# SPOT: Self-Training with Patch-Order Permutation for Object-Centric Learning with Autoregressive Transformers

## Supplementary Material

Ioannis Kakogeorgiou[1] Spyros Gidaris[2] Konstantinos Karantzalos[1] Nikos Komodakis[3,4,5]

[1]National Technical University of Athens [2]valeo.ai
[3]University of Crete [4]IACM-Forth [5]Archimedes/Athena RC

## Contents

## A. Discussion about FG-ARI unreliability

FG-ARI has been a common metric in assessing predicted object masks against ground-truth segmentation in previous works. However, several recent works have raised concerns about its reliability [8, 17, 25, 28, 36]. Notably, FG-ARI is criticized for its unreliability, which may favor either over-segmentation [8, 25] or under-segmentation. Additionally, the fact that it ignores background pixels makes it unable to probe a model's effectiveness in object segmentation [17, 25].

Consequently, FG-ARI can be misleading in assessing segmentation quality. We illustrate this in Tab. 6, where we show that we can achieve the highest 49.9 FG-ARI score in the Pascal dataset *by trivially assigning all pixels to a single slot-mask (1-block mask)*. This highlights the metric's unreliability, particularly in datasets with scenes featuring few or single objects.

To further examine FG-ARI, Fig. 7 compares qualitative results of SPOT with an autoregressive transformer decoder (the standard setup) and SPOT with an MLP-based decoder. While the standard SPOT setup outperforms SPOT with an MLP decoder in mBO$^i$, mBO$^c$, and mIoU metrics, SPOT with MLP achieves a higher FG-ARI score (42.5 vs. 37.8) (see Table 1 and Table 4 in the main paper). However, the qualitative results in Fig. 7 indicate that SPOT with MLP is distinctly inferior to the standard SPOT. This is particularly evident in the first three images, showing notable over-segmentation (as seen with the bear, horse, and dog) and incorrect object grouping (such as the horse and rider).

These observations emphasize the inadequacy of FG-ARI in measuring the segmentation quality of predicted object masks. Unsupervised object-centric methods should place greater reliance on mBO and mIoU metrics.

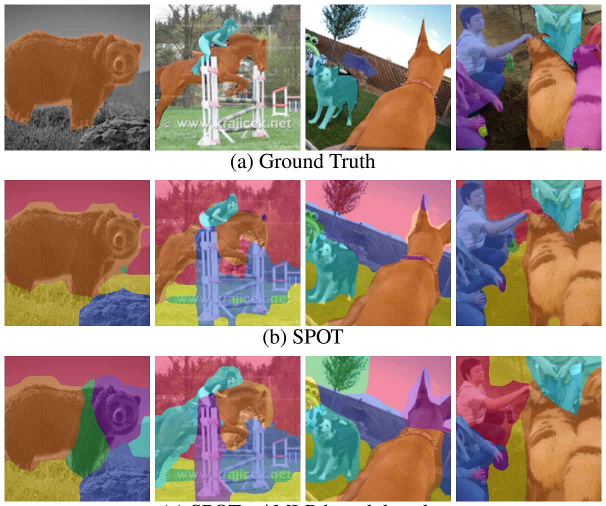

(a) Ground Truth
(b) SPOT
(c) SPOT w/ MLP-based decoder

Figure 7. Example results on COCO 2017, using 7 slots.

## B. Additional experimental results

### B.1. Comparison with prior object-centric methods

In Table 6, we present extended benchmark results, including the mean Intersection over Union (mIoU) and Foreground Adjusted Rand index (FG-ARI) across all datasets. We show results for SPOT on FG-ARI derived from both decoder and slot encoder attention masks, offering a more comprehensive view of its capabilities.

