Inverted-Attention Transformers can Learn Object Representations: Insights from Slot Attention

Yi-Fu Wu[*]
Rutgers University

Klaus Greff
Google DeepMind

Gamaleldin F. Elsayed
Google DeepMind

Michael C. Mozer
Google DeepMind

Thomas Kipf
Google DeepMind

Sjoerd van Steenkiste[†]
Google Research

Abstract

Visual reasoning is supported by a causal understanding of the physical world, and theories of human cognition suppose that a necessary step to causal understanding is the discovery and representation of high-level entities like objects. *Slot Attention* is a popular method aimed at object-centric learning, and its popularity has resulted in dozens of variants and extensions. To help understand the core assumptions that lead to successful object-centric learning, we take a step back and identify the minimal set of changes to a standard Transformer architecture to obtain the same performance as the specialized Slot Attention models. We systematically evaluate the performance and scaling behaviour of several "intermediate" architectures on seven image and video datasets from prior work. Our analysis reveals that by simply inverting the attention mechanism of Transformers, we obtain performance competitive with state-of-the-art Slot Attention in several domains.

1 Introduction

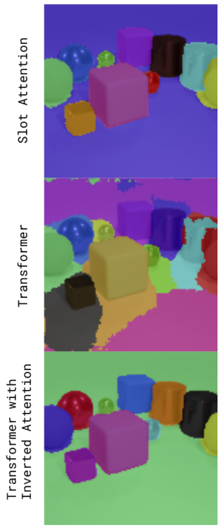

Figure 1

Human understanding of the natural world is rooted in the perception of entities like objects, which form the basic building blocks for causal prediction and reasoning in everyday situations [1, 2]. In contrast, standard neural network architectures like Transformers only partially succeed at learning representations that separately encode information about individual objects, especially in the absence of instance-level supervision [3, 4, 5]. To overcome this issue, a vast literature has emerged on more specialized *object-centric neural networks*, capable of discovering and representing information about objects in a self-supervised manner [6, 7, 8, 9, 10, 11, 12, 13]. (For an overview, see Greff et al. [3].)

Though there are notable exceptions [e.g., 14, 15], many recent approaches follow a fairly standard recipe derived from *Slot Attention* [16]. In Slot Attention, an image—encoded as a set of input tokens—is soft partitioned into K object *slots*. (The term *queries* is also used in related literature [17, 18].) Partitioning is a recurrent mechanism in which slots are initialized to values sampled from a distribution (with learnable parameters) and are then iteratively updated via scaled dot-product attention to the input tokens [19]. Neural network components that apply the updates, typically GRUs [20], share parameters between iterations. Notably, attention maps are computed via a kind of *inverted attention* [21], which induces competition

[*] Work done while the author was a student researcher at Google Research.

[†] Correspondence to svansteenkiste@google.com.

Accepted to the NeurIPS 2023 Workshop on Causal Representation Learning.

Algorithm 1: Slot Attention

```
inputs = LayerNorm(inputs)
for _ in range(N):
    updates = ScaledDotProductAttention(
                        q=LayerNorm(slots),
                        kv=inputs,
                        axis="queries",
                        renormalization=True
                )
    slots = GRU∗(slots, updates)
    slots += MLP∗(LayerNorm(slots))
```

Algorithm 2: Transformer (cross-attention)

```

for _ in range(N):
    updates = MultiHeadDotProductAttention(
                        q=LayerNorm(slots),
                        kv=inputs,
                        axis="keys"

                )
    slots += Dropout(updates)
    slots += Dropout(MLP(LayerNorm(slots)))
slots = LayerNorm(slots)
```

Figure 2: Comparing the Slot Attention and Transformer algorithms. * is used to indicate weight-sharing between iterations in Slot Attention. Other notable differences include the normalization axis used in the cross-attention operation (and subsequent renormalization), and the gated update using a GRU in place of a residual update.

between the slots to explain the input tokens. A prominent account for why this may lead to slot representations that capture individual objects stems from its connection to Soft K-Means or Neural EM [7, 22], relatedly a more general theory of the feasibility of learning object representations was recently proposed [23]. In combining Slot Attention with different encoders and decoders, its capabilities for learning representations of abstract entities have been extended to video [24, 25], 3D scenes [26, 27, 28], action sequences [29], and morphemes in language [30].

Although many variants and extensions of Slot Attention have been proposed [31, 32, 33, 34], the core assumptions that lead to successful object-centric learning remain elusive. Here we take a step back and ask what aspects of Slot Attention are actually critical for object discovery. We tackle this question by drawing a connection between Slot Attention and Transformers (Figure 2) which allows us to identify a minimal set of changes to a standard Transformer decoder architecture [19, 35] that unlock the capacity for object discovery. In particular, we perform an extensive analysis of architectural variants of Slot Attention and Transformers on a range of commonly-used synthetic and real-world datasets. Our study reveals that by simply 'inverting' the attention mechanism of Transformers, we obtain competitive performance for object-representation learning (see Figure 1). We further demonstrate that it is possible to substitute this *Inverted-Attention Transformer* for Slot Attention in SAVi [24] and OSRT [26] while obtaining comparable and sometimes improved performance.

