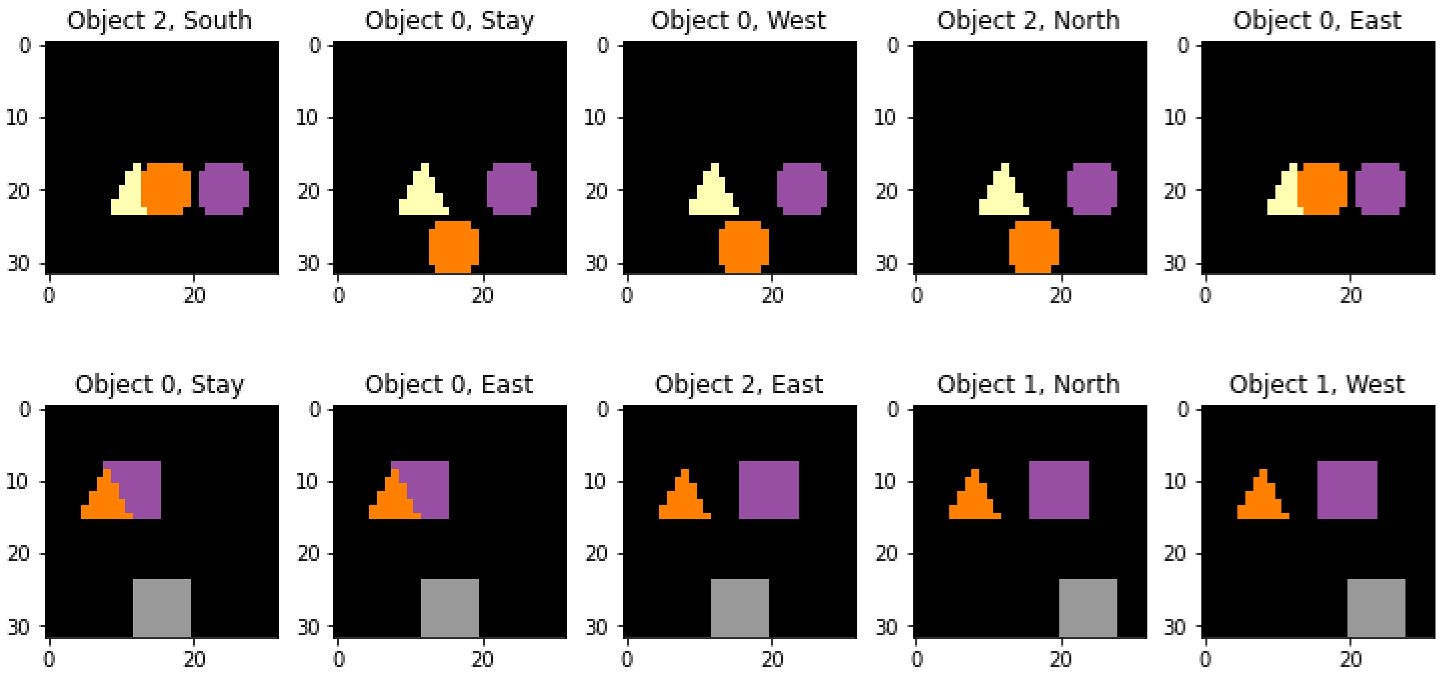
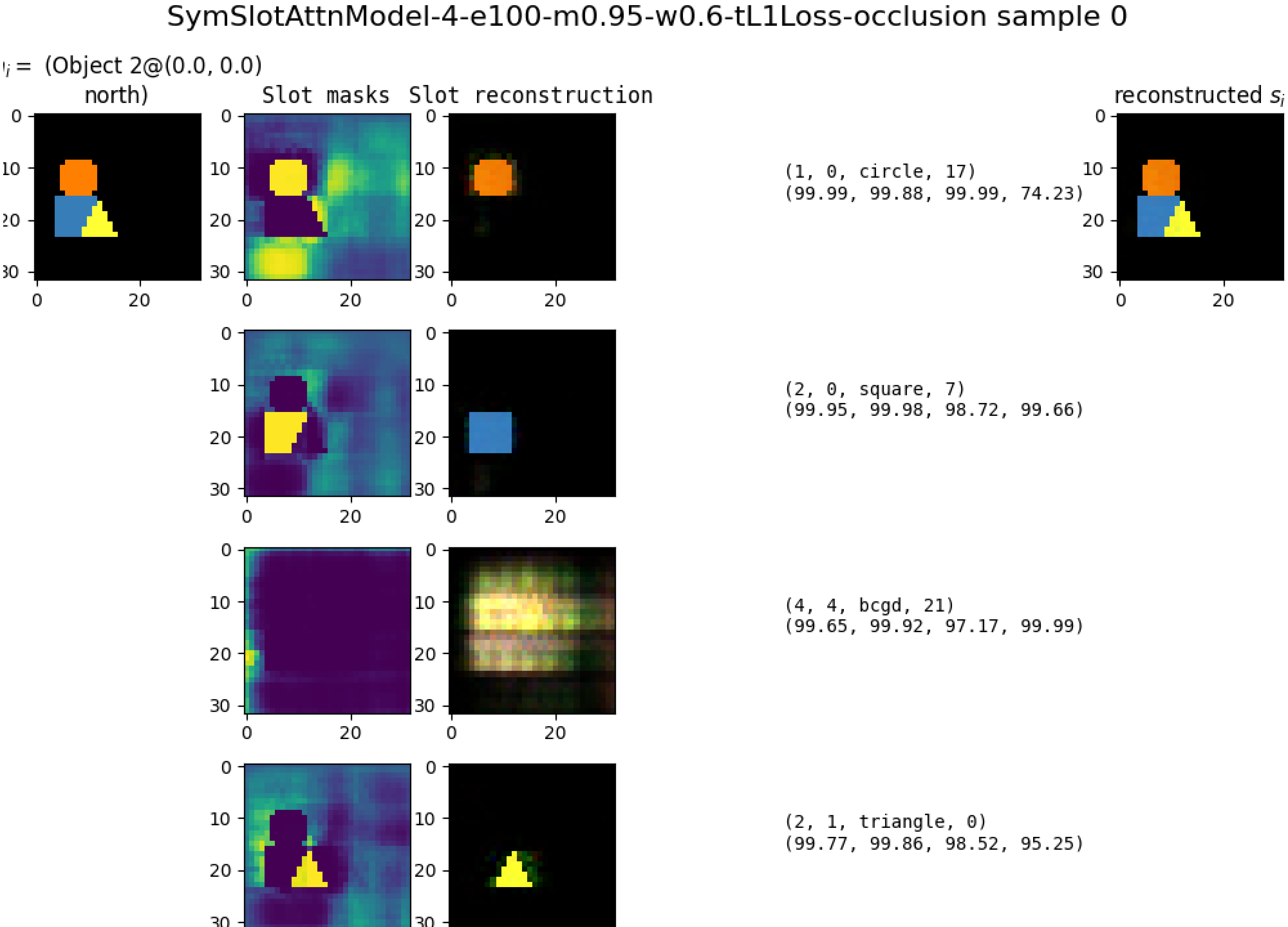
Unsupervised Neural Representation Learning

