Supplementary Material for Enhancing Interpretable Object Abstraction via Clustering-based Slot Initialization

Ning Gao[1,2]
ning.gao@de.bosch.com
Bernard Hohmann[2]
bernard.hohmann@student.kit.edu
Gerhard Neumann[2]
gerhard.neumann@kit.edu

[1] Bosch Center for Artificial Intelligence
Renningen, Germany

[2] Autonomous Learning Robots
Karlsruhe Institute of Technology,
Karlsruhe, Germany

In the appendix, we show the implementation details and qualitative results.

1 Implementation details

We show the implementation details and the architectures of different variants here. **Slot Attention:** This section provides a detailed explanation of all the methods presented in chapter 2. The Slot Attention architecture in Figure 6 is extended by a clusterization algorithm, that can be either k-means or mean shift, and by a mapping algorithm, being one of *Direct*, *Small MLP*, *Large MLP* or *Pseudoweights*. The encoder can be a U-Net or a size preserving convolution network. The extension initializes slots conditioned on the perceptual input and not like the original Slot Attention architecture from random gaussian distributions. During the iterative slot attention process, the initialized slots are updated to attend to certain feature pixels, while ignoring others. This is described by the bright yellow markings in the attention masks in Figure 6. The Slot attention uses three iterations to update the slots. Each slot is decoded into a rgb-image and an α-mask. The renderer calculates, with a weighted sum, the output according to the slotwise rgb-image and the α-mask.

IODINE: The extension for IODINE resemble the same structure as in the slot attention architecture in Figure 7. The only difference is that the mapping algorithm has to map between the cluster centers of dimension D to two parameters μ, σ of the Gaussian distribution. That is why *Direct* mapping is impossible for IODINE. Slot initializations are now drawn out of the perceptual conditioned gaussian distribution and have dimension D. A decoder calculates, in the same fashion as for slot attention, for each slot a rgb-image and an α-mask. The render outputs the reconstructed image, that will be compared to the groundtruth image to produce a loss. The loss is used in a refinement network, with auxiliary inputs, to update the gaussian parameters μ, σ. This process is repeated five times.

Direct mapping: This simple permutation equivariant approach depicted in Figure 8 directly injects the cluster centers determined by the clusterization algorithms into the slots.

© 2023. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

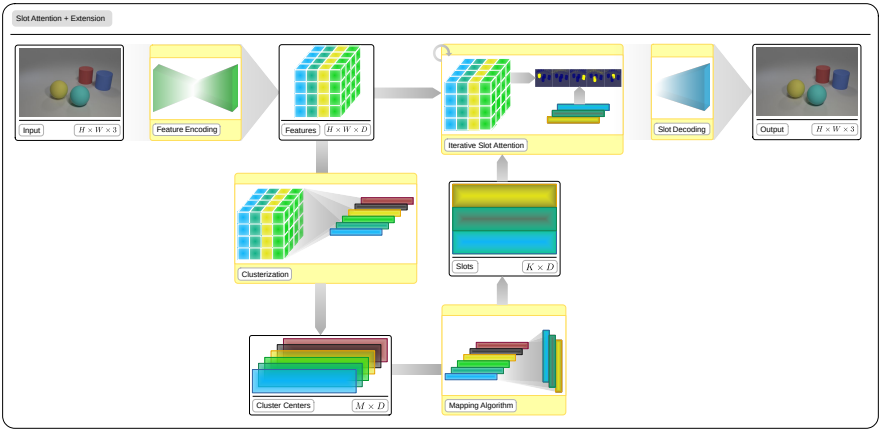

Figure 6: The framework architecture for slot initialization for slto attention. The top row is the original architecture.

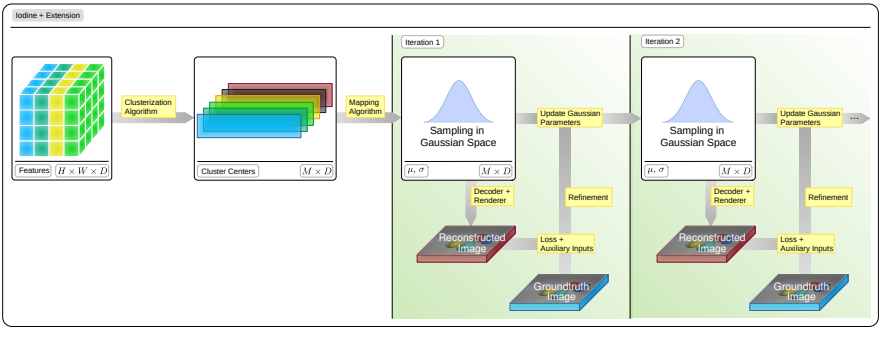

Figure 7: The framework architecture for IODINE based extensions. The original starts directly at iteration 1 with slots drawn out of the standard gaussian distribution with $(\mu, \sigma) = (0, 1)$.