2 Comparing Slot Attention to Transformers

Figure 2 presents a side-by-side comparison of Slot Attention [16] and Transformer Decoders [19]. Because Slot Attention only performs cross-attention, we consider pre-LayerNorm Transformer Decoders without self-attention [35]. Given an encoding of the input as a set of tokens and a set of initialized slots (or queries), the two algorithms proceed in a similar manner:

Multiple Iterations. Slots are updated over N iterations. In case of Slot Attention these iterations use shared weights, which lend the interpretation of them being state updates that converge on fixed-point attractors [31, 36], while in Transformers each iteration corresponds to a layer having potentially different weights.

Scaled Dot-Product Attention. Slots are updated by individually attending to the input tokens using a form of scaled dot-product attention [19], typically using multiple heads in case of Transformers. Here an important distinction is the normalization of the attention map, which in Slot Attention proceeds by transposing the axis over which the softmax is taken followed by a renormalization step to ensure the attention weights into each slot sum to one (see also Appendix B.1). This operation can also be understood as a kind of "Inverted Dot-Product Attention Routing" (without the extra LayerNorm) [21] followed by an additional renormalization step.

Update and Transformation. After attending, the slots are updated by a learned transformation of the resulting attention vectors. In Transformers, a residual update is applied followed by

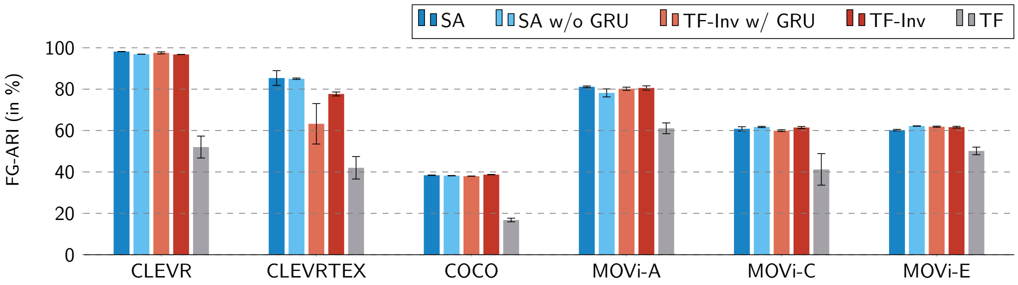

Figure 3: FG-ARI of Slot Attention and Transformer variants on six image datasets.

transformation by an MLP. Here Slot Attention leverages a *gated* update implemented as a GRU [20] that includes a transformation. Slot Attention also includes an additional optional transformation with an MLP.

We note that various smaller differences exist as well, though these are not expected to significantly affect the behavior of the model. For example, Slot Attention starts by applying Layer Normalization (LN) [37] to the inputs, while pre-LN Transformers add LN at the very end. Further, Transformers sometimes include Dropout [38] in the update and transformation steps, though no consistent improvement was found in applications to images [39].

3 Experiments

Complete details of our experiments can be found in Appendix B.

3.1 Object Discovery in Images

Experimental Setup. We focus on the task of unsupervised object discovery, which is commonly used to evaluate the capacity for object-centric representation learning [7, 16, 22]. We adopt the object discovery architecture from Locatello et al. [16] and conduct experiments on a wide range of image datasets: CLEVR with masks [40, 41], CLEVRTex [42], COCO [43], and individual frames from MOVi-A, MOVi-C, and MOVi-E [44]. For CLEVRTex, we use a ResNet34 encoder which was previously found to work well on this dataset [45]. For the COCO and MOVi datasets, we adopt the DINOSAUR [13] architecture whereby features from a pre-trained DINO-ViT [4, 39] are used as inputs to Slot Attention and as the reconstruction target. In other cases we train models to reconstruct pixels directly. To initialize slots/queries, we sample from a Gaussian distribution with learned mean and variance as in Locatello et al. [16]. We report 3 seeds for all our experiments and train for 300K steps. Unless otherwise stated we use $N = 3$ iterations for (variations of) Slot Attention and Transformers.

Benchmarking Variations of Slot Attention and Transformers. To determine what aspects in Slot Attention are essential for successful object-centric learning, we lay out a space of models that includes Slot Attention and Transformers and evaluate variations that are positioned in between. For these experiments we substitute a particular variant for Slot Attention, while keeping other components of the architecture, such as the encoder or decoder, fixed for each dataset (details are presented in Appendix B.1). To summarize, the variations include:

TF: a standard pre-LayerNorm Transformer decoder without self-attention.

TF-Inv: same as TF, except for using inverted attention and renormalization.

TF-Inv w/ GRU: TF-Inv with a GRU in each layer to perform the slot update (line 9 in Figure 2)[3].

SA w/o GRU: Slot Attention, but with the GRU replaced with a residual update.

SA: standard Slot Attention.

[3] We also explored GRU weight sharing across layers of *TF-Inv w/ GRU*, but we saw no significant effect.

