Appendix of *Adaptive Slot Attention: Object Discovery with Dynamic Slot Number*

1. More Implementation Details for MOVi-C/E and COCO

Vision backbones. We utilize the Vision Transformer backbone and leverage the pre-trained DINO weights available in the timm [8] library. Our specific configuration entails using ViT-B/16, which consists of 12 Transformer blocks. These blocks have a token dimensionality of 768, with a head number of 12 and a patch size of 16. In our pipeline, we take the output of the final block as the input of slot attention module and the reconstruction target.

Slot Attention. We adopt the slot attention bottleneck methodology based on the original work [4] for our implementation. The slot initialization process involves sampling from a shared learnable normal distribution $\mathcal{N}(\mu, \Sigma)$. Throughout all the experiments, we iterate the slot attention mechanism with 3 steps. The slot dimension is set to 128 for MOVi-C/E and 256 for COCO datasets. For the feed-forward network in Slot Attention, we utilize a two-layer MLP (Multi-Layer Perceptron). The hidden dimension of this MLP is set to 4 times the slot dimension.

Light Weight Network for Probability Prediction. We utilize a two-layer MLP for the probability prediction. The hidden dimension of this MLP is set to 4 times the slot dimension, and the output dimension is set to 2.

Decoder. We utilize a four-layer MLP with ReLU activations in our approach. The output dimensionality of the MLP is $D_{feat} + 1$, where D_{feat} represents the dimension of the feature, and the last dimension is specifically allocated for the alpha mask. The MLP's hidden layer sizes differ based on the dataset used. For the MOVi datasets, we employ hidden layer sizes of 1024. On the other hand, for COCO, we utilize hidden layer sizes of 2048.

Optimizer. In our main experiments, we train our models for 500k steps. Our model is initialized from a fixed K_{max} slot model trained for 200k steps. To optimize the model's parameters, we employ the Adam optimizer with a learning rate of $4e-4$. The β_0 and β_1 parameters are set to their default values $\beta_0 = 0.9, \beta_1 = 0.999$. For the ablation studies such as the necessity of Gumbel-Softmax and designs of masked slot decoders, we train our models for 200k steps.

To enhance the learning process, we incorporate a learning rate decay schedule with a linear learning rate warm-up of 10k steps. The learning rate follows an exponentially decaying pattern, with a decay half-life of 100k steps. Furthermore, we apply gradient norm clipping, limiting it to a maximum of 1.0, which aids in stabilizing the training procedure. The training of the models takes place on 8 NVIDIA T4 GPUs, with a local batch size of 8.

2. More Implementation Details for CLEVR10

For the experiment on toy dataset CLEVR10, we do pixel-level reconstruction instead of feature reconstruction. We utilized the CNN feature encoder and boardcast decoder in [4]. We set the slot dimension to 64, and set the hidden dimension to 128 for the feed-forward network in slot attention. The other setting closely follow the experiments on MOVi-C/E and COCO.

3. Detailed Results on Toy Dataset

In Tab. 1, we quantitatively compare our model with several fixed-slot models on the toy dataset CLEVR10 under pixel reconstruction setting. Moreover, we provide qualitative comparison among our model, 6-slot model, and 11-slot model in Fig. 1. The 11-slot model often assign one or more slots to represent the background, while 6-slot model can not properly segment all objects when the image have more than 6 objects.

Our model differs significantly from the 11-slot model in terms of handling the background, as observed from the visualizations. In the case of the 11-slot model, when the number of objects of an image is small, the 11-slot model tends to divide the background into several slots. However, this division does not segment the background into several regions. Instead, the segmentation of background is very even in terms of light and shadow.

On the contrary, our model takes a different approach of not utilizing a fixed background slot. Instead, it intelligently merge the background regions to the nearest foreground objects. It is reflected in the visualization that the shadow (which corresponds to background) around the object is much darker in our proposed model than the fixed slot model. The visualizations demonstrate that our model consistently outputs an appropriate number of slots for each image. In order to evaluate the accuracy of our model in determining the number of objects, we illustrate the heatmap of confusion matrix of segmentation number and the slot

