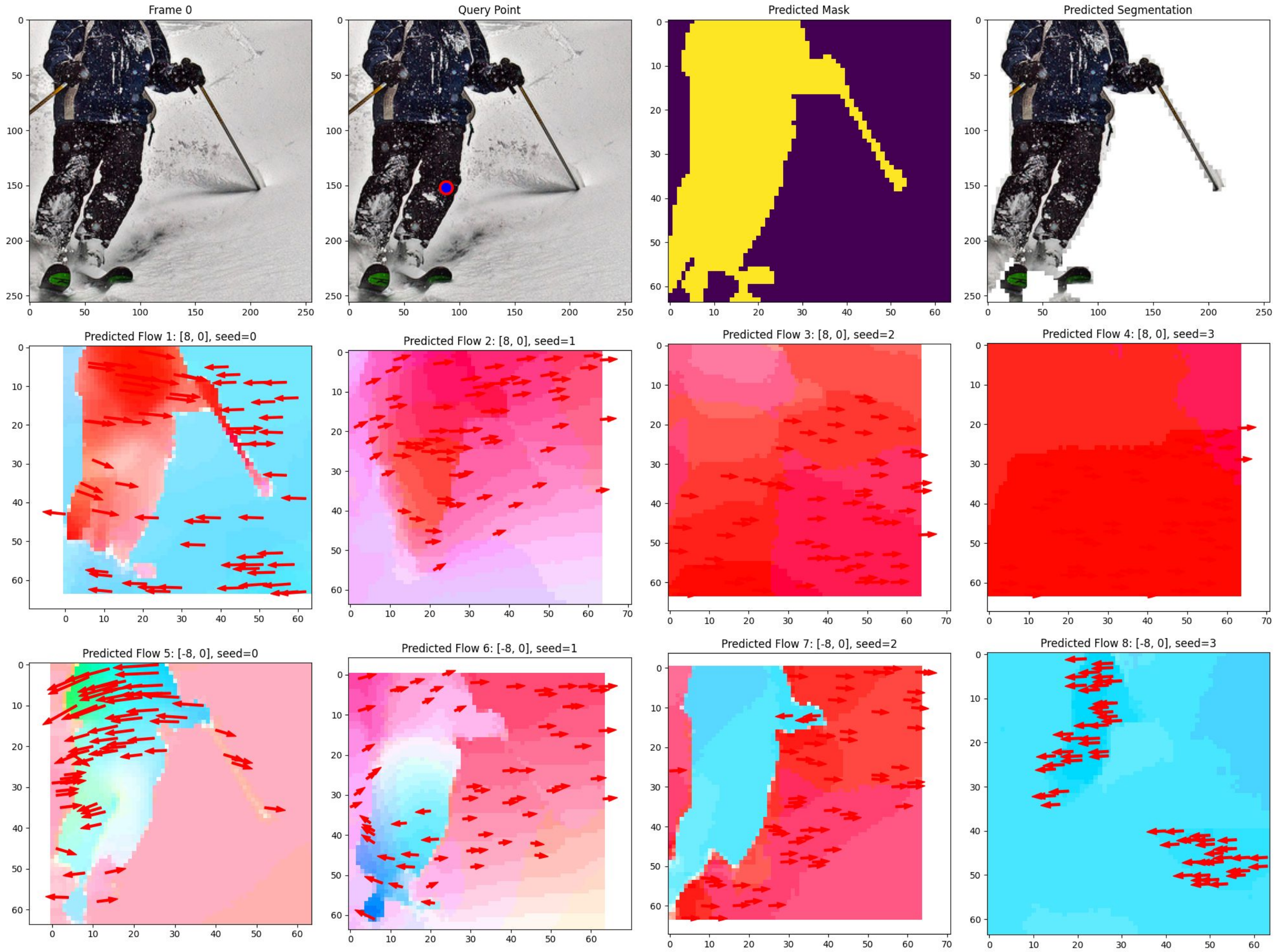
For each patch *p*, compute score S_p using *N* camera motions *C* and patch flows F_p:

$$S_p = \frac{1}{N} \sum_{i=1}^{N} \mathrm{sign}(F_p^{(i)} \cdot C^{(i)})$$

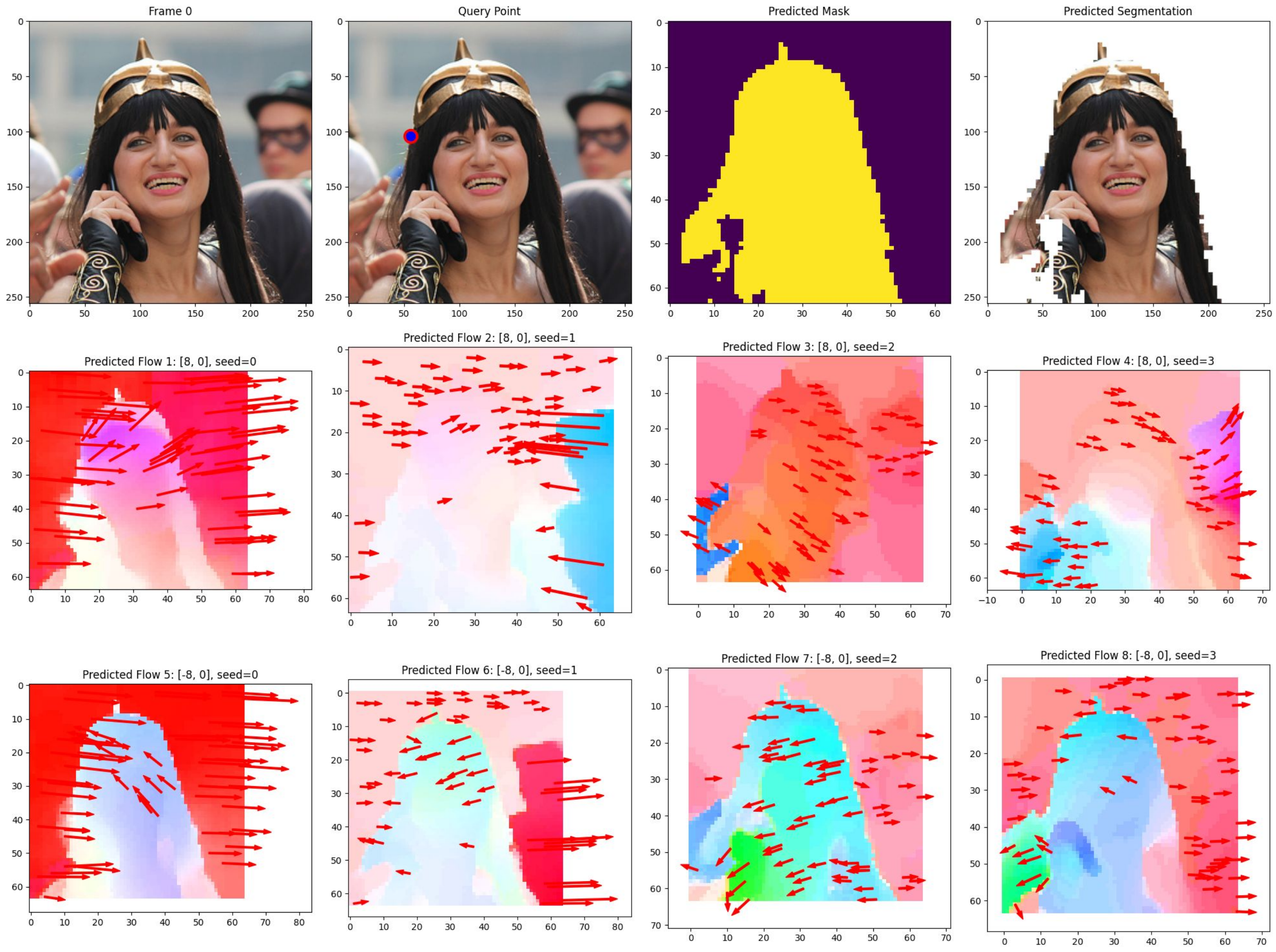
Score measures similarity of predicted flow to expected Spelke object flow.

