RAPID

Vladimir Frolov[1,2], Vitaliy Vorobyov[2,3], Leonid Ugadiarov[2,3], and Aleksandr Panov[2,3]

[1] Moscow Institute of Physics and Technology, Russia
[2] Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Russia
[3] AIRI Institute, Russia

Abstract. The essence of human perception and cognition lies in decomposing complex scenes into object-centric abstractions. For more effective understanding and utilization of graphical representations in computer vision, unsupervised object-centric learning is of paramount importance. However, contemporary models struggle to adapt to diverse domains, often being limited to either synthetic or real data. Recent advancements in unsupervised object segmentation enable the identification of intricate elements within images, leading to improved representation and enhanced performance across various computer vision tasks. Almost all modern models traditionally employ Slot Attention. A promising alternative to the classical Slot Attention, which utilizes a cross-attention mechanism, is the Slot Mixture module based on Gaussian mixture model. However, it demonstrates ambiguous results on real-world data. Based on this, a novel approach to training slot representations is proposed, which employs hierarchical concatenation of key features from the encoder, while the weights of pseudo-attention and mixture components, modeled probabilistically, undergo iterative refinement during a single slot update iteration.

Keywords: Object-Centric Learning · Unsupervised segmentation · Slot Mixture Model · Slot Attention.

1 Introduction

Humans understand the world in terms of discrete objects, which serve as compositional building blocks that can be processed independently and recombined. This compositional model of the world underpins higher-level cognitive abilities such as language, causal reasoning, mathematics, planning, etc., and is crucial for predictable and systematic generalization.

In object-centric learning, the model learns to decompose an image into several components without using any human-annotated data, and then reassemble them into a reconstructed image. Currently, there are many methods for training object-centric representations, as object-centric learning has the potential to enhance the reliability, efficiency, and generalization of models for new tasks, as

well as the interpretability of machine learning algorithms. A significant challenge at present is scaling these methods to complex real-world data, as most models perform well on synthetic data but fail on real-world scenarios.

In object-centric learning, the Slot Attention module [1] has already become the de facto standard for training slot representations. It is based on a cross-attention mechanism, where slots compete with each other in parallel and iteratively to occupy their own region in the pixel space. Slot Attention has improved learning speed and memory efficiency by incorporating parallel slot updates. All recent models utilize this module.

Recently, the Slot Mixture module [2] was introduced, which is based on Gaussian mixture model and demonstrates good results on synthetic datasets. However, attempts to train slot representations on real-world data often end in failure, as the model metrics begin to decline relentlessly at a certain point during training.

In this work, we overcome this limitation, and our main contribution lies in stabilizing the training of slot representations using a module based on Gaussian mixture model. This is demonstrated by metrics that surpass SOTA solutions in the discussed field. Our approach utilizes:

1. Hierarchical concatenation of key features from the encoder, which enhances the quality of the obtained slot representations and the robustness of the model.
2. A slot initializer, which is a transformer-based model that accelerates the convergence of the algorithm.
3. A reimagined algorithm for updating slot representations. Specifically, additional computations and updates to the pseudo-attention coefficients have significantly improved the quality of slot representations and the stability of the algorithm's convergence.

We named our method **R**obust **A**daptive **P**robabilistic **I**nference with **D**INO features (RAPID), with the acronym emphasizing the speed of training slot representations.

2 Related work

All modern models incorporate the Slot Attention module [1] as an indispensable element in their architecture. This module is based on a slot-normalized cross-attention mechanism [3], where slots compete with each other in parallel and iteratively for their own region in the pixel space. Subsequently, SAVi [4] and SAVi++ [5] were introduced based on Slot Attention. These two models are quite similar, but a fundamental difference lies in the use of optical flow [6–8] as

the predicted target in SAVi, while SAVi++ employs depth estimation [9–12] as the target signal.

Around the same time as SAVi, a model called SLATE was introduced [13]. This model serves as an analogue to DALL-E [14], which does not require textual descriptions for image generation. The authors of this work tested their development solely on images. However, shortly thereafter, the same team adapted this model for video and named it STEVE [15]. The fundamental architecture of the model remained unchanged; the authors merely scaled it to handle more complex video data.

In 2023, the scientific community was introduced to a model codenamed DINOSAUR [16], which scales to real-world data. The most notable feature of this development is that the authors used not the input image as the predicted target, but rather the features fed into the Slot Attention module. The authors came up with this idea by reflecting on the paradigm of knowledge sharing between student and teacher models [17].

