LEARNING INDUCTIVE OBJECT-CENTRIC SLOT INITIALIZATION VIA CLUSTERING

Anonymous authors
Paper under double-blind review

ABSTRACT

Object-centric representations using slots have shown the advances towards efficient, flexible and interpretable abstraction from low-level perceptual features in a compositional scene. Current approaches randomize the initial state of slots followed by an iterative refinement. As we show in this paper, the random slot initialization significantly affects the accuracy of the final slot prediction. Moreover, current approaches require a predetermined number of slots from prior knowledge of the data, which limits the applicability in the real world. In our work, we initialize the slot representations with clustering algorithms conditioned on the perceptual input features. This requires an additional layer in the architecture to initialize the slots given the identified clusters. We design permutation invariant and permutation equivariant versions of this layer to enable the exchangeable slot representations after clustering. Additionally, we employ mean-shift clustering to automatically identify the number of slots for a given scene. We evaluate our method on object discovery and novel view synthesis tasks with various datasets. The results show that our method outperforms prior works consistently, especially for complex scenes[1].

1 INTRODUCTION

Object-centric representations using slots have shown good performance in object detection (Li et al., 2021a; Locatello et al., 2020), segmentation (Kabra et al., 2021; Greff et al., 2019) and tracking (Wu et al., 2021; Kipf et al., 2022; Li et al., 2020) tasks. Slots are a set of latent variables. The common approach is to frame disentangled and structured slot representations of the compositional scene with some iterative refinement mechanisms in a self-supervised manner, e.g., using softmax-based attention (Locatello et al., 2020) or variational inference (Greff et al., 2019). The idea is to improve the sample efficiency and generalization of capturing the structured environment to unseen compositions or objects. However, most slot-based approaches have difficulties in representing complex scenes. Moreover, the number of slots needs to be specified beforehand on each dataset, which limits the generalization across datasets. In addition, a random slot initialization from a common distribution is widely used in prior works, which lacks consideration between the slots and the perceptual input. Consequently, the quality of the following iterative slot refinement is also affected by the sub-optimal initialization.

To overcome these challenges, instead of random sampling, it is intuitive to sample the initial slot representations conditioned on the perceptual input (see Figure 1). Hence, we employ the k-means clustering algorithm on the convolutional features of the input image. A set of cluster centers are specified based on the features. Afterwards, a set of slots are initialized given the cluster centers as input. Since the order of cluster centers changes randomly, we extend this idea with a permutation-invariant mechanism, where the initial slot representations remain invariant w.r.t. the order of clusters. To further evaluate the effect of permutation symmetry for slot representations, we employ another permutation equivariant model with mean-shift clustering algorithm, where the slot representations change accordingly with respect to the permutation of the clusters. Mean-shift identifies the number of clusters automatically based on each perceptual input, followed by an injective map-

[1] Code, data and evaluation results are available at `https://github.com/slot-initialization/linosic` for the review purpose.

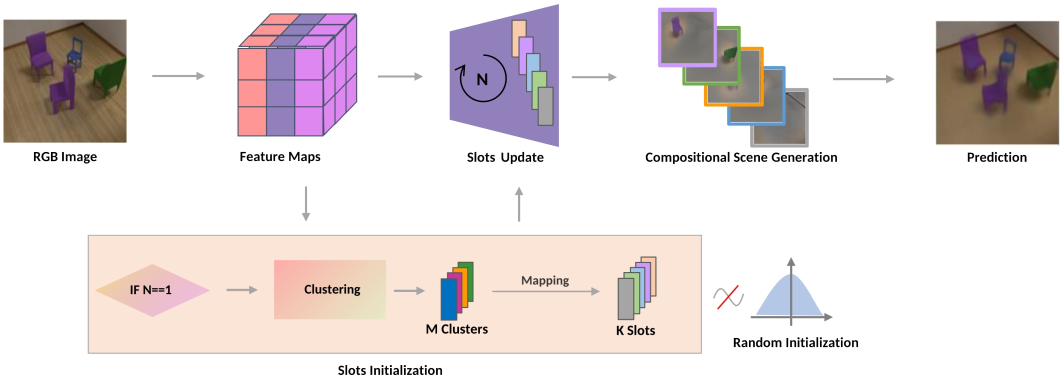

Figure 1: The network architecture. Instead randomizing slot initialization from a common distribution widely used in prior work, we initialize slot representations conditioned on the input features. A clustering algorithm and a mapping layer are adopted.

ping where each slot is considered as an output of each cluster individually. Thus, it does not require a fixed number of slots based on the whole dataset as prior works.

Our proposed method can be easily placed on top of existing slot-based approaches and trained in an end-to-end manner. In this work, we consider object discovery and novel view synthesis as downstream tasks. To evaluate the improvement and versatility of our method, we choose Slot Attention (Locatello et al., 2020) and IODINE (Greff et al., 2019) as baselines for object discovery task, and uORF (Yu et al., 2022) for novel view synthesis. The experiments are conducted on various datasets.

Our main contributions are as follows: i) We propose the conditional slot initialization using clustering algorithms instead of random initialization. ii) We analyze the effect of permutation symmetry including invariance and equivariance on the object-centric slot representations using different architectures and models. iii) We apply mean-shift clustering on the perceptual features which allows to generate flexible number of slots. iv) We demonstrate that, our proposed idea achieves significant improvement over all baselines, while the permutation equivariant mean-shift model presents notable advances especially for complex scenes.

2 Guiding Slot Initialization using Clustering

In this section, we will introduce i) the conditional slot initialization with k-means clustering (KM) in Section 2.1, ii) the permutation invariant version named *Pseudoweights* (PW) in Section 2.2, iii) and the permutation equivariant version with variable slot generation using the mean-shift clustering (MS) in Section 2.3. More details about implementations and architectures are shown in appendix A.1.

2.1 Image-Dependent Slot Initialization

Most slot-based methods typically sample from a standard Gaussian as the random initialization for the slot latent variables (see Figure 1). Although the slots are updated by the refinement mechanism incorporating the features from the perceptual input, it is inefficient to start from a random initialization and also limits the final accuracy. Since the perceptual input includes a strong inductive bias about the represented scene, it is straightforward to incorporate the perceptual input directly from the beginning. We first implement a non-permutation symmetric model using k-means clustering. K-means is applied on the pixel-wise convolutional perceptual feature $\mathbf{x} \in \mathbb{R}^{N \times D}$ to get the feature-based cluster centers: $\mathbf{c} = \text{K-means}(\mathbf{x}) \in \mathbb{R}^{M \times D}$ where N is number of pixels from the feature input, M is the number of clusters and D is the feature dimension. Afterwards, the cluster centers are flattened and mapped to the K slots using multi-layer perceptrons (MLPs):

$\mathbf{z}_{slots} = \text{MLP}(\mathbf{c}.\text{flat}()).\text{reshape}(K, D)$. Therefore, the number of slots is fixed beforehand like in prior works, as well as the amount of cluster centers.

2.2 Permutation-Invariant Slot Initialization

A good slot representation respects the permutation symmetry (Locatello et al., 2020). In our case, the order of the predicted slots should either remain the same (permutation invariance) w.r.t. the permutation of the cluster centers or change correspondingly in the same order as the cluster centers (permutation equivariance). Such symmetric behavior enables good generalization of slot representations to unseen world and objects. However, a simple mapping between M cluster centers and K slots as shown in Section 2.1 breaks the permutation symmetry and cannot generalize to more slots during evaluation for the scenes with more objects. To address this issue, we propose a permutation invariant model named *Pseudoweights*. To identify different slots, we use a sine-cosine positional encoding $\mathbf{p}_k$ for the k-th slot as follows:

$$\mathbf{p}_k = \Big(\sin\big(\frac{\pi}{D'}k\big), \cos\big(\frac{\pi}{D'}k\big), \sin\big(\frac{2\pi}{D'}k\big), \cos\big(\frac{2\pi}{D'}k\big), ..., \sin\big(\pi k\big), \cos(\pi k)\Big),\ k = 1, ..., K, \quad (1)$$

where $D' = \frac{D}{2}$ and D denotes the embedding length. Afterwards, the cluster centers are broadcasted along the slot dimension $\mathbf{c} \in \mathbb{R}^{K\times M\times D}$ and are concatenated with the broadcast of the positional encoding $\mathbf{p} \in \mathbb{R}^{K\times M\times D}$ to predict the weights $\mathbf{w} = \text{MLPs}([\mathbf{c}, \mathbf{p}]) \in \mathbb{R}^{K\times M\times D}$, which allocate the importance of the cluster centers to the different slots. We use a soft-max layer such that the weights allocated for each slot are normalized as follows:

$$\sum_{m=1}^{M} w_{k,m,d} = 1,\ w_{k,m,d} \in [0, 1],\ k = 1, ..., K,\ m = 1, ..., M,\ d = 1, ..., D. \quad (2)$$

The slots are then initialized as the weighted sum over the cluster centers by $\mathbf{w}$:

$$\mathbf{z}_k = \sum_{m=1}^{M} \mathbf{w}_{k,m} \cdot \mathbf{c}_{k,m}. \quad (3)$$

Thus, the *Pseudoweights* mapping applies a permutation invariant assignment of cluster centers into the slots. Moreover, since the slots are identified by the positional encoding, it enables generalization on increasing objects during test by changing the defined number of slots K without increasing the model parameters. A detailed visualization of the architecture is depicted in appendix A.1.

2.3 Automatic Tuning of the Number of Slots using Mean-Shift

Both models introduced in Section 2.1 and Section 2.2 still require a fixed number of slots beforehand. Therefore, it is essential to apply an unsupervised clustering mechanism to determine the number of slots conditioned on the input features while keeping the permutation symmetry. Consequently, we perform the mean-shift clustering algorithm (Carreira-Perpiñán, 2015) over the feature space to determine the cluster centers. Mean-shift is an iterative procedure to approximate different modes of a distribution using kernel density estimation. Each mode is represented as a cluster which does not need to be determined beforehand. In our model, we use a Gaussian kernel $k(x, y) = \exp(-\frac{1}{\sigma^2}||x - y||^2)$ for the density estimation. σ is a hyperparameter which affects the granularity of the modes. A shared mapping layer is utilized to initialize the slots based on each cluster respectively $\mathbf{z}_i = \text{MLP}_{\text{shared}}(\mathbf{c}_i)$ where $i \in \{1, ..., K\}$. Thus, it holds the permutation equivariance but requires to have the same number of slots as the number of the predicted cluster centers $K = M$. Since the Gaussian kernel is predefined by a hyperparameter, an expressive learned convolutional feature space is crucial to output distinctive modes. This is achieved by a fully differentiable pipeline including the fully differentiable implementations of the mean-shift algorithm.