Segmentation includes all patches with positive score $S_p > 0$

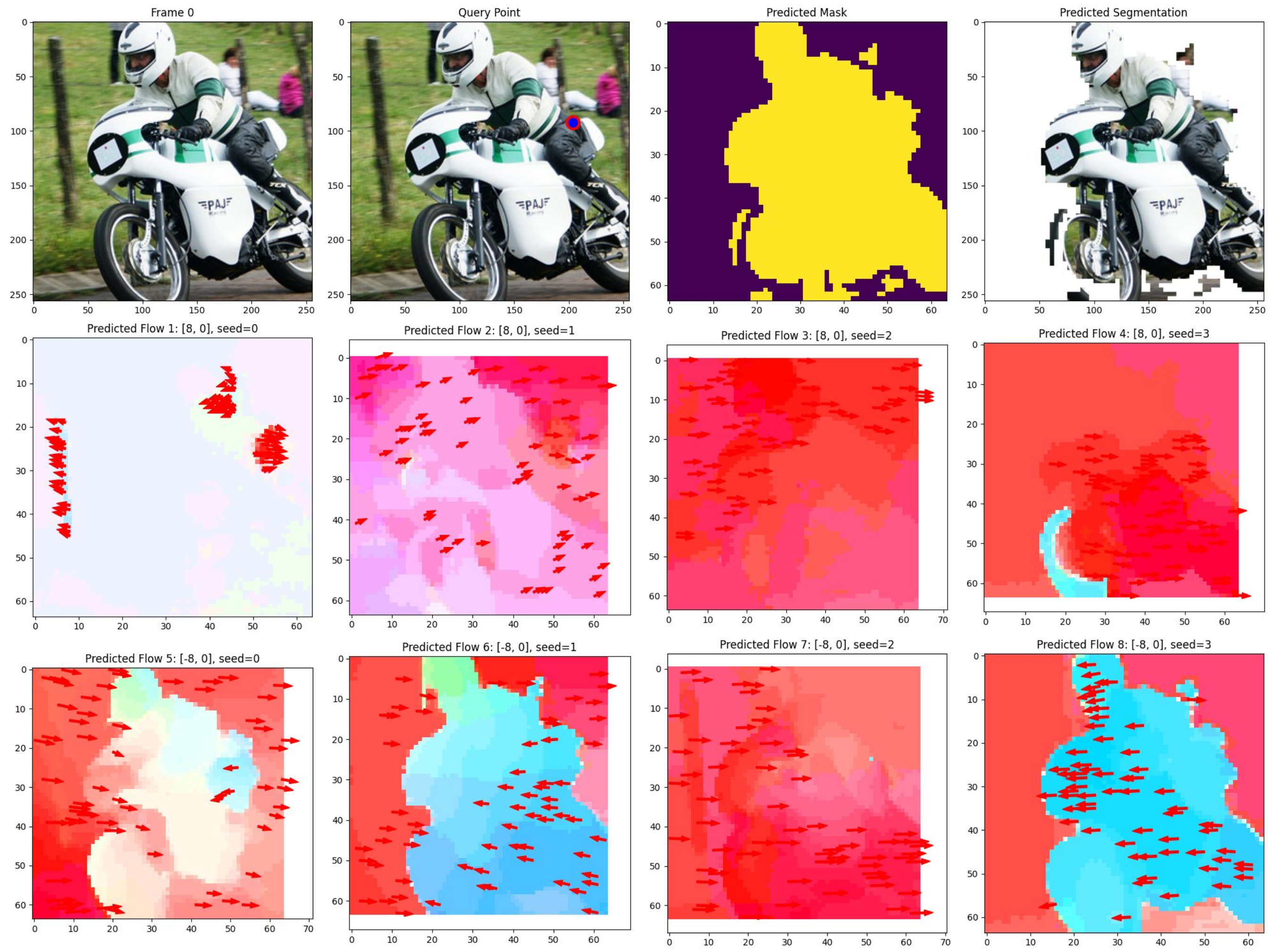

Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3



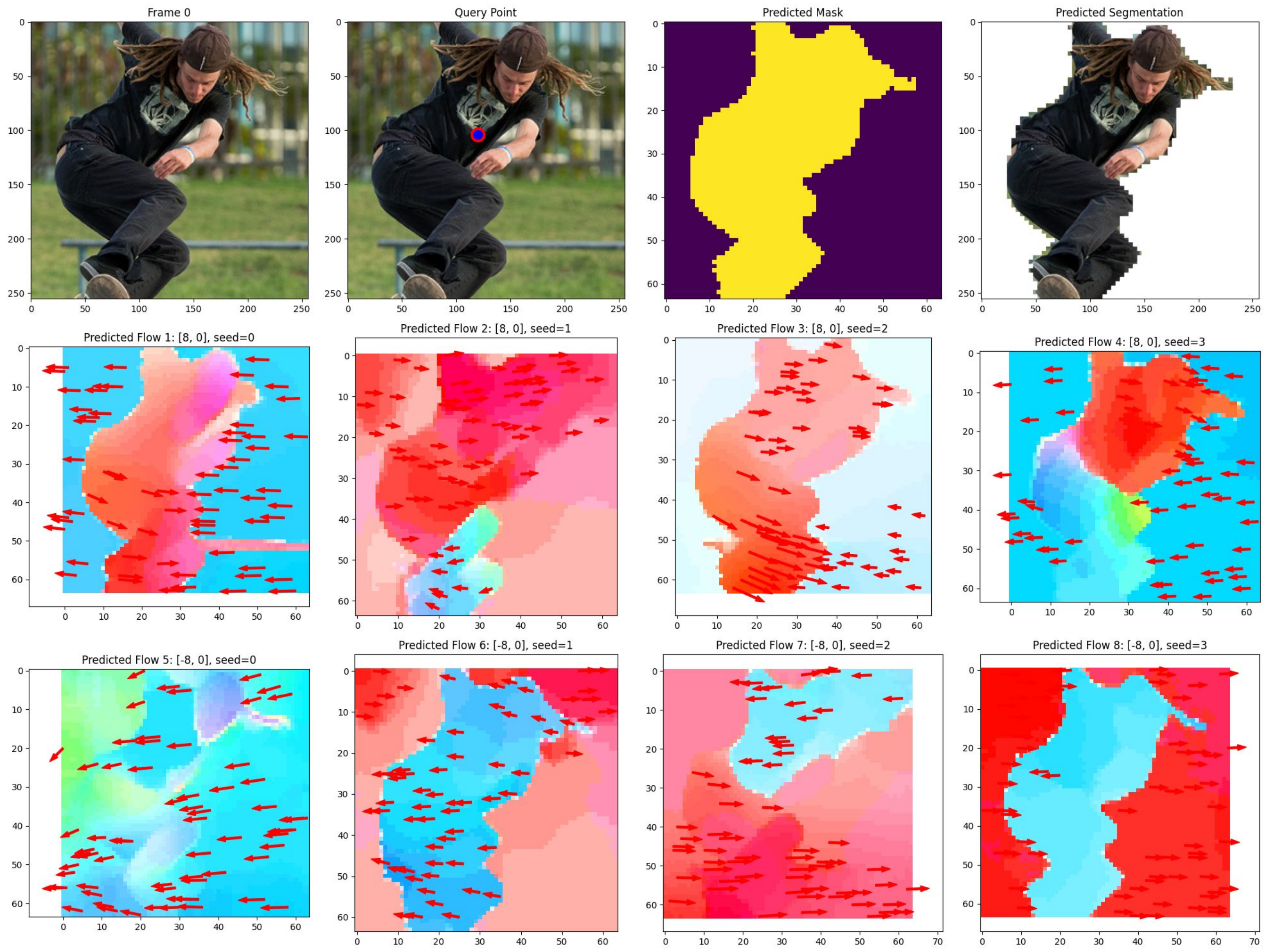
Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3