Another development emerged in 2023 — SLASH [18]. The authors propose two new modules, one of which acts as a low-pass filter, while the other directs the slots to the most important areas of the image. The incorporation of these two modules enhances stability, surpassing previous methods across several metrics; however, this model has only been tested on images.

3 Method

Our approach is architecturally similar to DINOSAUR and follows with the conventional design of object-centric models, akin to an autoencoder, as briefly illustrated in Figure 1. However, before delving into the method description (Section 3.2), we will discuss the classic Slot Mixture Module (Section 3.1), which is one of the key components of our model.

3.1 Slot Mixture Module

We aim to construct a probabilistic model based on the likelihood assumption, considering that all sample objects are independent and identically distributed. Consequently, the likelihood is nothing more than the product of probability densities; in the discrete case, it is the product of probabilities. More formally, we want to build a probabilistic model for an sample $\mathbf{x}_t \in \mathbb{X} = \{x_1, \ldots, x_T\}$, generated by a mixture of distributions with K components and their prior weights $\pi \in \mathbb{R}^K$:

$$\mathbf{x}_t \sim p(\mathbf{x}_t \mid \theta) = \sum_{k=1}^{K} \pi_k p(\mathbf{x}_t \mid \theta_k), \quad P(\mathbb{X} \mid \theta) = \prod_{t=1}^{T} p(\mathbf{x}_t \mid \theta), \quad \sum_{k=1}^{K} \pi_k = 1, \quad (1)$$

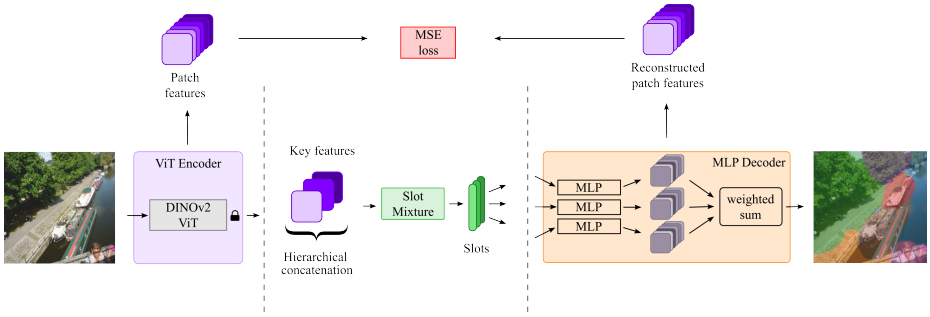

Fig. 1. Overview of the proposed architecture RAPID. The image is processed into a set of patch features $\mathbf{f}_t$ by a frozen DINO ViT model (pre-trained using the self-supervised DINO method). Slot Mixture Module groups the encoded features into a set of slots. The model is trained by reconstructing the DINO features from the slots, either independently per-slot (MLP decoder).

where θ_k are the parameters of the k-th component of the mixture.

We assume that each component of the mixture is modeled by a Gaussian distribution with its parameters, namely the mean μ and the covariance matrix Σ, which we consider to be diagonal. Using the Expectation-Maximization (EM) algorithm, we seek K groups of parameters θ_k that define each sample $\mathbf{x}_t$ and maximize the likelihood estimate of the model $P(\mathbb{X} \mid \theta)$.

E step:

$$P(\mathbb{X}, \mathbb{X}^* \mid \theta^*) = \prod_{t=1}^{T} \prod_{k=1}^{K} [\pi_k p(\mathbf{x}_t \mid \theta_k^*)]^{I(\mathbf{x}_t^* = \mathbf{x}_k^*)}, \tag{2}$$

$$Q(\theta^*, \pi^* \mid \theta, \pi) = \mathbb{E}_{P(\mathbb{X}^* \mid \mathbb{X}, \theta)} [\log P(\mathbb{X}, \mathbb{X}^* \mid \theta^*)], \tag{3}$$

$$p(\mathbf{x}_k^* \mid \mathbf{x}_t) = \frac{p(\mathbf{x}_k^*) p(\mathbf{x}_t \mid \theta_k)}{\sum_{k=1}^{K} p(\mathbf{x}_k^*) p(\mathbf{x}_t \mid \theta_k)} = \frac{\pi_k \mathcal{N}(\mathbf{x}_t \mid \mu_k, \Sigma_k)}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_t \mid \mu_k, \Sigma_k)} = \gamma_{k,t}, \tag{4}$$

$$\mathcal{A}_{k,t} = \frac{\gamma_{k,t}}{\sum_{d=1}^{D} \gamma_{d,k,t}}, \tag{5}$$

where $\mathbb{I}[\cdot]$ is the indicator function; $\gamma_{k,t}$ and $\mathcal{A}_{k,t}$ are the pre-normalized and normalized attention weight coefficients, respectively, for the k-th slot of the t-th sample.