Since there is no mapping network involved, this approach can not be used for IODINE, because the cluster centers have to be mapped to two gaussian parameters μ, σ.

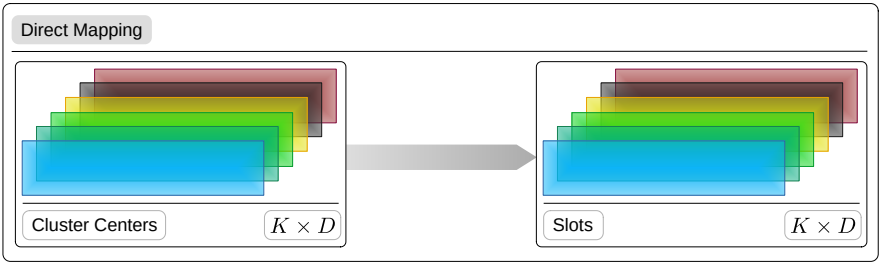

Figure 8: The Direct mapping approach. Slots are identical to the cluster centers chosen by the clusterization algorithm.

Small MLPs: This mapping extends *Direct*-mapping with a non linear network between the cluster centers and the slots, that is shared between all slots, as depicted in Figure 9. The *Direct*- and *Small MLPs*-mapping are used for their simplicity and the permutation equivariance. But they can only translate between the same number of cluster centers and slots.

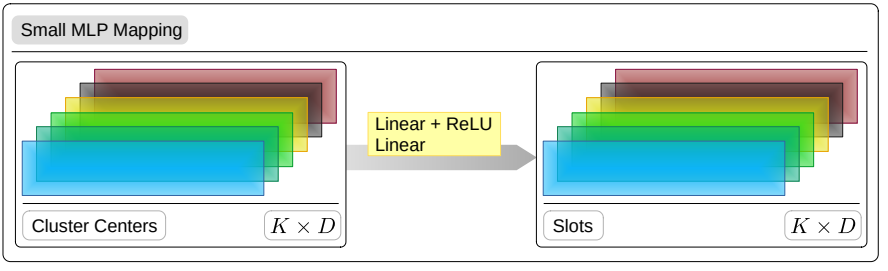

Figure 9: The Small MLPs mapping approach. It extends the direct mapping approach by a nonlinear network between cluster centers and slots.

Large MLPs: This network maps between a different number of cluster centers and slots, as provided in Figure 10. The reason for this is to increase the sampling amount of cluster centers from the perceptual input without increasing the model size noticeably, which scales linear with the amount of slots. It is not shared between the slots and thus it is not permutation symmetric. The cluster centers are flattened into one large vector and then mapped to a flattened representation of the slots. These slots are then reshaped to $M \times D$. A drawback of this design is, that it can not generalize to more slots, like all other mapping networks, because of the fixed large MLPs.

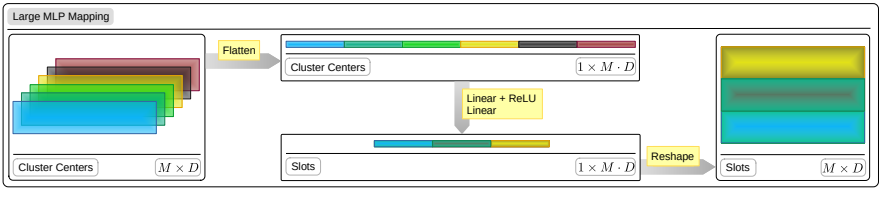

Figure 10: The Large MLPs approach.

Pseudoweights: This algorithm incorporates several concepts into one mapping approach. It can map between a different amount of cluster centers and slots, while being able to generalize to more slots and keeping permutation invariance. It has to be permutation invariant, because it is ambiguous to define permutation equivariance between two not equally large sets. This mapping sorts cluster centers into slots. It is aware in which slot it is, because of the position encoding of the K slots. Thus the segregation network before the pseudoweights tensor can decide, if a cluster center should be sorted into a particular slot, then the weights in the pseudoweights tensor will be high, other wise the weights will be low. This segregation network does the decision conditioned only on one cluster center and one position code for all possible $M \times K$ pairs. The last step calculates the weighted sum with the pseudoweights tensor and returns the initialized slots. An explanation of this process and a visual proof of permutation invariance is provided in Figure 12.