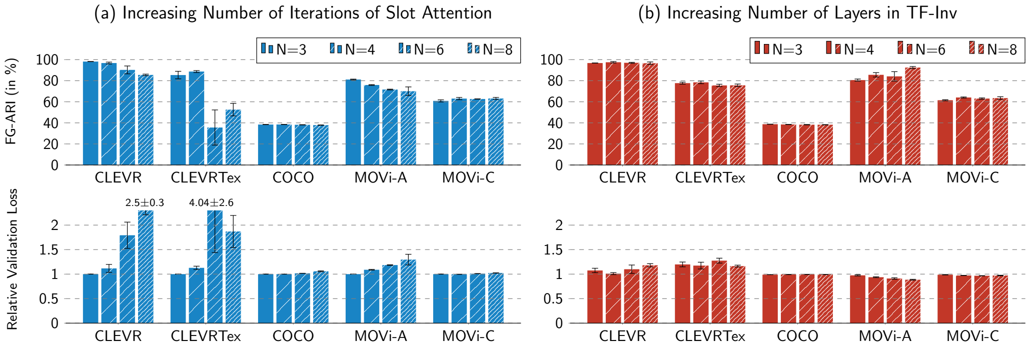

Figure 4: FG-ARI and Relative Validation Loss as we increase N for (a) *SA* and (b) *TF-Inv*. We report the validation loss relative to *SA* using $N = 3$ to be able to compare between datasets.

We evaluate these variants in terms of emergent instance segmentation. Figure 3 reports the foreground ARI [22, 46, 47] scores (hereafter, FG-ARI) for each of these variations, which are obtained by comparing ground-truth instance segmentations to those that were inferred after decoding the slots using a spatial broadcast decoder [22, 48] (e.g., as in Figure 1). It can be seen that *TF* generally performs the worst, though surprisingly, other variants are able to achieve an FG-ARI comparable to that of *SA*. Crucially, we note that simply inverting the attention operation is sufficient to facilitate object discovery with Transformers as *TF-Inv* generally performs as well as other variants closer to Slot Attention (e.g., those that share weights between iterations or incorporate the GRU). Foreground mIoU [49] reveals a similar trend, as shown in Figure 9.

Scaling by increasing N. We hypothesize that because the number of free parameters—and hence capacity—of *TF-Inv* increases linearly with N but it does not for *SA*, *TF-Inv* may have a performance advantage. We ran experiments for various values of N, using the same N during training and evaluation, across several datasets and present the results in Figure 4. We observe that for CLEVR and MOVi-A, the performance of *SA* does degrade as $N > 3$, both in terms of validation loss and in terms of FG-ARI. *TF-Inv*, on the other hand, generally maintains or slightly improves performance as its number of layers increases. This provides some evidence supporting our hypothesis that TF-Inv scales better to larger N, which opens up an interesting avenue for future work.

3.2 Application of *Inverted-Attention Transformers* to Other Domains

Slot Attention for Video (SAVi). We consider SAVi [24, 25], which relies on Slot Attention as a key component to enable the discovery and tracking of objects in videos. We plug *TF-Inv* into SAVi as a replacement and evaluate it on videos of the MOVi datasets. We plot the FG-ARI in Figure 5. It can be observed that SAVi with *TF-Inv* performs comparably on MOVi-A and MOVi-E, though it is unstable on MOVi-C. Interestingly, adding back the GRU helps *TF-Inv* to match SAVi performance, suggesting it is of some importance in this setting.

Object Scene Representation Transformer (OSRT). OSRT [26] is an object-centric model that uses Slot Attention for novel view synthesis of 3D scenes. Here, in addition to FG-ARI, we also report Peak Signal-to-Noise Ratio (PSNR), which captures the ability of the model to reconstruct *novel* views. We run a set of experiments on the MultiShapeNet-Hard (MSN-Hard) [50] dataset where we use *TF-Inv* instead of *SA* and report results in Table 1. We see that while *SA* performs better than *TF-Inv* when $N = 1$ (the standard setting in Sajjadi et al. [26]), increasing to $N = 5$ harms performance for Slot Attention, both in terms of PSNR and FG-ARI. In contrast, *TF-Inv* scales well with additional layers, yielding a higher FG-ARI and improved PSNR.

4 Discussion

In this work, we investigated the core assumptions underlying successful object-centric learning in attention-based architectures derived from Slot Attention. Through a comprehensive study, we

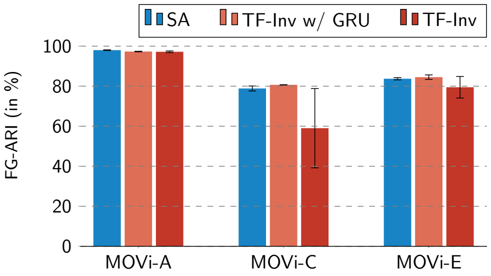

Figure 5: FG-ARI of Slot Attention, TF-Inv, and TF-Inv w/ GRU per Layer on the video version of the MOVi datasets.