M step:

$$(\theta, \pi) = \arg\max_{(\theta^*, \pi^*)} Q(\theta^*, \pi^* \mid \theta, \pi), \tag{6}$$

$$\pi_k^* = \frac{\sum_{t=1}^{T} \mathcal{A}_{k,t}}{T}, \qquad \mu_k^* = \frac{\sum_{t=1}^{T} \mathcal{A}_{k,t} \mathbf{x}_t}{\sum_{t=1}^{T} \mathcal{A}_{k,t}}, \tag{7}$$

$$\Sigma_k^* = \frac{\sum_{t=1}^{T} \mathcal{A}_{k,t}(\mathbf{x}_t - \mu_k)(\mathbf{x}_t - \mu_k)^T}{\sum_{t=1}^{T} \mathcal{A}_{k,t}}. \quad (8)$$

The Slot Mixture Module can be seen as an extension of the Slot Attention Module with the following key differences:

1. SMM updates not only the mean values but also the covariance values and prior probabilities.
2. The Gaussian density function is used instead of the dot-product attention.
3. Slots are considered not only as mean values of the cluster but as the concatenation of mean and covariance values.

3.2 RAPID

Encoder. Previous work demonstrates that powerful entity extraction tools help scale object-centric methods to more complex data. Therefore, we use a Vision Transformer (ViT), pre-trained using the DINO algorithm [19] on the ImageNet [20]:

$$\mathbf{f}_t = f_{\mathbf{enc}}(\mathbf{x}_t) \in \mathbb{R}^{N \times D_\mathrm{f}}, \qquad \mathbf{f}_t^* = \mathrm{Conv}(\mathrm{Concat}[\mathbf{f}_t^{l-1}, \mathbf{f}_t^{l-2}, \mathbf{f}_t^{l-3}]) \in \mathbb{R}^{N \times D_\mathrm{s}}, \quad (9)$$

where l is the number of blocks in ViT; N is the number of tokens; $D_\mathrm{f}, D_\mathrm{s}$ is the dimensionality of features and slots; $\mathbf{f}_t$ are the features obtained from the teacher model (DINO ViT).

Transformer initializer. As an initial approximation for the algorithm parameters, a zero mean is utilized, and the initialization of the log standard deviation parameter is conducted using the Xavier method with a uniform distribution. These parameters are subsequently refined through trainable transformer blocks and a multilayer perceptron (MLP). The transformer blocks employ a self-attention mechanism, while the MLP consists of a pair of linear layers with a ReLU non-linear activation function in between. The introduction of additional parameterization significantly enhances the convergence speed of the algorithm, reducing the stochasticity of the process.

Stabilized SMM. Instead of the classical Slot Attention, a stabilized version of the Slot Mixture Module is employed, the schematic representation of which is illustrated in Figure 2. The key idea behind the proposed enhancement is that after updating the key parameters of the mixture, the estimate of the prior distribution of the pseudo-attention coefficients γ, which play a crucial role in the model, is also updated. This rethinking of the EM algorithm helps to avoid degradation of the model's performance. This appears to occur because the standard Slot Mixture in modeling slot representations relies on the classical formulation of Gaussian mixtures, where some information is inadequately accounted for during the parameter update process. The estimation of the prior distribution is only updated once per iteration, whereas the algorithm, as it turns out, needs to adjust this estimate more frequently.

Algorithm 1: Stabilized Slot Mixture Module

Input: $\mathbf{f}^* \in \mathbb{R}^{N \times D_s}$ are features; μ, $\Sigma \in \mathbb{R}^{K \times D_s}$, $\pi \in \mathbb{R}^K$ are the parameters and weights of the mixture; $\mathcal{I} = \{1, \ldots, L\}$.