Clusterization Algorithms: The k-means algorithm used in the presented methods uses the k-means++ initialization, where the first center is randomly chosen and all other centers

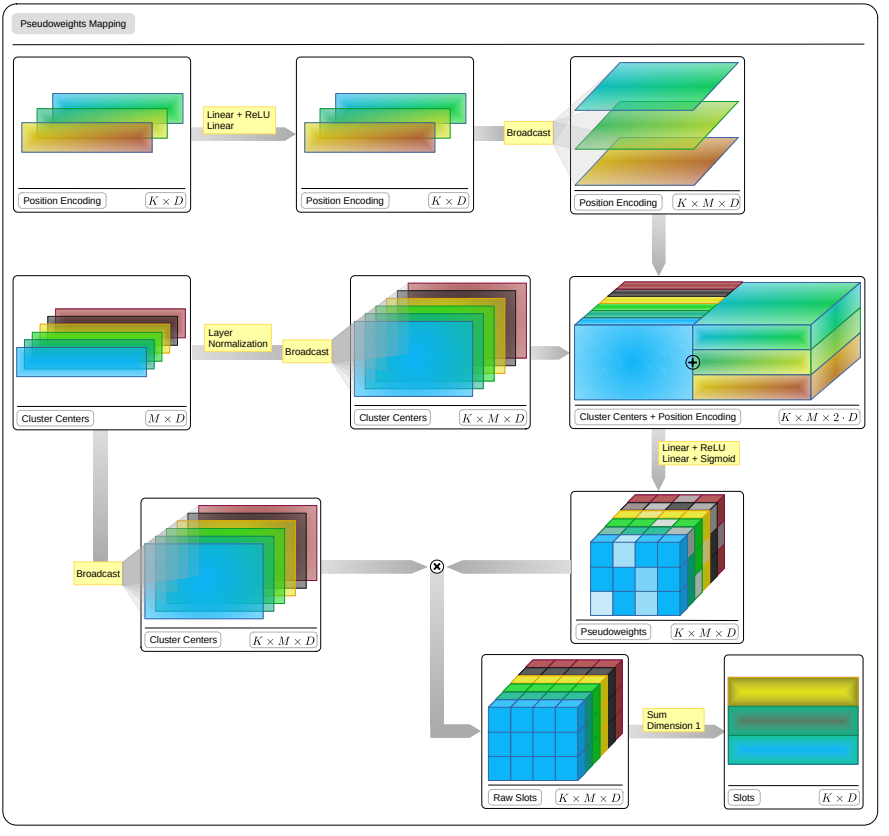

Figure 11: The permutation invariant Pseudoweights mapping.

are initialized iteratively at the data point being the farthest away from all current initialized centers. If k-means is used with the *Large MLP*, it requires a cluster dying prevention, because sometimes a cluster center will vanish, if all data points are closer to other cluster centers. In that case, a new cluster center is initialized with the k-means++ initialization. A pseudo code is provided in 1. The amount of cluster centers used in k-means is always initialized with the double amount of the maximum objects count in the dataset. So for CLEVR6, where there are up to six foreground objects and one background object, we initialize always 14 cluster centers at the start of k-means. The mean shift algorithm is initialized with 20 cluster centers for all datasets, because after mean shift converges an algorithm called *connected-components* is used to merge clusters centers, that are very close to each other in to one vector. This ability lets mean shift to determine the amount of slotsflexible. The hyper parameter ε is used to determine the radius in the *connected-components*, where all cluster centers within the ε-sphere are merged to one vector. Another hyper parameter used in mean-shift is σ and is used to determine the bandwidth of the gaussian kernel. A detailed pseudo code is provided in 1. We determine the hyperparameters dependent on the weight initialization of the network, so that from the beginning of the training, the output amount of slots fluctuates between 1 and 20, but will never be always 20 or always 1. This happens if σ or ε are too small, then mean shift will converge into every little mode, or if the