3 Related Work

Object-centric slot representations. Slot representations have been widely used in static scenes (Locatello et al., 2020; Greff et al., 2019; Carion et al., 2020; Burgess et al., 2019; Engelcke et al.,

2020) and videos (Li et al., 2021b; Yang et al., 2021; Kipf et al., 2022; Veerapaneni et al., 2020; Weis et al., 2021). Each slot represents a corresponding object in the scene. This can be achieved either by accumulating the evidence over time to maintain the consistent object slot (Weis et al., 2021) if a variational auto-encoder (Kingma & Welling, 2014) is employed, or using softmax-based attention mechanism (Locatello et al., 2020; Bao et al., 2022). However, all of these approaches require a fixed set of slot variables. The set size needs to be strictly equal or larger than the number of objects in the scene, which limits the generalization on real-world applications since the number of objects is changing dynamically over time and cannot be determined in advance.

Scene decomposition. Most works formulate scene decomposition as compositional generative model (Greff et al., 2019; Eslami et al., 2016; von Kügelgen et al., 2020) or a mixture of components (Locatello et al., 2020; Burgess et al., 2019; Engelcke et al., 2020). Recently, some works (Stelzner et al., 2021; Yu et al., 2022; Bing et al., 2022) extend 2D scene decomposition to 3D with the advances of Neural Radiance Field (NeRF) (Mildenhall et al., 2020). Chen et al. (2020) and Li et al. (2020) infer 3D scenes from multiple reference images and textureless background. In contrast, uORF (Yu et al., 2022) infer from a single image and test on complex objects with diverse textured background.

Object discovery. Object discovery requires to differentiate the objects and background in an unsupervised way. These methods typically model objects as a set of latent embeddings (Carion et al., 2020) and adopt topic modelling (Russell et al., 2006), group image patches (Tuytelaars et al., 2009; Grauman & Darrell, 2006) or clustering-based deep learning algorithms (Li et al., 2019; Vo et al., 2020). Some methods (Zhao & Wu, 2019; Vo et al., 2021) also apply saliency detection and region proposals on the entire image to group and localize the objects.

Novel view synthesis. Novel view synthesis aims to generate novel views of the given scene from a single (Greff et al., 2019; Eslami et al., 2018; Yu et al., 2022) or multiple (Li et al., 2020; Mildenhall et al., 2020) source views. Liu et al. (2021) employ a token-transformation module to synthesize the novel views from a single image without requiring the pose information. Chen et al. (2021) extend GQN (Eslami et al., 2018) with a Spatial Transformation Routing (STR) mechanism without requiring explicit camera intrinsic information. Lochmann et al. (2016) enable the real-time novel view inference with the advantage of volume rendering. Recently, Cao et al. (2022) replace the expensive computation of volumetric sampling in NeRF-like methods by pixel-wise depth prediction and a differentiable point cloud renderer.

Deep clustering. Clustering helps analyze unstructured and high-dimensional data into meaningful and low-dimensional representations, which has been improved with deep learning techniques in recent years (Xie et al., 2016). Guo et al. (2017) propose an iterative optimization of learning low-dimensional representations from an auto-encoder by minimizing the Kullback-Leibler divergence between the pixel-wise features to each cluster center. Ghasedi Dizaji et al. (2017) extend it with a classifier on top which predicts the probability over the k classes where k is the number of cluster centers. Yang et al. (2017) employ the objective of k-means as clustering loss in the feature space of deep neural networks while Fard et al. (2020) relax the cluster assignment problem by using a soft-assignment which can fully benefit from the efficiency of stochastic gradient decent (SGD). Genevay et al. (2019) propose a fully differentiable version with the cluster parameters while Cai et al. (2022) reduce the computational time by introducing a subspace-based clustering and improve the scalability of deep clustering.

4 Experiments

To evaluate our method, we choose two object-centric tasks: object discovery in Section 4.1 and novel view synthesis in Section 4.2. We employ our idea on top of three state-of-the-art methods: Slot Attention (Locatello et al., 2020), IODINE (Greff et al., 2019) and uORF (Yu et al., 2022). We show more details about implementations in appendix A.1 and qualitative results in appendix A.2.

Baselines. In the object discovery task, we use Slot Attention and IODINE as baselines and build our method on top of them. Both baselines use slot representations but with different procedures to refine the slots: Slot Attention uses simple but effective softmax-based attention mechanism while IODINE considers slots as probabilistic latent variables and employs variational inference to accumulate the evidence during iterations. For the novel view synthesis task, we choose uORF as baseline which

Table 1: Quantitative results on the object discovery task.

	MDS					CLEVR6					Chairs				
Model	ARI ↑	LPIPS$_A$↓	LPIPS$_V$↓	PSNR ↑	SSIM ↑	ARI ↑	LPIPS$_A$↓	LPIPS$_V$↓	PSNR ↑	SSIM ↑	ARI ↑	LPIPS$_A$↓	LPIPS$_V$↓	PSNR ↑	SSIM ↑
SL	0.9671	0.0693	0.1351	27.43	0.9237	0.9815	0.0748	0.1486	32.11	0.8908	0.9982	0.3144	0.4362	24.49	0.6035
SL + kmeans (direct)	0.9223	0.1074	0.1606	26.13	0.9095	0.9963	0.0381	0.1097	34.22	0.9161	0.9963	0.2971	0.4273	24.17	0.6024
SL + kmeans	0.9837	0.0519	0.1149	28.88	0.9417	0.9970	0.0313	0.1032	34.98	0.9255	0.6271	0.2948	0.4274	24.31	0.6034
SL + kmeans (shared MLPs)	0.9043	0.1174	0.1672	25.84	0.9019	0.9989	0.0320	0.1041	34.82	0.9255	0.9974	0.3173	0.4297	25.01	0.6199
SL + PW	0.9605	0.0834	0.1526	26.25	0.9104	0.9937	0.0371	0.1056	34.04	0.9251	0.9523	0.3052	0.4363	24.82	0.6104
SL + MS (direct)	0.9893	0.0448	0.1059	31.39	0.9559	0.6114	0.1098	0.1957	29.43	0.8555	0.9999	0.2757	0.3997	26.02	0.6341
SL + MS	**0.9944**	**0.0393**	**0.0919**	**32.17**	**0.9613**	**1.0000**	**0.0306**	**0.1022**	**35.32**	**0.9301**	**1.0000**	**0.2693**	**0.3774**	**26.03**	**0.6444**
ID	0.9362	0.0504	0.0888	30.91	0.9591	0.8990	0.0224	0.0500	37.5	0.9661	0.2185	0.2757	0.3843	24.27	0.6299
ID + kmeans (direct)	0.9910	0.0193	0.0492	36.03	0.9833	0.8791	0.0254	0.0559	36.86	0.9619	0.6881	0.2666	0.3842	24.25	0.6322
ID + kmeans	0.9962	0.0166	0.0415	37.06	0.9861	0.8325	0.0198	0.0479	37.725	0.9667	0.7281	0.2559	0.3744	24.31	0.6314
ID + PW	0.9930	0.0207	0.0440	36.42	0.9834	0.9818	0.0190	0.0483	37.725	0.9667	0.8792	0.2192	0.3712	29.025	0.6362
ID + MS	**0.9970**	**0.0143**	**0.0401**	**38.12**	**0.9921**	**0.9909**	**0.0141**	**0.0361**	**38.90**	**0.9753**	**0.9991**	**0.1645**	**0.3219**	**31.07**	**0.6995**

uses softmax-based attention module to update slots and generate slot-based compositional scenes with Neural Radiance Field (NeRF). Note that all these models use random slot initialization. In addition, we also design two ablated models where the slot initialization is conditioned on the input features. First, we employ the k-means initialization directly as slot representations without any mapping layers in between (*direct* model). Second, we design a simple and permutation equivariant model using shared MLPs to map the k-means cluster centers of the input features to the slots (*shared MLPs* model).

Datasets. We use three datasets for the object discovery task: Multi-dSprites (MDS), CLEVR and Chairs datasets. Each dataset contains multiple objects in the scene. Similar as Slot Attention, we extract the CLEVR dataset to have maximum 4, 6, and 10 objects respectively and denote them as CLEVR4, CLEVR6 and CLEVR10. The Chairs dataset originates from uORF(Yu et al., 2022), which includes 4 chairs in each scene. The dataset includes 1200 different shapes of chairs sampled from ShapeNet (Chang et al., 2015) and 50 different floor textures as background. To train the Slot Atttention related models, we use 5k images for CLEVR4 and 10k for MDS, CLEVR6 and Chairs. To train the IODINE related models, we use the same datasets except 13.9k images for MDS. Each dataset contains another 500 images for evaluation. For the novel view synthesis task: We only use the Chairs dataset but it includes 5k scenes for training and 500 scenes for testing, where each scene includes 4 images with different camera viewpoints. Therefore, there are in total 20k images for training and 2k images for testing.

Metrics. As prior works (Greff et al., 2019; Locatello et al., 2020; Burgess et al., 2019), for the object discovery task, we adapt the Adjusted Rand Index (ARI) score to be evaluated only on the pixels of the foreground objects and evaluate the predicted segmentation with the groundtruth mask. For the novel view synthesis, we follow uORF and adopt ARI on the fully reconstructed image, the foreground regions (Fg-ARI) and the synthesized novel view images (NV-ARI). Furthermore, we use LPIPS (Zhang et al., 2018), SSIM (Wang et al., 2004) and PSNR (Horé & Ziou, 2010) as perceptual metrics for both tasks.

