Neurosymbolic Object-Centric Learning with Distant Supervision

https://arxiv.org/abs/2506.16129

Presented by Jiong-Da Wang

2025-07-04

Contents

1. Introduction

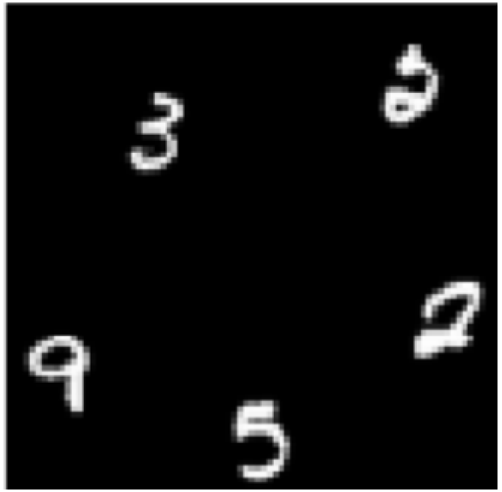
1.1 Authors

Stefano Colamonaco
PhD student at DTAI Lab, KU Leuven

David Debot
PhD candidate at DTAI Lab, KU Leuven

Giuseppe Marra
Assistant Professor
KU Leuven

1.2 Task



In Statistical Relational Learning (SRL) and Neurosymbolic AI, many works often make a critical assumption: *The input is already structured .* (e.g. a knowledge graph, a logical database)

This work only uses raw image input.

Object-centric learning tries to learn structured object-centric representation from low-level perceptual features. However, many models often rely on *object-level supervision*, meaning they require a label or annotation for each object in the image (It's difficult for most tasks).

This work only uses distant supervision from tasks.