Output: $s \in \mathbb{R}^{N \times 2D_s}$, $\gamma^{**} \in \mathbb{R}^{K \times D_f}$.

for *each* $i \in \mathcal{I}$ **do**

$\text{logits} = f_\theta(\mathbf{f}^*, \mu, \Sigma, \pi)$;

$\gamma = \text{Softmax}(\text{logits}, \text{dim} = 1)$;

$\mathcal{A} = \text{Softmax}(\gamma, \text{dim} = -1)$;

$\pi^* = \mathcal{A}.\text{mean}(\text{dim} = 0)$;

$\mu^* = \text{WeightedMean}(\text{weights} = \mathcal{A}, \text{values} = \mathbf{f}^*)$;

$\mu^* = \text{GRU}(\text{input} = \mu^*, \text{ hidden} = \mu)$;

$\mu^* = \text{MLP}[\text{LayerNorm}(\mu^*)] + \boxed{\text{LayerNorm}(\mu^*)}$;

$\boxed{\text{logits}^* = f_\theta(\mathbf{f}^*, \mu^*, \Sigma, \pi^*)}$;

$\boxed{\gamma^* = \text{Softmax}(\text{logits}^*, \text{dim} = 1)}$;

$\boxed{\mathcal{A}^* = \text{Softmax}(\gamma^*, \text{dim} = -1)}$;

$\Sigma^*_{\text{diag}} = \text{WeightedMean}(\text{weights} = \gamma, \text{values} = (\mathbf{f}^* - \mu^*)^2)$;

$\boxed{\pi^{**} = \mathcal{A}^*.\text{mean}(\text{dim} = 0)}$;

if $i = L$ **then**

$\boxed{\text{logits}^{**} = f_\theta(\mathbf{f}^*, \mu^*, \Sigma^*, \pi^{**})}$;

$\boxed{\gamma^{**} = \text{Softmax}(\text{logits}^{**}, \text{dim} = 1)}$;

$\boxed{\gamma_{\text{new}} = \text{WNConv}(\gamma^{**})}$;

$\text{slots} = \text{concat}([\mu^*, \Sigma^*])$;

return slots, γ_{new};

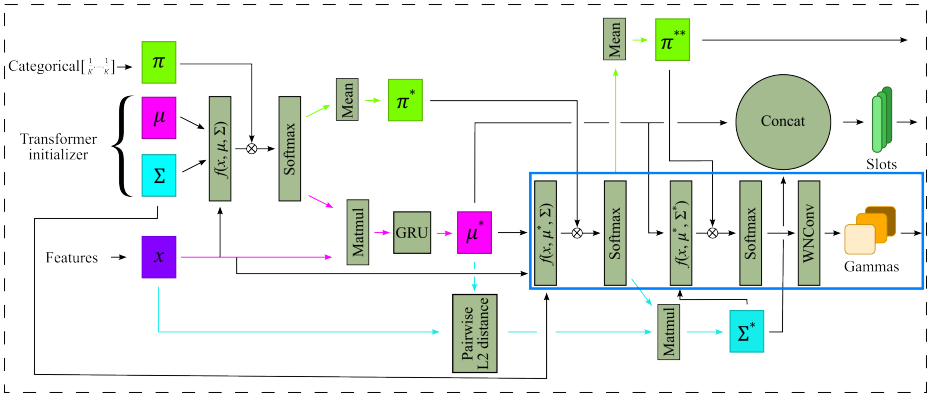

Fig. 2. Stabilized SMM. The proposed improvement is outlined in the blue rectangle.

Table 1. Overview of datasets.

	COCO	PASCAL VOC	ClevrTex
Train Size	118287	10582	35000
Validation Size	5000	2913	5000
Image/Crop Size	224	224	224
Random Horizontal Flip	Yes	Yes	No
Train Crop Type	Center	Random	Full
Validation Crop Type	Center	Center	Full

Decoder. We use a MLP decoder:

$$\hat{\mathbf{m}}_{t,k}, \hat{\mathbf{f}}_{t,k} = f_{\mathbf{dec}}(s_{t,k}), \quad \mathbf{m}_t = \underset{K}{\mathrm{Softmax}}(\hat{\mathbf{m}}_{t,k}), \quad \hat{\mathbf{f}}_t = \sum_{k=1}^{K} \mathbf{m}_{t,k} \odot \hat{\mathbf{f}}_{t,k}. \tag{10}$$

The optimization problem is formulated as follows:

$$\mathcal{L}_{\mathrm{rec}} = \min_{\hat{\mathbf{f}}_t \in \mathcal{F}} \sum_{t=1}^{T} \|\mathbf{f}_t - \hat{\mathbf{f}}_t\|_2^2. \tag{11}$$