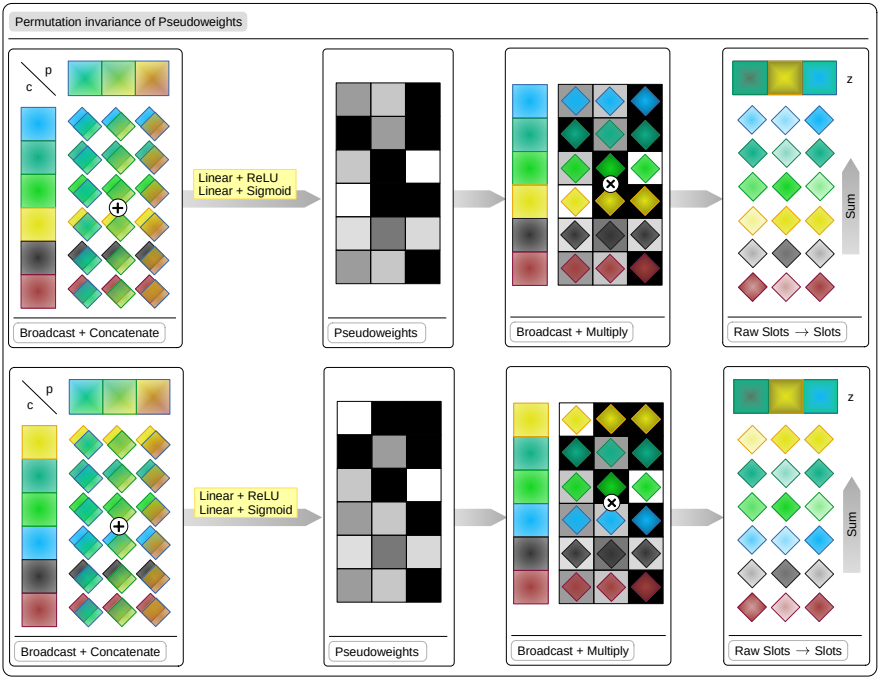

Figure 12: The permutation invariant mapping between 6 cluster centers and 3 slots. For this example all slots and cluster centers are of dimension D=1, to keep it simple. The pseudoweights tensor has high values in black squares and low values in white squares. If the blue and yellow slot change their position, the slots won't change their initialization.

hyperparameters are too large, then all cluster centers can merge into the same spot.

Algorithm 1 K-means algorithm with cluster dying prevention, that reinitializes a new cluster center as soon as one vanishes.

$c_i \leftarrow$ k-means++ initialization; $i \leq N$
repeat
  for each c_i **do**
    $C_i = \{x_j : d(x_j, c_i) \leq d(x_j, c_k); \forall x_j \wedge \forall k \neq i\}$
  end for
  for each C_i **do**
    if $C_i = \varnothing$ **then**
      $c_{inew} \leftarrow$ k-means++ reinitialization
    else
      $c_{inew} = \sum_{c_i \in C_i} \frac{c_i}{|C_i|}$
    end if
  end for
  if $d(c_i, c_{inew}) \leq tolerance \forall i$ **then**
    Return c_{inew}
  end if
until max iterations
Return c_{inew}

Algorithm 2 Mean shift algorithm, with the hyperparameters ε used in the connected-components algorithm and σ used in the gaussian kernel function.

for $n \in 1, ..., N$ **do**
  $x \leftarrow x_n$
  repeat
    $\forall n : p(n|x) \leftarrow \frac{exp(-0.5||\frac{x-x_n}{\sigma}||^2)}{\sum_{n'=1}^{N} exp(-0.5||\frac{x-x_n}{\sigma}||^2)})$
    $x \leftarrow \sum_{n'=1}^{N} p(n|x) \cdot x_n$
  until stop
  $z_n \leftarrow x$
end for
connected-components$(\{z_n\}_{n=1}^{N}, \varepsilon)$

2 Visualizations on object discovery task

We show some qualitative evaluation examples here for the object discovery task.

3 Visualizations on novel view synthesis task

We visualize the examples of novel view synthesis tasks in Figure 21.

