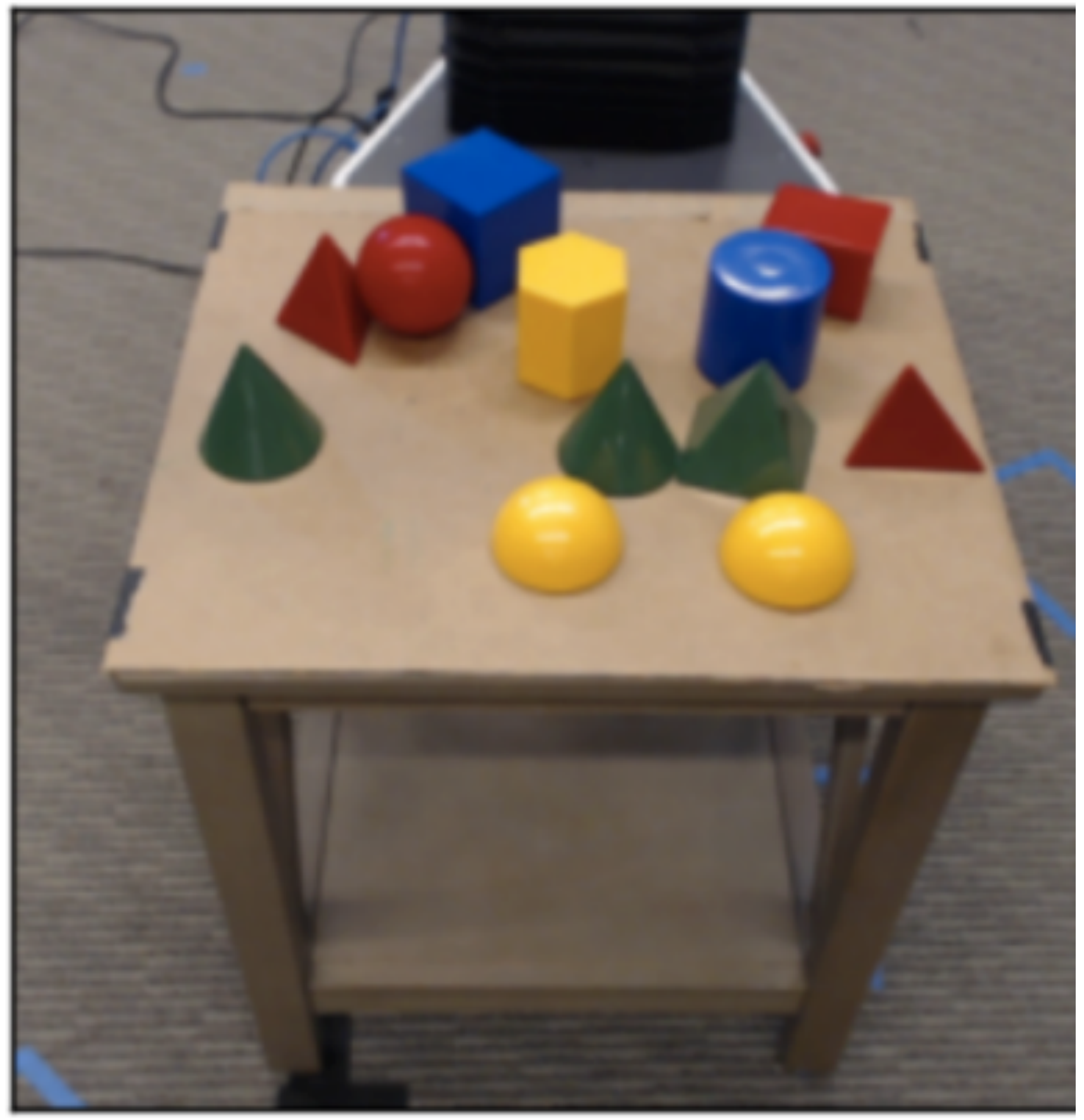
Geometry-Aware Neural Rendering

Josh Tobin, OpenAI Robotics, Pieter Abbeel

How to model complex robots scenes?

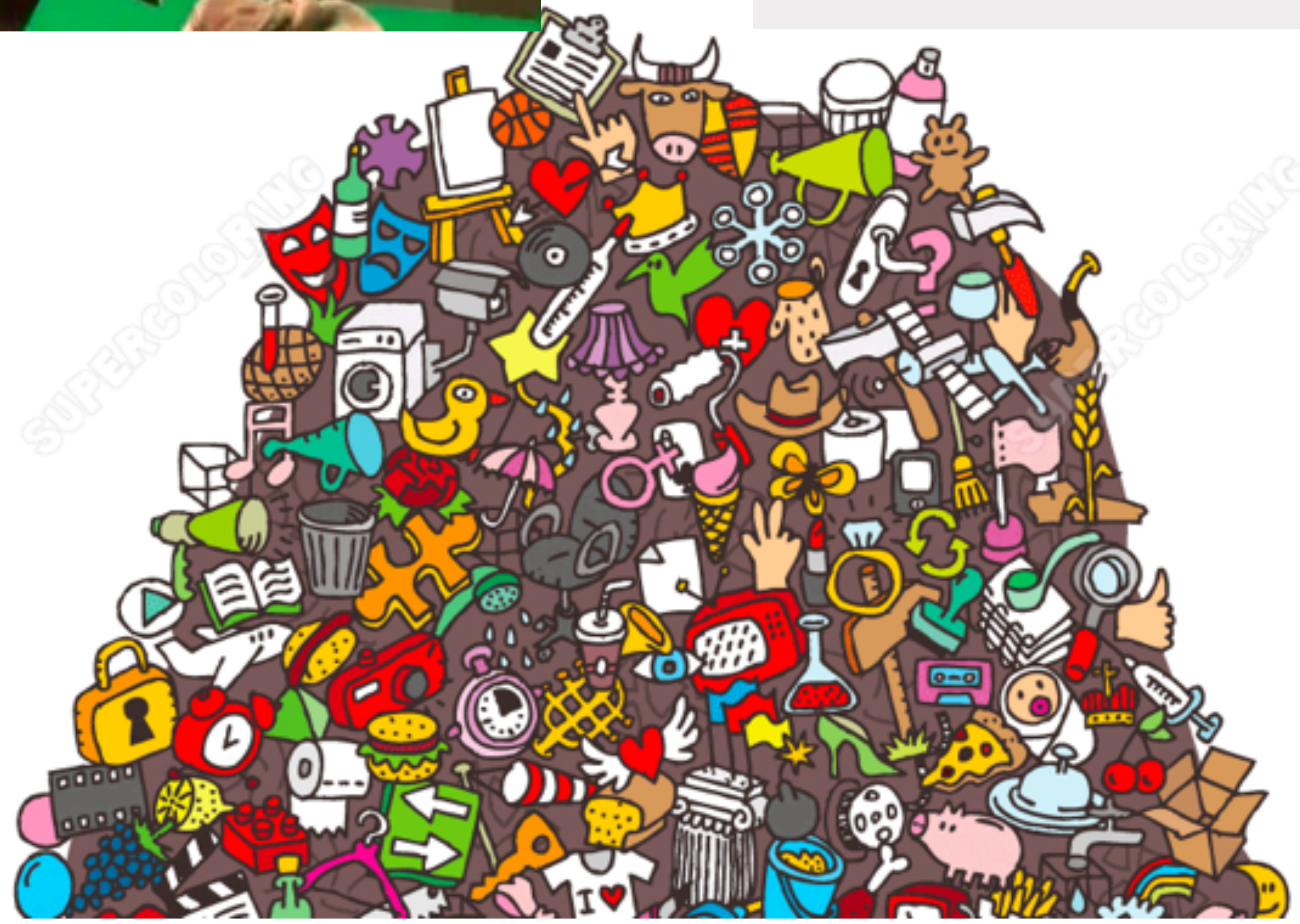
From...