| Method | mBO$^i$ | mBO$^c$ | mIoU | FG-ARI |
|---|---|---|---|---|
| | | MOVi-C | | |
| 11-block Mask | 19.5 | - | 18.2 | 42.7 |
| SA [21]† | 26.2±1.0 | - | - | 43.8±0.3 |
| SLASH [18] | - | - | 27.7±5.9 | 51.9±4.0 |
| SLATE [29]† | 39.4±0.8 | - | 37.8±0.7 | 49.5±1.4 |
| DINOSAUR (MLP) [28] | 39.1±0.2 | - | - | **68.6±0.4** |
| DINOSAUR [28] | 42.4 | - | - | 55.7 |
| LSD [16] | 45.6±0.8 | - | 44.2±0.9 | 52.0±3.5 |
| SPOT w/o ENS (ours) | 47.0±1.2 | - | 46.4±1.2 | 52.1±3.3/57.9±2.0 |
| SPOT w/ ENS (ours) | **47.3±1.2** | - | **46.7±1.3** | 52.3±3.3/57.9±2.0 |
| | | MOVi-E | | |
| 24-block Mask | 20.4 | - | 18.8 | 41.9 |
| SA [21]† | 24.0±1.2 | - | - | 45.0±1.7 |
| SLATE [29]† | 30.2±1.7 | - | 28.6±1.7 | 46.1±3.3 |
| DINOSAUR (MLP) [28] | 35.5±0.2 | - | - | **65.1±1.2** |
| SlotDiffusion [36] | 30.2 | - | 30.2 | 60.0 |
| LSD [16] | 39.0±0.5 | - | 37.6±0.5 | 52.2±0.9 |
| SPOT w/o ENS (ours) | 39.9±1.1 | - | 39.0±1.1 | 56.4±4.1/59.9±0.4 |
| SPOT w/ ENS (ours) | **40.1±1.2** | - | **39.3±1.2** | 56.8±4.3/59.9±0.4 |
| | | PASCAL | | |
| 1-block Mask | 19.1 | 23.0 | 17.4 | **49.9** |
| 6-block Mask | 24.7 | 25.9 | 24.2 | 13.9 |
| SA [21]† | 24.6 | 24.9 | - | 12.3 |
| SLATE [29]† | 35.9 | 41.5 | - | 15.6 |
| CAE [22]† | 32.9±0.9 | 37.4±1.0 | - | - |
| DINOSAUR (MLP) [28] | 39.5±0.1 | 40.9±0.1 | - | 24.6±0.2 |
| DINOSAUR [28] | 44.0±1.9 | 51.2±1.9 | - | 24.8±2.2 |
| Rotating Features [23] | 40.7±0.1 | 46.0±0.1 | - | - |
| SlotDiffusion [36] | **50.4** | 55.3 | - | 17.8 |
| SPOT w/o ENS (ours) | 48.1±0.4 | 55.3±0.4 | 46.5±0.4 | 19.4±0.7/19.7±0.4 |
| SPOT w/ ENS (ours) | 48.3±0.4 | **55.6±0.4** | **46.8±0.4** | 19.9±0.9/19.7±0.4 |
| | | COCO | | |
| 7-block Mask | 16.8 | 19.5 | 15.9 | 22.7 |
| SA [21]† | 17.2 | 19.2 | - | 21.4 |
| SLATE [29]† | 29.1 | 33.6 | - | 32.5 |
| DINOSAUR [28] | 32.3±0.4 | 38.8±0.4 | - | 34.3±0.5 |
| SlotDiffusion [36] | 31.0 | 35.0 | - | 37.2 |
| Stable-LSD [16] | 30.4 | - | - | 35.0 |
| SPOT w/o ENS (ours) | 34.7±0.1 | 44.3±0.3 | 32.7±0.1 | 36.6±0.3/**37.8±0.5** |
| SPOT w/ ENS (ours) | **35.0±0.1** | **44.7±0.3** | **33.0±0.1** | 37.0±0.2/**37.8±0.5** |

Table 6. *Comparison with object-centric methods on COCO, PASCAL, MOVi-C and MOVi-E datasets.* SPOT results are the mean and std over 3 seeds. For SPOT FG-ARI, we report results from both *decoder/slot encoder* masks. DINOSAUR uses an autoregressive decoder and DINO [2] ViT encoder (ViT-B/16 for PASCAL and MOVi-C, ViT-S/8 for COCO). DINOSAUR-MLP uses an MLP decoder and DINO ViT encoder (ViT-B/16 for COCO and PASCAL, ViT-S/8 for MOVi-C/E). †: COCO and PASCAL results of SA and SLATE are from [36], MOVi-C/E results are from [28] for SA and from [16] for SLATE, PASCAL results of CAE are from [23]. We **bold** the best and underline the second-best results.