4 Experiments

4.1 Datasets and metrics

Here, we consider three datasets: PASCAL VOC 2012 [21], MS COCO 2017 [22], and ClevrTex [23]. An overview of the datasets is presented in table 1. We train RAPID using the Adam optimizer [24], with a learning rate of 4×10^{-4} and a linear warm-up of the learning rate over 5×10^3 iterations. After that, the learning rate decreases according to a cosine schedule until the scaling factor of the learning rate reaches 0.2. Following this, there is a smooth transition to an exponential schedule—linear growth with cosine decay and an exponential tail. During training, the gradient norm is capped at 1.0. In all experiments, a batch size of 64 is used. For the PASCAL VOC dataset, 6 slots are used, while 7 slots are used for the COCO datasets. A patch size of 16 is used for all datasets.

We evaluate the quality of the models using the Foreground Adjusted Rand Index (FG-ARI), a metric that measures the similarity of clusters while ignoring the background, and classic ARI:

$$\mathrm{RI} = \frac{2(a+b)}{n(n-1)}, \qquad \mathrm{ARI} = \frac{\mathrm{RI} - \mathbb{E}[\mathrm{RI}]}{\max \mathrm{RI} - \mathbb{E}[\mathrm{RI}]}, \tag{12}$$

where a is the number of pairs of objects with the same labels in both partitions $\mathbb{Y}$ and $\hat{\mathbb{X}}$; b is the number of pairs of objects with different labels in both partitions; and n is the number of objects in the sample.

Table 2. Comparison with classic SMM.

	ARI	mBO
Default SMM	42.33 ± 0.23	28.07 ± 0.13
Default SMM with scheduling	43.26 ± 0.23	28.91 ± 0.11
Default SMM with hierarchical concat.	44.41 ± 0.19	29.15 ± 0.12
RAPID (Ours)	$\mathbf{47.04 \pm 0.46}$	$\mathbf{30.48 \pm 0.10}$

Additionally, we compute a metric based on Intersection over Union (IoU) called mean Best Overlap (mBO) [25]. mBO assesses how well the masks fit the objects and is calculated by assigning each ground truth mask to the predicted mask with the highest overlap, and then averaging the values of the assigned mask pairs:

$$\text{mBO} = \frac{1}{TC} \sum_{t=1}^{T} \sum_{c=1}^{C} \frac{\mathbf{y}_{c,t} \cap \arg\max_{\hat{\mathbf{x}}_{c,t} \in \hat{\mathbb{X}}} \mathbf{y}_{c,t} \cap \hat{\mathbf{x}}_{c,t}}{\mathbf{y}_{c,t} \cup \arg\max_{\hat{\mathbf{x}}_{c,t} \in \hat{\mathbb{X}}} \mathbf{y}_{c,t} \cap \hat{\mathbf{x}}_{c,t}}, \tag{13}$$

where C is the number of classes; $\mathbf{y}_{c,t}$ is the true label for the instance with index t in class c; and $\hat{\mathbf{x}}_{c,t}$ is the prediction for instance t in class c.

4.2 Results

Sequential experiments on the COCO dataset demonstrate that all discussed innovations are valid and play their role in the algorithm. The proposed rethinking of the EM algorithm helps to avoid degradation of the model's performance. This seems to occur because the standard Slot Mixture in modeling slot representations relies on the classical formulation of Gaussian mixtures, where some information is underutilized over the chosen number of iterations. In such cases, a more accurate parameterization is lost due to the fact that we process information through subsamples (batches).

It is worth noting that the probabilistic formulation ensures the robustness of the algorithm; additional updates to the pseudo-attention coefficients and mixture component weights do not lead to overfitting, which is also very important. Most importantly, these innovations allow us to circumvent the limitation associated with the drastic decline in clustering quality in the standard implementation of the Slot Mixture module, as shown in Table 2.

Unlike the standard implementations with the Slot Mixture module, where a trend of significant quality deterioration emerges quite rapidly, our method converges more stably and does not exhibit degradation.

We also demonstrate superiority over DINOSAUR on the COCO dataset, with the results presented in Table 3.

Table 3. Comparison with DINOSAUR on COCO dataset.