...To

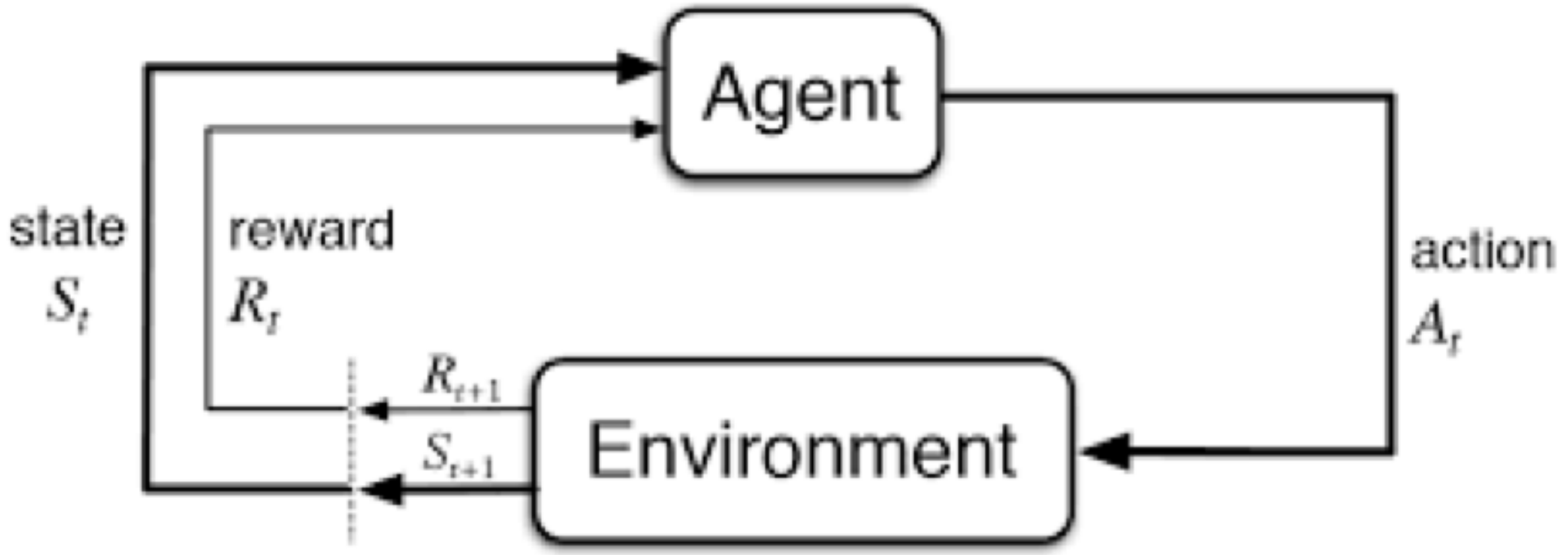
Model the state of all objects?

- Scales with scene complexity
- How to deal with complex internal state?
- How to deal with out-of-distribution?



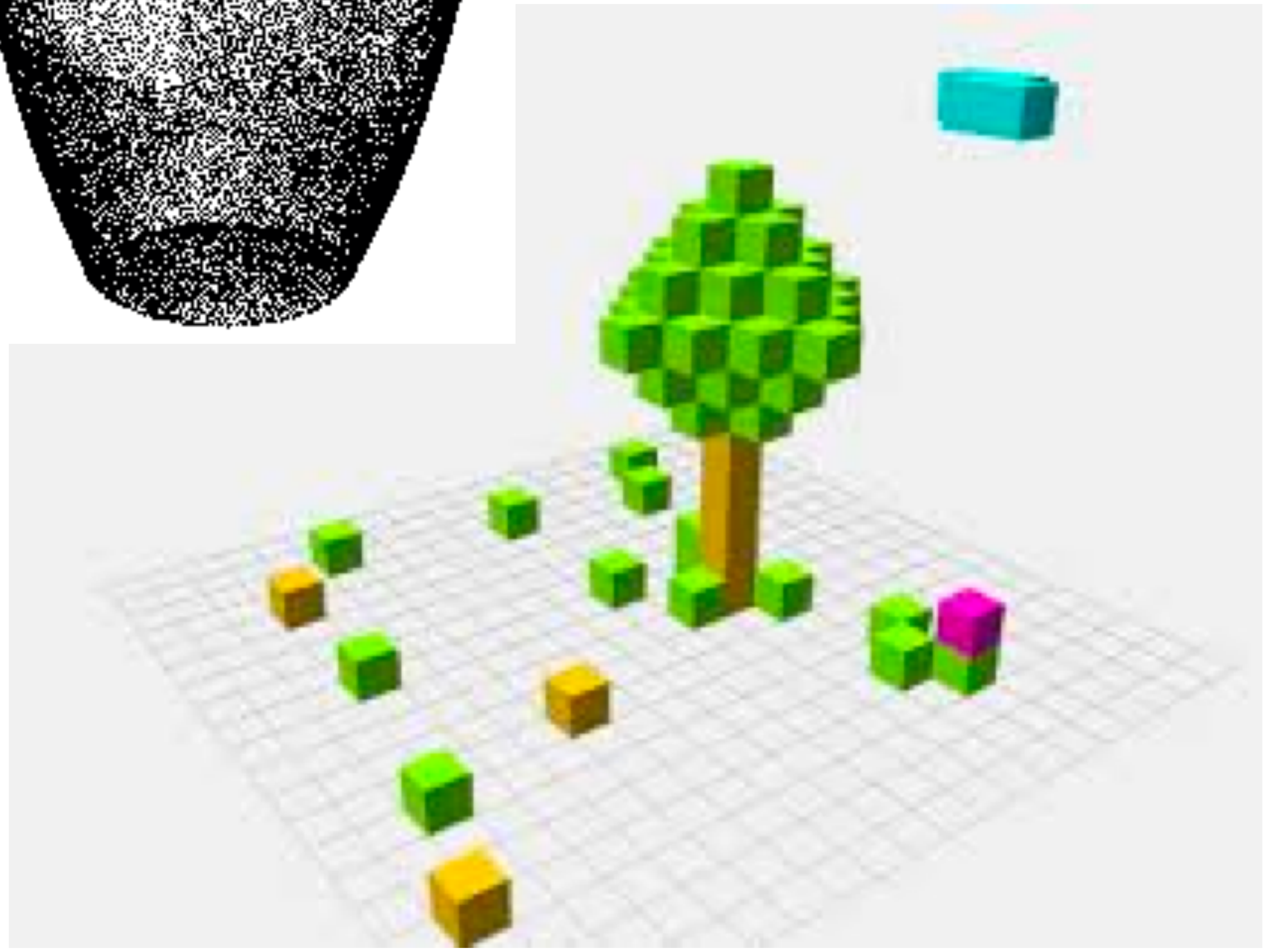
Only use the state implicitly?