OSRT Model	PSNR (dB)	FG-ARI (%)
w/ SA ($N = 1$)	22.07 ± 0.20	$\mathbf{75.66} \pm 2.09$
w/ SA ($N = 5$)	21.73 ± 0.14	68.19 ± 1.04
w/ TF-Inv ($N = 1$)	21.20 ± 0.57	73.58 ± 2.06
w/ TF-Inv ($N = 5$)	$\mathbf{22.21} \pm 0.07$	75.39 ± 0.28

Table 1: PSNR and FG-ARI when substituting *SA* in OSRT [26] with *TF-Inv*.

discovered that inverted dot-product attention is a crucial component, which can readily be integrated in Transformers by changing the axis along which the attention normalization takes place (i.e., the 'query' axis instead of the 'key' axis) and subsequently renormalizing as in Locatello et al. [16]. In particular, a modified pre-LayerNorm transformer (cross-attention only, i.e., *TF-Inv*), was shown to perform on par with Slot Attention on a variety of datasets, to scale well with model depth, and to be broadly applicable to other domains where Slot Attention is used such as SAVi and OSRT. In the medium term, there is an exciting opportunity to apply these insights to applications of Transformers more broadly, i.e., outside the context of unsupervised object-centric representation learning, including related architectures for object detection [17, 51, 52].

Open Questions

Although Inverted-Attention Transformers make considerable progress toward simplifying Slot Attention (and extending its applicability), our investigations revealed several additional factors that influence the performance of both *TF-Inv* and *SA*. Although these observations require more thorough study, we present our initial investigations below hoping that they spur further insights from the research community.

Slot Initialization. To initialize the slots, our previous experiments sample from a shared Gaussian distribution with learnable parameters. Alternatively, it is possible to learn the set of slot initializations directly [17, 24, 25, 32]. We demonstrate the effect of this change in Figure 6 on the CLEVR, CLEVRTex, and MOVi-A datasets. It can be seen that when learning the set of slot initializations, *TF-Inv* generally achieves slightly lower FG-ARI on CLEVR and MOVi-A and considerably lower FG-ARI on CLEVRTex. Moreover, the GRU gating improves performance of *TF-Inv* in this setting, especially on the CLEVRTex dataset. *TF* can also be seen to perform considerably better in this setting, though a consistent ordering among the model variants across datasets can not be established.

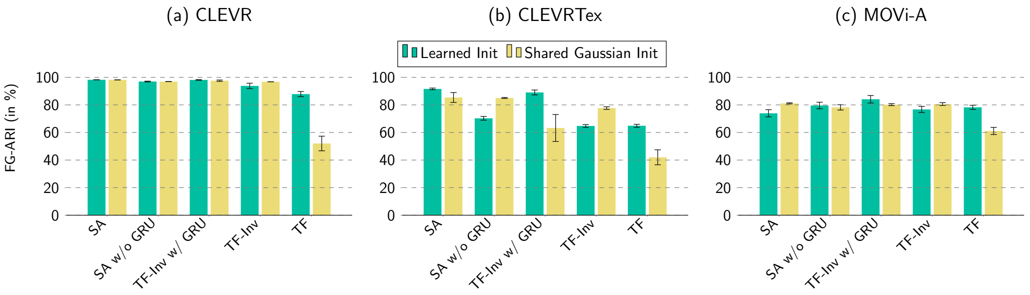

Figure 6: FG-ARI of the different model variants when using learned initializations per slot or learned Gaussian initializations shared between slots.

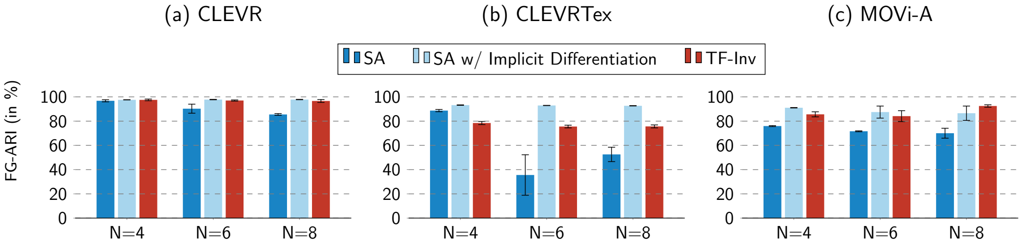

Figure 8: Comparing *SA*, *SA-ID* and *TF-Inv* in terms of FG-ARI as N increases.

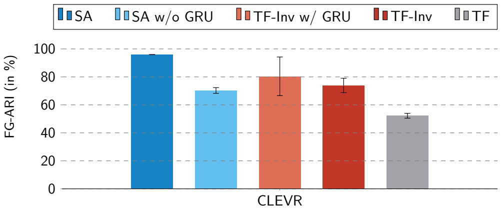

Figure 7: FG-ARI of Slot Attention and Transformer variants using a Transformer Decoder on CLEVR.

Sensitivity to Decoder. In our previous experiments we adopted the Spatial Broadcast Decoder [48] to reconstruct the image from the slots following Locatello et al. [16]. Though this is arguably the standard approach, it is a relatively weak decoder that incorporates a strong inductive bias for modeling images as a composition of parts. Recently, Singh et al. [33] proposed using a more powerful Transformer Decoder that cross-attends into the slots in place of the Spatial Broadcast Decoder and subsequent work [53] has shown this type of decoder facilitates object discovery in more visually complex scenes.