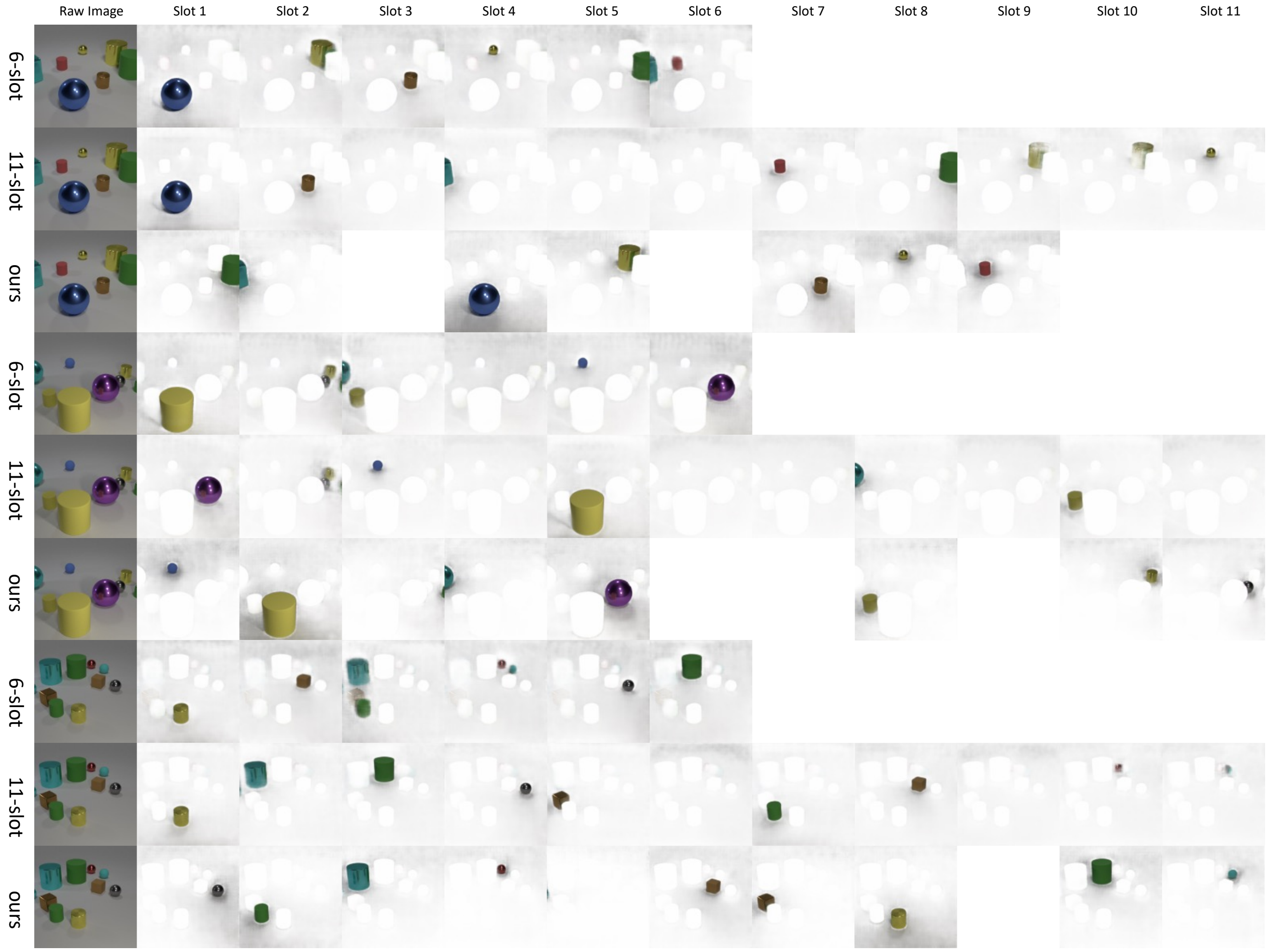


Figure 1. **Visualization of instance-level adaptive slot number selection** by per-slot segmentation on CLEVR10. We compare our model and two fixed-slot models. The results show that our model can select the slot number for each instance adaptively.

distribution of the models in Fig. 2. Our models exhibit a prediction distribution that almost perfectly aligns with the ground truth. Additionally, the heatmap revealed an excellent diagonal relationship, indicating that our method can roughly resolves the challenge of unsupervised object counting on CLEVR10. The diagonal of the heatmap reveals the instance-level adaptability of our model.

As for the metrics, our model achieves comparable object grouping results to both the 11-slot and 9-slot models. However, when it comes to localization, our model exhibits slightly lower performance. Nonetheless, we would like to suggest that this discrepancy can be attributed to the distinct approach we take in handling the background. Our model tends to merge the shadows around objects with the foreground, which, in turn, results in slightly lower IoU scores for the object masks predicted by our model. Consequently, this leads to drops in metrics such as mBO and CorLoC.

4. More Analysis on COCO

Similarly, we present the heatmap of the confusion matrix of segmentation number and the slot distribution of the models in Figure 3. However, It is worth noting that the COCO dataset has incomplete annotations, which means that not all objects have been annotated. In this case, we make our method solely focus on predictions related to the foreground. In other words, we only consider slots whose masks intersect with foreground objects. Besides, we limit our analysis to images that contain no more than 10 objects, since a significant majority of COCO images contain fewer than 10 objects. These particular images play a crucial role in determining an appropriate value for the fixed slot number, as 6 slot number reached the best results on COCO. Among the three models, our model shows better correlation between the ground truth object number and the predicted slot number. In contrast, the fixed-slot models fail

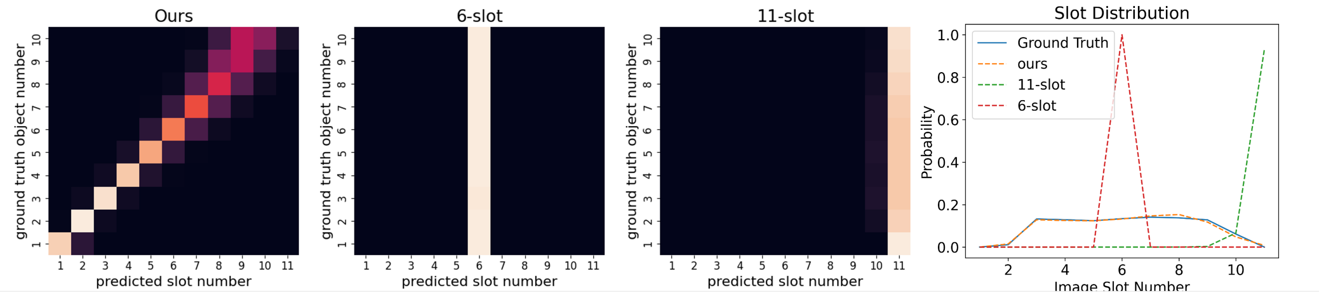


Figure 2. **Comparison between ground truth and predicted object numbers.** Heatmap of confusion matrix and slot distribution of our models and two fixed slot models on CLEVR10. For heatmap, y-axis corresponds to the number of objects of ground truth, and x-axis is the predicted object number by models. Due to imbalanced ground truth object numbers, we normalized the row and visualize the percentage. The brighter the grid, the higher the percentage. The slot distribution graph shows the probability density of grounded and predicted object numbers. **The results show that our model can choose the slot number almost perfectly on CLEVR10.**

Table 1. Results on CLEVR10 dataset.