4.1 Object Discovery

Training. We follow the same training setup of Slot Attention and IODINE. We use Adam optimizer (Kingma & Ba, 2015) with a learning rate of 4×10^{-4} for Slot Attention based models and 3×10^{-4} for IODINE related models. We train the Slot Attention related models with 2 NVIDIA Tesla V100-32GB GPUs and a batch size of 32 on each GPU. For IODINE related models, we use 4 GPUs since IODINE requires more computation and gpu memory. We train each model for 1000 epochs with a warm-up training strategy (Goyal et al., 2017) and an exponential learning rate decay. We use $K = 5$ for MDS, CLEVR4 and Chairs datasets since there are maximum 4 objects in each scene, and $K = 7$ for CLEVR6. The cluster number is set to $M = 2K$ except for the *mean-shift*, *direct* and *shared MLPs* versions which require $M = K$.

Results. Quantitative results are shown in Table 1 and qualitative results in Figure 2. In general, learning inductive slot initialization from input features improve the performance on both baselines, where *mean-shift* models achieve the best performance consistently over all datasets. **Well-recovered details:** Surprisingly, all our IODINE-based variants achieve higher resolution even than the groundtruth image for MDS dataset, while the original IODINE is struggled with the data prior and cannot reconstruct the shape of objects. Furthermore, in Figure 2, we observe that only the

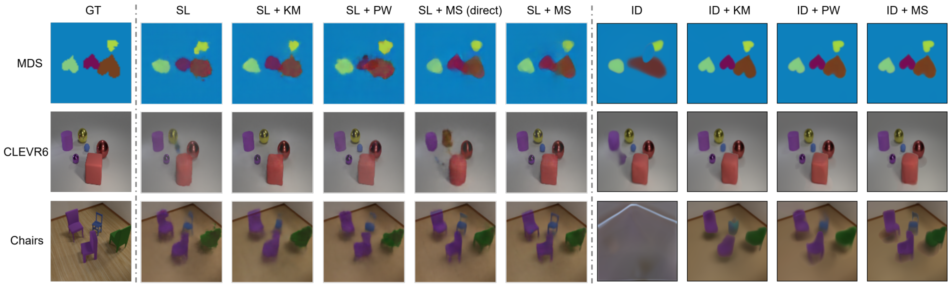

Figure 2: Qualitative results on the object discovery task. We provide the groundtruth image (GT) and the reconstructed scenes using different methods. Notably, our models especially the *mean-shift* (MS) versions can recover detailed appearance over all datasets with even better quality than original input for IODINE-based models in MDS dataset.

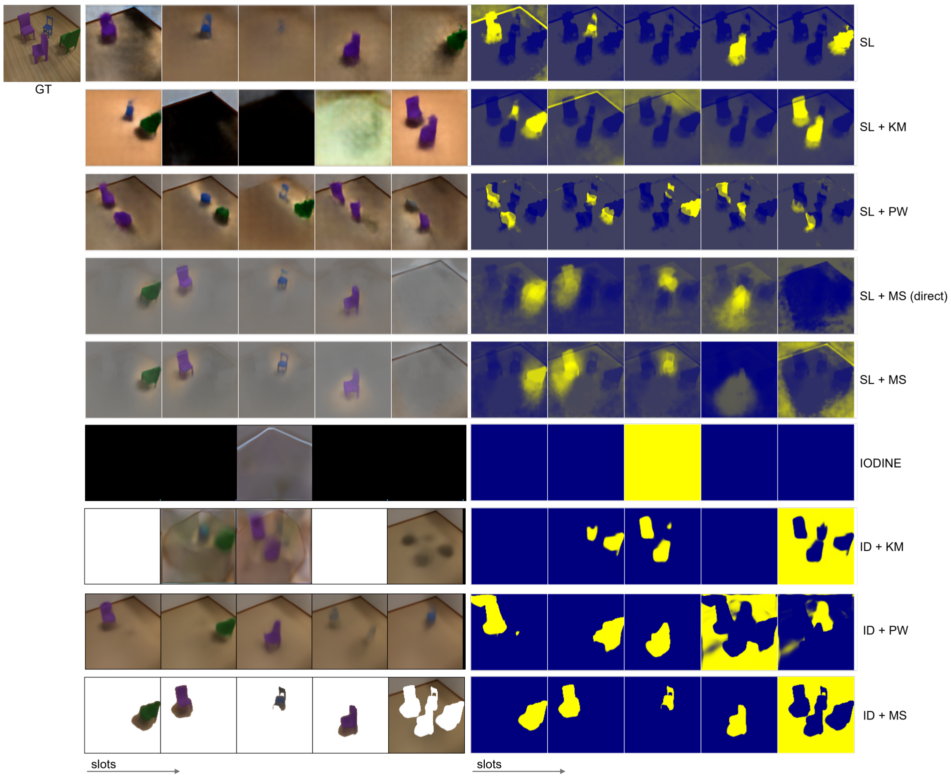

Figure 3: Qualitative results of slot-wise reconstructed scenes (left) and masks (right). *Mean-shift* models disentangle the objects better than others and recover more details.

mean-shift models can capture the details of objects for Slot Attention based models. For example, it captures the “heart” objects in MDS while others struggle with the data prior. In particular, our models (especially for *mean-shift models*) can reconstruct the appearance in very good details, e.g., the small blue sphere in CLEVR6 and the legs and rims of various chairs in Chairs dataset. **Slots disentanglement:** We also visualize the slot-wise reconstructed scenes and masks in Figure 3. From the masks, we observe that only the *mean-shift* models can fully disentangle the objects and background where the highlighted area indicates large attention. In contrast, original Slot Attention mixes the background and a chair in slot #1 while IODINE cannot even work with textured background. *Pseudoweights* and *k-means* models also entangle the chairs into one slot even though the overall reconstructed performance is still better than the baselines (Table 1 and Figure 2). The slot-wise reconstructed scenes also reveal our conclusion that *mean-shift* models contain more ap-

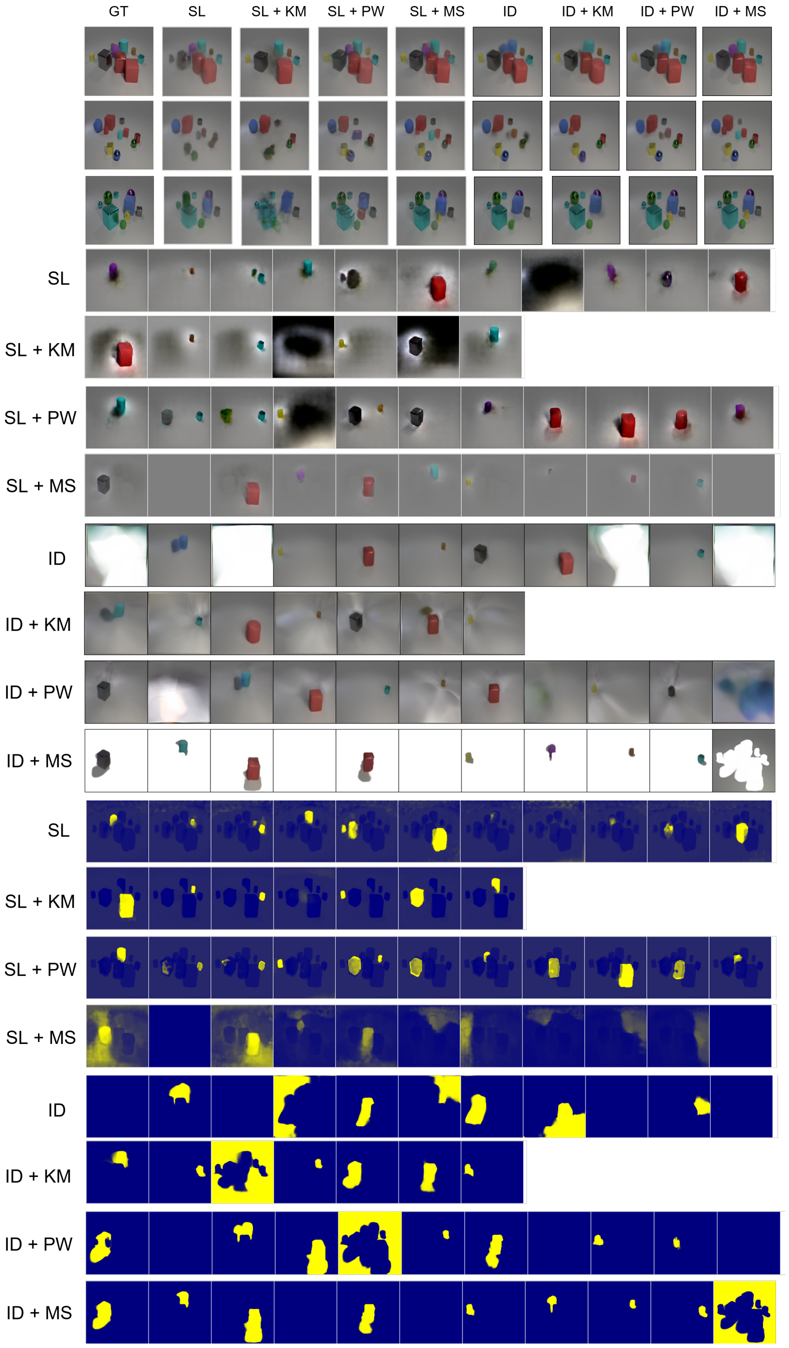

Figure 4: Qualitative results on increasing objects. The models are trained with CLEVR6 but evaluated on CLEVR10 with larger number of objects. We demonstrate three examples in the first 3 rows follwed by the slot-wise reconstructed scenes and masks of the first example.

pearance details with fully disentangled slots. **Mapping between clusters and slots:** Furthermore, our ablation studies demonstrate that the *k-means* models using non-linear mapping layers between the clusters and slots gain additional benefits compared to the *direct* models (in Table 1). Additionally, the permutation equivariant model (*shared MLPs*) performs better than the non-permutation symmetric model (*k-means*) on CLEVR6 and Chairs datasets, indicating the benefits of permutation symmetry on complex scenes, though it is not as good as the *mean-shift* models especially on MDS dataset. **Generalization on increasing objects:** In addition, we evaluate the generalization on more

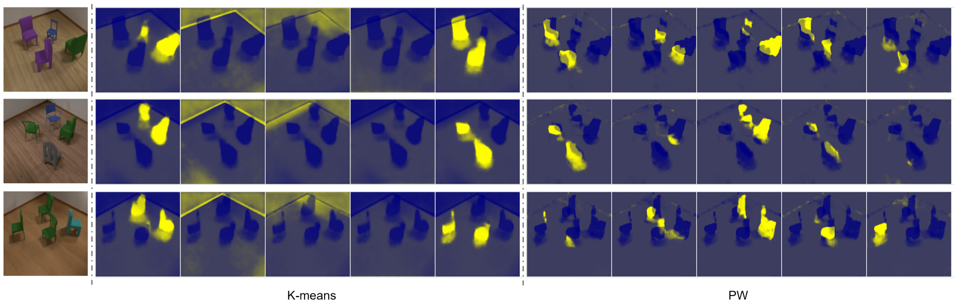

Figure 5: Failure cases on Chairs dataset where *k-means* and *Psuedoweights* (PW) cannot disentangle the objects and use each individual slot for specific areas.