| | SP | ST | Decoder | | | Slot Attention | | |
|---|---|---|---|---|---|---|---|---|
| | | | mBO$^i$ | mIoU | FG-ARI | mBO$^i$ | mIoU | FG-ARI |
| (a) | | | 45.3±1.8 | 44.6±1.7 | 50.6±4.3 | 42.8±0.1 | 42.0±0.1 | 55.4±0.6 |
| (b) | ✓ | | 46.1±0.9 | 45.2±0.9 | 51.8±2.7 | 42.5±0.1 | 41.6±0.1 | 57.6±0.7 |
| (c) | ✓ | ✓ | **47.3±1.2** | **46.7±1.3** | **52.3±3.3** | **46.2±0.8** | **45.4±0.7** | **57.9±2.0** |

Table 7. *Ablation study on MOVi-C.* Metrics for slot masks generated by Decoder and Slot Attention. Results are mean and standard dev. over 3 seeds. SP: sequence permutation with ensembling of nine permutations at test-time, ST: self-training.

| BO-QSA [15] | ST | mBO$^i$ |
|---|---|---|
| | | Collapse |
| ✓ | | 30.7±2.2 |
| ✓ | ✓ | **34.7±0.1** |

Table 8. *Image encoder training stability results on COCO.* Results are the mean and standard deviation of the decoder's mBO$^i$ over 3 seeds. ST: self-training studies the impact of the distillation loss $L_{\text{ATT}}$, and BO-QSA the impact of using trainable initialization of slots along with bi-level optimization [3, 15]. All models use sequence permutation in the autoregressive decoder.

SPOT outperforms other methods in mBO$^i$, mBO$^c$ and mIoU across all datasets, except for the mBO$^i$ in the Pascal setting, where it ranks second-best. Concerning FG-ARI, SPOT achieves the best results in COCO and second-best in MOVi-C and MOVi-E (within the standard deviation). However, as discussed in Sec. A, FG-ARI is unreliable. For instance, in the Pascal dataset, the 1-block mask (*i.e.*, the trivial solution where the entire image is covered by a single mask) achieves the highest FG-ARI score, doubling the score of the second-best.

## B.2. Ablations in MOVi-C

In Tab. 7, we analyze the effects of self-training and sequence permutations on the MOVi-C dataset. Both of these approaches enhance performance, resulting, for instance, in a 2-point increase in mBO$^i$.

## B.3. Image encoder training stability analysis

In Tab. 8, we notice that fine-tuning the image encoder, without self-training, can avoid training collapse by using trainable initial slots and bi-level optimization (BO-QSA [15]). However, the achieved mBO$^i$ score, 30.7, is notably lower compared to not fine-tuning the image encoder and using randomly initialized slots, where the mBO$^i$ is 32.7 (refer to Tab. 1 entry (b) in the main paper). Furthermore, fine-tuning the image encoder solely with BO-QSA and without self-training exhibits a high standard deviation of 2.2, indicating there is still a training stability issue. The introduction of self-training not only enhances performance significantly (from 30.7 to 34.7) but also stabilizes the training process, as evident from the drop in standard deviation

|  | SlotDiffusion [36] |  |  | DINOSAUR [28] |  |  | SPOT |  |
|---|---|---|---|---|---|---|---|---|
|  | Offic. | Reprod. | w/ BO | Offic. | Reprod. | w/ BO | w/o ENS | w/ ENS |
| MBO$^c$ | 35.0 | 36.3 | 34.7 | 38.8 | 42.1 | 39.7 | <u>44.3</u> | **44.7** |
| MBO$^i$ | 31.0 | 30.5 | 29.5 | 32.3 | 32.3 | 31.5 | <u>34.7</u> | **35.0** |

Table 9. *Results on COCO from integrating SlotDiffusion [36] and DINOSAUR [28] frameworks with BO-QSA, referred to as BO.* Except for DINOSAUR (OFFIC.), which is trained for 270 epochs, all others are trained for 100 epochs.