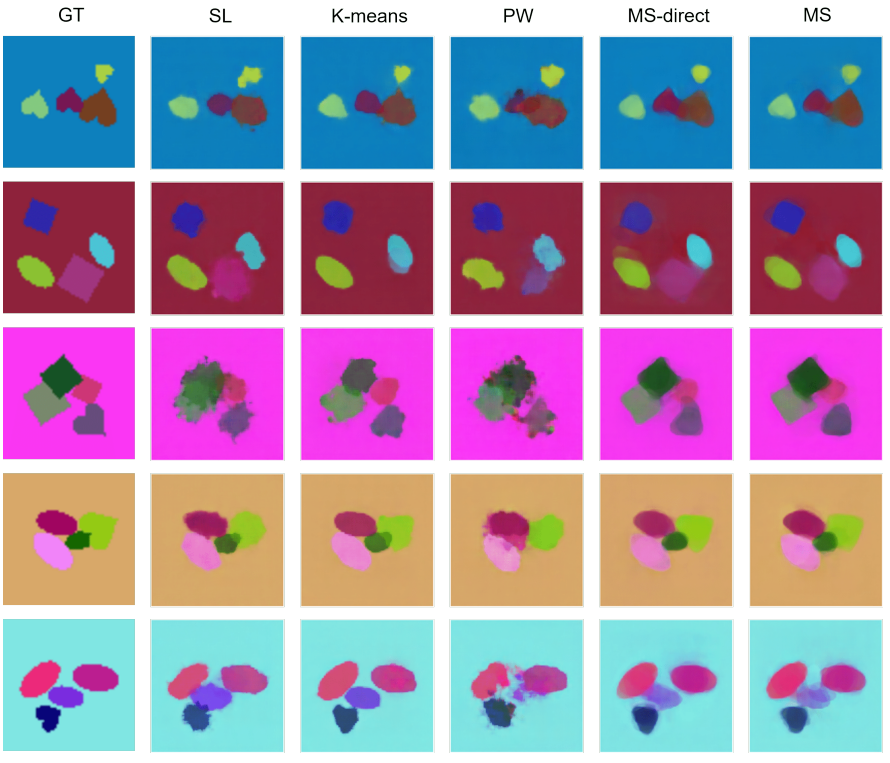

Figure 13: Qualitative results on MDS dataset.

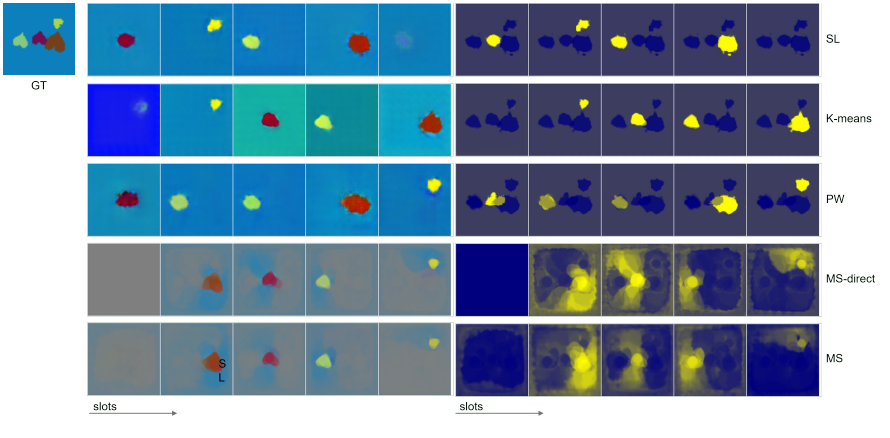

Figure 14: The slot-wise predicted masks and reconstructed scenes on MDS dataset.

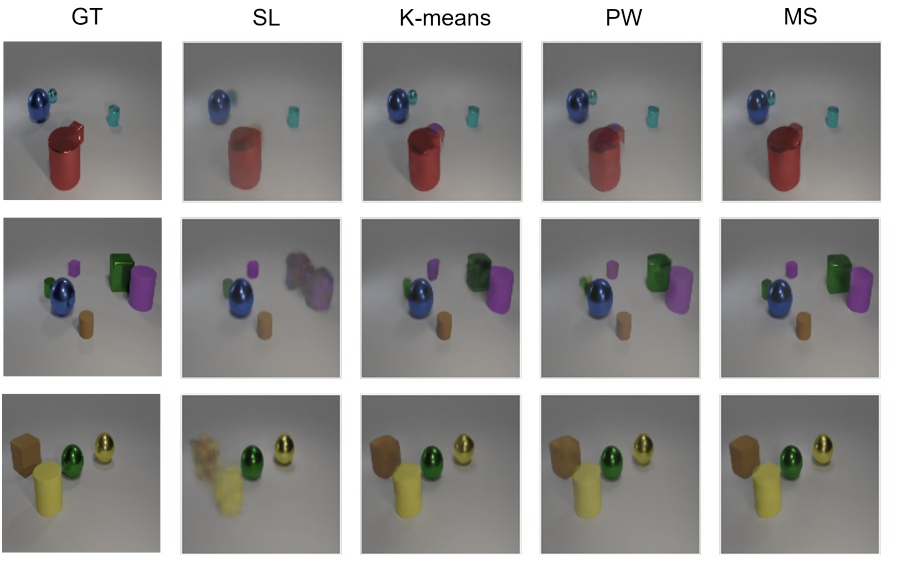

Figure 15: The original Slot Attention model struggles with overlapped objects.