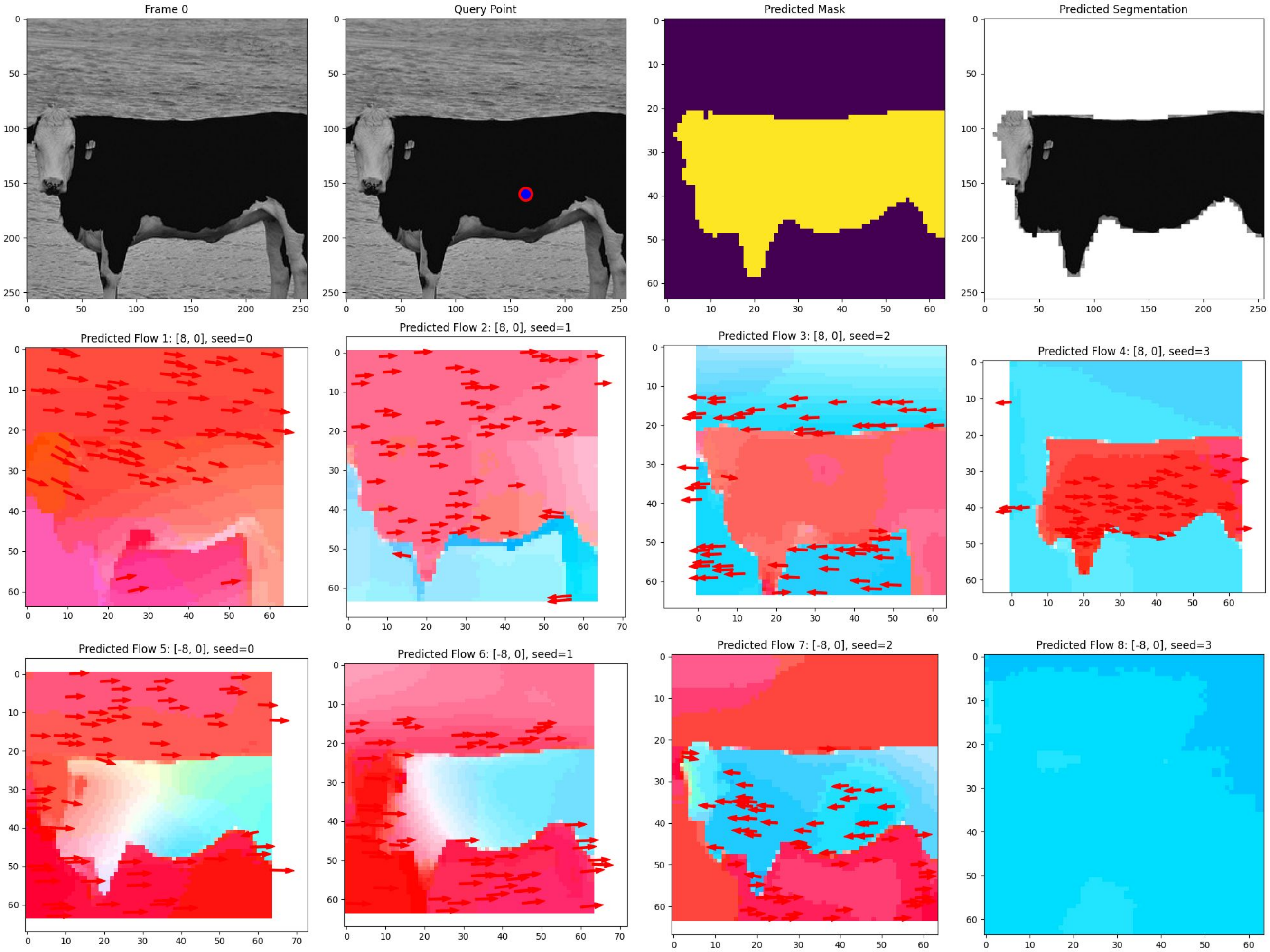


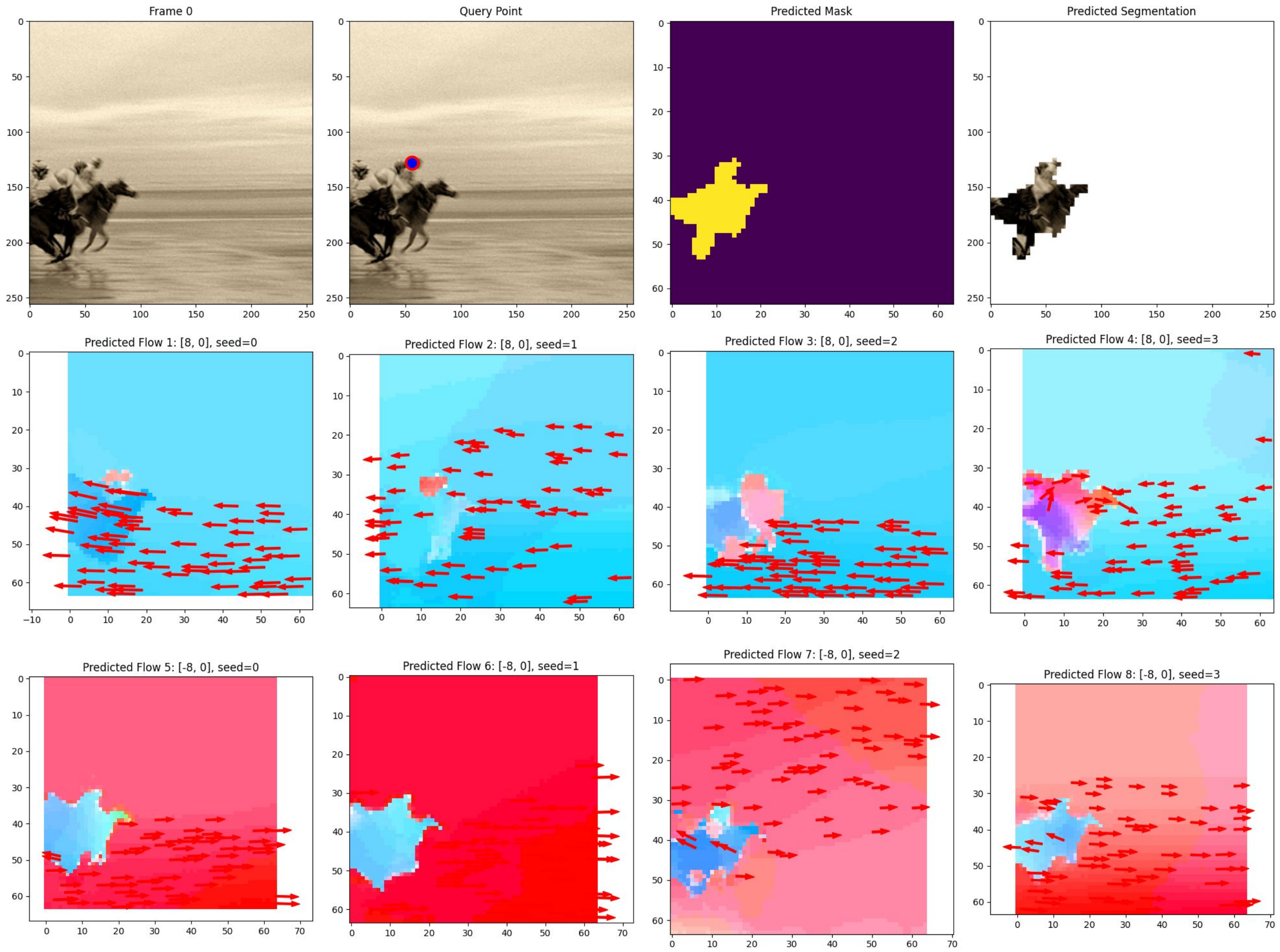
Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3