To investigate the sensitivity to this choice, we evaluated each model variant using a Transformer Decoder instead of the Spatial Broadcast Decoder on the CLEVR dataset. From Figure 7, we observe that while *TF-Inv* still outperforms *TF* in this setting, it performs considerably worse than *SA* overall. Interestingly, the GRU gating appears to have a significant effect in this case. Finally, we note that all models perform worse with the Transformer Decoder than with the Spatial Broadcast Decoder (see Figure 3), suggesting that the more complex Transformer Decoder is not helpful for this dataset containing images of simple geometric shapes.

Slot Attention with 'Implicit Differentiation' Chang et al. [31] recently proposed using 'implicit differentiation' to improve upon Slot Attention, which effectively comes down to propagating the gradient through only the last iteration when training the model (i.e., it uses 1-step truncated backpropagation through time [54]). Further, Jia et al. [32] propose a straight-through estimator [55] to extend this approach to learned slot initializations. In Figure 8 we compare to *SA w/ Implicit Differentiation* (*SA-ID*) for varying N and confirm that it often improves over vanilla *SA*. Although we observe that *TF-Inv* outperforms or matches *SA-ID* when scaling to a larger number of layers for CLEVR and MOVi-A, it fails to do so on the CLEVRTex dataset, where *SA-ID* achieves over 90% FG-ARI.

Surprisingly, in preliminary experiments applying *SA-ID* to the OSRT setting, we observed that using $N = 5$ iterations yields 23.64 dB PSNR and 78.13% FG-ARI, which is a substantial improvement over both *SA* and *TF-Inv*. At the moment is not entirely clear why *SA-ID* can yield significant improvements in some conditions, but not consistently. Hence, it would be interesting to better understand the conditions under which improvements are obtained and to apply any insights to *TF-Inv*.

Acknowledgments and Disclosure of Funding

We thank Aravindh Mahendran for general advice and detailed feedback on an early version of this paper.

References

[1] Elizabeth S Spelke and Katherine D Kinzler. Core knowledge. *Developmental science*, 10(1):89–96, 2007.

[2] Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021.

[3] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

[4] 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] Luke Melas-Kyriazi, Christian Rupprecht, Iro Laina, and Andrea Vedaldi. Deep spectral methods: A surprisingly strong baseline for unsupervised semantic segmentation and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8364–8375, 2022.

[6] SM Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Geoffrey E Hinton, et al. Attend, infer, repeat: Fast scene understanding with generative models. *Advances in neural information processing systems*, 29, 2016.

[7] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. Neural expectation maximization. *Advances in Neural Information Processing Systems*, 30, 2017.

[8] Sjoerd van Steenkiste, Michael Chang, Klaus Greff, and Jürgen Schmidhuber. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *International Conference on Learning Representations*, 2018.

[9] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

[10] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: unsupervised object-oriented scene representation via spatial attention and decomposition. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

[11] Jindong Jiang, Sepehr Janghorbani, Gerard De Melo, and Sungjin Ahn. Scalor: Generative world models with scalable object representations. In *International Conference on Learning Representations*, 2019.

[12] Gautam Singh, Yeongbin Kim, and Sungjin Ahn. Neural systematic binder. In *The Eleventh International Conference on Learning Representations*, 2022.

[13] 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 *The Eleventh International Conference on Learning Representations*, 2023.

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

[15] Martin Engelcke, Oiwi Parker Jones, and Ingmar Posner. Genesis-v2: Inferring unordered object representations without iterative refinement. *Advances in Neural Information Processing Systems*, 34:8085–8094, 2021.

[16] 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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[17] 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*, pages 213–229. Springer, 2020.

[18] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2021.

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[20] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[21] Yao-Hung Hubert Tsai, Nitish Srivastava, Hanlin Goh, and Ruslan Salakhutdinov. Capsules with inverted dot-product attention routing. In *International Conference on Learning Representations*, 2019.

[22] 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*, pages 2424–2433. PMLR, 2019.

[23] Jack Brady, Roland S. Zimmermann, Yash Sharma, Bernhard Schölkopf, Julius Von Kügelgen, and Wieland Brendel. Provably learning object-centric representations. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 3038–3062. PMLR, 23–29 Jul 2023.

[24] 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 *International Conference on Learning Representations*, 2021.

[25] 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 *Advances in Neural Information Processing Systems*, 2022.

[26] Mehdi SM Sajjadi, Daniel Duckworth, Aravindh Mahendran, Sjoerd van Steenkiste, Filip Pavetic, Mario Lucic, Leonidas J Guibas, Klaus Greff, and Thomas Kipf. Object scene representation transformer. *Advances in Neural Information Processing Systems*, 35:9512–9524, 2022.

[27] Karl Stelzner, Kristian Kersting, and Adam R Kosiorek. Decomposing 3d scenes into objects via unsupervised volume segmentation. *arXiv preprint arXiv:2104.01148*, 2021.