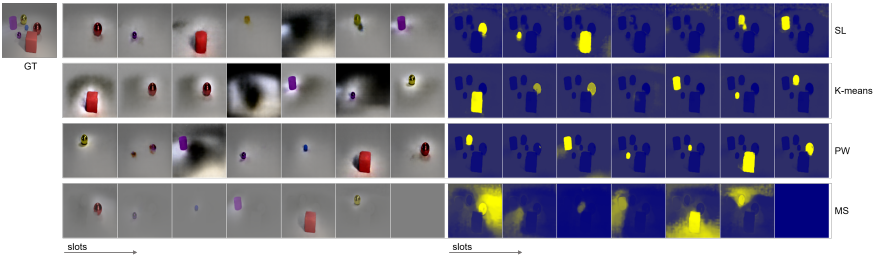

Figure 16: The slot-wise predicted masks and reconstructed scenes on CLEVR6 dataset.

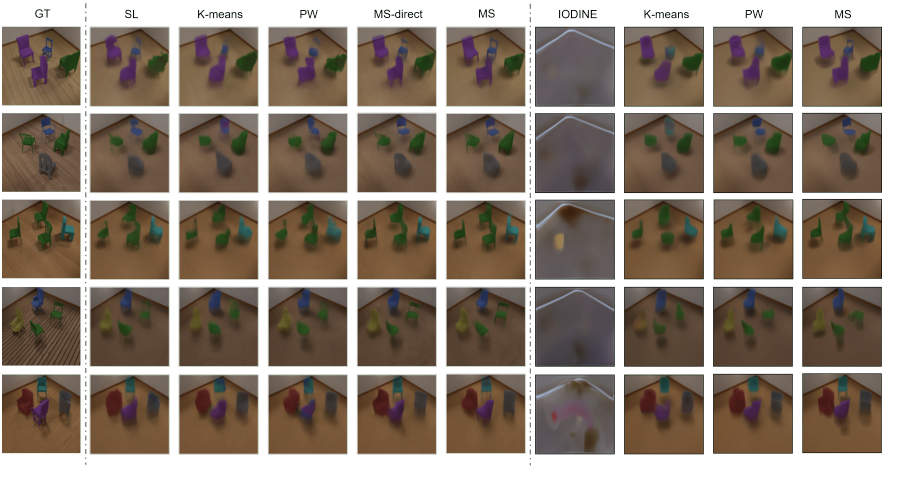

Figure 17: Qualitative results on Chairs dataset.

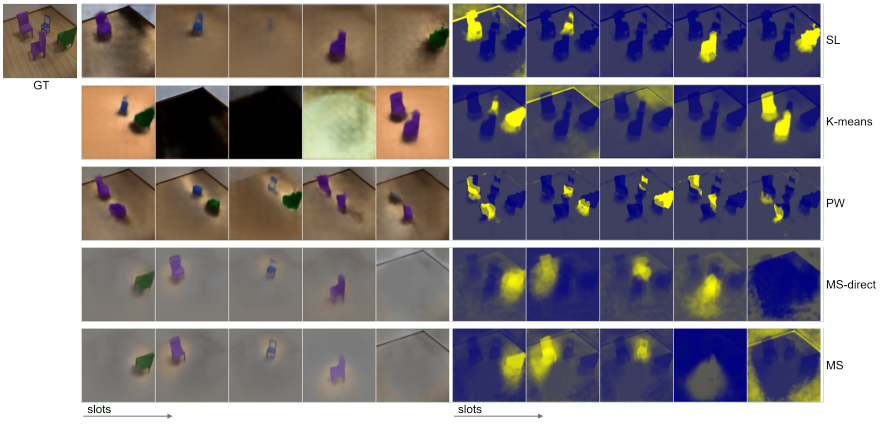

Figure 18: The slot-wise predicted masks and reconstructed scenes on Chairs dataset.