	Pair-Counting				Matching			Information	
Models	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
3	59.00	60.85	93.17	72.22	10.33	0.08	70.09	66.36	66.41
6	90.77	89.26	98.13	93.08	19.35	19.45	91.81	92.32	92.34
9	97.59	97.86	**98.55**	98.14	*26.45*	*45.72*	97.81	97.39	97.40
11	**98.06**	**98.77**	98.35	**98.51**	**27.39**	**47.15**	**98.27**	**97.90**	**97.90**
Ours	*97.65*	*98.19*	*98.36*	*98.21*	22.51	37.00	*98.03*	*97.50*	*97.51*

to exhibit this diagonal pattern, further highlighting the efficacy of our approach.

As for the distribution of total slot number, all three models' predictions deviate from the ground truth. However, our model demonstrates the closest approximation to the ground truth distributions. This is substantiated by the visual examples presented in Figure 8, where our model showcases its ability to generate semantically coherent and meaningful segmentations. Notably, our model demonstrates adaptability by adjusting the slot number according to the complexity of the images, thereby further enhancing the quality of its predictions.

Figure 8 provides valuable insights into the reasons behind the deviations of the distributions from the ground truth. Let's consider the first column of Fig. 8, where our model demonstrates successful segmentation of the raw image into distinct regions, including the head of the girl, the T-shirts, the glove, and the background. The separation of the T-shirt and the head seems to be an over-segmentation compared to annotation, which may lead to low metric score. However, each segmented region exhibits semantic coherence and is still visually reasonable.

Real-world datasets often encompass complex part-whole hierarchies within objects. Without the availability of human annotations, accurately segmenting objects into the expected part-whole hierarchy becomes extremely challenging. Since many objects consist of multiple parts, just like the human body, it is expected that our model's predictions will slightly surpass the ground truth in terms of the number of slots. As a result, our model's prediction will be slightly more than ground truth.

5. Ablation

We conduct a series of ablation studies on MOVi-E dataset to investigate the components and design choices of our method.

Comparison of three design choices of masked slot decoder. In our main paper, we proposed several design choices of the masked slot decoder, and we focused on the zero mask variant. In Tab. 2 and Fig. 4, we compare the three variants in both quantitative and qualitative ways. The results show that our zero mask method effectively improves most metrics compared to the original slot attention model with 24 slots. However, in zero slot and learnable slot strategy, simply changing the manipulation on the mask to the manipulation on the slot makes the model collapse. Both zero slot and learnable slot strategy tend to group all pixels together instead of making a segmentation. If we do not explicitly remove the effect of the dropped slot by setting their alpha masks to zero, the zero/learnable slot

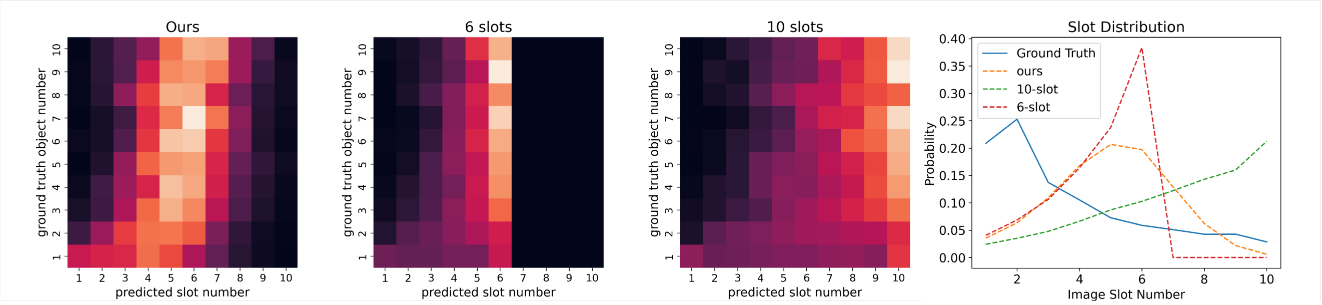


Figure 3. **Comparison between ground truth and predicted object numbers.** Heatmap of confusion matrix and slot distribution of our models and two fixed slot models on COCO. The sole distinction is that we consider the ground truth masks and predicted masks on the foreground. **Our model outperforms two fixed slot models for slot number selection.**

will still contribute to the reconstruction. Some instance-irrelated information will be introduced and may mislead the slot selection. As a result, zero/learnable slot tend to group all pixels together.

The Necessity of Gumbel Softmax. In the main paper, we utilized the hard zero-one mask:

$$Z = \mathrm{GumbelSoftmax}(\pi)_{:,1}. \tag{1}$$

To verify the necessity of Gumbel-Softmax, we provide experiments that keep the same masked slot decoder but replace the hard mask with a soft mask without Gumbel Softmax:

$$Z_{soft} = \pi_{:,1}. \tag{2}$$

The results are displayed in Fig. 5 and Tab. 3. Notably, without Gumbel Softmax, although the model provides slightly better mBO, all the other metrics are kept at the same level as the original slot attention model. Moreover, from the visualization, without Gumbel Softmax we can not achieve adaptive instance-level slot selection but produce segmentation with $K_{max} = 24$ masks. This failure is due to the landscape of the soft mask. Consider the following case:

$$\pi_{1,1} = \pi_{2,1} = \cdots = \pi_{K,1}, \quad \text{and} \quad \pi_{K,1} \to 0. \tag{3}$$

The regularization term approach zero $\mathcal{L}_{reg} \to 0$, and $\tilde{m}_i \approx m_i$. Therefore, our method is reduced to ordinary slot attention reconstruction. This simple case shows that without Gumbel Softmax, we can not easily suppress the information of unselected slots, leading to the failure of slot selection. With Gumbel Softmax, when $\pi_{i,1} \to 0$, $Z_i = 0$ and $\tilde{m}_i = 0$ happens with higher probability. The information of S_i will be totally removed. This difference leads to our success.