- Can be data inefficient
- May require learning from scratch (which can be dangerous)
- Often lacks reusability



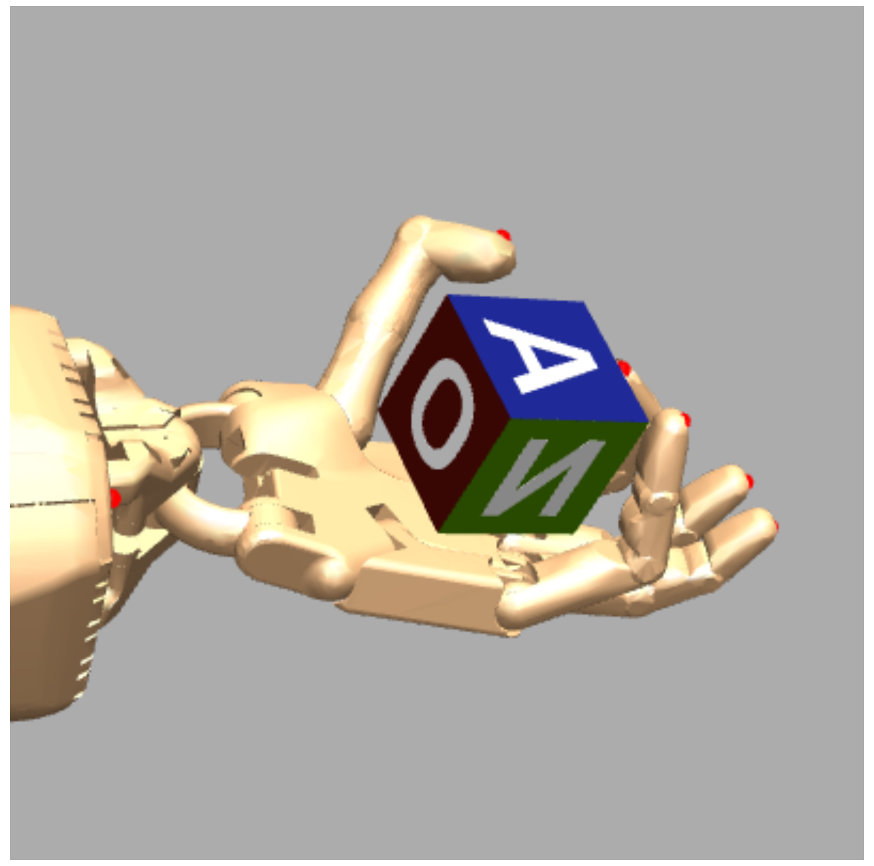
Model the 3D structure of the scene?

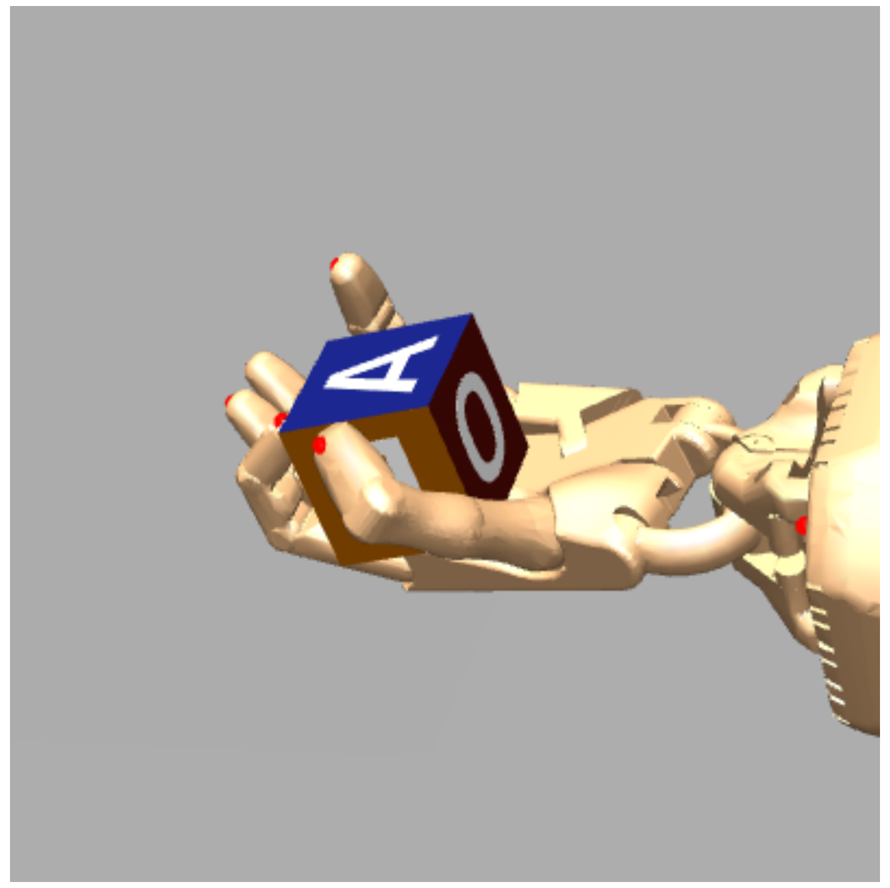
- High-dimensional representations - scale poorly with scene detail
- No notion of semantics



Model the 3D structure implicitly?