2. Related Works

2.1 Slot attention

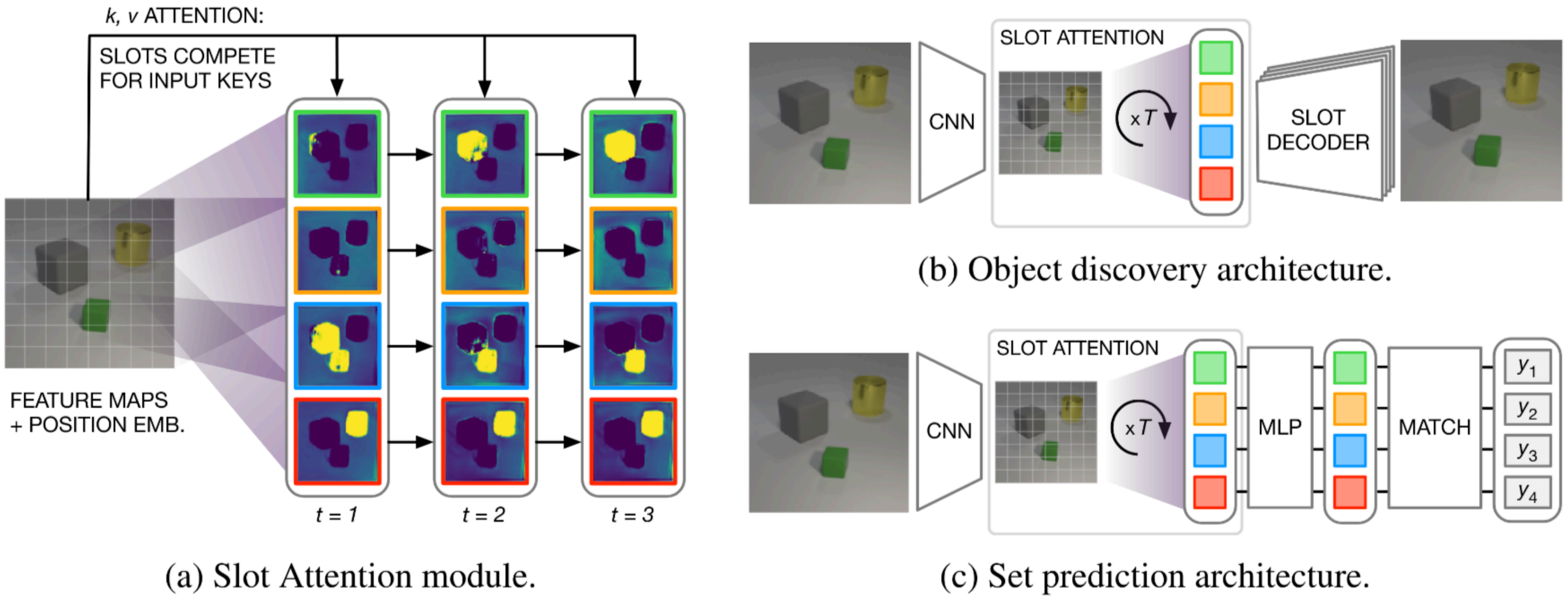


Figure 1: (**a**) Slot Attention module and example applications to (**b**) unsupervised object discovery and (**c**) supervised set prediction with labeled targets y_i. See main text for details.

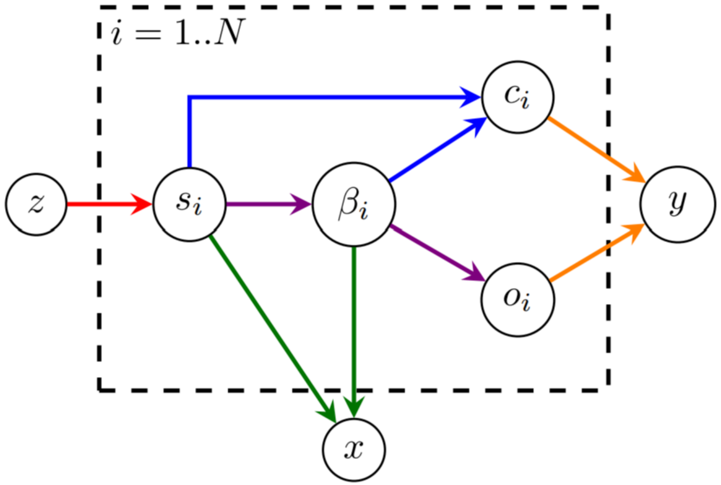
TL;DR: A slot (vector) can store (and bind to) any object in the input.

2.2 Neurosymbolic systems

Neurosymbolic systems

3. Methods

3.1 DeepObjectLog (Highlevel architecture)



- Objects encoder: $z \to s_i (i \in 1, ..., N)$
  z is a global representation.
  s_i describes a potential object in scene.
- Object detector: $s_i \to \beta_i, \beta_i \to o_i$
  β_i is the prob that s_i corresponds to a meaningful object. o_i is an *objectness* flag.
- Object classifier: $s_i, \beta_i \to c_i$, c_i is the class of s_i.
- Probabilistic Logic Reasoning: $c_i, o_i \to y$
  y is the downstream class label computed by reasoning on a logic theory L.
- Image Decoder: $s_i, \beta_i \to x$
  x is a probability distribution over images.

3.2 Probabilistic Graphical Model

$z \in \mathbb{R}^{d_z}$: a latent representation of the image

For $i \in 1, \ldots, N$, each i is a potentially present object.

$$p(y, x, \boldsymbol{s}, \boldsymbol{\beta}, \boldsymbol{o}, \boldsymbol{c}, z) =$$

$$p(z)p_L(y|\boldsymbol{o}, \boldsymbol{c})p(x|\boldsymbol{s}, \boldsymbol{\beta}) \left[\prod_{i=1}^{N} p(s_i|z)p(\beta_i|s_i)p(o_i|\beta_i)p(c_i|\beta_i, s_i)\right]$$

Image decoder

Object detector

Probabilistic logic reasoning

Slot extractor

Object classifier

3.2 Probabilistic Graphical Model

Four inductivbe assumptions:

- Object-level structure is latent and uncertain
- Symbolic abstraction (o_i, c_i, y) and perceptual grounding (x) are complementary flows
- Relational reasoning can be performed over learned representations
- Direct supervision at the object level can be avoided as it can be inferred from distant observations on x and y.