Table 2: Evaluation with different number of iterations (5 iterations are used for training). In particular, our models achieve significant improvement already at the first iteration.

	Iter 1				Iter 3				Iter 7			
Model	LPIPS$_A$↓	LPIPS$_V$↓	PSNR ↑	SSIM ↑	LPIPS$_A$↓	LPIPS$_V$↓	PSNR ↑	SSIM ↑	LPIPS$_A$↓	LPIPS$_V$↓	PSNR ↑	SSIM ↑
ID	0.4415	0.6071	12.72	0.3820	0.4477	0.5804	16.33	0.4908	0.4363	0.5646	19.53	0.5001
ID + kmeans	0.2108	0.3768	27.05	0.6202	0.1956	0.3607	28.75	0.6533	0.1884	0.3545	29.33	0.6656
ID + PW	0.2269	0.3734	27.57	0.6297	0.1973	0.3531	29.33	0.6642	0.1885	0.3461	29.92	0.6768
ID + MS	**0.1798**	**0.3545**	**28.39**	**0.6467**	**0.1602**	**0.3343**	**30.16**	**0.6828**	**0.1528**	**0.3273**	**30.68**	**0.6951**

objects and slots (CLEVR10) while the models are trained on CLEVR6. The qualitative results are shown in Figure 4. We observe that the original baselines struggle with closed or overlapped objects by missing, mixing or predicting wrong color of objects, while our models (especially the *mean-shift* models) can detect the overlapped objects perfectly without missing any object even for extremely difficult scenes. For example, i) the *mean-shift* Slot Attention model (SL + MS) can detect the brown cylinder behind the purple sphere in the first example even though it is hardly visible, ii) both *mean-shift* models (SL + MS and ID + MS) and *k-means* IODINE (ID + KM) can detect the red small cylinder in front of the red cube in the second example, though the objects are overlapped and with the same color, and iii) both *mean-shift* models and *Pseudoweights* IODINE (ID + PW) can reconstruct the yellow cylinder in the third example. We believe the benefits come from the inductive slot initialization conditioning on the perceptual input features, which gives expressive slot representations used in the following slot refinement. Note that *k-means* models can merely detect 6 objects from the scene since the slot number is by design not scalable. **Generalization on increasing iterations:** Furthermore, Table 2 shows the evaluation with increasing number of iterations up to 7 while the models are trained with 5 iterations. All models are capable of generalizing on more iterations with performance gains. In particular, using inductive slot initialization enables notable improvement at the first iteration, which indicates the efficiency of the learned inductive slot initialization. **Failure cases:** We further investigate the cases when *k-means* and *Pseudoweights* are failed to disentangle objects in Chairs dataset. Examples are shown in Figure 5. Interestingly, we find they learned structured slot representations not always based on the objects. The slot representations of *k-means* model are not generalize due to the non-permutation symmetry. Thus, it always uses the same slot to represent specific area, e.g., the first slot to represent the objects in the top right area, the second and third slots for walls. On the other hand, *Pseudoweights* outputs the same slot representations while changing the object positions due to the permutation invariance. As a result, it neglects the object-centric spatial features in the scene. Thus, the model tends to reconstruct the scenes by assigning fixed spatial area to each individual slot. Such undesirable behaviors occur especially on Chairs dataset where each scene includes 4 images with changing viewpoints. In contrast, a good permutation equivariant model such as *mean-shift* can alleviate this issue and decouple the objects (as shown in Figure 3).

4.2 Novel View Synthesis

Setup. The Chairs dataset contains 4 images from different viewpoints of each scene. During training, we randomly pick one image from each scene as input and reconstruct the images for the

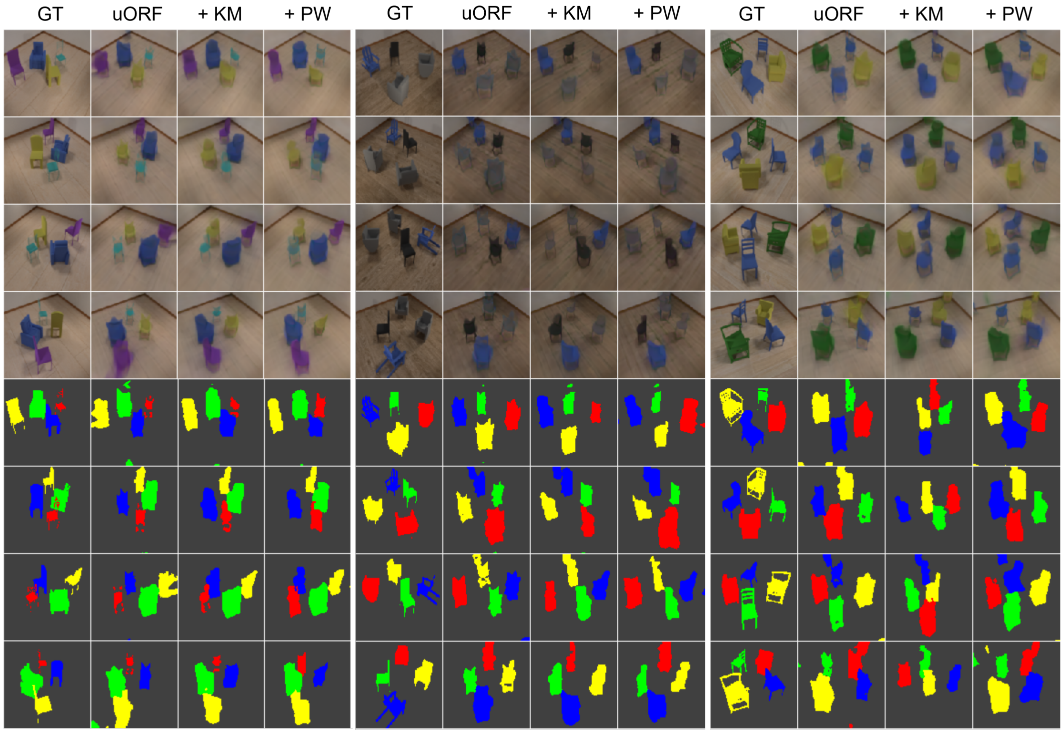

Figure 6: Qualitative results on novel view synthesis. Our models can represent the chairs with more details than the original uORF.

Table 3: Results of novel view synthesis on Chairs-diverse.

Model	ARI ↑	Fg-ARI ↑	NV-ARI ↑	LPIPS ↓	SSIM ↑	PSNR ↑
uORF	0.4974	0.5347	0.4291	0.2417	0.6862	24.9712
uORF + kmeans	**0.651**	**0.7346**	**0.5304**	**0.1894**	**0.7176**	**26.1833**
uORF + PW	0.5784	0.6943	0.4773	0.221	0.703	25.6277

other 3 viewpoints. We use the same training loss functions and strategies as uORF (Yu et al., 2022). uORF is a memory-extensive model which only works with a batch size of 1 on NVIDIA Tesla V100-32GB. Meanwhile, mean-shift also consumes large memory for the intermediate tensors due to its iterative optimizations. Therefore, we cannot build a mean-shift algorithm on top of uORF with our available hardware. We consider this as a limitation of our *mean-shift* model.

Results. We show quantitative results in Table 3 and qualitative results in Figure 6 (more results in appendix A.2). Overall, our models outperform the original uORF consistently over all metrics. In particular, our models can better reconstruct the chairs pointed to the right direction while original uORF cannot build a clear shape for most chairs.

5 Conclusion

We propose to learn an inductive slot initialization from the input instead of using a random initialization which is widely used in the prior works for the slot-based methods. To evaluate the importance of permutation symmetry over slots, we design various models with non-permutation symmetry, permutation invariance and permutation equivariance into consideration. In particular, our proposed permutation equivariant mean-shift model enables additional flexibility without requiring a fixed number of slots in advance, while it achieves notable improvements on the reconstructed perception details.

Reproducibility Statement

We have provided the code repository for training and evaluation with the links to download the required datasets and pre-trained models. We have also provided the details of network architectures and implementations in Appendix. The researchers can use the provided repository to reproduce the results or evaluate on their own datasets.

References

Zhipeng Bao, Pavel Tokmakov, Allan Jabri, Yu-Xiong Wang, Adrien Gaidon, and Martial Hebert. Discovering objects that can move. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11789–11798, June 2022.

Wang Bing, Lu Chen, and Bo Yang. Dm-nerf: 3d scene geometry decomposition and manipulation from 2d images. *arXiv preprint arXiv:2208.07227*, 2022.

Christopher P. Burgess, Loïc Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew M. Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *ArXiv*, abs/1901.11390, 2019.

Jinyu Cai, Jicong Fan, Wenzhong Guo, Shiping Wang, Yunhe Zhang, and Zhao Zhang. Efficient deep embedded subspace clustering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–10, June 2022.

Ang Cao, Chris Rockwell, and Justin Johnson. Fwd: Real-time novel view synthesis with forward warping and depth. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15713–15724, June 2022.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*, pp. 213–229, 2020.

Miguel Á. Carreira-Perpiñán. A review of mean-shift algorithms for clustering. *ArXiv*, abs/1503.00687, 2015.

Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, L. Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015.

Chang Chen, Fei Deng, and Sungjin Ahn. Learning to infer 3d object models from images. *ArXiv*, abs/2006.06130, 2020.

Wen-Cheng Chen, Min-Chun Hu, and Chu-Song Chen. Str-gqn: Scene representation and rendering for unknown cameras based on spatial transformation routing. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5946–5955, 2021.

Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *International Conference on Learning Representations (ICLR)*, 2020.

S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, koray kavukcuoglu, and Geoffrey E Hinton. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

S. M. Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S. Morcos, Marta Garnelo, Avraham Ruderman, Andrei A. Rusu, Ivo Danihelka, Karol Gregor, David P. Reichert, Lars Buesing, Théophane Weber, Oriol Vinyals, Dan Rosenbaum, Neil C. Rabinowitz, Helen King, Chloe Hillier, Matthew M. Botvinick, Daan Wierstra, Koray Kavukcuoglu, and Demis Hassabis. Neural scene representation and rendering. *Science*, 360:1204 – 1210, 2018.

Maziar Moradi Fard, Thibaut Thonet, and Éric Gaussier. Deep k-means: Jointly clustering with k-means and learning representations. *ArXiv*, abs/1806.10069, 2020.

Aude Genevay, Gabriel Dulac-Arnold, and Jean-Philippe Vert. Differentiable deep clustering with cluster size constraints. *ArXiv*, abs/1910.09036, 2019.

Kamran Ghasedi Dizaji, Amirhossein Herandi, Cheng Deng, Weidong Cai, and Heng Huang. Deep clustering via joint convolutional autoencoder embedding and relative entropy minimization. In *International Conference on Computer Vision (ICCV)*, Oct 2017.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *ArXiv*, abs/1706.02677, 2017.

Kristen Grauman and Trevor Darrell. Unsupervised learning of categories from sets of partially matching image features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:19–25, 2006.

Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pp. 2424–2433, 2019.

Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1753–1759, 2017.

Alain Horé and Djemel Ziou. Image quality metrics: Psnr vs. ssim. *International Conference on Pattern Recognition*, pp. 2366–2369, 2010.

Rishabh Kabra, Daniel Zoran, Goker Erdogan, Loic Matthey, Antonia Creswell, Matt Botvinick, Alexander Lerchner, and Chris Burgess. Simone: View-invariant, temporally-abstracted object representations via unsupervised video decomposition. In *Advances in Neural Information Processing Systems*, volume 34, pp. 20146–20159, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations (ICLR)*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014.

Thomas Kipf, Gamaleldin F. Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-Centric Learning from Video. In *International Conference on Learning Representations (ICLR)*, 2022.

B. Li, Zhengxing Sun, Qian Li, Yunjie Wu, and Anqi Hu. Group-wise deep object co-segmentation with co-attention recurrent neural network. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8518–8527, 2019.

Liangzhi Li, Bowen Wang, Manisha Verma, Yuta Nakashima, Ryo Kawasaki, and Hajime Nagahara. Scouter: Slot attention-based classifier for explainable image recognition. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1026–1035, 2021a.

Nanbo Li, Cian Eastwood, and Robert Fisher. Learning object-centric representations of multi-object scenes from multiple views. In *Advances in Neural Information Processing Systems*, volume 33, pp. 5656–5666, 2020.

Nanbo Li, Muhammad Ahmed Raza, Wenbin Hu, Zhaole Sun, and Robert Fisher. Object-centric representation learning with generative spatial-temporal factorization. In *Advances in Neural Information Processing Systems*, volume 34, pp. 10772–10783, 2021b.

Bingzheng Liu, Jianjun Lei, Bo Peng, Chuanbo Yu, Wanqing Li, and Nam Ling. Novel view synthesis from a single image via unsupervised learning. *ArXiv*, abs/2110.15569, 2021.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pp. 11525–11538, 2020.

Gerrit Lochmann, Bernhard Reinert, Arend Buchacher, and Tobias Ritschel. Real-time Novel-view Synthesis for Volume Rendering Using a Piecewise-analytic Representation. In *Vision, Modeling and Visualization (VMV)*, 2016.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020.

Bryan C. Russell, William T. Freeman, Alexei A. Efros, Josef Sivic, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2:1605–1614, 2006.

Karl Stelzner, Kristian Kersting, and Adam R. Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *ArXiv*, abs/2104.01148, 2021.

Tinne Tuytelaars, Christoph H. Lampert, Matthew B. Blaschko, and Wray L. Buntine. Unsupervised object discovery: A comparison. *International Journal of Computer Vision*, 88:284–302, 2009.

Rishi Veerapaneni, John D. Co-Reyes, Michael Chang, Michael Janner, Chelsea Finn, Jiajun Wu, Joshua Tenenbaum, and Sergey Levine. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning*, pp. 1439–1456, 2020.

Huy V. Vo, Patrick P'erez, and Jean Ponce. Toward unsupervised, multi-object discovery in large-scale image collections. In *European Conference on Computer Vision (ECCV)*, 2020.

Van Huy Vo, Elena Sizikova, Cordelia Schmid, Patrick Pérez, and Jean Ponce. Large-scale unsupervised object discovery. In *Advances in Neural Information Processing Systems*, volume 34, pp. 16764–16778, 2021.

Julius von Kügelgen, Ivan Ustyuzhaninov, Peter V. Gehler, Matthias Bethge, and Bernhard Schölkopf. Towards causal generative scene models via competition of experts. *International Conference on Learning Representations (ICLR) Workshop: "Causal learning for decision making"*, 2020.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Marissa A. Weis, Kashyap Chitta, Yash Sharma, Wieland Brendel, Matthias Bethge, Andreas Geiger, and Alexander S. Ecker. Benchmarking unsupervised object representations for video sequences. *Journal of Machine Learning Research*, 22(183):1–61, 2021.

Yizhe Wu, Oiwi Parker Jones, Martin Engelcke, and Ingmar Posner. Apex: Unsupervised, o'bject-centric scene segmentation and tracking for robot manipulation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3375–3382, 2021.

Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning (ICML)*, pp. 478–487, 2016.

Bo Yang, Xiao Fu, Nicholas D. Sidiropoulos, and Mingyi Hong. Towards k-means-friendly spaces: Simultaneous deep learning and clustering. In *International Conference on Machine Learning (ICML)*, pp. 3861–3870, 2017.

Charig Yang, Hala Lamdouar, Erika Lu, Andrew Zisserman, and Weidi Xie. Self-supervised video object segmentation by motion grouping. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7177–7188, 2021.

Hong-Xing Yu, Leonidas J. Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. In *International Conference on Learning Representations (ICLR)*, 2022.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Ting Zhao and Xiangqian Wu. Pyramid feature attention network for saliency detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

A APPENDIX

A.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 7 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 7. 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.

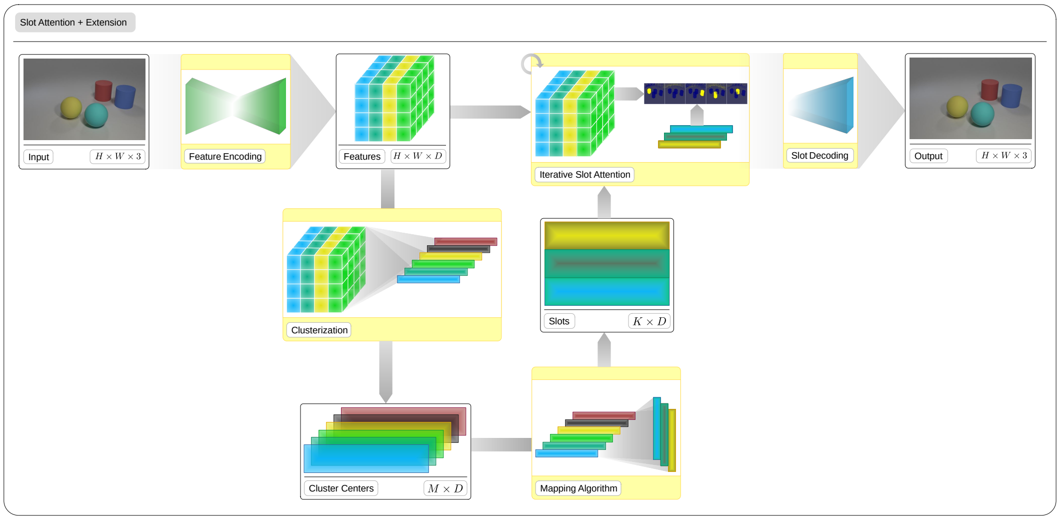

Figure 7: The framework architecture for slot initialization for slto attention. The top row is the original architecture.

IODINE: The extension for IODINE resemble the same structure as in the slot attention architecture in Figure 8. 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 9 directly injects the cluster centers determined by the clusterization algorithms into the slots. 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.

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 10. 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.

Large MLPs: This network maps between a different number of cluster centers and slots, as provided in Figure 11. The reason for this is to increase the sampling amount of cluster centers from the

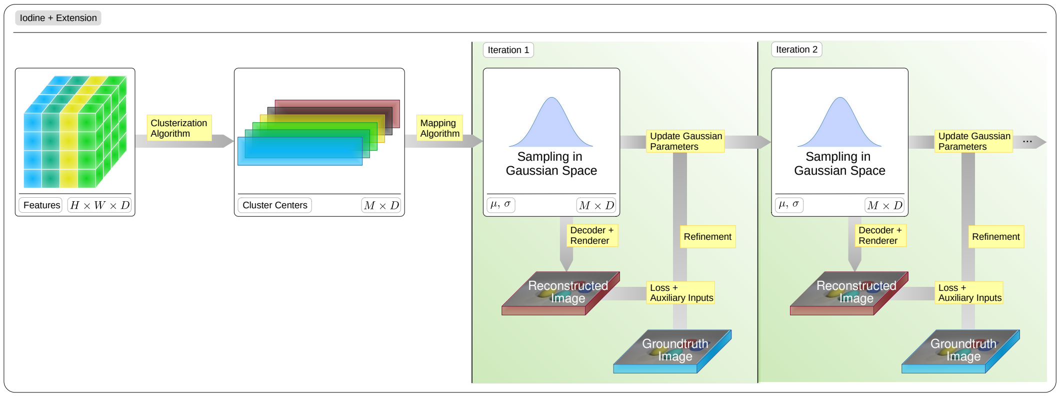

Figure 8: The framework architecture for IODINE based extensions. The original starts directly at iteration 1 with slots drawn out of the standard gaussian distribution with $(\mu, \sigma) = (0, 1)$.