[28] Allan Jabri, Sjoerd van Steenkiste, Emiel Hoogeboom, Mehdi SM Sajjadi, and Thomas Kipf. Dorsal: Diffusion for object-centric representations of scenes *et al.*. *arXiv preprint arXiv:2306.08068*, 2023.

[29] Anand Gopalakrishnan, Kazuki Irie, Jürgen Schmidhuber, and Sjoerd van Steenkiste. Unsupervised learning of temporal abstractions with slot-based transformers. *Neural Computation*, 35(4):593–626, 2023.

[30] Melika Behjati and James Henderson. Inducing meaningful units from character sequences with slot attention. *arXiv preprint arXiv:2102.01223*, 2021.

[31] Michael Chang, Tom Griffiths, and Sergey Levine. Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. *Advances in Neural Information Processing Systems*, 35:32694–32708, 2022.

[32] Baoxiong Jia, Yu Liu, and Siyuan Huang. Improving object-centric learning with query optimization. In *The Eleventh International Conference on Learning Representations*, 2022.

[33] Gautam Singh, Fei Deng, and Sungjin Ahn. Illiterate dall-e learns to compose. In *International Conference on Learning Representations*, 2021.

[34] Yan Zhang, David W. Zhang, Simon Lacoste-Julien, Gertjan J. Burghouts, and Cees G. M. Snoek. Unlocking slot attention by changing optimal transport costs. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 41931–41951. PMLR, 23–29 Jul 2023.

[35] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020.

[36] Richard S Zemel and Michael C Mozer. Localist attractor networks. *Neural Computation*, 13(5):1045–1064, 2001.

[37] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[38] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[39] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[40] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017.

[41] Rishabh Kabra, Chris Burgess, Loic Matthey, Raphael Lopez Kaufman, Klaus Greff, Malcolm Reynolds, and Alexander Lerchner. Multi-object datasets, 2019.

[42] Laurynas Karazija, Iro Laina, and Christian Rupprecht. Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[43] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, volume 8693 of *Lecture Notes in Computer Science*, pages 740–755. Springer, 2014.

[44] Klaus Greff, Francois Belletti, Lucas Beyer, Carl Doersch, Yilun Du, Daniel Duckworth, David J Fleet, Dan Gnanapragasam, Florian Golemo, Charles Herrmann, Thomas Kipf, Abhijit Kundu, Dmitry Lagun, Issam Laradji, Hsueh-Ti (Derek) Liu, Henning Meyer, Yishu Miao, Derek Nowrouzezahrai, Cengiz Oztireli, Etienne Pot, Noha Radwan, Daniel Rebain, Sara Sabour, Mehdi S. M. Sajjadi, Matan Sela, Vincent Sitzmann, Austin Stone, Deqing Sun, Suhani Vora, Ziyu Wang, Tianhao Wu, Kwang Moo Yi, Fangcheng Zhong, and Andrea Tagliasacchi. Kubric: a scalable dataset generator. 2022.

[45] Ondrej Biza, Sjoerd van Steenkiste, Mehdi S. M. Sajjadi, Gamaleldin Fathy Elsayed, Aravindh Mahendran, and Thomas Kipf. Invariant slot attention: Object discovery with slot-centric reference frames. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 2507–2527. PMLR, 2023.

[46] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical association*, 66(336):846–850, 1971.

[47] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2:193–218, 1985.

[48] Nicholas Watters, Loïc Matthey, Christopher P. Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *CoRR*, abs/1901.07017, 2019.

[49] Paul Jaccard. Distribution de la flore alpine dans le bassin des dranses et dans quelques régions voisines. *Bull Soc Vaudoise Sci Nat*, 37:241–272, 1901.

[50] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas Funkhouser, and Andrea Tagliasacchi. Scene Representation Transformer: Geometry-Free Novel View Synthesis Through Set-Latent Scene Representations. *CVPR*, 2022.

[51] Georg Heigold, Matthias Minderer, Alexey Gritsenko, Alex Bewley, Daniel Keysers, Mario Lučić, Fisher Yu, and Thomas Kipf. Video owl-vit: Temporally-consistent open-world localization in video. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13802–13811, October 2023.

[52] Michael E Sander, Pierre Ablin, Mathieu Blondel, and Gabriel Peyré. Sinkformers: Transformers with doubly stochastic attention. In *International Conference on Artificial Intelligence and Statistics*, pages 3515–3530. PMLR, 2022.

[53] Gautam Singh, Yi-Fu Wu, and Sungjin Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. In *Advances in Neural Information Processing Systems*, 2022.

[54] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[55] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

[56] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[57] Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, et al. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, pages 7480–7512. PMLR, 2023.

[58] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[59] Mehdi S. M. Sajjadi, Henning Meyer, Etienne Pot, Urs Bergmann, Klaus Greff, Noha Radwan, Suhani Vora, Mario Lucic, Daniel Duckworth, Alexey Dosovitskiy, Jakob Uszkoreit, Thomas A. Funkhouser, and Andrea Tagliasacchi. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pages 6219–6228. IEEE, 2022.