Random viewpoints



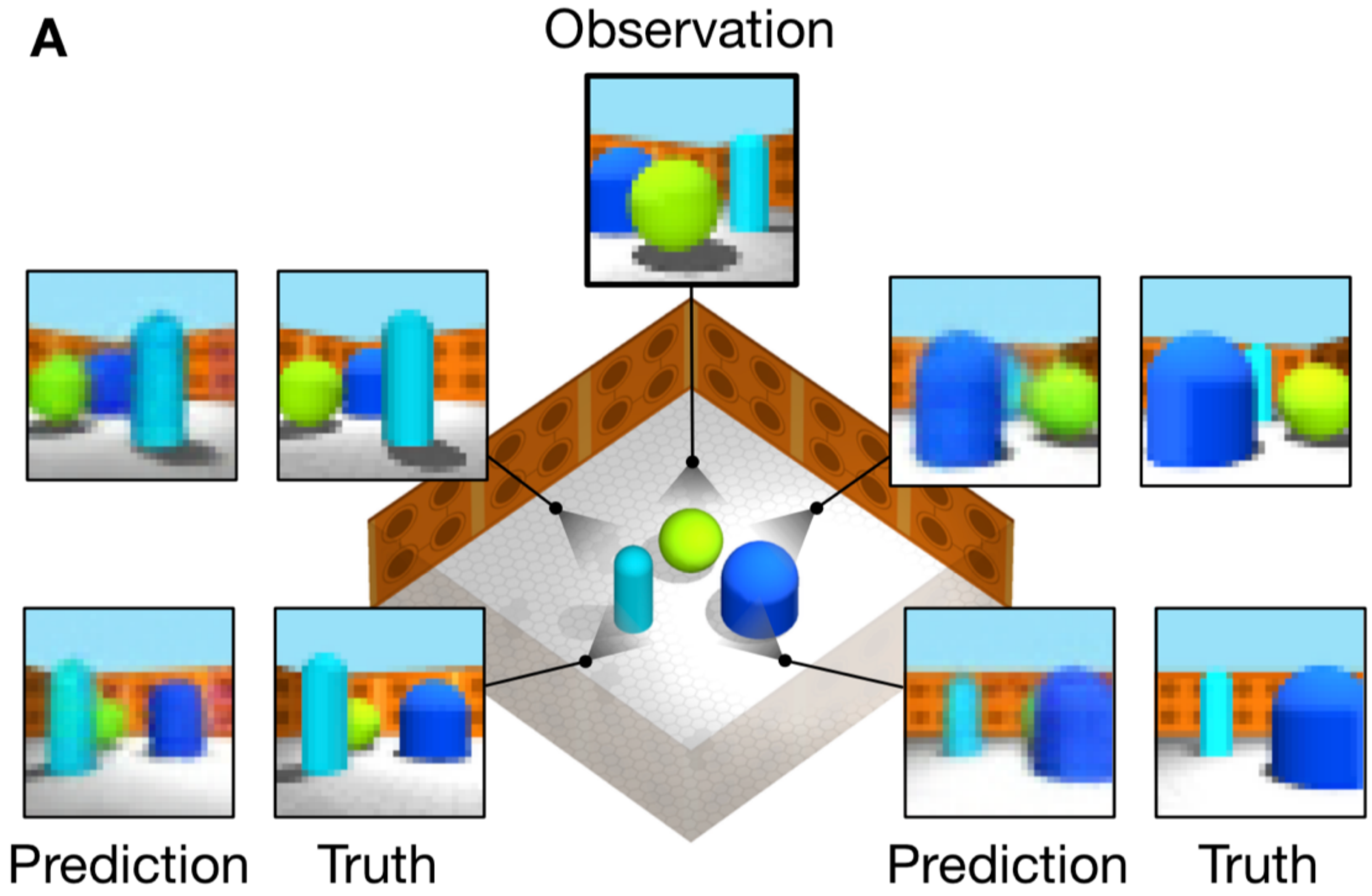


Model

Arbitrary “query” viewpoint

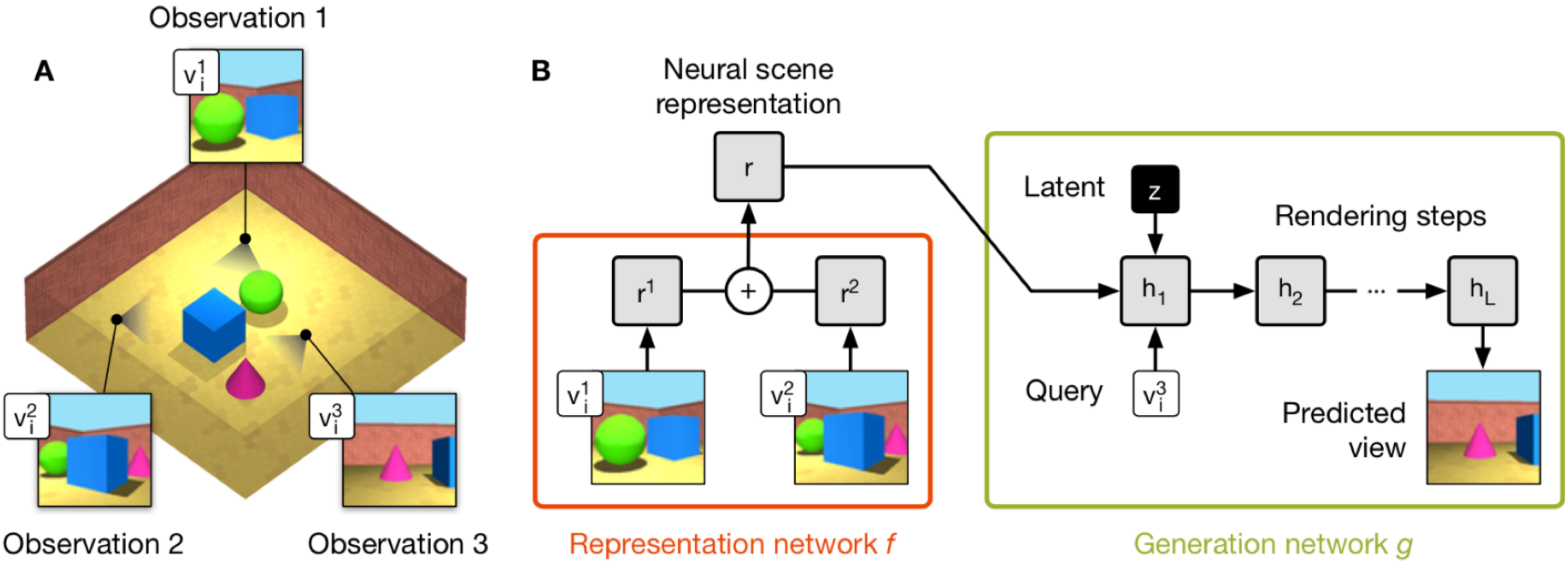
The *Neural Rendering* problem

Motivation: Generative Query Nets



Eslami, SM Ali, et al. "Neural scene representation and rendering." Science 360.6394 (2018): 1204-1210.

Motivation: Generative Query Nets



Eslami, SM Ali, et al. "Neural scene representation and rendering." Science 360.6394 (2018): 1204-1210.