| ENCODER | METHOD | MBO$^i$ | MBO$^c$ | MIOU | FG-ARI |
|---|---|---|---|---|---|
| DINO | DINOSAUR | 31.6$\pm$0.7 | 39.7$\pm$0.9 | - | 34.1$\pm$1.0 |
|  | SPOT w/o ENS | 34.7$\pm$0.1 | 44.3$\pm$0.3 | 32.7$\pm$0.1 | 36.6$\pm$0.3 |
|  | SPOT w/ ENS | **35.0$\pm$0.1** | **44.7$\pm$0.3** | **33.0$\pm$0.1** | 37.0$\pm$0.2 |
| MoCo-v3 | DINOSAUR | 31.4$\pm$0.2 | 38.5$\pm$0.5 | - | 35.2$\pm$0.2 |
|  | SPOT w/o ENS | 32.7$\pm$0.2 | 41.7$\pm$0.4 | 30.7$\pm$0.2 | 34.4$\pm$0.2 |
|  | SPOT w/ ENS | 32.9$\pm$0.2 | 42.0$\pm$0.4 | 30.9$\pm$0.2 | 34.8$\pm$0.3 |
| MAE | DINOSAUR | 30.2$\pm$1.8 | 33.2$\pm$1.8 | - | 32.8$\pm$3.7 |
|  | SPOT w/o ENS | 33.3$\pm$0.3 | 40.7$\pm$0.7 | 31.4$\pm$0.3 | 37.5$\pm$1.0 |
|  | SPOT w/ ENS | 33.4$\pm$0.3 | 40.9$\pm$0.7 | 31.6$\pm$0.3 | **37.7$\pm$1.0** |

Table 10. *Evaluation with various pre-trained encoders on COCO.* SPOT results are the mean and standard deviation over 3 seeds.

from 2.2 to 0.1. This underscores the crucial role of self-training in ensuring training stability.

### B.4. Impact of bi-level optimized query on other frameworks

In Tab. 9, we show that BO-QSA [15] without our SPOT's self-training does not perform well on DINOSAUR [28] and SlotDiffusion [36] frameworks.

### B.5. Comparing with other pre-trained image features

In our experiments, we used DINO [2] features for the image encoder and reconstruction targets $Y$. We also explored MOCO-v3 [4] and MAE [13] as alternatives, comparing SPOT with DINOSAUR on the COCO dataset. Results in Tab. 10 show SPOT outperforming DINOSAUR across all examined pre-trained encoders, except with MOCO-v3 and FG-ARI metric. DINO excels in MBO$^i$, MBO$^c$, and MIOU metrics, while MAE performs best in the FG-ARI metric.

### B.6. Performance across different instance sizes

In Fig. 8, we present a detailed analysis of SPOT's performance across various instance sizes. We observe that SPOT performs optimally when instances occupy between 20% and 80% of the input image area. For larger instances (exceeding 80%), SPOT continues to yield favorable results. Conversely, a decline in performance is noted as the size of the instances decreases. We note that this decrease in performance for smaller instance sizes is expected due to the coarse resolution of ViT encoders.

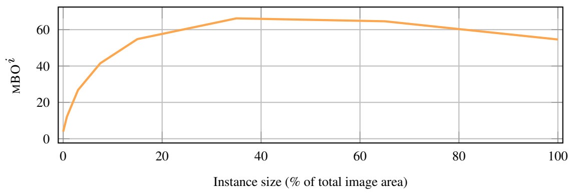
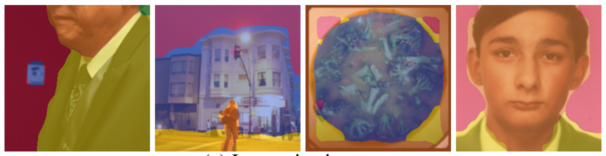
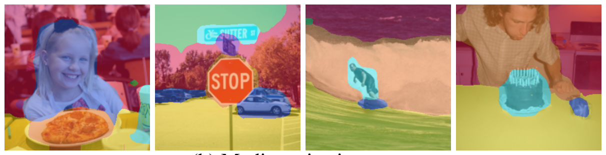