3.3 Object detector & classifier

Object detector:

$s_i \to \beta_i$, $\beta_i \in [0, 1]$ is the uncertainty scores. β_i is a determinstic value (the output of a neural network $\beta_i^* = f_\beta(s_i \mid \theta_\beta)$)

$\beta_i \to o_i$, $o_i \in \{0, 1\}$ is the objectness flag. (a Bernoulli distribution)

The classifier $s_i, \beta_i \to c_i$: a Bernoulli or a Categorical distribution. This distribution is parameterized by a NN $f_c(\beta_i s_i; \theta_c)$

s_i multiplies the uncertainty β_i elementwise, allows the model to downweigh uncertain or irrelevant representations.

3.4 Probabilistic logic programming (ProbLog)

Example:

```
% Probabilistic
facts:
0.5::heads1.
0.6::heads2.
% Rules:
someHeads :- heads1.
someHeads :- heads2.
% Queries:
query(someHeads).
% Result: 0.8
```

A ProbLog program L is defined over two sets of syntactic constructs: probabilistic facts and rules.

$$p(y) = \sum_{\boldsymbol{f}} \boxed{p(y|\boldsymbol{f}; R)} \prod_{f_i \in \boldsymbol{f}} p(f_i)$$

probabilistic facts **f**

a deterministic distribution stating whether y can be obtained from f applying the rules in R.

3.5 ProbLog for object-centric learning

Using ProbLog to model 3 components of the DeepObjectLog:

1. The object detectors $p(o_i|\beta_i)$: `object(i)`
2. The object classifiers $p(c_i|s_i, \beta_i)$: `class(i)` (binary classifier) or `class(i,k)` (categorical classifier)
3. The conditional task distribution $p_L(y| \boldsymbol{c}, \boldsymbol{o})$: Given the objectnesses `object(i)` and the classes of all objects `class(i, k)`, use all the logic rules to obtain the task y.

3.5 ProbLog for object-centric learning

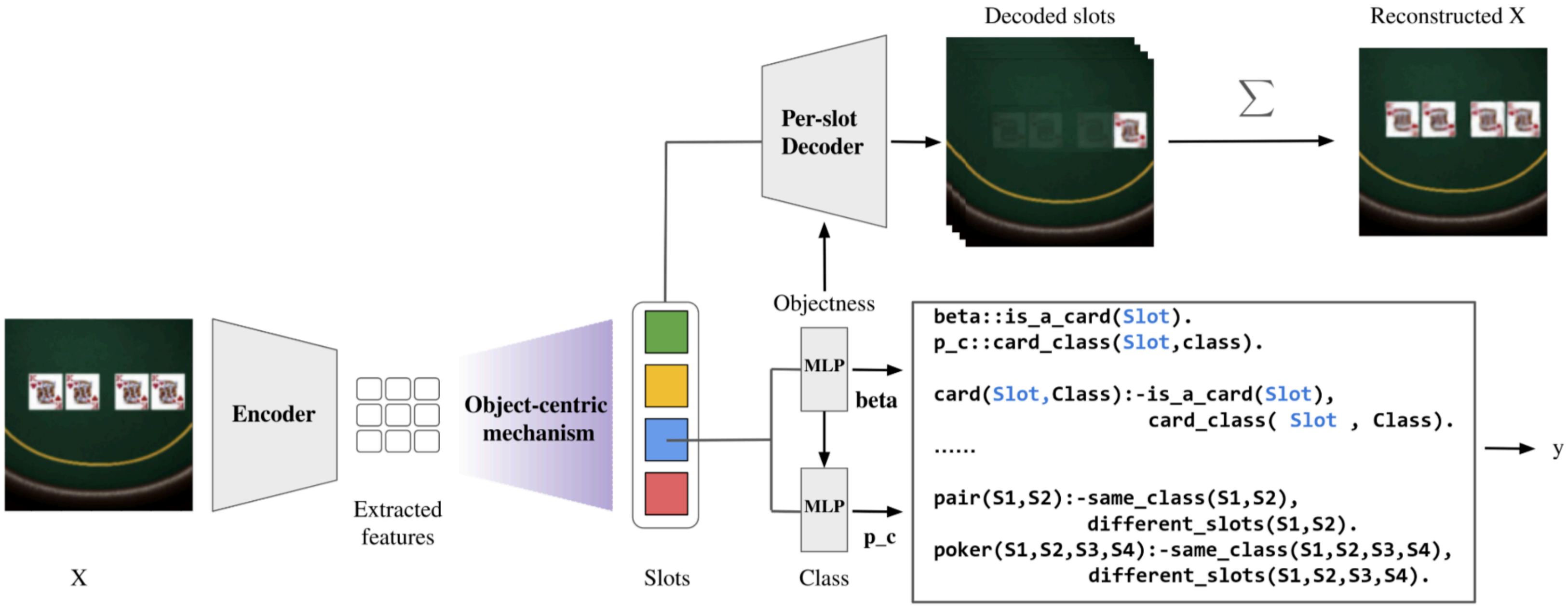
Example. Addition of at most two digits.