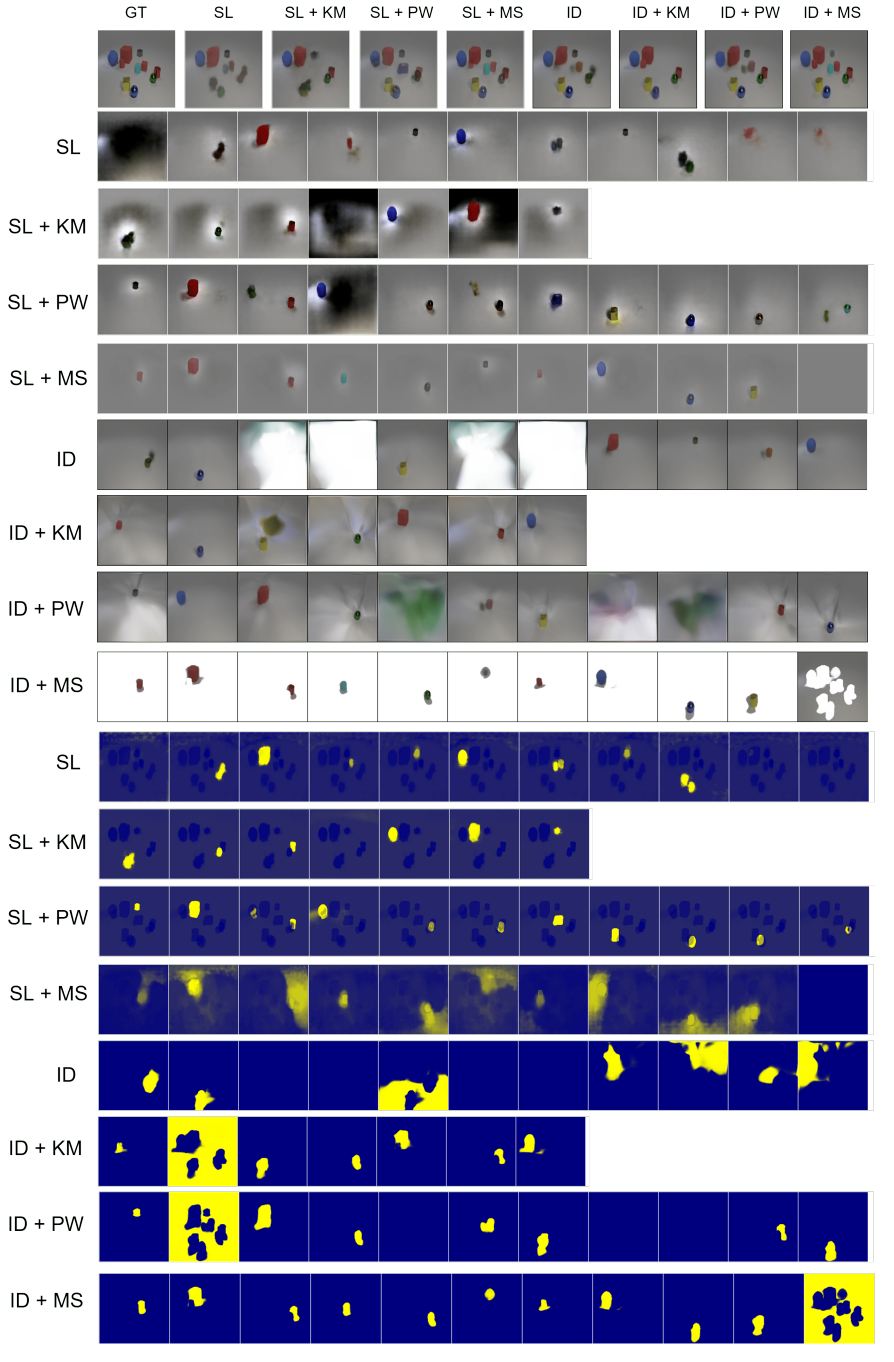

Figure 19: Qualitative comparison of generalization on CLEVR10 while the models are trained with CLEVR6.