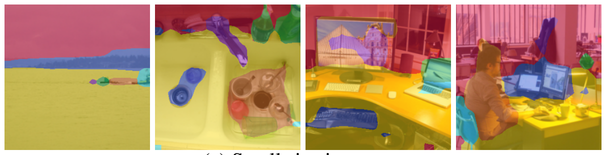
Figure 8. *Analysis of SPOT performance across different instance sizes.* We demonstrate SPOT's performance, measured in MBO$^i$ on COCO, across varied instance sizes. The instance sizes are expressed as a percentage of the total image area, categorized into distinct bins: 0-0.5%, 0.5-1%, 1-5%, 5-10%, 10-20%, 20-50%, 50-80%, 80-100%.



(a) Large size instances



(b) Medium size instances



(c) Small size instances

Figure 9. Example SPOT's results on COCO 2017, using 7 slots, for large, medium and small instance sizes.

In Fig. 9, we provide examples for large, medium and small instance mask sizes. For the large and medium-sized examples, we observe good performance. For small instances, there is often a tendency to be grouped together or be part of the background.

## C. Further Discussion

### C.1. About the autoregressive decoder

**Why use an autoregressive decoder design?** For object-centric learning, autoregressive (AR) design was shown to be superior for handling complex scenes [28–30] compared to spatial-broadcast MLPs that make a strong assumption that patches are independent when conditioned on slots. In contrast, AR imposes no assumption: from the chain rule of probability, any joint distribution over random variables (e.g., patches) can be expressed as the product of conditional distributions AR-style. While not all AR factorization orderings are effective, it is crucial to note that effective ones can be defined for images. This is supported by a significant body of work in image synthesis [9, 10, 26, 27, 32, 33, 38, 39] and recent successes in self-supervised pre-training [6] and generalist image models [1]. Also, our AR decoder has access to the slots that encode information from the *entire image*.

**Does using permutations suggest causal AR-parsing is irrelevant for 2D data?** The use of permutations allows the AR decoder to learn from multiple factorization orders (which has shown positive effects even in text [37]). *Sequence permutations aim to tackle an overfitting issue*: AR transformers rely less on input slots when predicting later tokens in the sequence. With the permutations, due to shared parameters across all sequence orders, the model learns to equally rely on slots across all token positions. *We perceive this permutation strategy as more natural for implementation with 2D data*, as all used orderings are equally meaningful.

### C.2. Limitations and future work

As in prior works, SPOT extracts a fixed number of slots, a constraint that should be addressed in future works. Additionally, integrating online teacher-student training (e.g., via momentum teachers) during the second training stage of SPOT could potentially enhance both training efficiency and performance. Furthermore, as depicted in Fig. 8, there is a decrease in performance with small-sized objects, indicating a need to explore strategies for handling higher-resolution images in future studies. Finally, SPOT shows a preference for semantic over instance segmentation, as evidenced by its performance on the $\mathrm{MBO}^i$ and $\mathrm{MBO}^c$ metrics, which is linked to the broader issue of object definition ambiguity in real-world images.

## D. Implementation details

### D.1. MLP decoder

In the MLP-based decoder, we employ the spatial broadcast mechanism as described in prior works [28, 35], in which each slot is expanded into $n$ tokens that correspond to the patches. Next, learnable positional information is added to these tokens for spatial identification. These tokens are then processed individually by a four-layer MLP with ReLU activations. This process yields both the reconstruction and an alpha map for each slot ($d_y$ + 1 dimensions). The alpha map serves as an attention map that indicates the active regions of the slot. The final reconstructed output is obtained by aggregating these individual slot reconstructions, using the alpha maps as weighting factors. In the MLP decoder, the slot-attention masks are the predicted alpha maps.