Agenda:

- Recap
- How does slot attention handle occlusion:
    - Dataset
    - Results
- Do symbolic rules "help" the vision system.
- Next Steps

Slot Attention - Occlusion Dataset

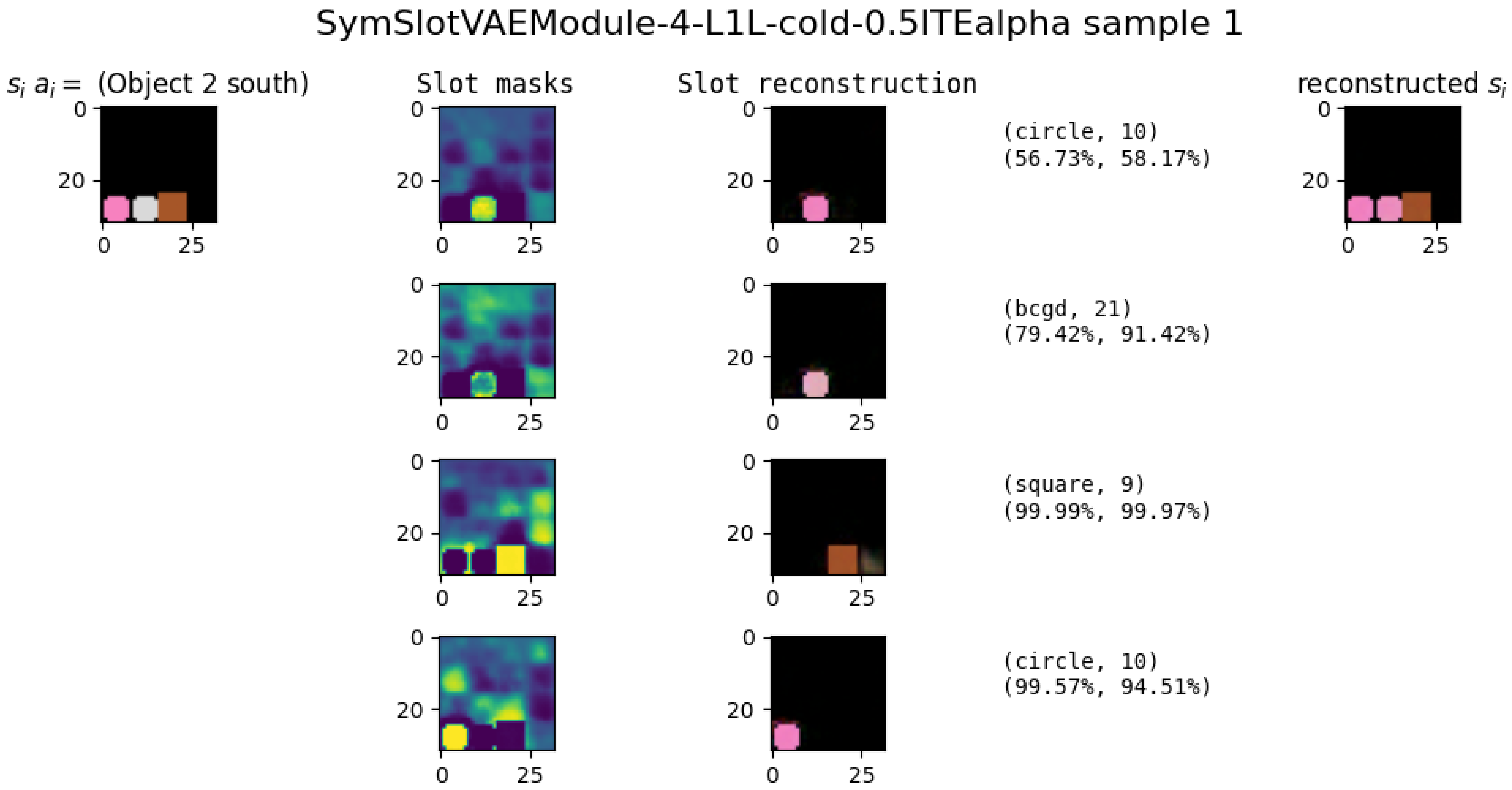



SymSlotAttnModel-4-e100-m0.95-w0.6-tL1Loss-occlusion sample 0
s_i = (Object 2@(0.0, 0.0) north)
Slot masks
Slot reconstruction
reconstructed s_i
(1, 0, circle, 17)
(99.99, 99.88, 99.99, 74.23)
(2, 0, square, 7)
(99.95, 99.98, 98.72, 99.66)
(4, 4, bcgd, 21)
(99.65, 99.92, 97.17, 99.99)
(2, 1, triangle, 0)
(99.77, 99.86, 98.52, 95.25)