An image contains zero tw two digits that can only be 0 or 1. The task is to get the sum of digits in the image. We can encode this task as follows:

```
p_o_id :: object(ID) % distribution of objectness of the slot ID
p_c_id :: class(ID, C) % distribution of corresponding classes of the slot ID
digit(ID, Val) :- object(ID), class(ID, Val).
digit(ID, 0) :- \+ object(ID)
add(Z) :- digit(1, Y1), digit(2, Y2), Z is Y1 + Y2.
query(add(2))
```

3.6 Masked detector

The image decoder $p(x|\boldsymbol{s}, \boldsymbol{\beta})$ model the image as a mixture of Multivariate Gaussian distributions. Each slot can be decoded as $\mathcal{N}(\mu, I)$ that is parameterized with a NN $f_x(s_i\beta_i; \theta_x)$. The mixing weights w of slots are parameterized by a NN $f_w(s_i\beta_i; \theta_w)$.



3.7 Learning

$D = \{x, y\}_K$ is a dataset of pairs (x, y). $\theta_p = \{\theta_s, \theta_\beta, \theta_c, \theta_x, \theta_w\}$

$$\max_{\theta_p} \sum_{x, y \in D} \log p(x, y; \theta_p)$$

$$p(x, y) = \sum_{\boldsymbol{o}, \boldsymbol{c}} \int dz d\boldsymbol{s} d\boldsymbol{\beta} p(x, y, \boldsymbol{s}, \boldsymbol{\beta}, \boldsymbol{o}, \boldsymbol{c}, z)$$

For $\boldsymbol{o}, \boldsymbol{c}$, we use ProbLog to solve. And s, β are determinstic. So:

$$p(x, y) = \int dz p(x, y, z) = \int dz p(z) p(x | \boldsymbol{s}^*(z), \boldsymbol{\beta}^*(z)) p(y | \boldsymbol{s}^*(z))$$

z is global representation. We can't consider every z. How to solve it?

3.7 Learning

Since we are interested in a discriminative setting only, we can approximate the MAP state $z^* = \arg\max_z \log p(x, y, z)$ by a NN encoder $z^* \simeq q(x; \phi)$, where ϕ is set of parameters of the NN.

$$\max_{\theta_p, \phi} \sum_{(x,y)\in D} \log p\left(y | \boldsymbol{s}^*(q(x)), \boldsymbol{\beta}^*(q(x))\right)[1] +$$

$$\log p\left(x | \boldsymbol{s}^*(q(x)), \boldsymbol{\beta}^*(q(x))\right)[2] + \log p(q(x))[3]$$

[1]: A maximum likelihood term for the observed label y.

[2]: A standard reconstruction term of the observed x.

[3]: An activation regularization of $q(x)$.