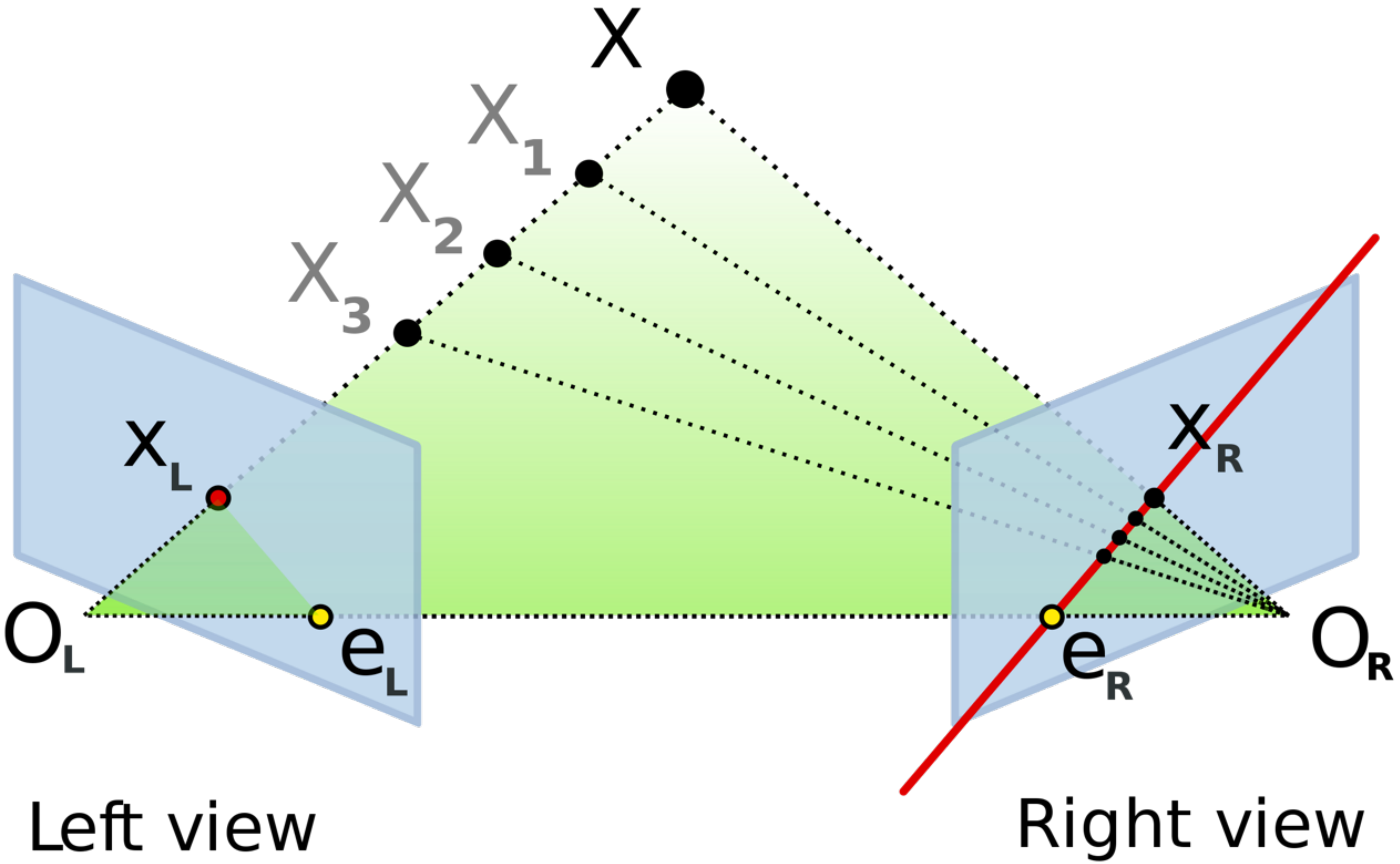
Key questions

- Can it scale to high(er)-dimensional images (GQN is 64x64)?
- Does it work for objects with complex state?
- Does it work for a wide range of realistic objects?
- Is it useful for downstream robotics tasks?

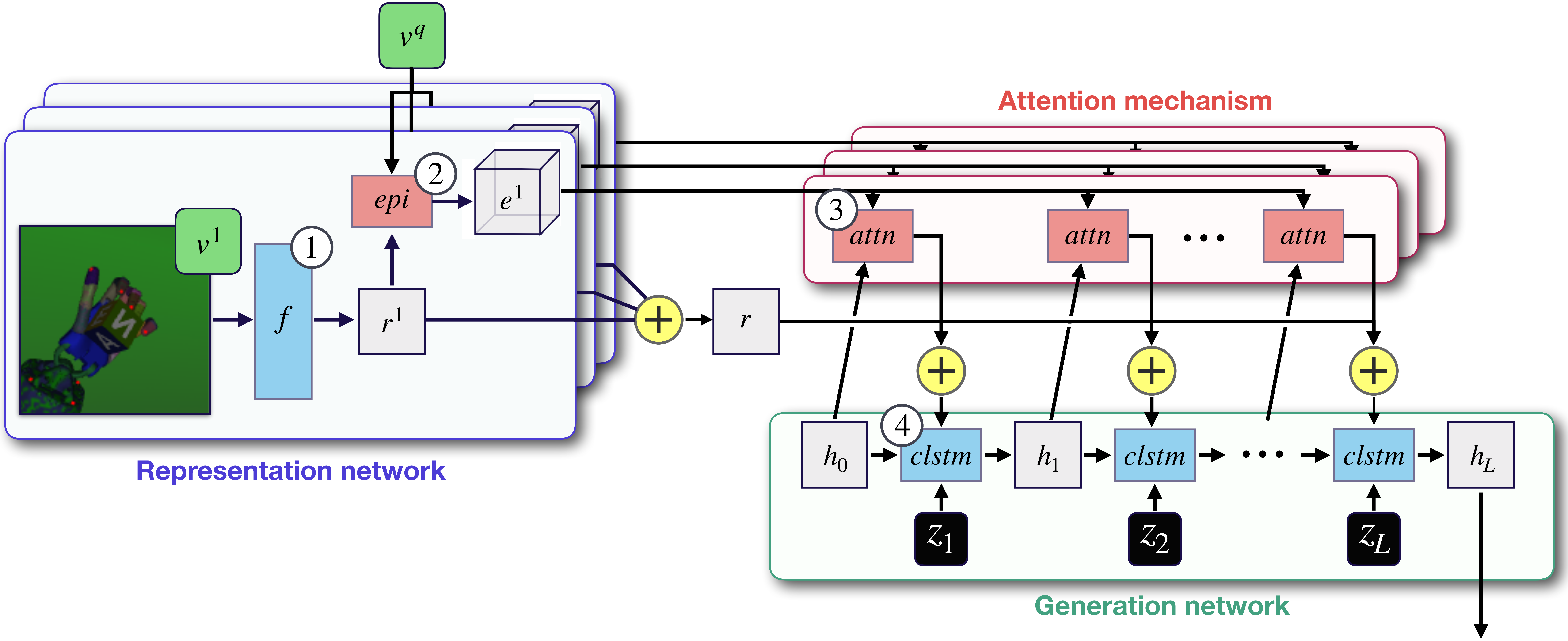
Potential limitations

- Scene representation is a sum — each feature contains only local information
- Rendering process cannot interact with the full representations (except through backprop)