[60] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.

A Additional Results

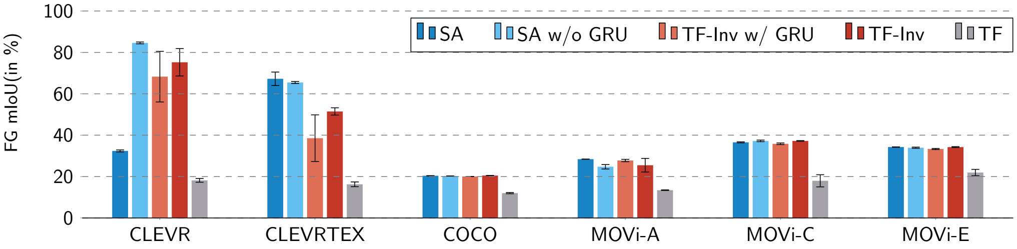

Figure 9: Foreground mIoU of Slot Attention and Transformer variants on the image datasets

B Implementation Details

B.1 Architectures

In the following we provide an overview of the architecture details for our main results.

Slot Initialization. For all object discovery experiments (except the Slot Initialization experiments), we initialize the slots by sampling from a shared Gaussian distribution with learnable mean and standard deviation as in Locatello et al. [16]. We use a slot size of 128 for CLEVR and MOVi, 64 for CLEVRTex, and 256 for COCO.

Encoder. For CLEVR, we use the 4-layer CNN from Locatello et al. [16]. Each layer has a 5×5 kernel size with 64 channels. The first layer has a stride of 2 and the remaining layers have a stride of 1. A learned positional embedding is added to the resulting spatial feature grid, followed by a LayerNorm and a single layer MLP with hidden dimension of 64.

For CLEVRTex, we use a ResNet34 [56] as the encoder, as suggested by Biza et al. [45]. We then add a learned positional embedding and apply LayerNorm, followed by an MLP with hidden size 128.

For COCO and the MOVi datasets, we use a ViT [39] pre-trained with DINO [4] as the image encoder, as proposed in DINOSAUR [13]. We follow the setup in the original DINOSAUR paper and do not add an additional positional embedding to the output of the ViT. We use the recommended architecture and hyperparameters from the original paper, using a ViT-B/16 for COCO and a ViT-B/8 for the MOVi datasets.

Slot Attention (and Variants). ***TF*** uses a pre-LayerNorm cross-attention-only Transformer. In the cross-attention, the slots are used as queries and the encoder features are used as the keys and the values. We additionally apply LayerNorm on the inputs and do not apply the final LayerNorm on the queries, similar to what is done in standard Slot Attention. In our experiments, we use a single head and do not use dropout.

TF-Inv uses the same architecture as *TF*, except we apply the 'inverted' version of attention that is used in Slot Attention. Specifically, during the scaled dot-product attention operation, when we normalize the attention map, we perform the softmax over the queries (ie. the slots) instead of the keys as is normally done [19]. This is followed by a renormalization step to ensure the attention weights for each slot sum to one, essentially taking a weighted average of the values instead of a weighted sum. This technique was proposed in Locatello et al. [16] to help stabilize the inverted attention operation. Alternatively, we experimented with using LayerNorm in place of renormalization as in Tsai et al. [21] and using key, query normalization as in Dehghani et al. [57] though we did not observe meaningful differences in performance throughout our investigation. Only the results reported for OSRT using *TF-Inv* included those changes.

TF-Inv w/ GRU uses a GRU to update the slots after the cross-attention instead of the normal residual connection. This weights of the GRU are not shared across layers of the Transformer, essentially acting as a gating mechanism for the slot updates. We had also experimented with using a shared GRU across layers, but empirically did not observe a meaningful difference.

SA is the standard Slot Attention algorithm with the MLP update.

SA w/o GRU replaces the GRU that is normally used to update the weights with a residual update.

Decoder. For CLEVR and CLEVRTex, we use the Spatial Broadcast Decoder [48] proposed in Locatello et al. [16]. For CLEVR, we use a grid of size 8×8, followed by a 4-layer Transposed Convolutional network, each with kernel size 5×5, hidden dimension 64, and a stride of 2. For CLEVRTex, we use a grid of size 16×16, followed by a 5-layer Transposed Convolutional network, each with kernel size 5×5, and hidden dimension 64. The first three layers have a stride of 3 and the last two have a stride of 2. For the COCO and the MOVi datasets, we use the MLP decoder with the same hyperparameters as proposed in the original DINOSAUR paper [13].

SAVi. For the SAVi experiments, we follow the setup from the enhanced SAVi++ [25]. Specifically, we initialize the slots using the ground truth bounding box information of the objects. We use a single layer MLP with hidden dimension 256 to encode the bounding box into the set of initial slots. We use the modified ResNet34 encoder as described in SAVi++ [25]. For both the Slot Attention and Transformer variants used for the ‘corrector’, we use a slot size of 128 and MLP hidden dimension of 256. Similarly, the Transformer used for the ‘predictor’ uses a slot size of 128 and MLP hidden dimension of 256. We use the same 4-layer Spatial Broadcast Decoder as we use in the CLEVR object discovery experiments. Instead of pixel reconstruction, we predict the optical flow and depth, as described in the original SAVi++ paper.