### D.2. SPOT models

We employ the Adam optimizer [19] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, no weight decay, and a batch size of 64. The learning rate ($lr$) follows a linear warm-up from 0 to a peak value for 10000 training iterations and then decreases via a cosine annealing schedule. For experiments on COCO and PASCAL using DINO and MoCo-v3 encoders and for both training stages, the peak learning rate is $4 \times 10^{-4}$ and the low value is $4 \times 10^{-7}$. For both stages with MAE on COCO and the first stage of MOVi-C/E experiments, the peak value is $2 \times 10^{-4}$ and the low value is $4 \times 10^{-5}$. For the second stage of MOVi-C/E experiments, the peak value is $2 \times 10^{-4}$ and the low value is $1.5 \times 10^{-4}$. For each training stage on COCO and PASCAL, we use 50 and 560 training epochs, respectively. For MOVi-C/E experiments, we use 65 and 30 epochs for the first and second stages, respectively.

We implement our SPOT models using ViT-B/16 [5] for the encoder (by default initialized with DINO [2]), without applying the final layer norm. In the autoregressive transformer decoder, we use 4 transformer blocks each one with 6 heads. For the MLP-based decoder, we use 2048 hidden layer size. For all experiments, we use 3 iterations in the slot attention module with the dimension of slot $d_u$ being 256 and the slot attention's MLP hidden dimension being 1024. Unless stated otherwise, loss weight $\lambda$ is 0.005. For MLP-based decoder, MoCo-v3 [4], and MAE [13] encoders, $\lambda$ is 0.001. At the self-training stage, we fine-tune the last four ViT encoder's blocks.

Following [28], on COCO, PASCAL, MOVi-C, and MOVi-E we use 7, 6, 11, and 24 slots, respectively.

In the main paper, we explored a training approach inspired by CapPa [31]. To tune this approach, we experimented with different percentages of parallel decoding —25%, 50%, and 100%— and observed $\mathrm{MBO}^i$ scores of 27.8, 25.9, and 23.6, respectively. We found that a lower percentage of parallel decoding correlates with better performance. Consequently, we used a 25% parallel decoding for the experiment in the main paper.

### D.3. Datasets and evaluation

Here, we provide details about the employed datasets and the evaluation protocol.

**COCO** We use COCO 2017 dataset for training, which consists of 118,287 images, and evaluate SPOT on the validation set of 5,000 images. During training, we resize the minor axis of the images to 224 pixels, perform center cropping at 224×224 and then random horizontal flipping with 0.5 probability. For the evaluation, we use both types of masks: instance masks for the $\mathrm{MBO}^i$, mIoU, FG-ARI and segmentation masks for the $\mathrm{MBO}^c$ metrics. For consistency with previous studies [11, 14, 28], we scale the minor axis to 320 pixels, perform center cropping, and evaluate the masks at 320×320 resolution.

**PASCAL** We use PASCAL VOC 2012 “trainaug” set for training, which consists of 10,582 images. The "trainaug" variant contains 1,464 images from the segmentation train set and 9,118 from the SBD dataset [12]. We use this split, for consistency with previous works [24, 28, 34]. During training, we resize the minor axis of the images to 224 pixels, perform simple random cropping at 224×224 and then random horizontal flipping with 0.5 probability. We evaluate on the official segmentation validation set, which consists of 1,449 images. We ignore the unlabeled pixels during evaluation. Similar to COCO, we resize the minor axis to 320 pixels, perform center cropping, and evaluate the masks at 320×320 resolution.

**MOVi-C/E** We transform MOVi-C/E datasets, originally consisting of videos, into image datasets by selecting nine random frames from each clip of the train set. From this process, we obtain 87,633 images for training with MOVi-C and 87,741 images with MOVi-E. During training, we resize the images at 224×224 resolution. For evaluation, we use every frame from the 250 clips of the validation set consisting of 6,000 images, being consistent with prior works [7, 20, 28]. We evaluate the masks at the full 128×128 resolution.

**n-block Mask** : As in [28], to generate block mask patterns, we first divide the image into columns and then subdivide these columns to obtain the required number of slot masks. Specifically, for MOVi-C, we utilize 3 columns to produce a total of 11 block masks. For MOVi-E, we use 4 columns to accommodate 24 masks. In the case of PASCAL VOC 2012, we use 2 columns for 6 masks. For COCO, we use 2 columns for 7 masks. The number of block masks is aligned with the number of slots utilized for each respective dataset. For the 1-block mask, we employ just one column and one mask; in other words, the entire image is covered by a single mask.