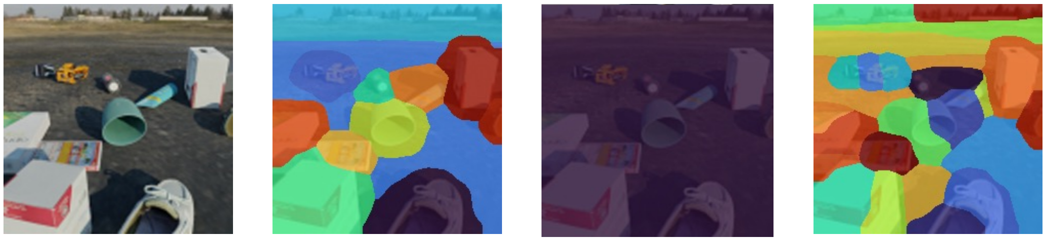
Influence of λ. We test how the regularization strength λ influences the results on MOVi-E. We compare 7 possible values of λ, ranging from $1e-2$ to 1 in Tab. 4, keeping other parameters unchanged compared with the main experiments. Generally, larger regularization prefers fewer slots left and grouping more patches. Recall and λ exhibit a positive correlation, while Precision and λ exhibit a negative correlation. For foreground grouping, the two metrics reach the balance around $\lambda = 0.1$ and $\lambda = 0.2$, which leads to the highest ARI and F_1 score. The grouping results have the best agreement with ground truth, which can also be proven by the highest AMI, NMI and Purity score. However, if we continue increasing λ, these metrics will decrease and drop to an abysmal level. When $\lambda = 1$, the model simply merges all tokens into a single group, which leads to perfect Recall but inferior results for all other metrics. For localization, $\lambda = 0.1$ have the best CorLoc score and performs well on mBO.

Influence of K_{max}. Our model includes a hyperparameter, denoted as K_{max}, which determines the maximum number of segmentations/slots that the model can produce. Ideally, K_{max} should be approximately equal to the highest number of objects present in any image within the dataset. Nonetheless, our model still yields satisfactory results even when K_{max} is set higher than this ideal value. We have conducted a comparative analysis of five distinct K_{max} settings on MOVi-E dataset, as detailed in Table 5. Our findings indicate that when K_{max} exceeds the actual maximum object count(MOVi-E includes at most 23 objects), most performance metrics tend to decrease as K_{max} increases. However, most metrics keep robust and consistently outperform the fixed-slot model. Notably, the metric mBO even shows improvement with very large values of K_{max}. The experiments demonstrate the robustness of our model to variations in K_{max}.

Comparison with oracle model. An oracle model for slot number selection is that we provide the ground-truth object number of each instance for DINOSAUR. We compare our model with this oracle model. The comparative analysis, presented in Table 6, reveals that our model not only matches but in some cases surpasses the performance of the oracle model. This is particularly noteworthy as our model achieves these results without access to the exact ob-

Table 2. Ablation study on the designs of masked slot decoder.

	Pair-Counting				Matching			Information	
Models	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
24 slots	*61.98*	**88.09**	57.82	*67.91*	**30.54**	*85.15*	*68.96*	*77.93*	*78.14*
Zero Mask	**75.30**	*84.74*	*78.64*	**80.20**	*29.47*	**90.09**	**80.12**	**82.32**	**82.45**
Zero Slot†	0.00	21.19	**100.00**	33.93	2.21	0.08	33.87	0.00	0.00
Leanrnable Slot†	0.00	21.19	**100.00**	33.93	2.21	0.08	33.87	0.00	0.00



Raw Image

Ours

Zero/learnable slot

24 slot

Figure 4. Illustration of the segmentation mask of three designs of mask slot decoders and ordinary 24-slot model.

ject count per instance. Such findings prove the effectiveness of our approach in slot number determination.

6. Results of semantic-level masks on COCO

In the main paper, we evaluate the metrics on COCO according to the instance mask. Moreover, we report the results based on semantic-level masks in Tab. 7 for further understanding. Compared with instance-level results, grouping metrics like ARI and F_1 score are lower, indicating that the model prefers instance-level object discovery to class-level. Overall, the results of semantic-level and instance-level masks are consistent.

7. Comparison with Unsupervised Multiple Instance Segmentation Method

Our work falls in unsupervised object discovery, which aims to locate and distinguish between different objects in the image without supervision. However, it does not necessarily provide fine-grained segmentation of each object. In different granularity, unsupervised instance segmentation aims to get a detailed mask for each localized object, clearly demarcating its boundaries.

Most unsupervised object segmentation methods follow a pipeline: initially creating pseudo masks using a self-supervised backbone and subsequently training a segmentation model based on these pseudo masks. In our discussion, we will primarily concentrate on the *initial stage* of these models. We compare our model with MaskCut proposed in CutLER [6], since it can generate multiple instance segmentation while other methods either segment only one object from each image [1, 7], or generate overlapping masks [5]. To accelerate MaskCut's inference, we work with a fixed subset here.

Table. 8 demonstrate that our model is great at distinguishing objects apart, whereas MaskCut is good at creating masks that closely match objects (thought some masks might cover more than one object). Unlike our model, MaskCut is based on iterative application of Normalized Cuts, which assumes images have very clear foreground and background distinctions, with only a few objects standing out in the foreground. But this assumption does not hold true for MOVi-E/C datasets. As a result, MaskCut produces high-quality masks that capture object shapes well (higher mBO on MOVi-C and COCO), but it struggles to tell different objects apart (lower ARI). This happens because it often groups multiple objects as foreground in each iteration of Normalized Cuts.

Additionally, our model can do object grouping in real-time, which is another advantage compared to MaskCut.