	FG-ARI	mBO
DINOSAUR	40.5 ± 0.2	27.7 ± 0.2
RAPID (Ours)	$\mathbf{52.23 \pm 0.18}$	$\mathbf{30.48 \pm 0.10}$

PASCAL VOC

ClevrTex

5 Conclusions

In this work, we propose an approach to stabilize the training of slot representations for SMM, called RAPID. This method aggregates key features from the encoder through hierarchical concatenation. The initialization of slots occurs from a normal distribution, with parameters derived from the outputs of a fully parameterized model that uses features from the encoder after hierarchical concatenation as its basis. The EM algorithm has been rethought; it is suggested that during one iteration, after updating the key mixture parameters, the distribution of pseudo-attention coefficients—crucial for the model—should also be updated. Experimental results indicate that this approach works effectively not only in terms of stabilizing training but also in achieving results that surpass DINOSAUR across selected metrics.

References

1. F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, T. Kipf: Object-Centric Learning with Slot Attention. NeurIPS 2020
2. D. Kirilenko, V. Vorobyov, A. Kovalev, A. Panov: Object-Centric Learning with Slot Mixture Module. ICLR 2024
3. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, Illia Polosukhin: Attention Is All You Need. NeurIPS 2017
4. T. Kipf, G. F. Elsayed, A. Mahendran, A. Stone, S. Sabour, G. Heigold, R. Jonschkowski, A. Dosovitskiy, K. Greff: Conditional Object-Centric Learning from Video. ICLR 2022
5. G. F. Elsayed, A. Mahendran, S. v. Steenkiste, K. Greff, M. C. Mozer, T. Kipf: SAVi++: Towards End-to-End Object-Centric Learning from Real-World Videos. NeurIPS 2022
6. A. Mahendran, J. Thewlis, A. Vedaldi: Cross pixel optical-flow similarity for self-supervised learning. Asian Conference on Computer Vision 2019
7. J. Walker, A. Gupta, M. Hebert: Dense optical flow prediction from a static image. ICCV 2015

8. D. Sun, X. Yang, M. Liu, J. Kautz: PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. CVPR 2018
9. H. Laga, L. V. Jospin, F. Boussaid, M. Bennamoun: A survey on deep learning techniques for stereo-based depth estimation. Transactions on Pattern Analysis and Machine Intelligence 2020
10. Y. Ming, X. Meng, C. Fan, H. Yu: Deep learning for monocular depth estimation: A review. Neurocomputing 2021
11. L. Wang, J. Zhang, O. Wang, Z. Lin, H. Lu: SDC-depth: Semantic divide-andconquer network for monocular depth estimation. CVPR 2020
12. H. Fu, M. Gong, C. Wang, K. Batmanghelich, D. Tao: Deep ordinal regression network for monocular depth estimation. ICLR 2022
13. G. Singh, F. Deng, S. Ahn: Illiterate DALL-E Learns to Compose. ICLR 2022
14. A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, I. Sutskever: Zero-Shot Text-to-Image Generation. ICML 2021
15. G. Singh, Y. Wu, S. Ahn: Simple Unsupervised Object-Centric Learning for Complex and Naturalistic Videos. NeurIPS 2022
16. M. Seitzer, M. Horn, A. Zadaianchuk, D. Zietlow, T. Xiao, C. Simon-Gabriel, T. He, Z. Zhang, B. Schölkopf, T. Brox, F. Locatello: Bridging the Gap to Real-World Object-Centric Learning. ICLR 2023
17. G. E. Hinton, O. Vinyals, J. Dean: Distilling the Knowledge in a Neural Network. NeurIPS 2014
18. J. Kim, J. Choi, H. Choi, S. J. Kim: Shepherding Slots to Objects: Towards Stable and Robust Object-Centric Learning. CVPR 2023
19. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR 2021
20. J. Deng, W. Dong, R. Socher, L. Li, K. Li, L. Fei-Fei: ImageNet: A large-scale hierarchical image database. CVPR 2009
21. M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman: The PASCAL Visual Object Classes Challenge 2012 (VOC2012). URL http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html, 2012
22. T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick: Microsoft COCO: Common Objects in Context. ECCV 2014
23. L. Karazija, I. Laina, C. Rupprecht: Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. CoRR, abs/2111.10265, 2021. URL https://arxiv.org/abs/2111.10265
24. D. P. Kingma, J. Ba: Adam: A Method for Stochastic Optimization. URL https://arXiv.org/abs/1412.6980
25. J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, J. Malik: Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation. Transactions on Pattern Analysis and Machine Intelligence 2017
26. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016). https://doi.org/10.10007/1234567890
27. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999)
28. Author, A.-B.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010)
29. LNCS Homepage, http://www.springer.com/lncs. Last accessed 4 Oct 2017