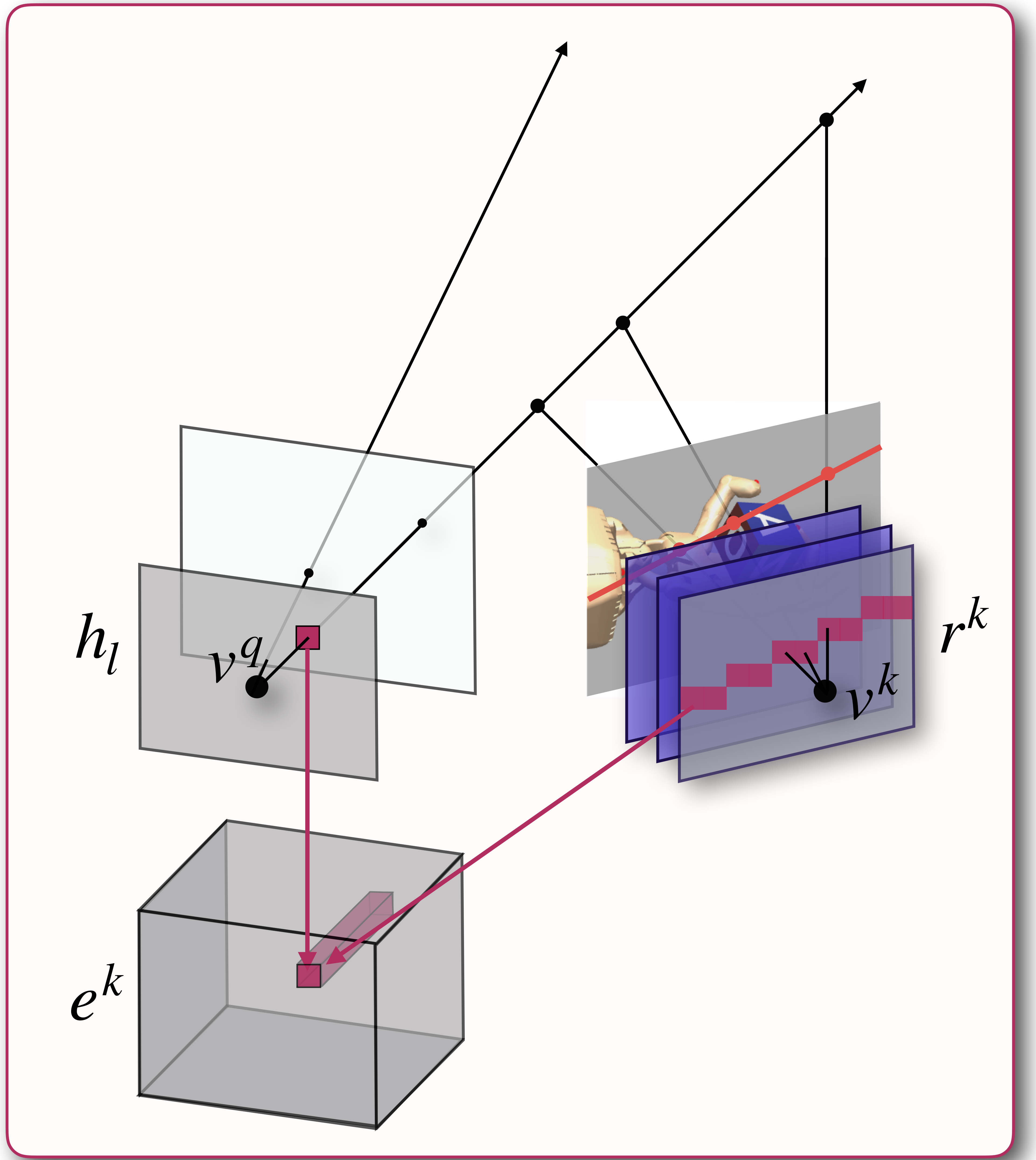
Background: Epipolar Geometry



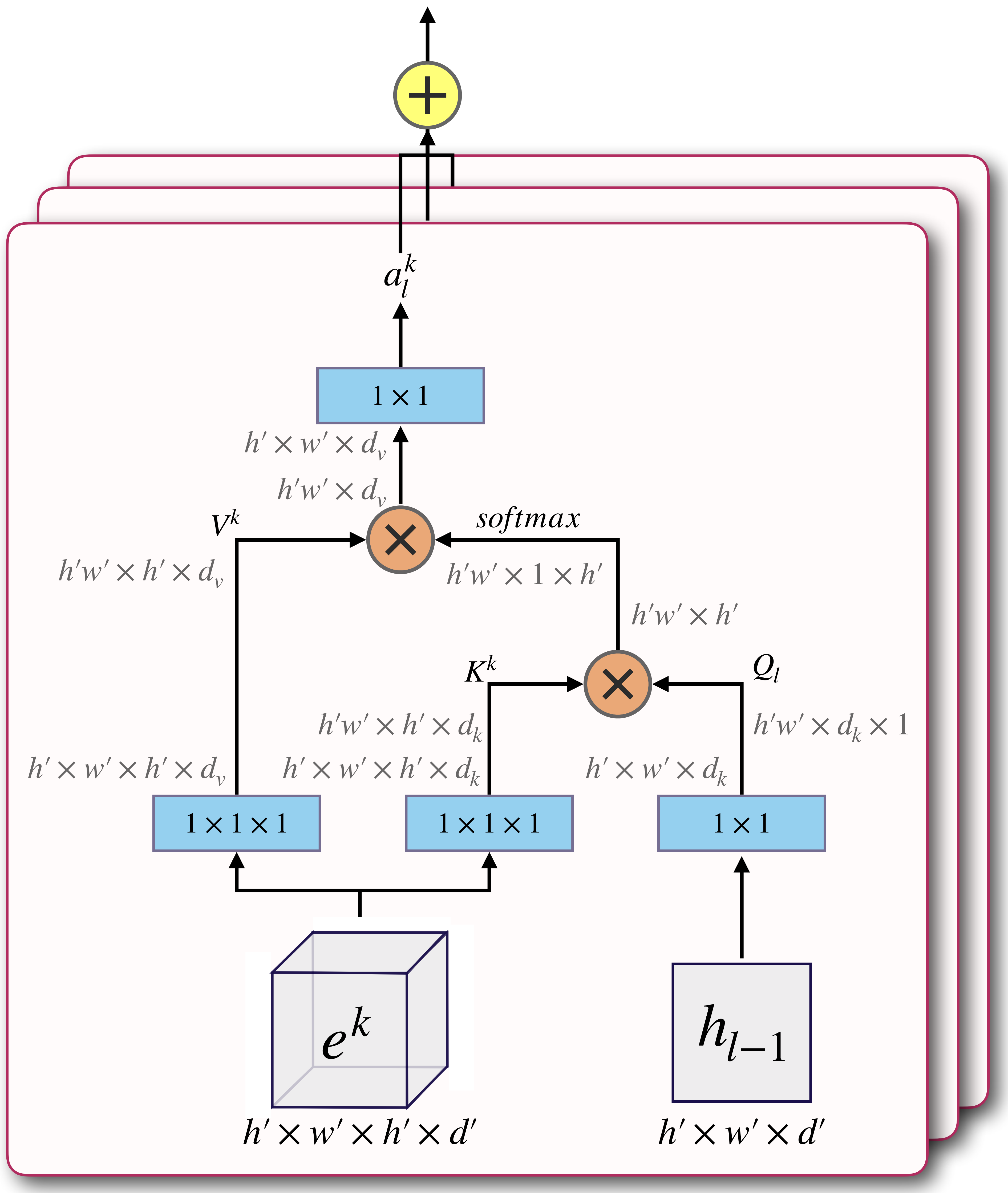
Our Approach