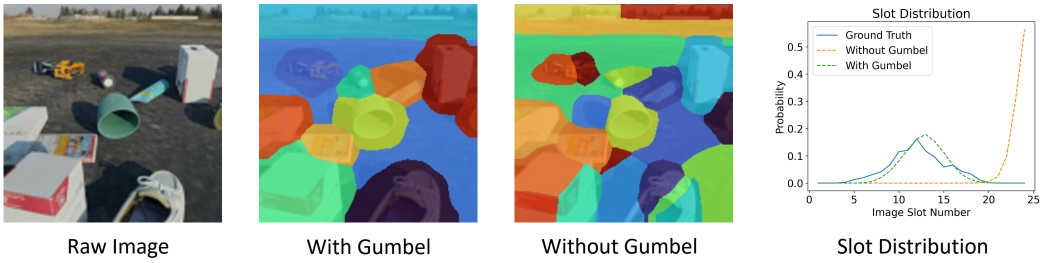
8. Results on Object Property Prediction on CLEVR10

In addition to MOVi-C, we study the usefulness of the adaptive slot attention of property prediction on CLEVR10. Following the setting of [2], we provide experiments of object position regression and color prediction.

Our experiments employ a one-hidden layer MLP as the downstream model. The model operates independently on the retained slots. Specifically, a kept slot serves as the

Table 3. Ablation study on the necessity of Gumbel Softmax.

	Pair-Counting				Matching			Information	
Models	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
24 slot	*61.98*	**88.09**	57.82	*67.91*	*30.54*	85.15	*68.96*	*77.93*	*78.14*
With Gumbel	**75.30**	84.74	**78.64**	**80.20**	29.47	**90.09**	**80.12**	**82.32**	**82.45**
wo Gumbel	61.76	*87.49*	*57.88*	67.74	**31.31**	*88.85*	68.85	77.51	77.73



Raw Image

With Gumbel

Without Gumbel

Slot Distribution

Figure 5. Illustration of the segmentation mask without Gumbel softmax and with Gumbel softmax respectively.

model's input, yielding a vector containing property predictions for that particular slot. We employ cross-entropy loss for color prediction and mean squared error (MSE) loss for coordinate regression. When both tasks are undertaken, we sum these two losses to calculate the total loss. We align predictions with targets with the Hungarian algorithm [3], minimizing the total loss of the assignment.

We present results in terms of the regression R^2 score for position estimation. For the color prediction task, to better compare the results for the model with different slot numbers, we provide the precision, recall and the Jaccard index. The results are provided in Tab. 9.

In our experiments with CLEVR10, the 6-slot model achieved the best Jaccard index among fixed-slot models. Notably, our model yields a superior Jaccard index to all fixed slot models. This demonstrates the effectiveness of our adaptive slot attention mechanism.

Additionally, our model demonstrates superior performance in terms of R^2 score for coordinate regression on CLEVR10. It is worth noting that the 3-slot model fails to predict the object coordinate well, with R^2 score less than 0. This highlights the importance of the slot number. With an improper slot number, the model may be not able to fit the data.

9. GENESIS-V2 with DINO backbone

In the main paper, we inherit the official implementation of GENESIS-V2 with UNet encoder. For better comparison, we provide results with the same DINO ViT/B-16 backbone as our models in Tab. 10. GENESIS-V2 with DINO backbone shows consistent improvement across all metrics, particularly in ARI. However, it significantly falls behind our method, further validating our approach's effectiveness. As illustrated in Fig 6, 7 & 8, compared to GENESIS-V2 with DINO backbone, our model can better determine the proper slot number and generate object mask closer to the boundary of an object, which makes our AdaSlot better on various metrics, especially mBO.

Moreover, other than the *heuristic stopping rule* of GENESIS-V2, our method introduces a novel *end-to-end approach* to selecting the slot numbers.

10. More Visualization

To provide a more comprehensive understanding of our methods, we have included additional visualizations in Fig. 6, Fig. 7 and Fig. 8. For each dataset, we select five examples and compare our model with GENESIS-V2 and fixed-slot DINOSAUR. Our model segments the raw image into regions that are not only semantically coherent but also highly meaningful. Moreover, our model showcases adaptability by dynamically adjusting the slot number in accordance with the complexity of the images.

11. Discussion and Limitations

Our model primarily applies to cases with clearly defined and thoroughly segmented objects. For situations similar to COCO, with numerous complex objects and incomplete annotations, the learned objects may not necessarily align with manual annotations. Additionally, due to the characteristics of the feature reconstruction, the performance on dense small objects is not very outstanding. When compare our model of K_{max} with the fixed slot model of $K = K_{max}$, our model produces fewer masks, and more small objects will be missed. However, the fixed-slot counterpart will also over-segment one object into multiple parts. Further, the

Table 4. Ablation on the influence of different λ

	Pair-Counting				Matching			Information	
λ	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
0.01	62.99	*87.44*	59.50	68.93	**30.47**	85.54	69.84	78.05	78.26
0.02	63.68	**87.49**	60.36	69.55	*30.16*	85.08	70.48	78.35	78.55
0.05	70.95	85.67	71.48	76.32	29.46	*86.79*	76.67	80.87	81.02
0.1	*75.30*	84.74	78.64	*80.20*	29.47	**90.09**	*80.12*	**82.32**	**82.45**
0.2	**76.07**	78.79	86.51	**81.30**	26.28	86.68	**80.49**	*81.50*	*81.61*
0.5	33.52	38.04	*89.27*	51.74	9.05	13.26	50.40	46.62	46.74
1.00	0.01	21.20	**99.96**	33.93	2.21	0.08	33.87	0.03	0.03

Table 5. Experiments under different choices of K_{max}. **Our model is robust to K_{max} and outperforms fixed slot model by a large margin.**

	Pair-Counting				Matching			Information	
K_{max}	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
24	76.73	85.21	80.31	81.42	29.83	91.03	81.28	83.08	83.20
28	75.24	86.46	77.04	80.06	30.03	90.33	80.25	82.75	82.88
32	73.77	87.17	74.34	78.71	30.29	89.63	79.10	82.32	82.46
36	71.87	88.08	70.94	76.93	30.57	89.39	77.60	81.77	81.92
40	70.64	88.61	68.87	75.81	31.03	89.07	76.63	81.35	81.51
24(*fixed*)	61.98	88.09	57.82	67.91	30.54	85.15	68.96	77.93	78.14

part-whole hierarchy in real-world scenes brings additional complexity and challenge to unsupervised object discovery. We leave improvements regarding this challenge for future works.