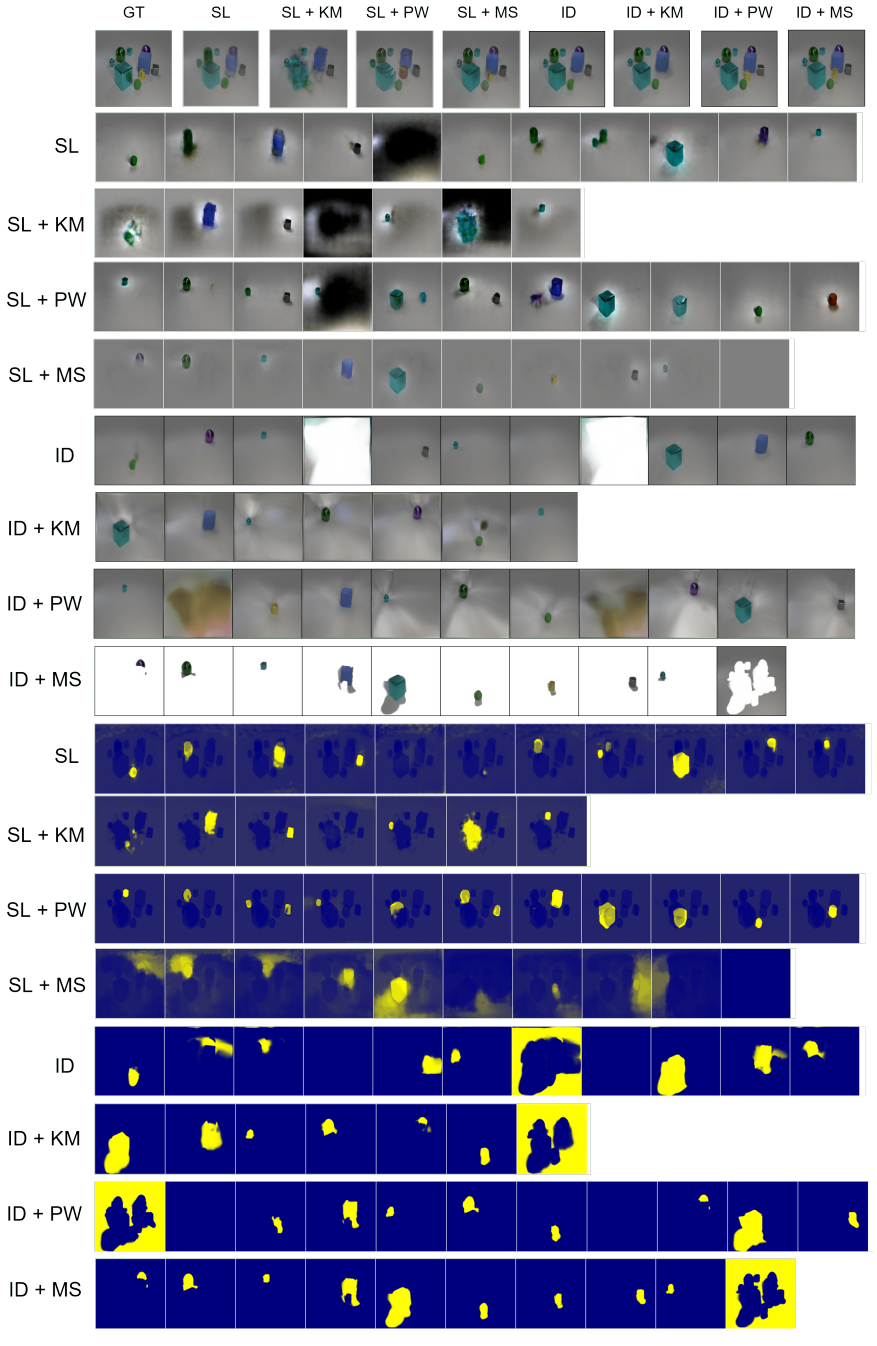

Figure 20: Another qualitative comparison of generalization on CLEVR10.

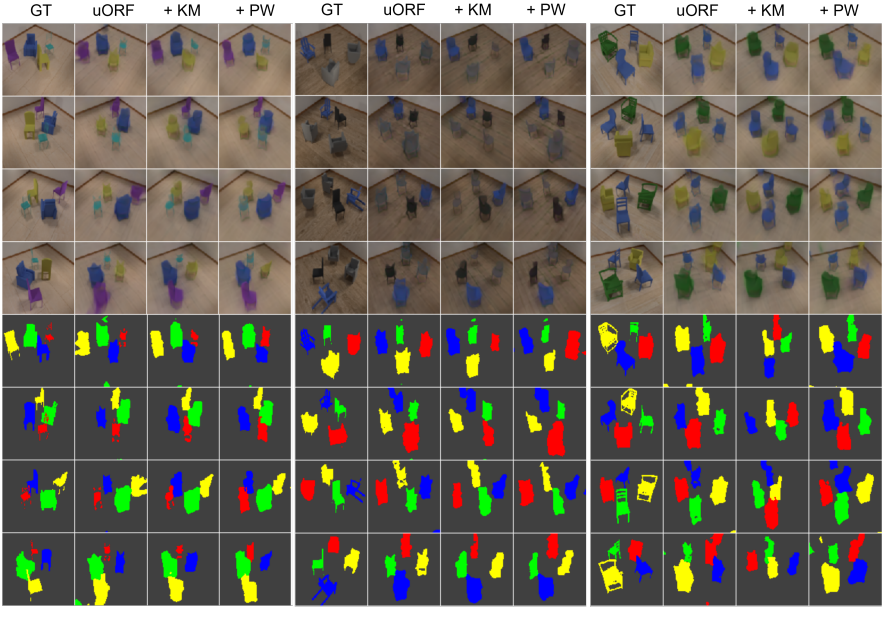

Figure 21: Qualitative results on novel view synthesis. Our models can represent the chairs with more details than the original uORF.