4. Experiments

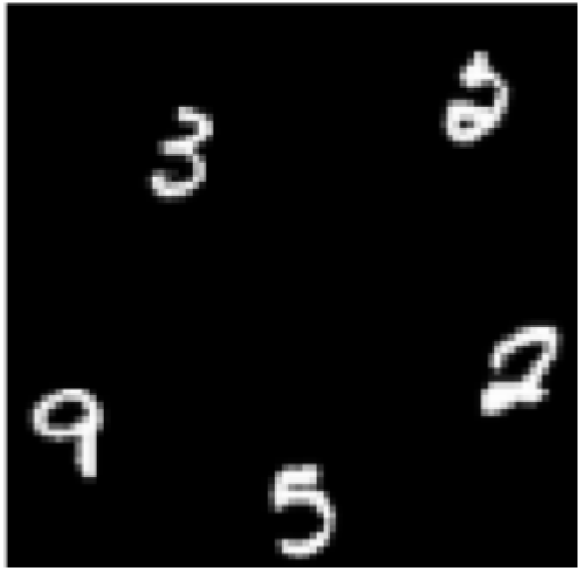
4.1 Three rearch questions

1. **Compositional generalization**: Can the model recombine familiar objects in novel ways beyond those encountered during training?
2. **Task generalization**: Can the model generalize to new tasks or output classes not present in the training data?
3. **Object count generalization**: Can the model detect, classify, and reason over a varying number of objects, including configurations larger (extrapolation) or smaller (interpolation) than those seen during training?

4.2 Datasets

MultiMNIST-Addition Dataset

Input: an image containing sevel digits. **Output**: the sum of the digits image contains.



PokerRules Dataset

Input: an image containing sevel poker cards. **Output**: The poker combination form (Nothing, pair, straight, etc.).

4.3 Results

Task Accuracy and Concept Accuracy (digit-level decomposition) for models evaluated on MultiMNIST-Addition.
The model is trained on images containing at most 3 digits and evaluated on images containing 4 and 5 digits (extrapolation).

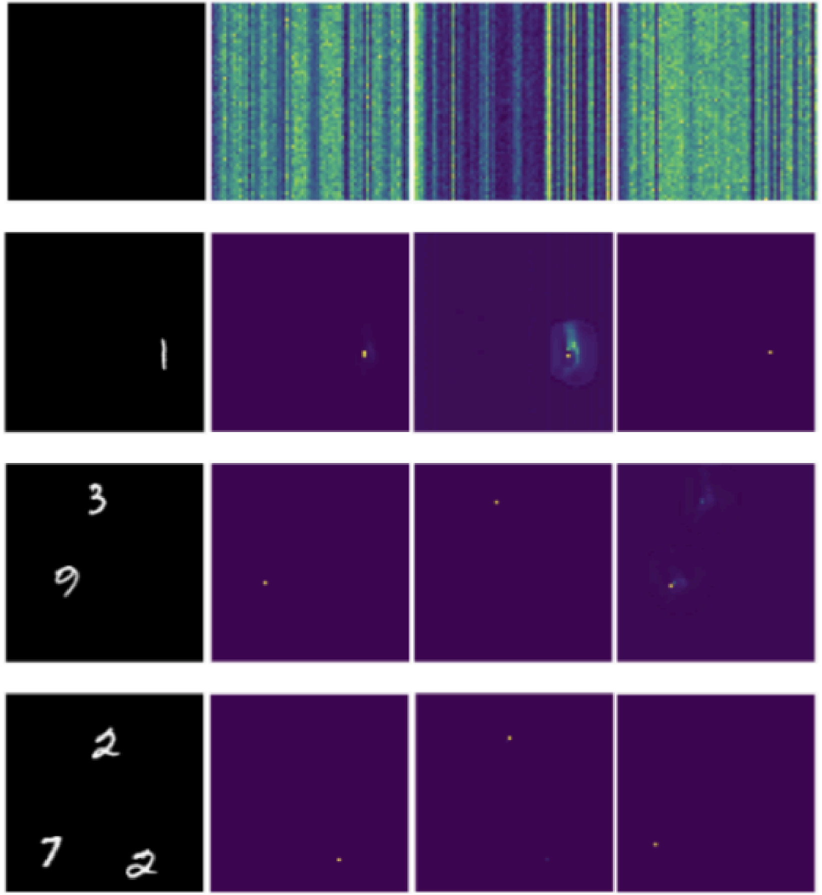
	Task Acc.		Concept Acc.		Extrapolation	
	Test	OOD	Test	OOD	4 digits	5 digits
CNN	79.90±0.17	3.43±0.15	-	-	22.16±0.90	13.16±0.96
SA	**98.90**±0.34	13.30±3.14	-	-	36.23±4.44	9.76±6.11
SA-MESH	98.86±0.47	18.26±1.33	-	-	37.50±1.15	12.00±0.51
CoSA	93.00±1.92	36.2±15.10	-	-	52.20±14.01	25.06±12.82
NeSy	90.70±4.02	35.76±6.24	55.43±26.40	23.83±3.82	-	-
Ours	94.26±2.00	**90.00**±3.01	**85.16**±2.45	**65.46**±5.70	**69.73**±10.74	**44.06**±3.85