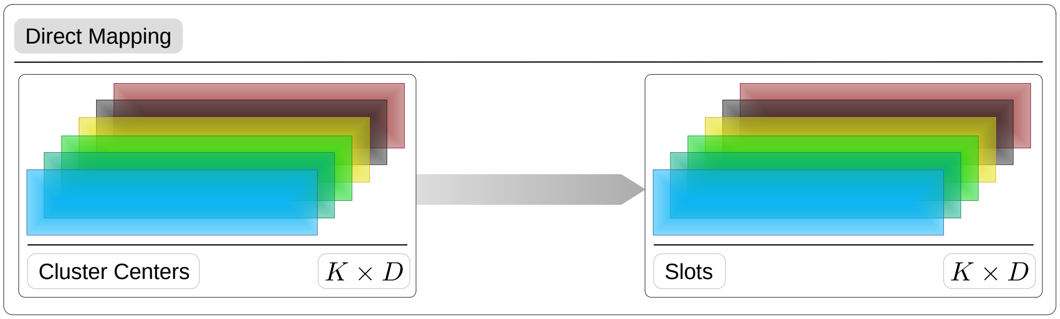

Figure 9: The Direct mapping approach. Slots are identical to the cluster centers chosen by the clusterization algorithm.

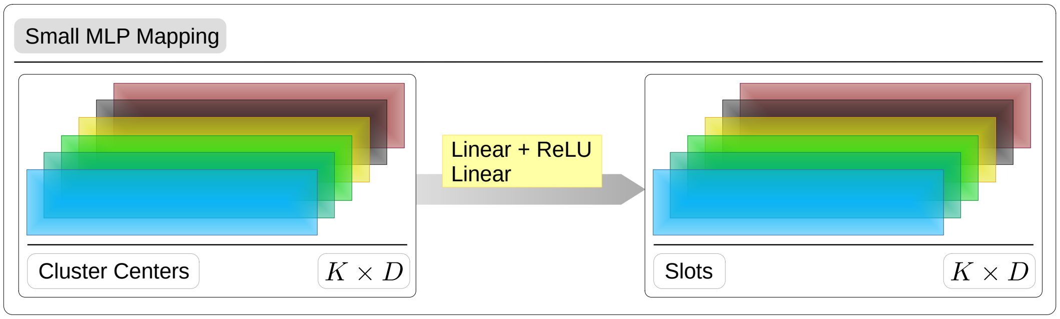

Figure 10: The Small MLPs mapping approach. It extends the direct mapping approach by a non-linear network between cluster centers and slots.

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.

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

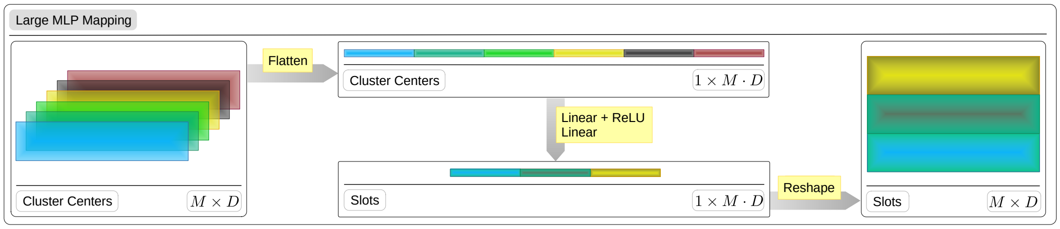

Figure 11: The Large MLPs approach.

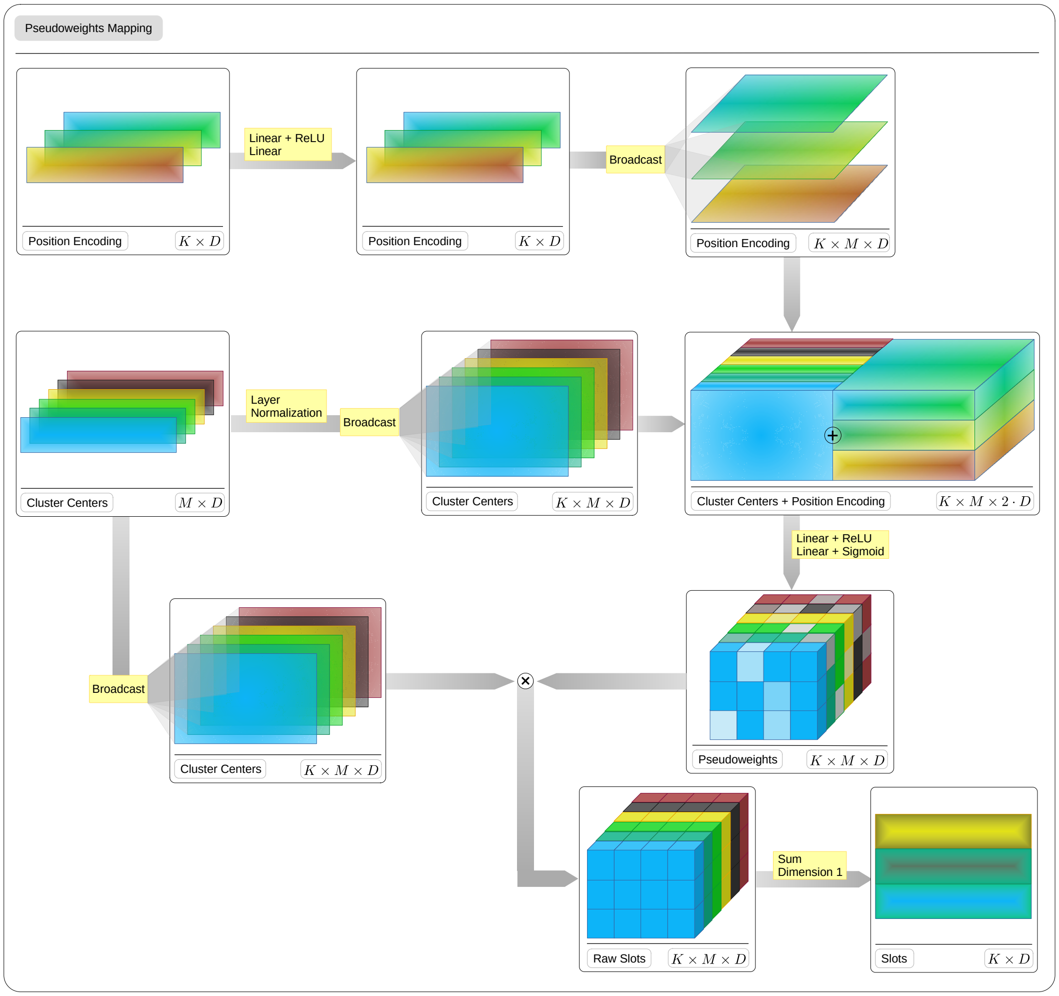

Figure 12: The permutation invariant Pseudoweights mapping.

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

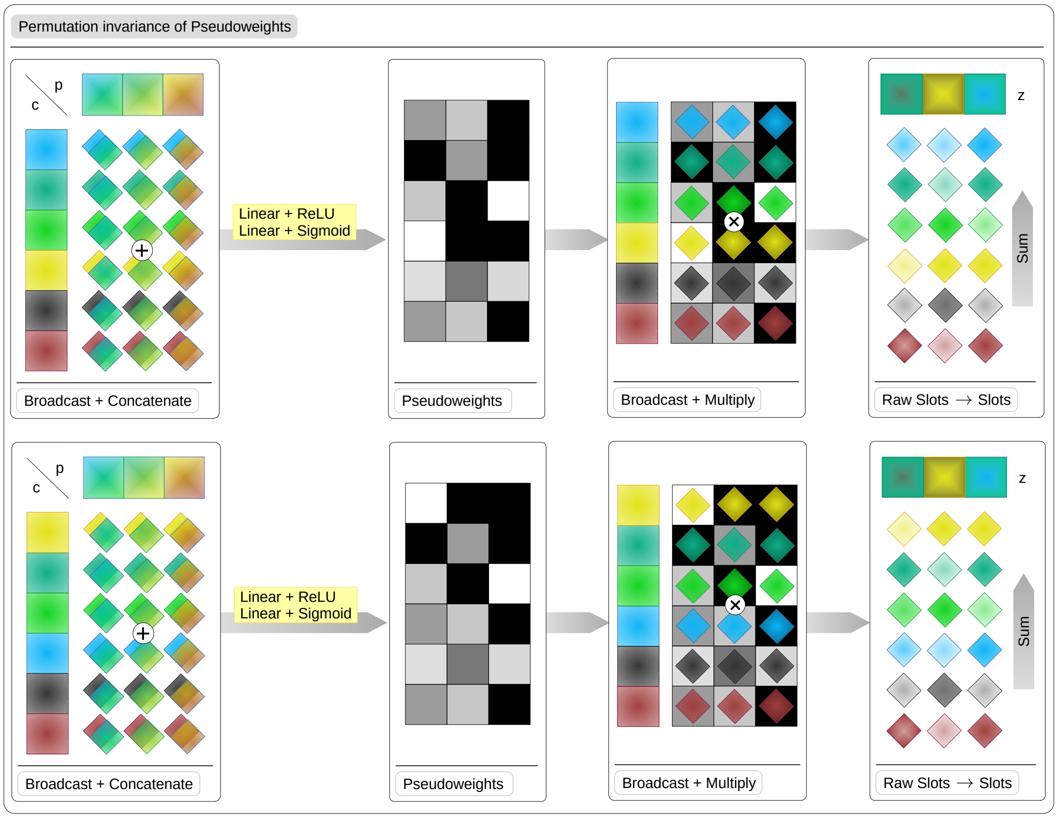

Figure 13: The permutation invariant mapping between 6 cluster centers and 3 slots. For this example all slots and cluster centers are of dimension D=1, to keep it simple. The pseudoweights tensor has high values in black squares and low values in white squares. If the blue and yellow slot change their position, the slots want change their initialization.

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 13.