## References

[1] Yutong Bai, Xinyang Geng, Karttikeya Mangalam, Amir Bar, Alan Yuille, Trevor Darrell, Jitendra Malik, and Alexei A Efros. Sequential modeling enables scalable learning for large vision models. *arXiv preprint arXiv:2312.00785*, 2023. 4

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 2, 3, 4

[3] Michael Chang, Tom Griffiths, and Sergey Levine. Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. In *NeurIPs*, 2022. 2

[4] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 3, 4

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Jakob Uszkoreit, Mostafa Dehghani Neil Houlsby, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 4

[6] Alaaeldin El-Nouby, Michal Klein, Shuangfei Zhai, Miguel Angel Bautista, Alexander Toshev, Vaishaal Shankar, Joshua M Susskind, and Armand Joulin. Scalable pre-training of large autoregressive image models. *arXiv preprint arXiv:2401.08541*, 2024. 4

[7] Gamaleldin Fathy Elsayed, Aravindh Mahendran, Sjoerd van Steenkiste, Klaus Greff, Michael Curtis Mozer, and Thomas Kipf. SAVi++: Towards end-to-end object-centric learning from real-world videos. In *NeurIPS*, 2022. 5

[8] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. Genesis: Generative scene inference and sampling with object-centric latent representations. In *ICLR*, 2020. 1

[9] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 4

[10] Oran Gafni, Adam Polyak, Oron Ashual, Shelly Sheynin, Devi Parikh, and Yaniv Taigman. Make-a-scene: Scene-based text-to-image generation with human priors. In *ECCV*, 2022. 4

[11] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T. Freeman. Unsupervised semantic segmentation by distilling feature correspondences. In *ICLR*, 2022. 4

[12] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 5

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 3, 4

[14] Xu Ji, Joao F. Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *ICCV*, 2019. 4

[15] Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization. In *ICLR*, 2022. 2, 3

[16] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. In *NeurIPS*, 2023. 2

[17] Laurynas Karazija, Iro Laina, and Christian Rupprecht. Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. In *NeurIPS Datasets and Benchmarks Track*, 2021. 1

[18] Jinwoo Kim, Janghyuk Choi, Ho-Jin Choi, and Seon Joo Kim. Shepherding slots to objects: Towards stable and robust object-centric learning. In *CVPR*, 2023. 2

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 4

[20] Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional object-centric learning from video. In *ICLR*, 2022. 5

[21] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020. 2

[22] Sindy Löwe, Phillip Lippe, Maja Rudolph, and Max Welling. Complex-valued autoencoders for object discovery. *Transactions on Machine Learning Research*, 2022. 2

[23] Sindy Löwe, Phillip Lippe, Francesco Locatello, and Max Welling. Rotating features for object discovery. *NeurIPs*, 2023. 2

[24] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *CVPR*, 2022. 5

[25] Tom Monnier, Elliot Vincent, Jean Ponce, and Mathieu Aubry. Unsupervised Layered Image Decomposition into Object Prototypes. In *ICCV*, 2021. 1

[26] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 4

[27] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *NeurIPS*, 2019. 4

[28] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023. 1, 2, 3, 4, 5

[29] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate dall-e learns to compose. In *ICLR*, 2022. 2

[30] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In *NeurIPS*, 2022. 3

[31] Michael Tschannen, Manoj Kumar, Andreas Peter Steiner, Xiaohua Zhai, Neil Houlsby, and Lucas Beyer. Image captioners are scalable vision learners too. In *NeurIPs*, 2023. 4

[32] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *NeurIPS*, 2016. 4

[33] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *ICML*, 2016. 4

[34] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. Unsupervised semantic segmentation by contrasting object mask proposals. In *ICCV*, 2021. 5

[35] Nick Watters, Loic Matthey, Chris P. Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for disentangled representations in VAEs. In *ICLR workshops*, 2019. 4

[36] Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. In *NeurIPS*, 2023. 1, 2, 3

[37] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPs*, 2019. 4

[38] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *TMLR*, 2022. 4

[39] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multimodal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2023. 4