Table 6. Comparison between our models and oracle models. **Our model not only matches but in some cases surpasses the performance of the oracle model.**

		Pair-Counting				Matching			Information	
Dataset	Model	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
Movi-C	Ours	75.59	84.64	86.67	84.25	35.64	76.80	85.21	78.54	78.60
Movi-C	Oracle	75.68	85.67	84.99	84.30	33.82	72.36	85.48	78.55	78.62
Movi-E	Ours	76.73	85.21	80.31	81.42	29.83	91.03	81.28	83.08	83.20
Movi-E	Oracle	74.97	84.02	78.44	80.08	29.33	90.67	79.54	81.92	82.05

Table 7. Experiments of Semantic-level masks on COCO datasets. **The semantic-level results are consistent with instance-level results.**

		Pair-Counting				Matching			Information	
Model	K	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
DINOSAUR	4	20.72	85.02	52.18	61.47	20.61	13.89	59.32	24.93	24.96
	6	28.93	89.92	58.04	67.12	30.85	41.00	65.43	32.35	32.38
	7	27.43	90.66	54.15	64.17	31.10	39.79	62.48	31.72	31.75
	8	25.32	91.29	48.89	59.89	29.93	34.69	58.31	30.74	30.78
	10	23.02	92.16	43.58	55.26	29.75	32.47	53.84	29.71	29.75
	12	20.99	92.89	38.80	50.79	29.17	30.70	49.68	28.75	28.79
	20	15.72	94.12	27.97	39.43	26.44	23.33	39.35	25.77	25.82
	33	11.60	95.07	19.99	30.04	24.03	18.87	30.97	23.14	23.21
Ours		26.60	89.71	55.30	64.90	30.53	37.74	63.62	30.56	30.59

Table 8. Comparsion between our model and MaskCut.

		Pair-Counting				Matching			Information	
Model	Dataset	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
MaskCut	MOVi-E	54.14	55.59	86.49	65.48	25.28	92.80	65.56	63.87	63.99
	MOVi-C	59.05	75.60	88.08	79.19	40.84	88.71	78.36	60.80	60.88
	COCO	29.18	73.58	74.47	69.73	33.95	71.88	69.23	32.20	32.25
Ours	MOVi-E	77.48	86.18	80.19	81.83	30.43	93.20	81.67	84.08	84.19
	MOVi-C	72.81	86.13	86.08	84.33	37.33	80.16	83.81	75.97	76.03
	COCO	40.38	81.26	67.16	68.55	26.94	47.12	67.33	45.53	45.59

Table 9. Experiments of **object property prediction** on CLEVR-10.

Slot	Recall	Precision	Jaccard	R^2
3	26.84	58.19	26.84	-0.35
6	72.43	78.51	64.95	0.57
9	88.26	63.78	62.91	0.60
11	91.88	54.32	54.32	0.69
Ours	89.06	93.03	87.25	0.76

Table 10. Experiments of GENESIS-V2 with DINO backbone

			Pair-Counting				Matching			Information	
Dataset	Model	K	ARI	P.	R.	F_1	mBO	CorLoc	Purity	AMI	NMI
MOVi-C	GENESIS-V2	6	68.48	77.77	87.05	80.54	29.47	63.07	81.60	71.86	71.93
		11	52.60	67.51	83.20	72.02	20.29	41.78	73.09	57.73	57.81
	Ours	11	**75.59**	**84.64**	**86.67**	**84.25**	**35.64**	**76.80**	**85.21**	**78.54**	**78.60**
MOVi-E	GENESIS-V2	9	72.99	74.70	86.28	78.89	16.39	48.60	77.89	79.32	79.43
		24	65.76	72.64	78.08	73.23	21.39	72.42	73.30	74.20	74.35
	Ours	24	**76.73**	**85.21**	**80.31**	**81.42**	**29.83**	**91.03**	**81.28**	**83.08**	**83.20**
COCO	GENESIS-V2	33	24.30	75.93	53.30	56.17	19.76	22.37	55.74	32.05	32.11
	Ours	33	**39.00**	**81.86**	**66.42**	**68.37**	**27.36**	**47.76**	**67.28**	**44.11**	**44.17**