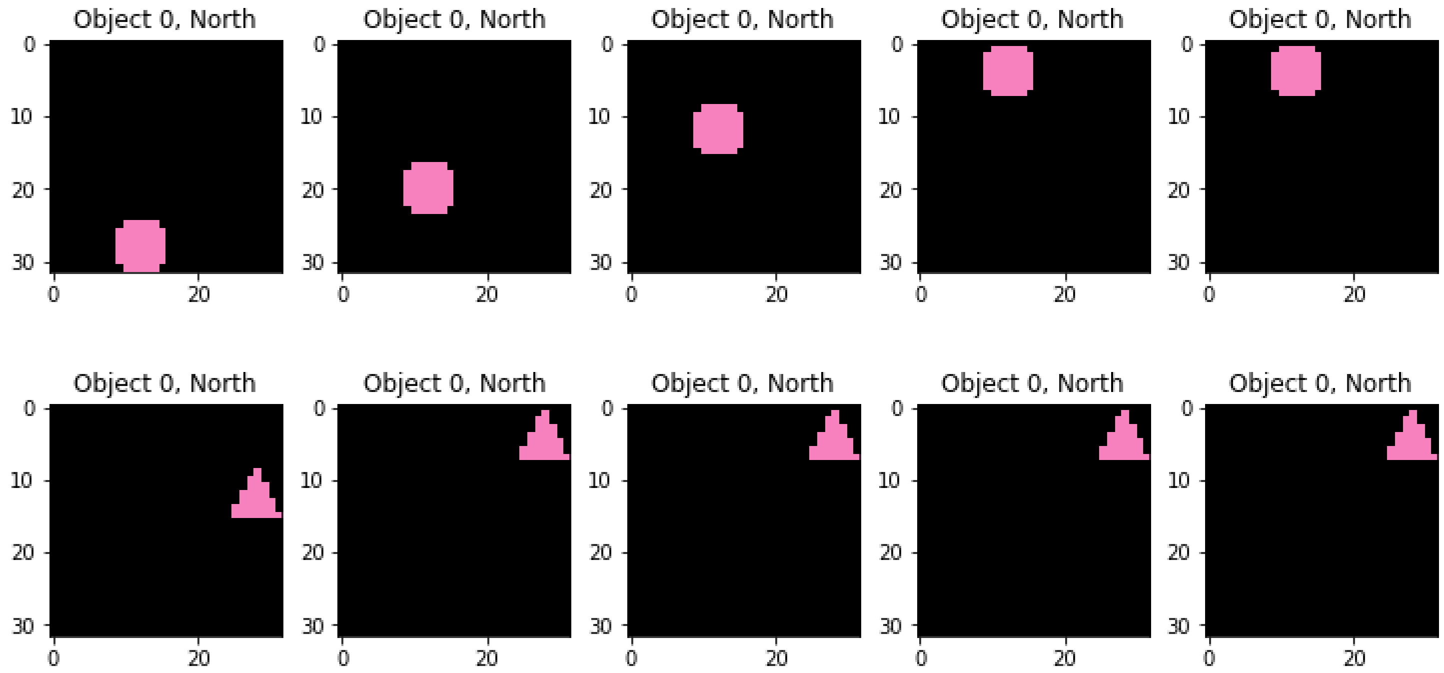
Fixed Symbolic Rules

- Pure DNN based approaches struggle to disassociate shapes with colors.
  - The (shape, color) pairs in training is different from (shape, color) pairs in testing.
  - Sample 1 on next slide shows an effect of this disassociation on a purely neural model (with no rules).
- Hypothesis: Given a fixed transition function, can a hybrid neurosymbolic model produce better reconstructions and converge faster than a purely symbolic model.
  - Three models:
    - A vanilla resnet autoencoder reconstructiong $(s_i, a_i) \rightarrow (s_{i+1})$
    - A slot attention based autoencoder (with an MLP for the transition layer) reconstructiong $(s_i, a_i) \rightarrow (s_{i+1})$
    - A slot attention based autoencoder (with a fixed symbolic program for the transition layer) reconstructiong $(s_i, a_i) \rightarrow (s_{i+1})$

Sample 1


SymSlotVAEModule-4-L1L-cold-0.5ITEalpha sample 1
s_i a_i = (Object 2 south)
Slot masks
Slot reconstruction
reconstructed s_i
(circle, 10)
(56.73%, 58.17%)
(bcgd, 21)
(79.42%, 91.42%)
(square, 9)
(99.99%, 99.97%)
(circle, 10)
(99.57%, 94.51%)

Dataset

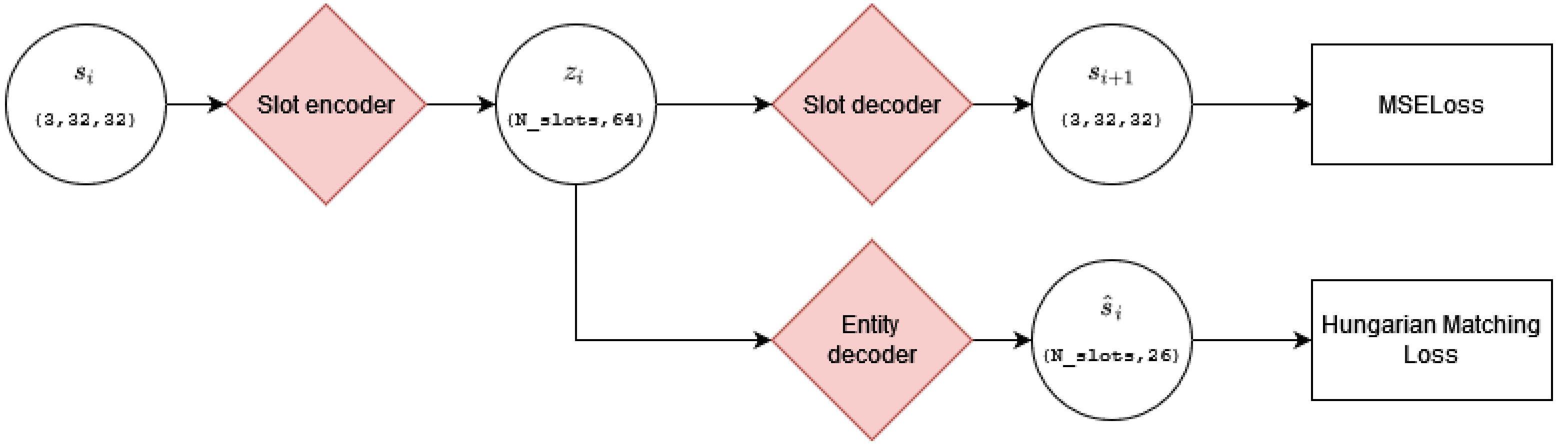

Object 0, North
Object 0, North
Object 0, North
Object 0, North
Object 0, North
Object 0, North
Object 0, North
Object 0, North
Object 0, North
Object 0, North
0
10
20
30
0
20

Next steps

- Slot Attention w/ Occlusion on mice domain.
  - MABE Videos.
- Results for the symbolic rule hypothesis.