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 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 A.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 A.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 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($\{z_n\}_{n=1}^{N}, \epsilon$)
```

A.2 Visualizations on object discovery task

We show some qualitative evaluation examples here for the object discovery task. We put more results in our repository `https://github.com/slot-initialization/linosic`.

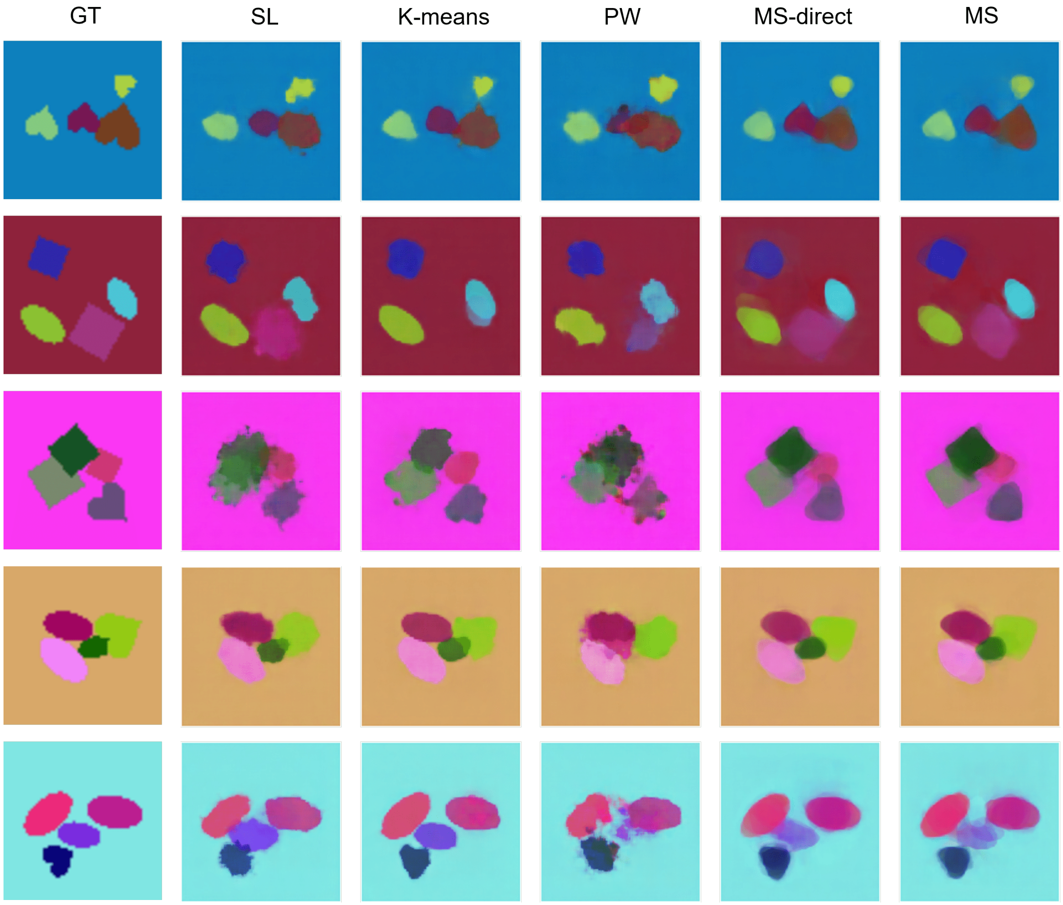

Figure 14: Qualitative results on MDS dataset.

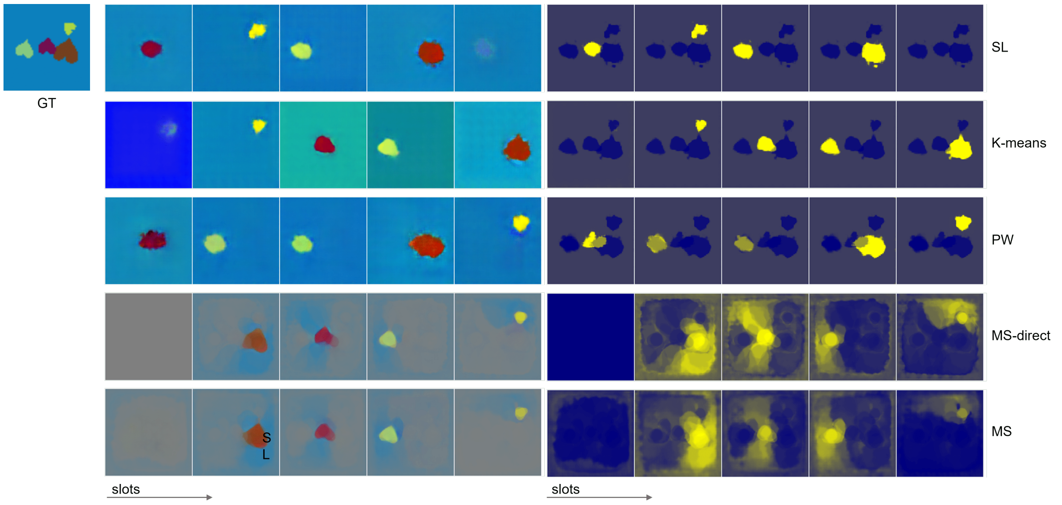

Figure 15: The slot-wise predicted masks and reconstructed scenes on MDS dataset.