4.3 Results

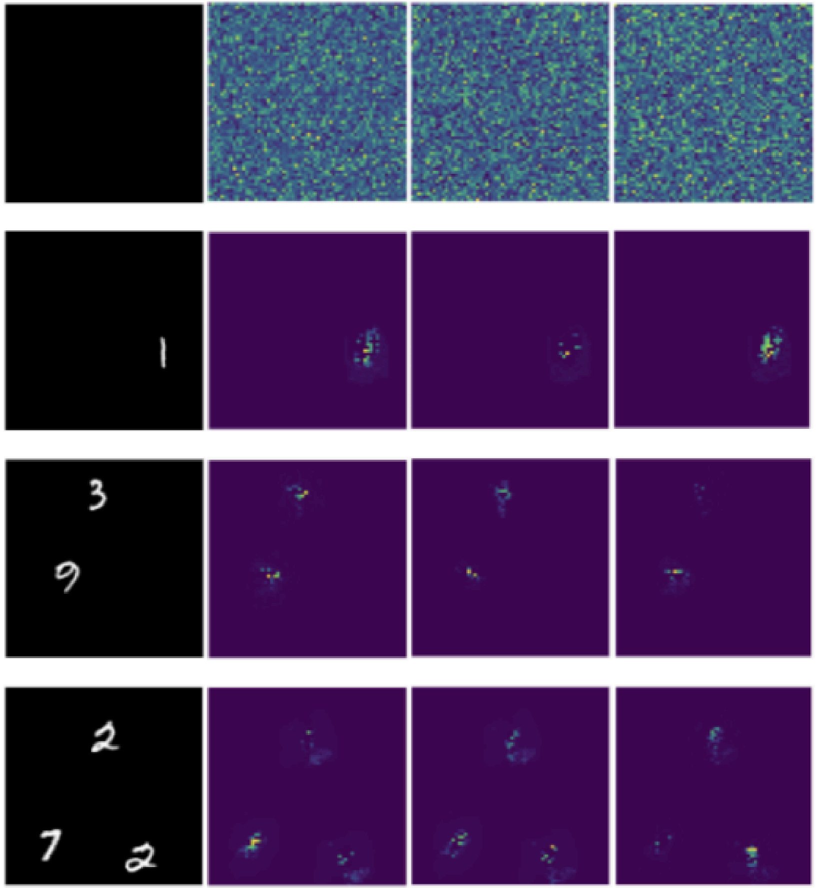
Task Accuracy for models evaluated on PokerRules:

			Extrapolation: 5 cards	
	Test	OOD class	In-distribution class	OOD class
CNN	81.96±1.10	-	22.03±5.27	-
SA	99.30±1.21	-	35.86±8.72	-
SA-MESH	**99.93**±0.11	-	37.80±4.91	-
CoSA	95.46±3.90	-	44.26±17.16	-
NeSy	80.23±2.11	0.46±0.05	-	-
Ours	97.90±1.17	**72.23**±16.72	**78.53**±4.68	**78.23**±19.24

4.3 Results



Ours



SA-MESH

4.3 Results