Slides from Last Week

Slot Attn Decoder



Changes

[Submitted on 15 Jun 2019 (v1), last revised 24 Apr 2020 (this version, v6)]

Deep Set Prediction Networks

Yan Zhang, Jonathon Hare, Adam Prügel-Bennett

Current approaches for predicting sets from feature vectors ignore the unordered nature of sets and suffer from discontinuity issues as a result. We propose a general model for predicting sets that properly respects the structure of sets and avoids this problem. With a single feature vector as input, we show that our model is able to auto-encode point sets, predict the set of bounding boxes of objects in an image, and predict the set of attributes of these objects.

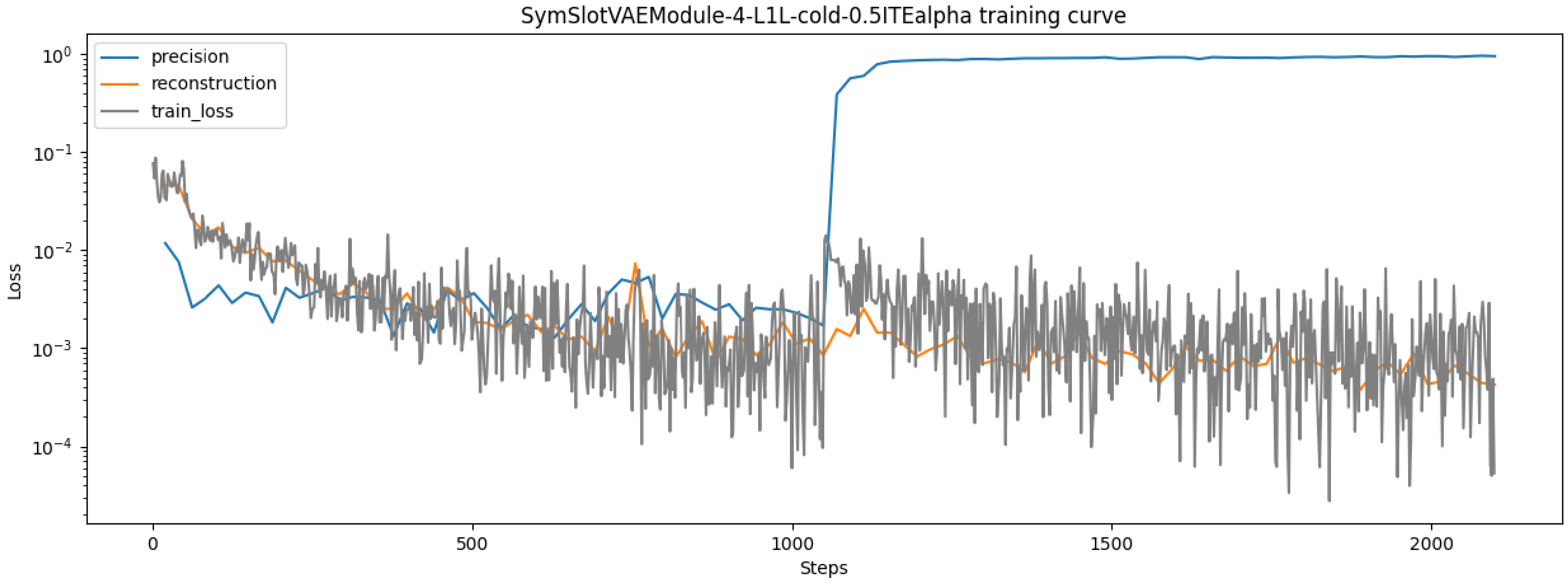
Comments:	Appendix C contains an erratum
Subjects:	**Machine Learning (cs.LG)**; Machine Learning (stat.ML)
Journal reference:	Advances in Neural Information Processing Systems 32 (NeurIPS 2019)
Cite as:	arXiv:1906.06565 **[cs.LG]**
	(or arXiv:1906.06565v6 **[cs.LG]** for this version)

Changes:

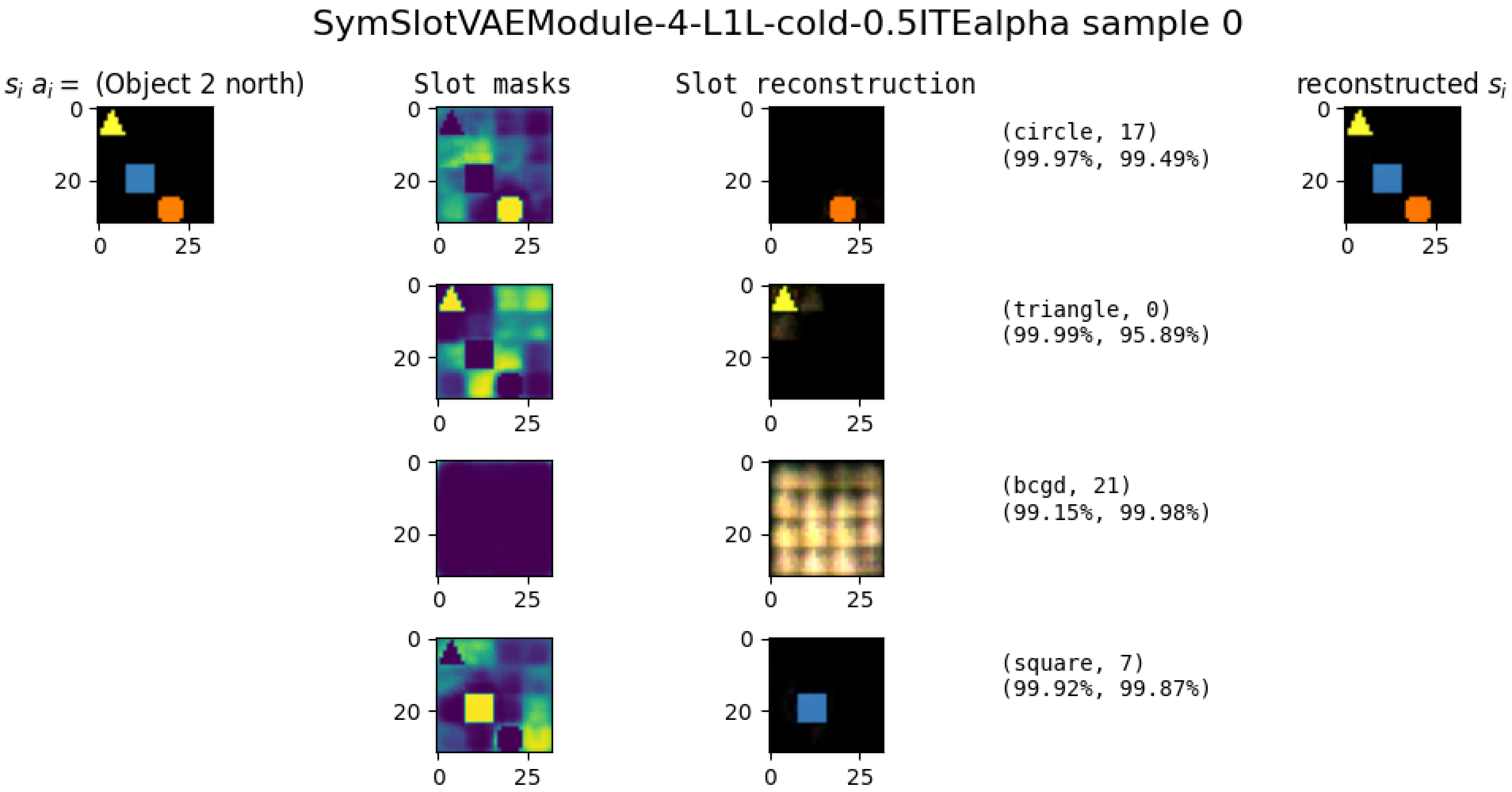
- Use smoothL1 loss instead of L1 loss.
- Use the same loss function for matching and backprop.