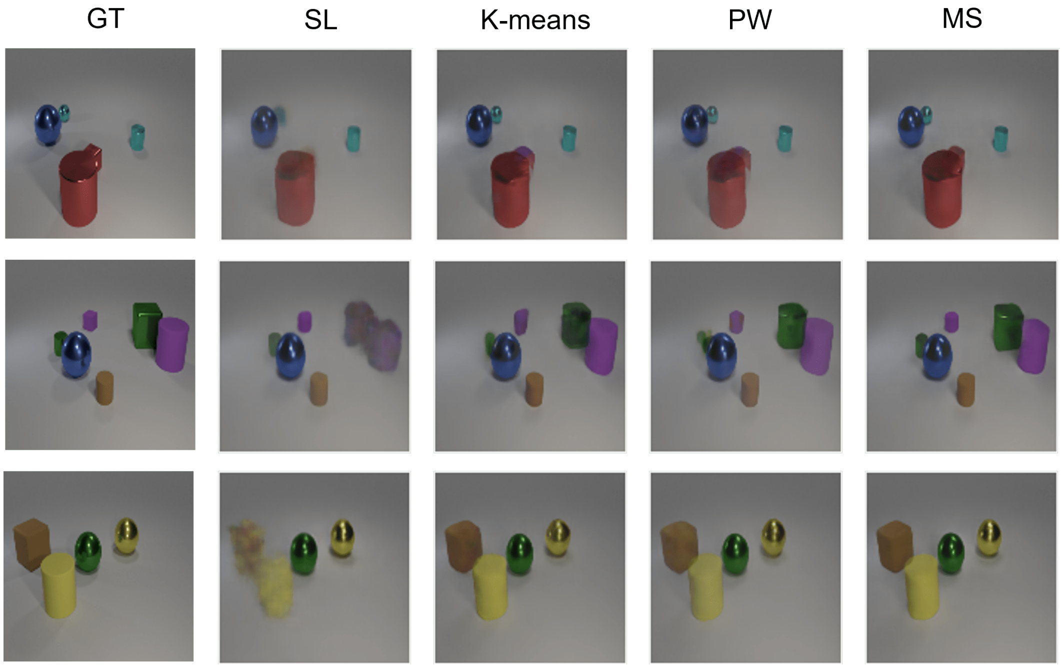

Figure 16: The original Slot Attention model struggles with overlapped objects.

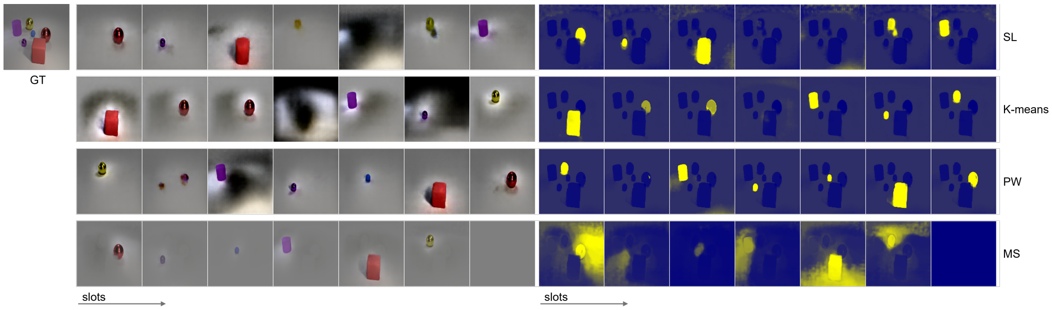

Figure 17: The slot-wise predicted masks and reconstructed scenes on CLEVR6 dataset.

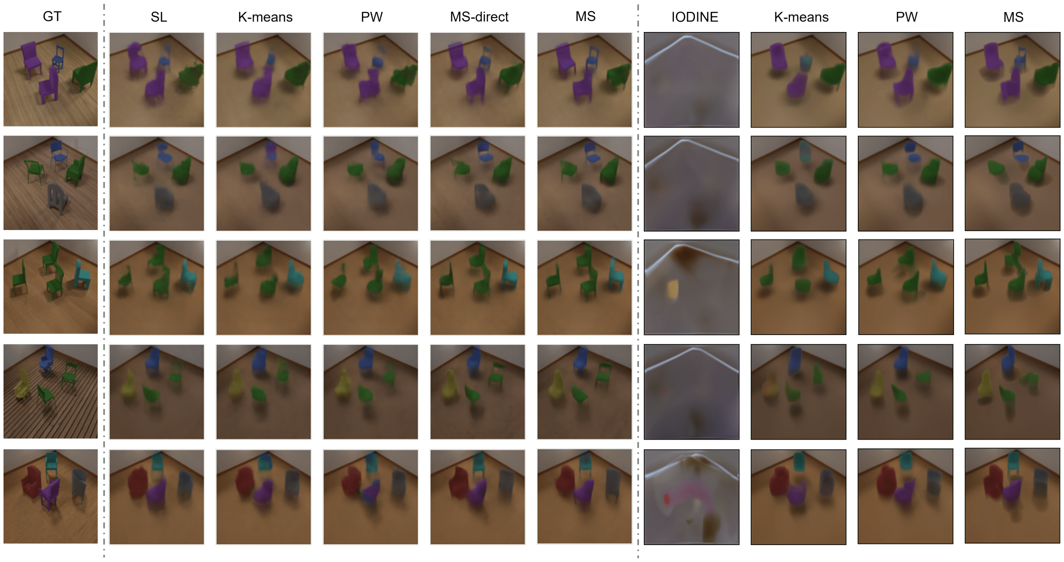

Figure 18: Qualitative results on Chairs dataset.

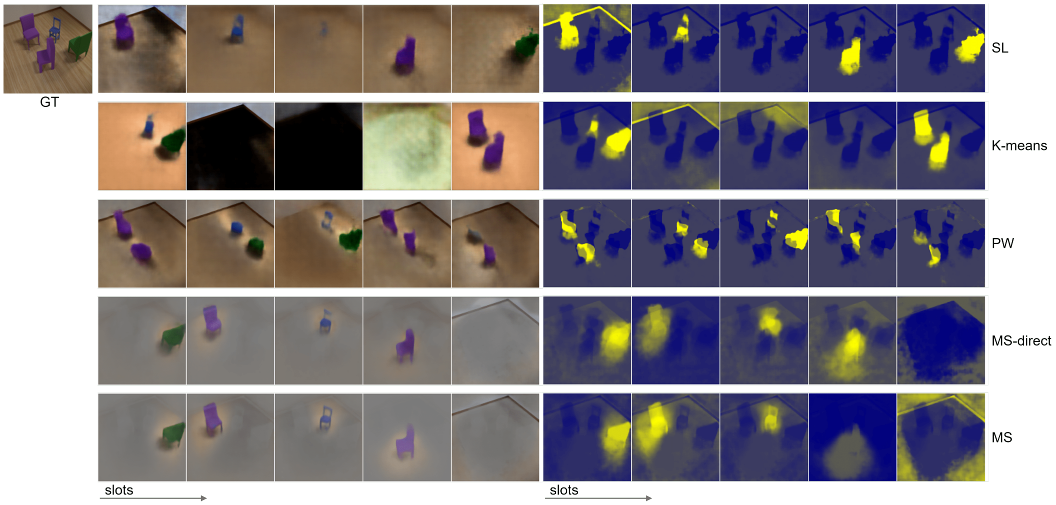

Figure 19: The slot-wise predicted masks and reconstructed scenes on Chairs dataset.

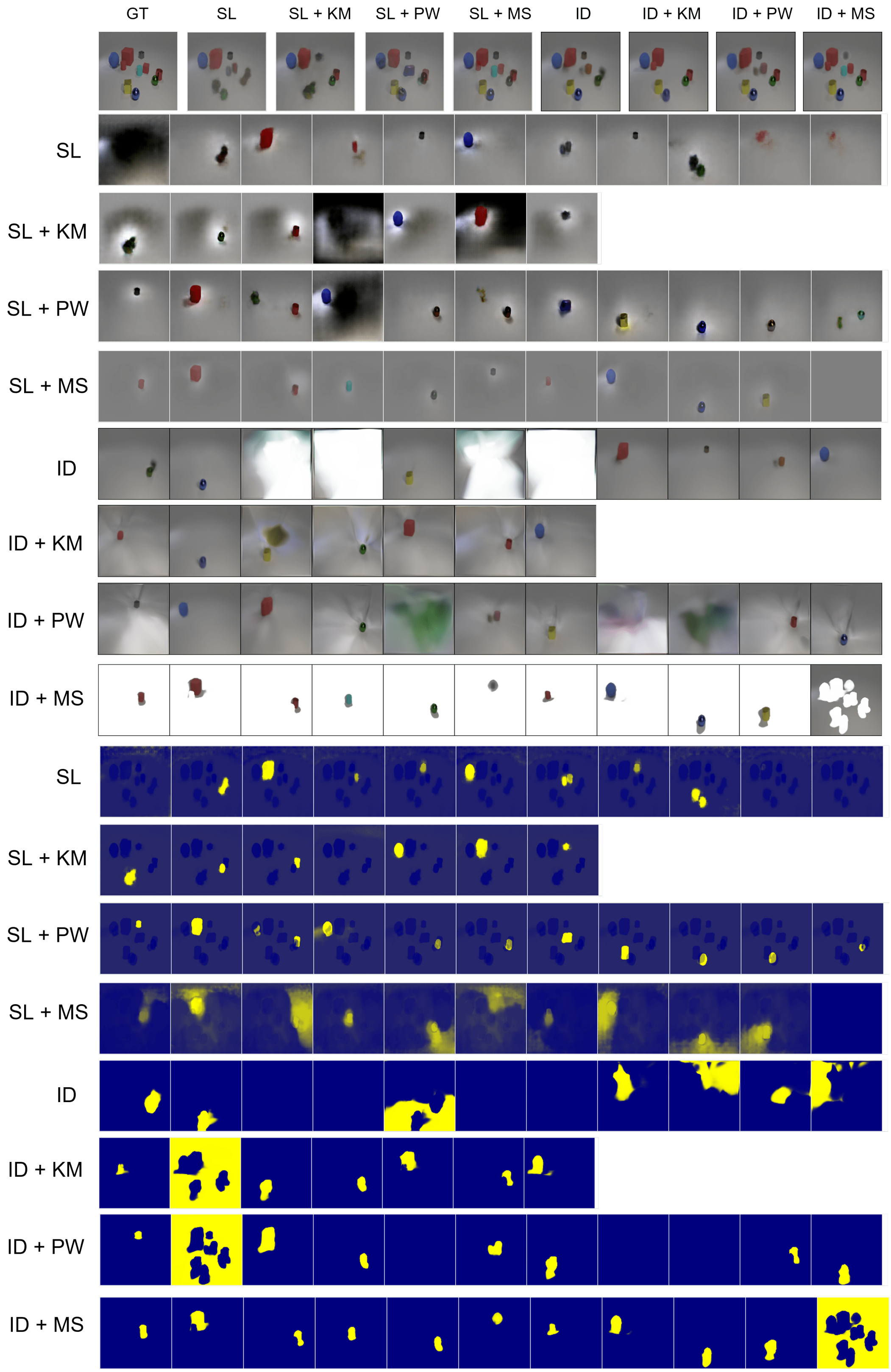

Figure 20: Qualitative comparison of generalization on CLEVR10 while the models are trained with CLEVR6.

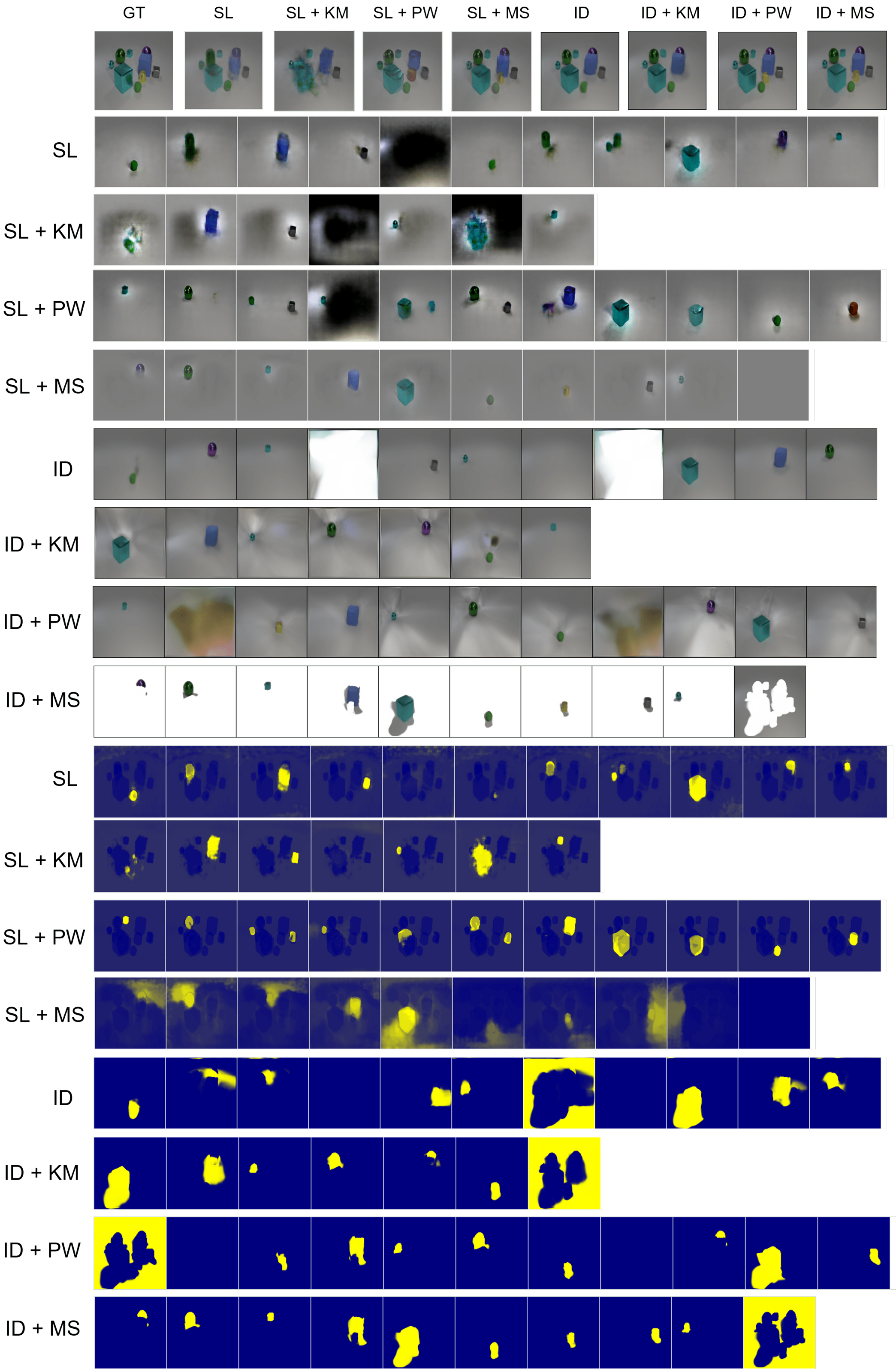

Figure 21: Another qualitative comparison of generalization on CLEVR10.