Transformer Decoder. For our experiments with the Transformer Decoder, we follow the architecture and hyperparameters from Singh et al. [33, 53], except instead of using the DVAE encoder for the input features, we use the same CNN encoder as we use on the CLEVR dataset in our other experiments. The DVAE architecture is unchanged from the original paper and we use a Transformer Decoder with 8 layers, 4 heads, model size of 192, MLP hidden size of 768, and 0.1 Dropout probability.

OSRT. For OSRT, we adopt the default configuration on the MSN-H dataset outlined in Sajjadi et al. [26]. Given a set of input views of a scene rendered from different camera angles, the model is trained to predict novel views of the scene.

B.2 Experimental Details

B.2.1 Datasets

For CLEVR, we use the version of the dataset with masks [41], splitting it into 70,000 images for training and 10,000 for testing. For CLEVRTex, we use a training set of 40,000 images and test set of 5,000 images. For COCO, we use the official splits of 118,287 images for training and 5,000 images for testing. Following DINOSAUR [13], we ignore pixels belonging to overlapping ground truth segments during evaluation. For MOVi-Image, we split each 24-frame MOVi video into individual frames, resulting in a training set of 232,872 images and a test set of 6,000 images. Note that we use the official “validation” set for testing since the official “test” set is out-of-distribution, which is a setting that we not concerned with in our experiments. For the SAVi experiments, we randomly sample a subsequence of 6 frames from each video during training. We run evaluation on the entire 24-frame videos.

B.2.2 Training Details

Learning Rates. For all object discovery experiments, except the Transformer Decoder experiments, we train to 300,000 steps. We use a batch size of 64 and the Adam optimizer [58] for all datasets. For CLEVR and the MOVi video datasets, we linearly warm up the learning rate from 0 to 2e-4 over 2,500 steps and then anneal the learning rate with a Cosine schedule for the rest of the training steps. We additionally clip the gradient norm to 0.05. For CLEVRTex, we instead use a base learning rate of 4e-4 and 10,000 warmup steps, but keep the rest of the training configurations the same. For COCO and the MOVi image datasets, we also use a base learning rate of 4e-4 and 10,000 warmup steps, but exponentially decay the learning rate with a half-life of 100,000 steps. We clip the gradient norm at 1.0. For the Transformer Decoder experiments, we train to 200,000 steps and clip the gradient norm at 0.05. We train the DVAE with a constant learning rate of 3e-4. We train the Slot Attention / Transformer module with a base learning rate of 1e-4 and 30,000 warmup steps, followed by exponential decay wth a half-life of 250,000 steps. We train the Decoder with the same schedule, but with a base learning rate of 3e-4.

For OSRT, we train to 1 million steps and otherwise follow the training procedure of the original paper, which is based off of SRT [59].

Number of Slots. Table 2 shows the number of slots we used for each dataset in the object discovery experiments. We make sure to set the number of slots to be large enough to cover the maximum number of objects in the dataset.

Dataset	**Number of Slots**
CLEVR	11
CLEVRTex	11
COCO	7
MOVi-A	11
MOVi-C	11
MOVi-E	24

Table 2: Number of slots used in the object discovery experiments.

Data Preprocessing and Augmentations. For the CLEVR and CLEVRTex datasets, we take a 192×192 center crop of each image before resizing it to 128×128. For COCO, we randomly flip the image horizontally during training before resizing the image such that the smaller side is 224 pixels while maintaining the aspect ratio. We then take a 224×224 center crop. For the MOVi image datasets, we resize the images to 224×224. For the MOVi video datasets, we follow the data augmentation strategy provided in SAVi++ [25].

B.2.3 Evaluation Metrics

To evaluate the quality of the segmentations in the object discovery task, we use the foreground adjusted rand index (FG-ARI) [22, 46, 47]. ARI measures the similarity of clusters and assigns a high score if pixels belonging to the same ground truth cluster are also in the same predicted clusters. As is common practice in the literature, we compute ARI only on the foreground ground-truth masks, ignoring the background pixels.

For the SAVi experiments, we use the video version of FG-ARI, which considers pixels corresponding to one object along the entire video as a single cluster. This requires each slot to predict the same object consistently throughout the video. Following SAVi [24], we do not include the first frame of the video during evaluation since we provide conditional bounding box information in the first frame. For OSRT, pixels corresponding to one object across different views are considered as a single cluster, requiring consistent predictions across views.

For our object discovery experiments, we additionally report foreground mean intersection over union (mIoU) in Table 9. To calculate this metric, we first use Hungarian matching [60] between the ground truth masks and the predicted masks to obtain a ground truth assignment for each predicted mask. We then calculate the IoU between the predicted and ground truth masks using this assignment. Unlike FG-ARI, this metric penalizes predictions that over-segment the objects.

For our OSRT experiments, we additionally report Peak Signal-to-Noise Ration (PSNR), which measures how well the model can reconstruct novel views.