Slot Attn Decoder

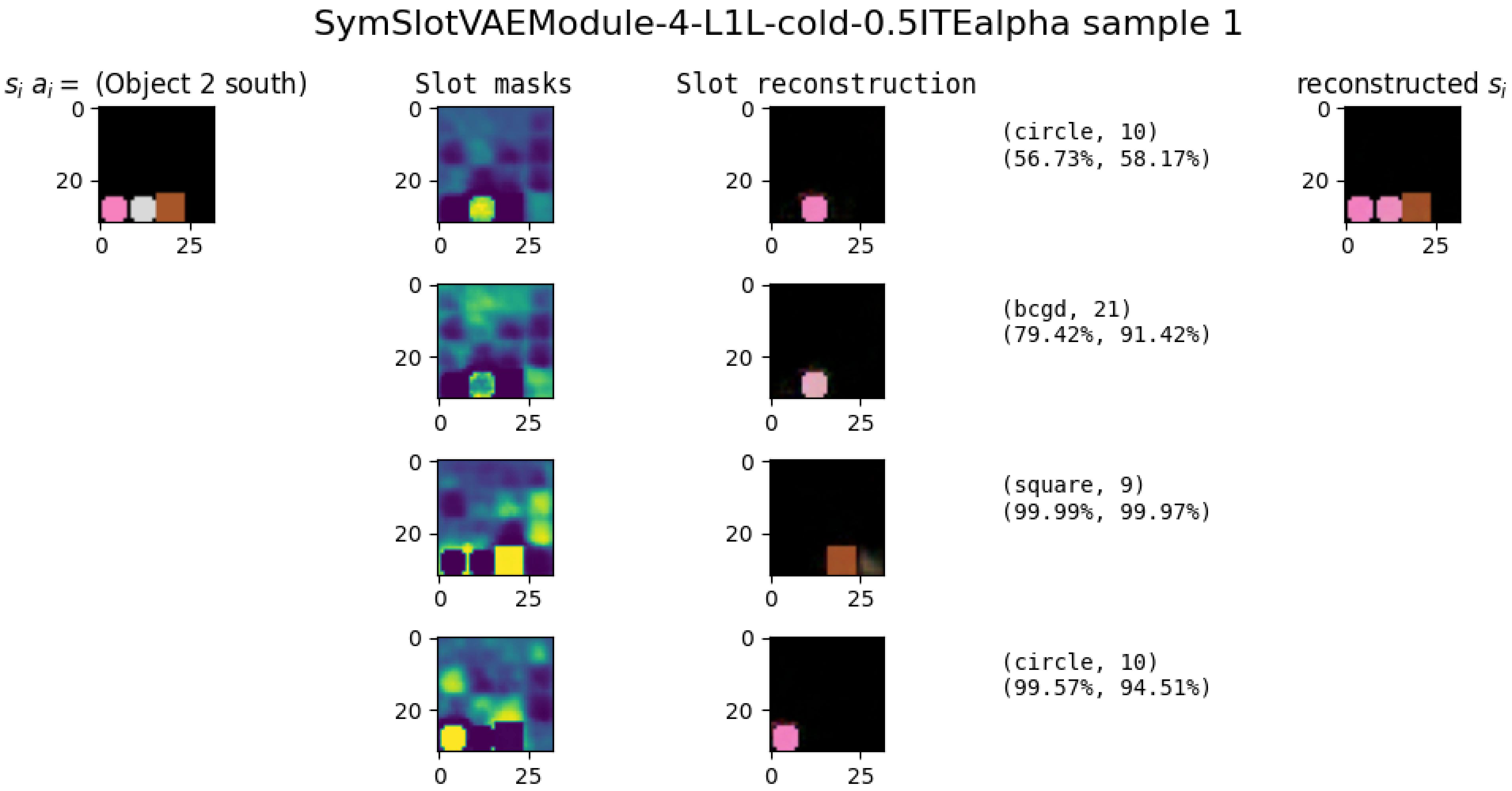
Training Curve



Sample 0

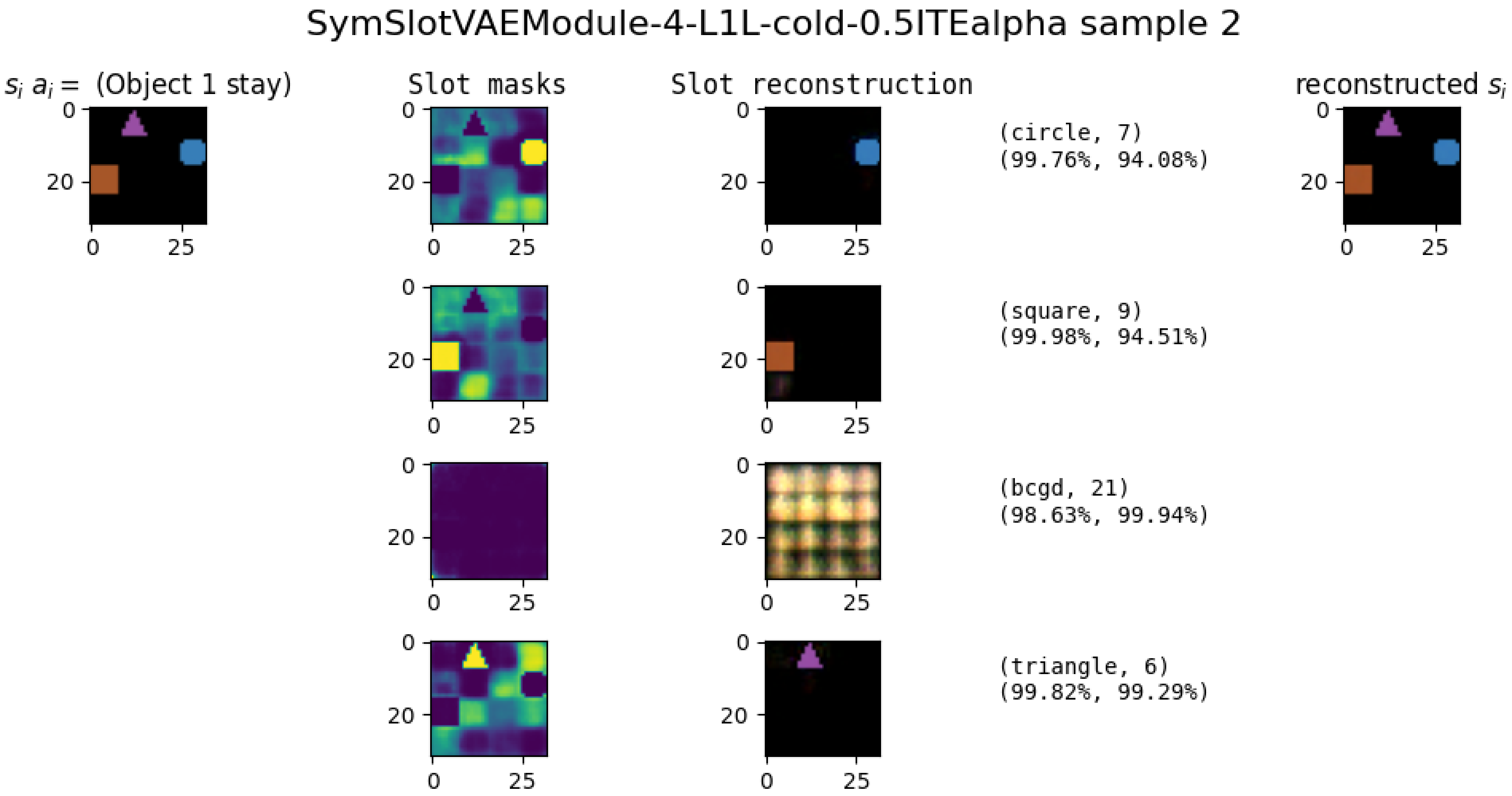

SymSlotVAEModule-4-L1L-cold-0.5ITEalpha sample 0
s_i a_i = (Object 2 north)
Slot masks
Slot reconstruction
reconstructed s_i
(circle, 17)
(99.97%, 99.49%)
(triangle, 0)
(99.99%, 95.89%)
(bcgd, 21)
(99.15%, 99.98%)
(square, 7)
(99.92%, 99.87%)

Sample 1



SymSlotVAEModule-4-L1L-cold-0.5ITEalpha sample 1
s_i a_i = (Object 2 south)
Slot masks
Slot reconstruction
reconstructed s_i
(circle, 10)
(56.73%, 58.17%)
(bcgd, 21)
(79.42%, 91.42%)
(square, 9)
(99.99%, 99.97%)
(circle, 10)
(99.57%, 94.51%)

Sample 2


SymSlotVAEModule-4-L1L-cold-0.5ITEalpha sample 2
s_i a_i = (Object 1 stay)
Slot masks
Slot reconstruction
reconstructed s_i
(circle, 7)
(99.76%, 94.08%)
(square, 9)
(99.98%, 94.51%)
(bcgd, 21)
(98.63%, 99.94%)
(triangle, 6)
(99.82%, 99.29%)
0
20
0
25

Dealing with multiple objects

```
S0: s_i |> (enc) |> (slot)                                            (dec) |> s_{i+1}
S1:                           |> (entitiy dec) |> (a_i) |> (???) |>
```

dPads

- Given a differentiable DSL, dPads aims to derive a program that maps an input stream to an output stream.
- $$\arg\min_{\theta,\alpha} \mathbb{E}_{\mathbf{i},\mathbf{o}\in D}[l(P(\mathbf{i};\alpha,\theta),o)] + c(\alpha)$$
- alpha is a program derivation graph

DSL

Inputs:

```
pos :: [int, int]
color :: int
shape :: {circ, sq, tri}
slot :: [color, shape, pos]
```