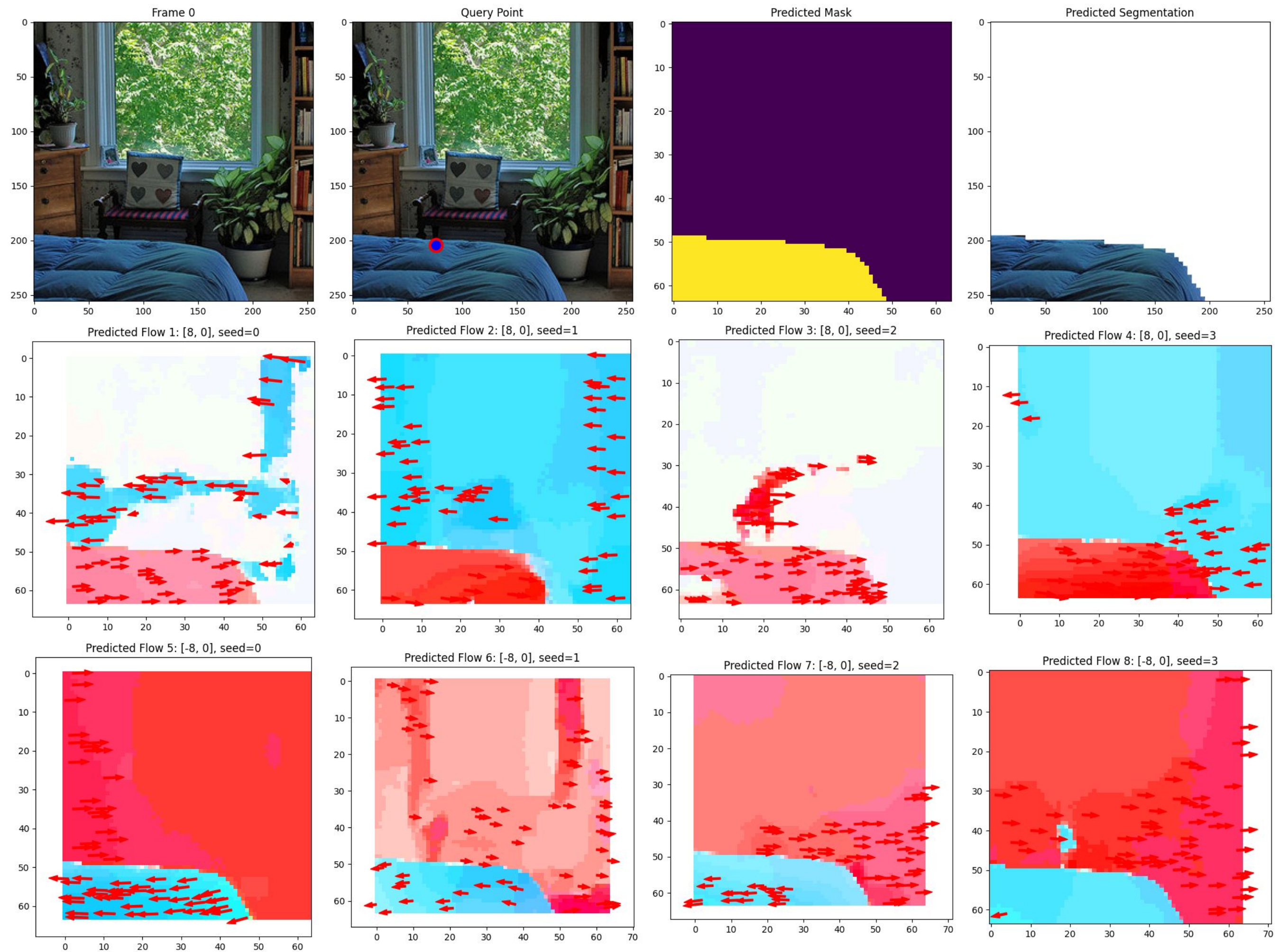


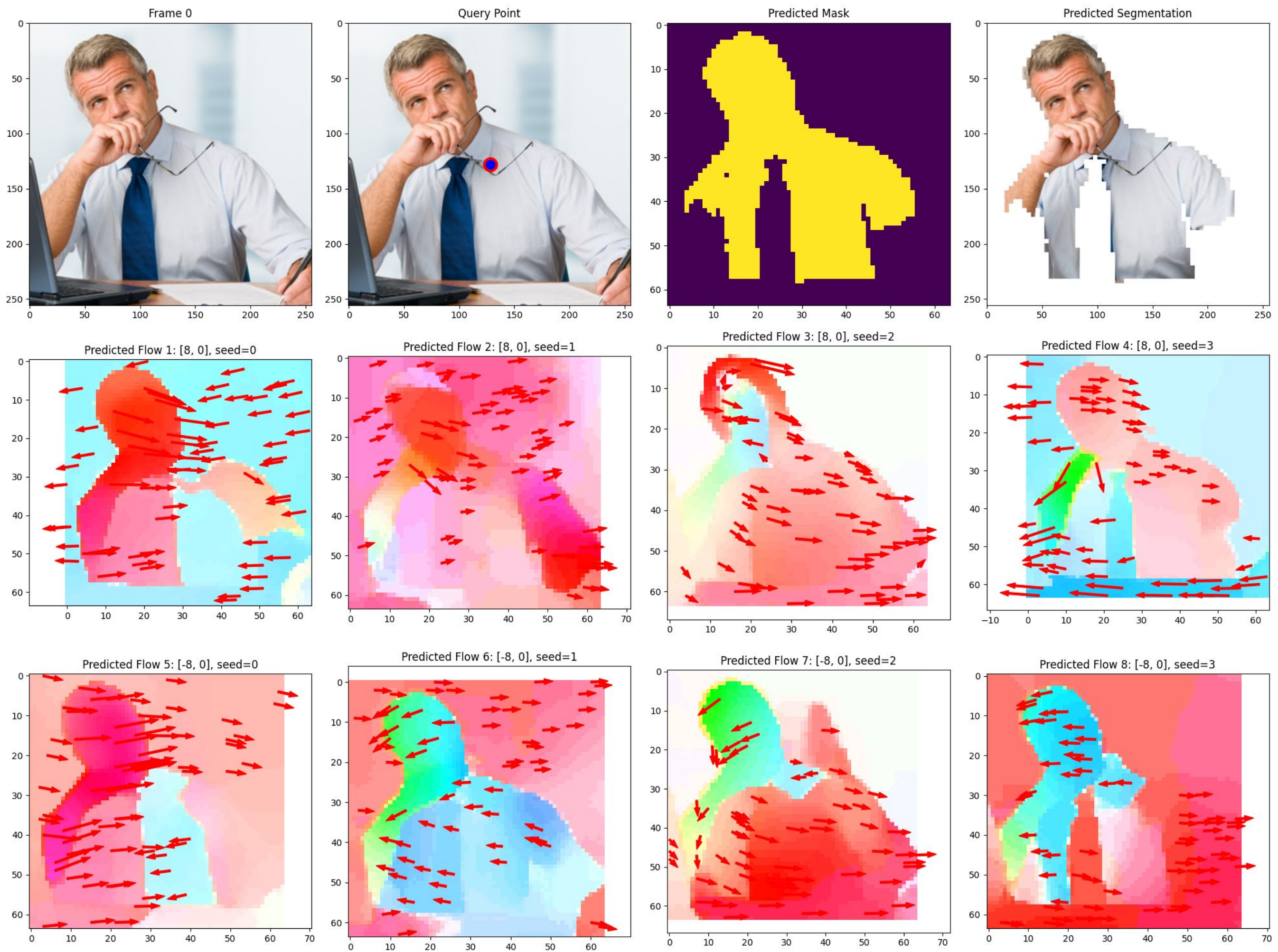
Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3


Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3

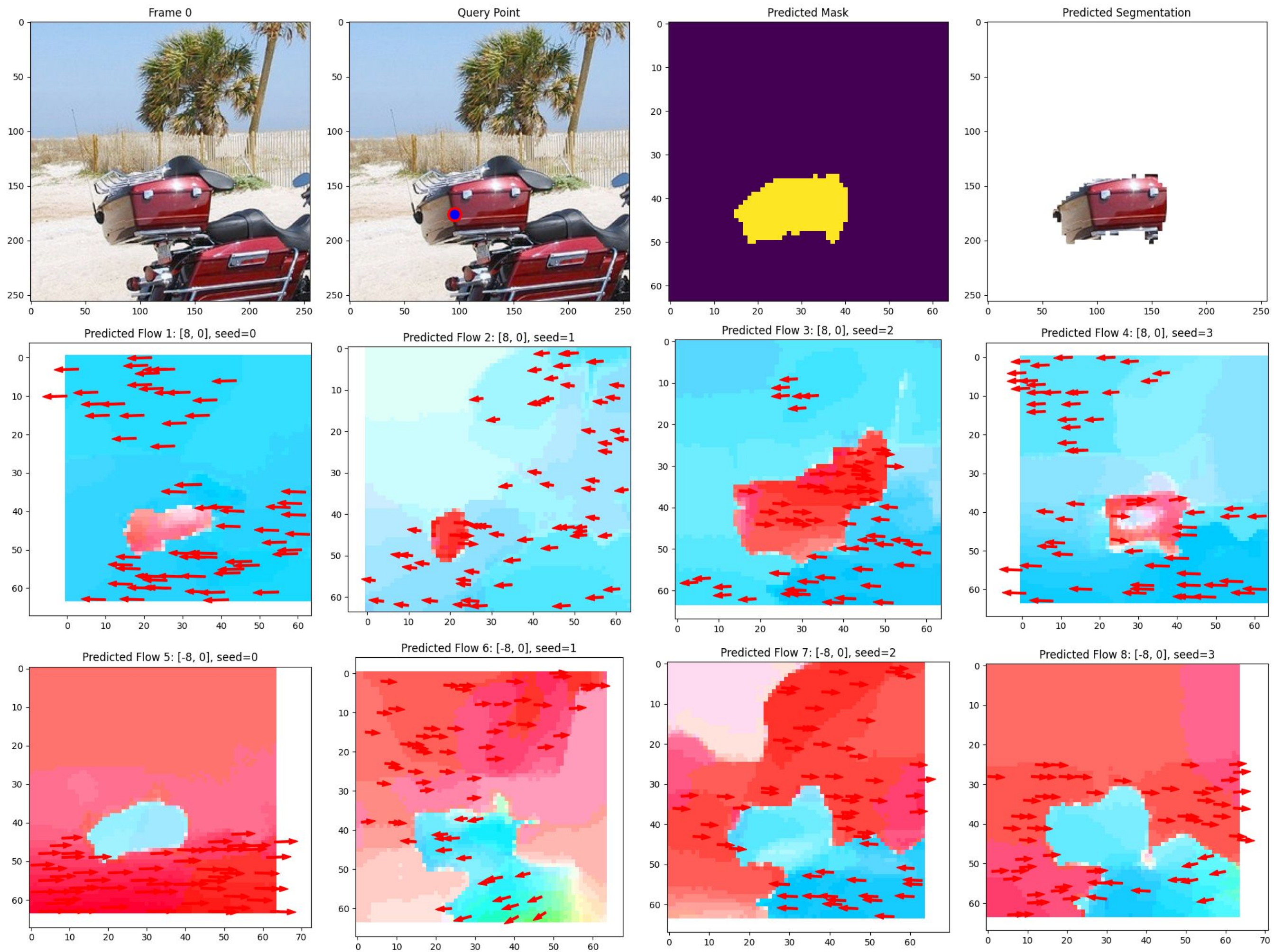

Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3


Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3

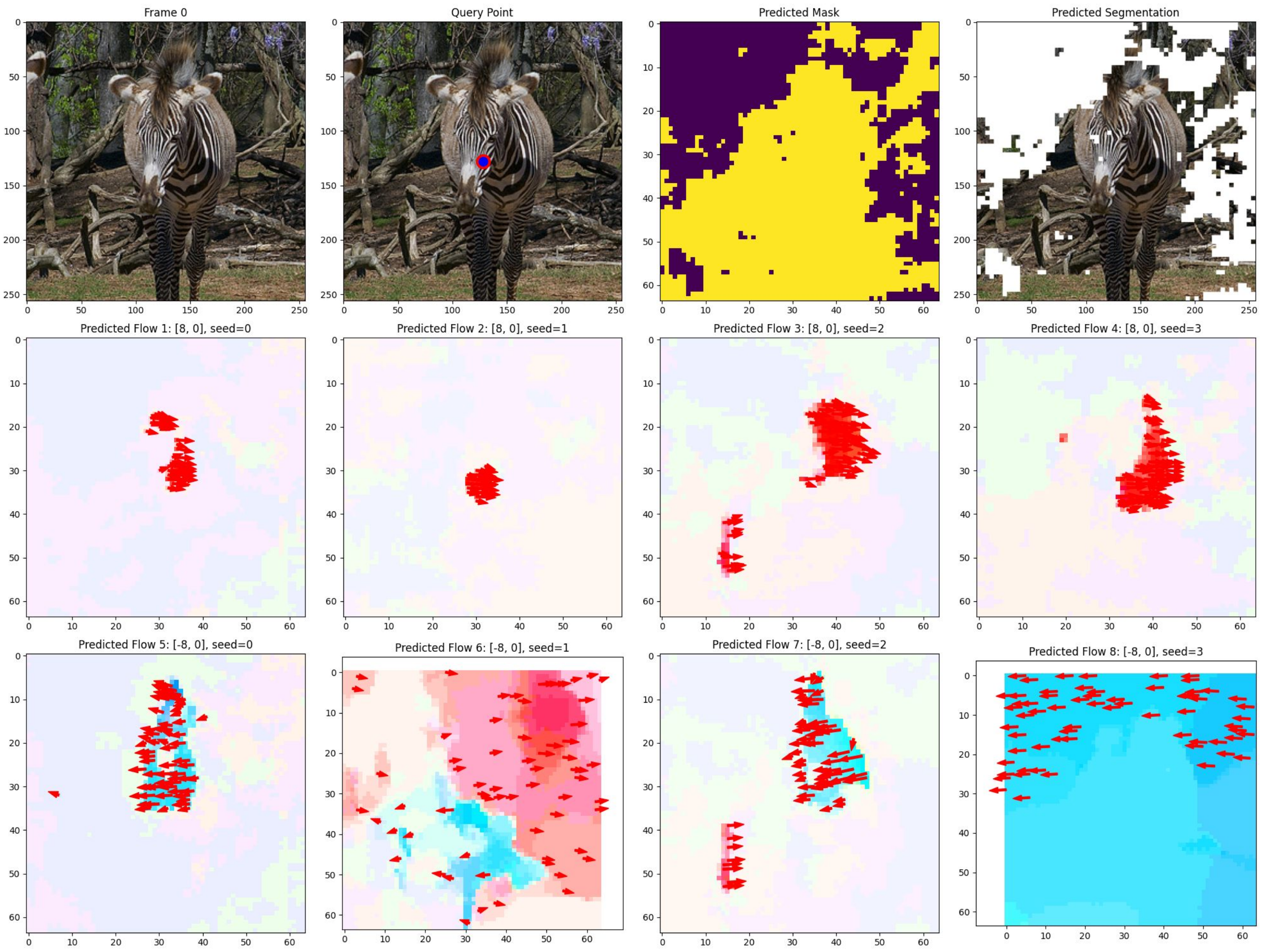

Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3



Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3



Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3


Frame 0
Query Point
Predicted Mask
Predicted Segmentation
Predicted Flow 1: [8, 0], seed=0
Predicted Flow 2: [8, 0], seed=1
Predicted Flow 3: [8, 0], seed=2
Predicted Flow 4: [8, 0], seed=3
Predicted Flow 5: [-8, 0], seed=0
Predicted Flow 6: [-8, 0], seed=1
Predicted Flow 7: [-8, 0], seed=2
Predicted Flow 8: [-8, 0], seed=3

Shortcomings

- Lots of variance, highly dependent on seed
  - Rollout order matters!
  - More sophisticated rollout algorithms may improve consistency

- Model predictions aren’t always very good
  - Samples were produced using a 100M parameter model
  - Currently training a 1B model and plan to train a 7B model

Thank you!

Many thanks to everyone working on CCWM:

- Klemen Kotar
- Honglin Chen
- Wanhee Lee
- Rahul Mysore Venkatesh
- Daniel L. K. Yamins

[1] Daniel M. Bear, Kevin Feigelis, Honglin Chen, Wanhee Lee, Rahul Venkatesh, Klemen Kotar, Alex Durango, and Daniel L. K. Yamins. Unifying (machine) vision via counterfactual world modeling, 2023.

[2] Wanhee Lee, Jared Watrous, Honglin Chen, Klemen Kotar, Tyler Bonnen, Daniel L. K. Yamins. A biologically plausible route to learn 3D perception. Cognitive Computational Neuroscience. Boston, Massachusetts, 2024.

[3] Honglin Chen, Rahul Venkatesh, Yoni Friedman, Jiajun Wu, Joshua B Tenenbaum, Daniel LK Yamins, and Daniel M Bear. Unsupervised segmentation in real-world images via spelke object inference. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX, pages 719–735. Springer, 2022.