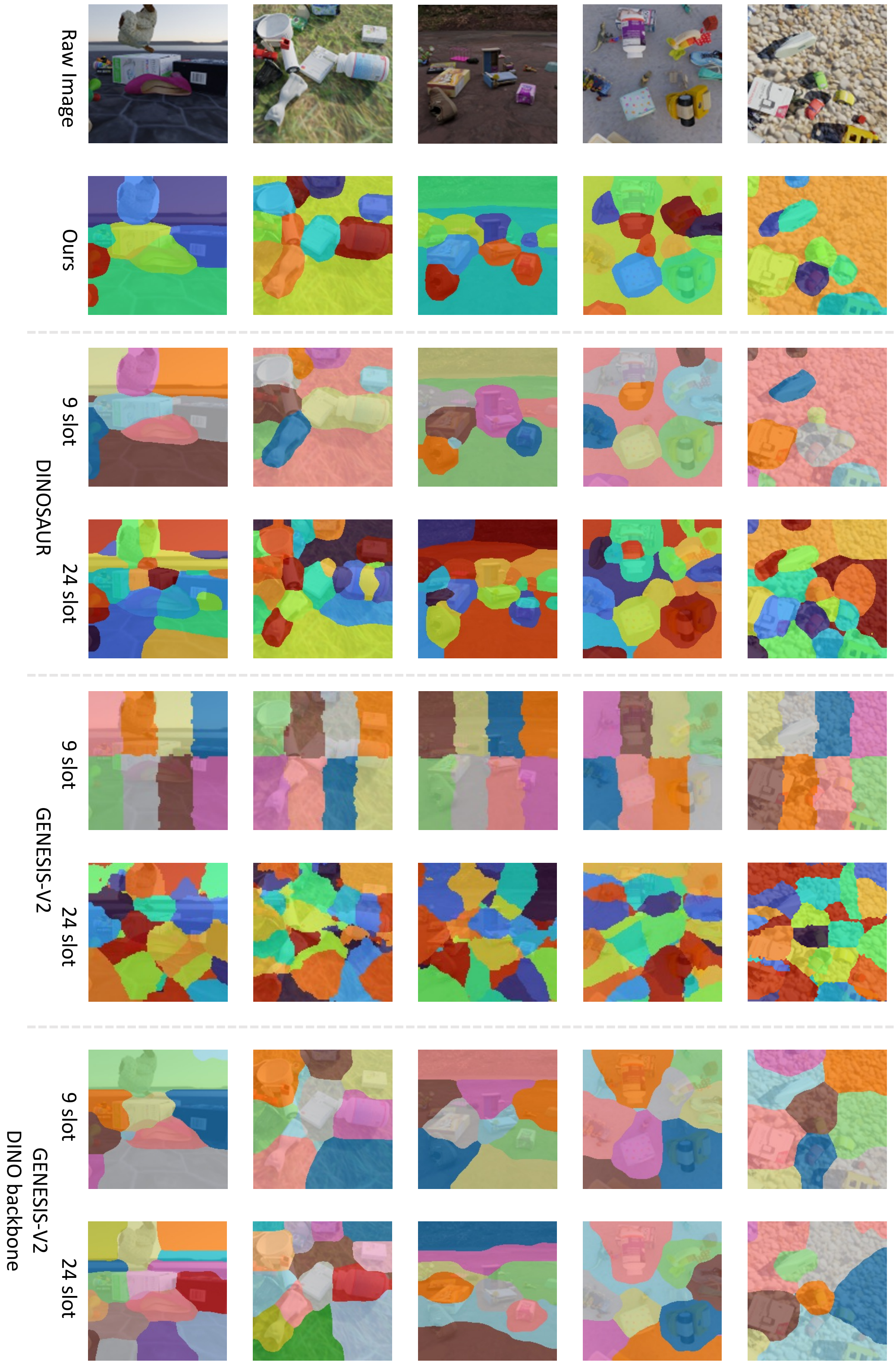


Figure 6. More visualizations on MOVi-E

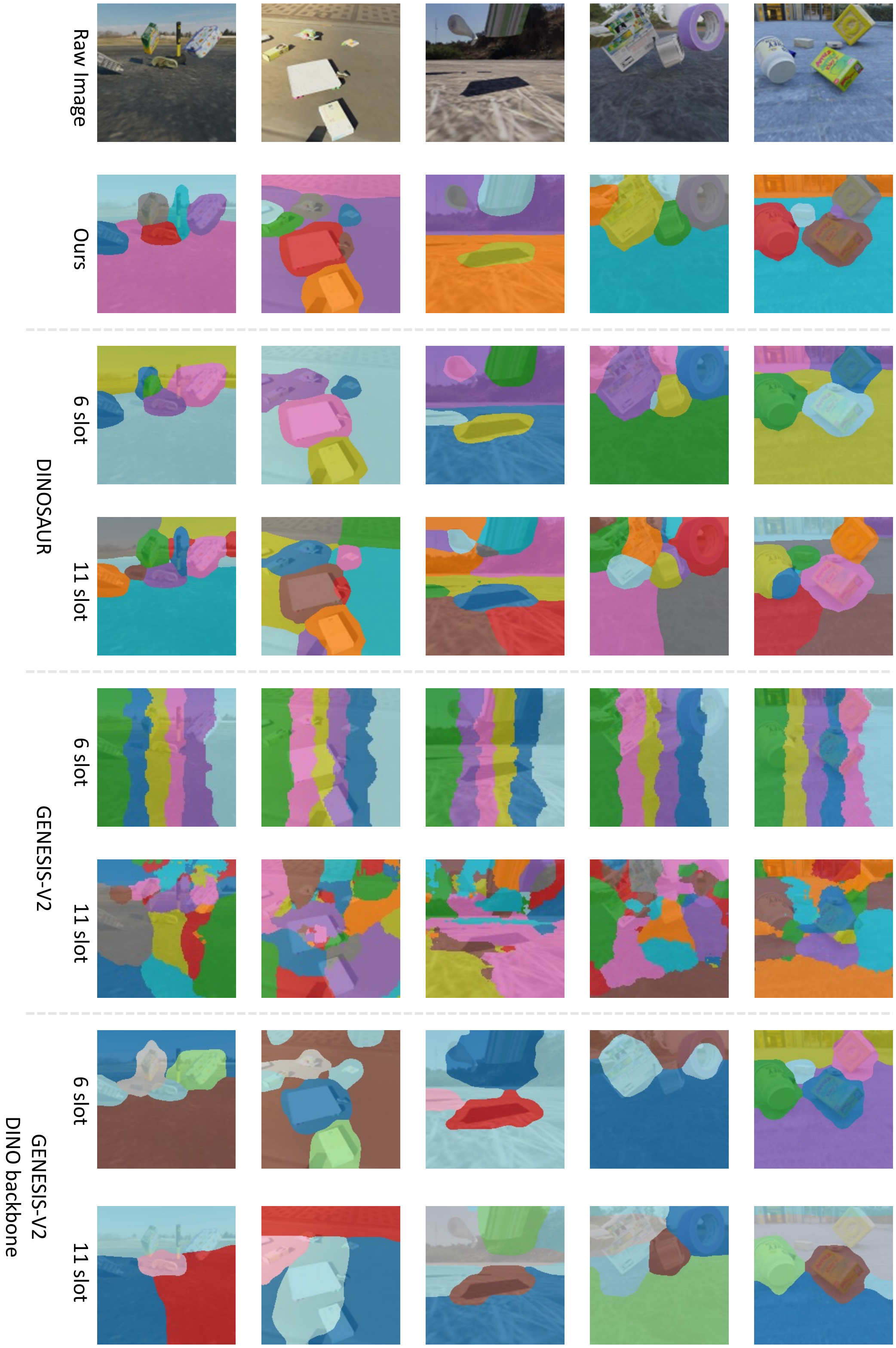


Figure 7. More visualizations on MOVi-C

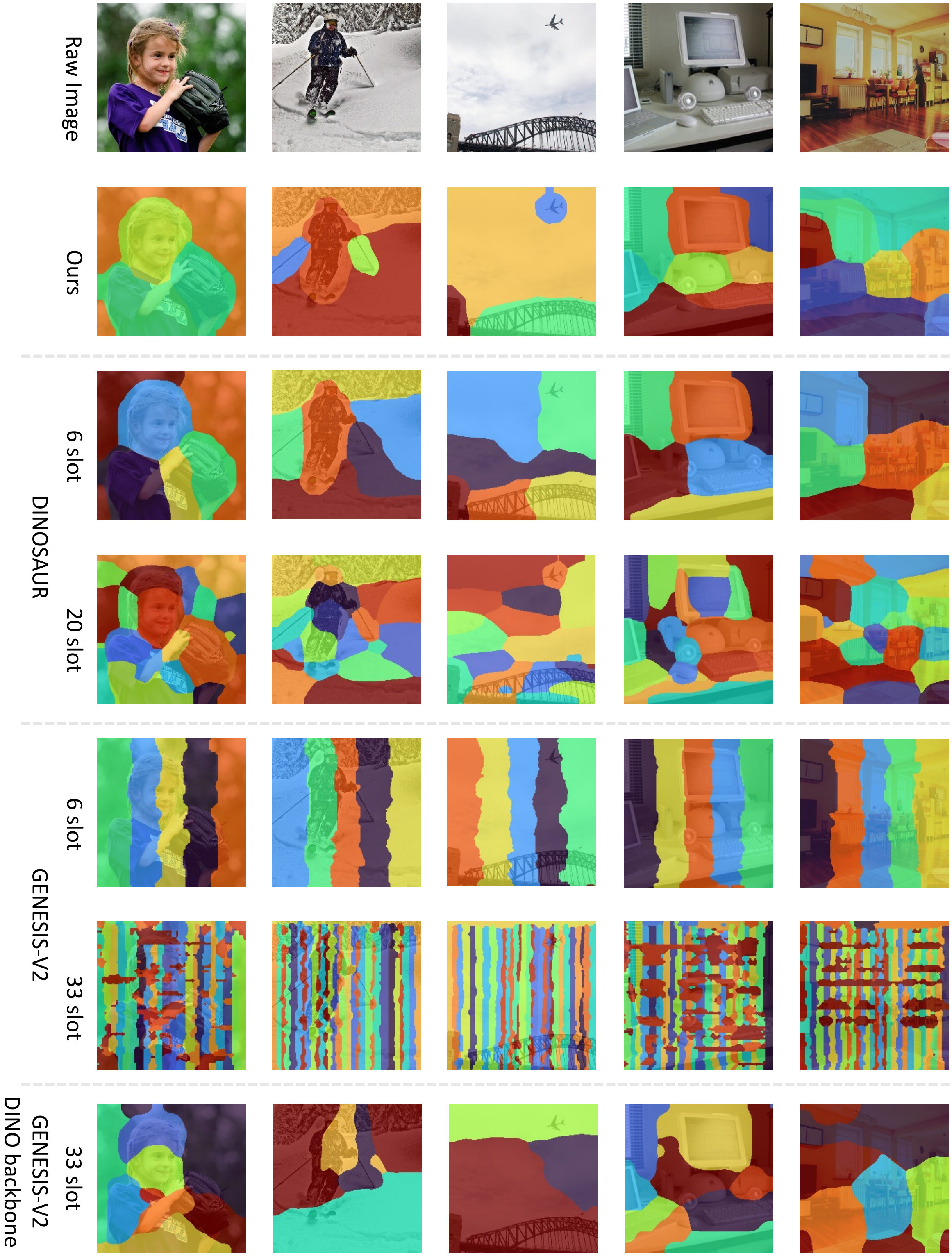


Figure 8. More visualizations on COCO

References

[1] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 5

[2] Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. *arXiv preprint arXiv:2107.00637*, 2021. 5

[3] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[4] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 1

[5] Xinlong Wang, Zhiding Yu, Shalini De Mello, Jan Kautz, Anima Anandkumar, Chunhua Shen, and Jose M Alvarez. Freesolo: Learning to segment objects without annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14176–14186, 2022. 5

[6] Xudong Wang, Rohit Girdhar, Stella X Yu, and Ishan Misra. Cut and learn for unsupervised object detection and instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3124–3134, 2023. 5

[7] Yangtao Wang, Xi Shen, Yuan Yuan, Yuming Du, Maomao Li, Shell Xu Hu, James L Crowley, and Dominique Vaufreydaz. Tokencut: Segmenting objects in images and videos with self-supervised transformer and normalized cut. *arXiv preprint arXiv:2209.00383*, 2022. 5

[8] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 1