Epipolar extraction



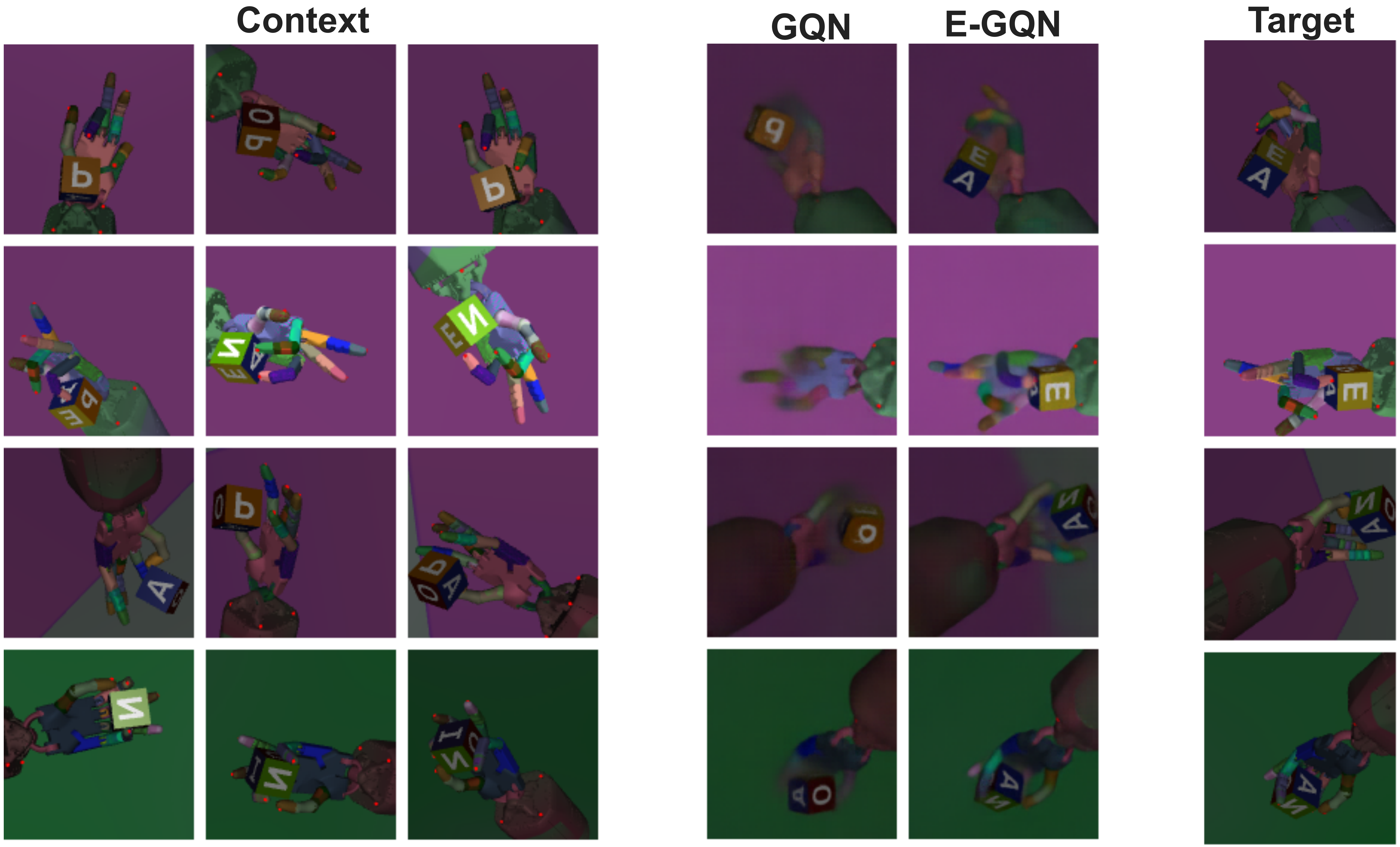
Attention mechanism



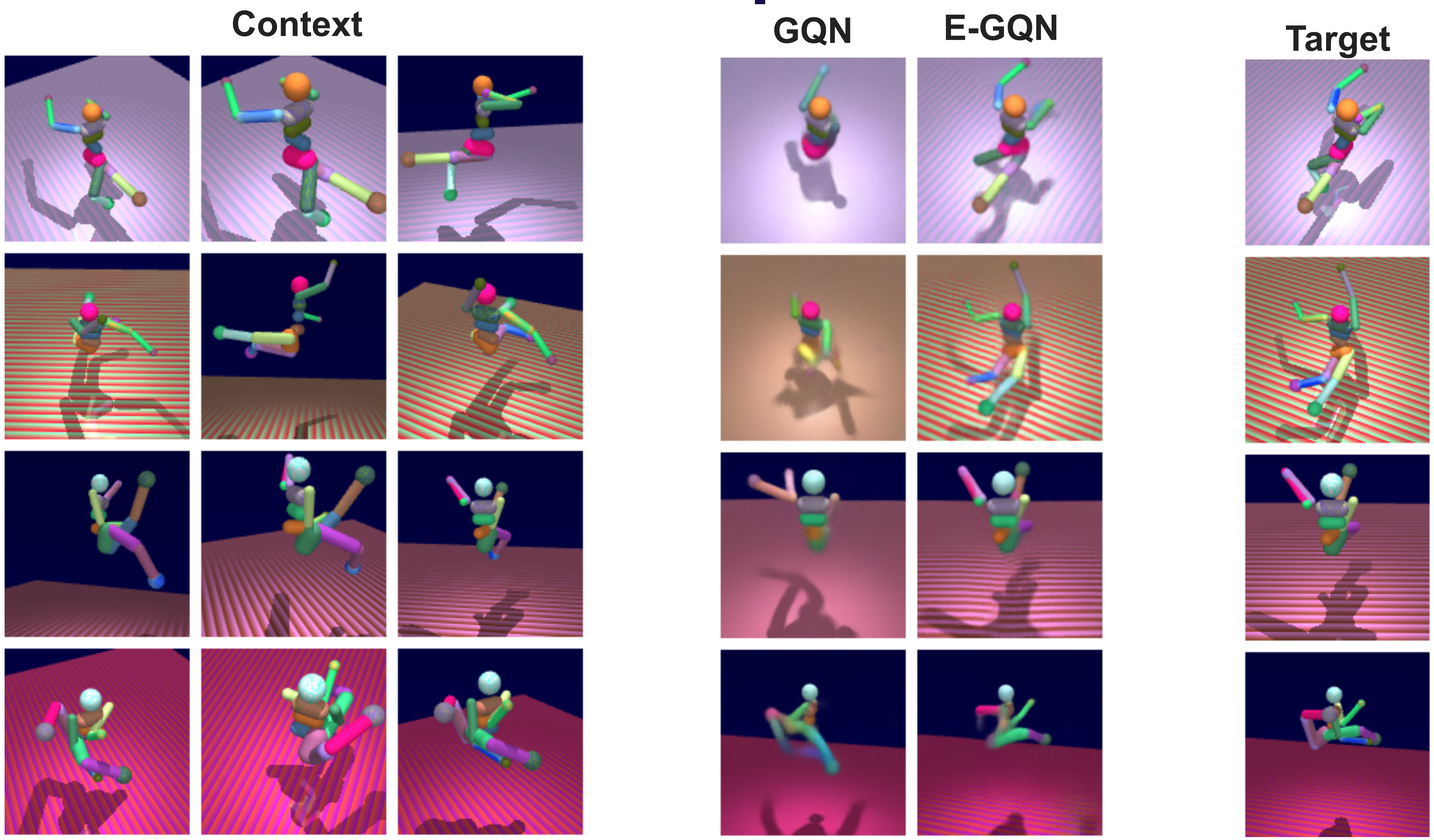
Does it help?

Dataset	Mean Absolute Error (pixels) GQN	Mean Absolute Error (pixels) E-GQN	Root Mean Squared Error (pixels) GQN	Root Mean Squared Error (pixels) E-GQN	ELBO (nats / dim) GQN	ELBO (nats / dim) E-GQN
rrc	7.40 ± 6.22	**3.59 ± 2.10**	14.62 ± 12.77	**6.80 ± 5.23**	0.5637 ± 0.0013	**0.5629 ± 0.0008**
rfc	12.44 ± 12.89	**12.05 ± 12.79**	**26.80 ± 21.35**	27.65 ± 20.72	**0.5637 ± 0.0011**	0.5639 ± 0.0012
jaco	4.30 ± 1.12	**4.00 ± 0.90**	8.58 ± 2.94	**7.43 ± 2.32**	0.5634 ± 0.0007	**0.5631 ± 0.0005**
sm7	3.13 ± 1.30	**2.14 ± 0.53**	9.97 ± 4.34	**5.63 ± 2.21**	0.5637 ± 0.0009	**0.5628 ± 0.0004**
oab	10.99 ± 5.13	**5.47 ± 2.54**	22.11 ± 8.00	**10.39 ± 4.55**	1.2587 ± 0.0018	**1.2569 ± 0.0011**
disco	18.86 ± 7.16	**12.46 ± 9.27**	32.72 ± 6.32	**22.04 ± 11.08**	1.2635 ± 0.0055	**1.2574 ± 0.0007**
rro	10.12 ± 5.15	**6.59 ± 3.23**	19.63 ± 9.14	**12.08 ± 6.52**	1.2573 ± 0.0011	**1.2566 ± 0.0009**

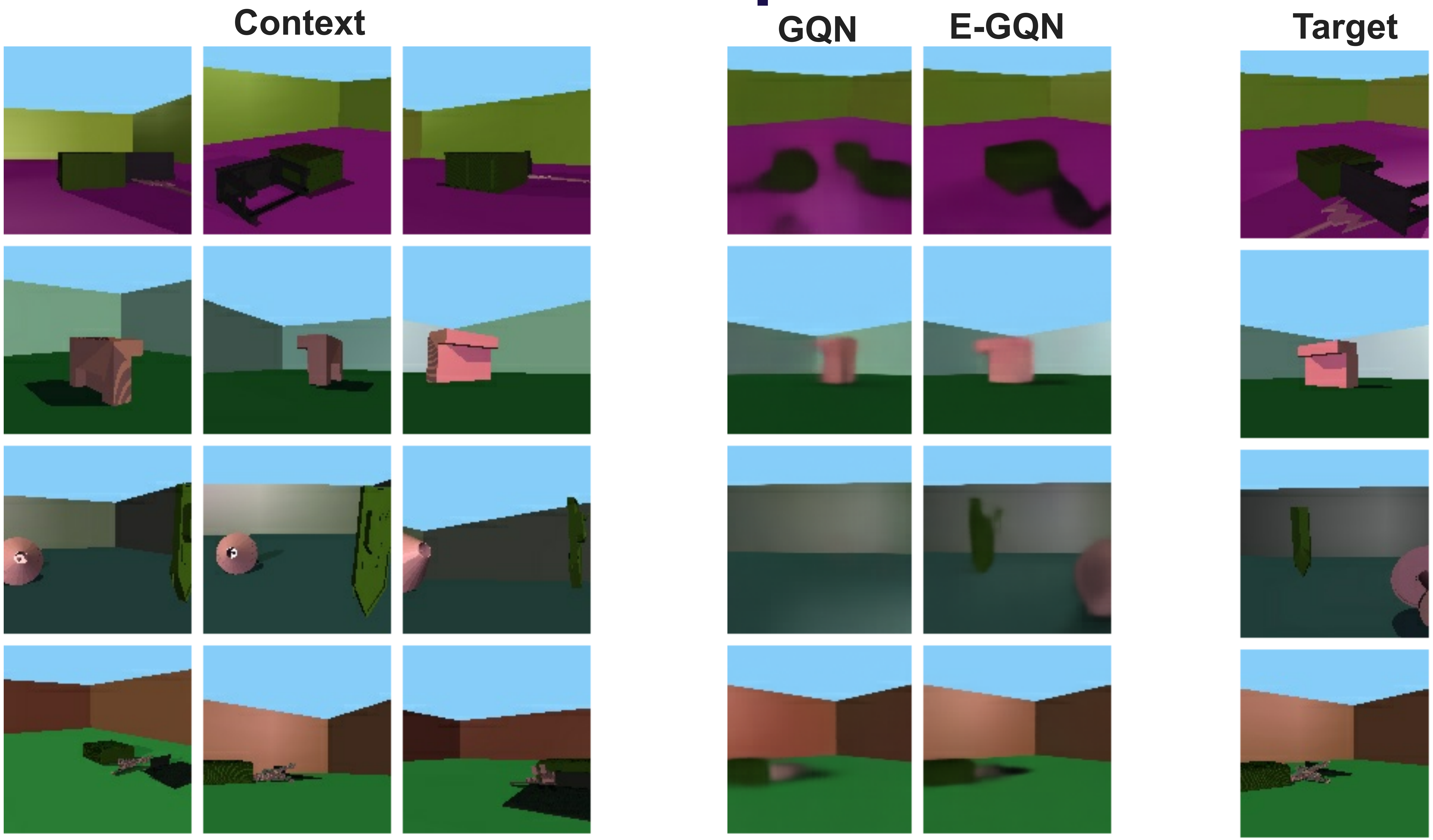
Examples



Examples



Examples



Conclusion

- Geometrically-inspired neural network primitives improve implicit 3D understanding
- Forcing the model to understand geometry can improve downstream robotic tasks
- How to go from this to general 3D understanding?