Towards Object-Centric Learning with General Purpose Architectures

Jack Brady[1,2], **Julius von Kügelgen**[3], **Sébastien Lachapelle**[4], **Simon Buchholz**[1,2], **Thomas Kipf** [† 5], **Wieland Brendel**[† 1,2,6]

[1]Max Planck Institute for Intelligent Systems, Tübingen, Germany
[2]Tübingen AI Center, Tübingen, Germany
[3]Seminar for Statistics, ETH Zürich, Switzerland
[4]Samsung - SAIT AI Lab, Montréal, Canada
[5]Google DeepMind
[6]ELLIS Institute Tübingen, Germany
[†]Shared last author

Abstract

Learning disentangled representations of objects in an image is a prerequisite for the robust compositional generalization in human intelligence. While progress has been made in learning such object-centric representations (OCRL), these methods rely on strong architectural priors which hinder scalability. In this work, we explore a more scalable approach for OCRL. Namely, we propose to use a *general purpose* architecture for OCRL and add inductive biases to the model via additional regularizers. To formulate suitable regularizers, we take inspiration from recent theoretical results in Brady et al. [6] which put forth two properties a model should satisfy to provably disentangle objects. We show that these properties can be scalably enforced using a VAE loss and a novel loss on the attention weights of a Transformer. We incorporate these regularizers into a general purpose Transformer autoencoder and attain competitive and often superior performance to existing methods in OCRL with stronger architectural priors.

1 Introduction

A core feature of human cognition is the ability to recompose known concepts to generalize far beyond direct experience [10, 11, 14, 25]. For example, humans can make sense of an image of a "penguin in a desert" by composing the concepts of "penguin" and "desert" to understand this novel combination. Such *compositional generalization* is non-trivial and requires first learning an internal model of different concepts in the world, e.g., "penguin", "desert". This implies learning a *separate* internal representation of each concept from sensory observations. In machine learning, this is commonly referred to as learning *disentangled representations* of concepts [4, 15, 39].

Recently, several works have shown remarkable empirical success in learning to disentangle [23, 33] and compose [7, 31, 32, 35, 36] visual concepts in images on web-scale data. These works rely on explicit supervision via segmentation masks or natural language descriptions of each concept. Notably, however, many species in human's evolutionary lineage disentangle concepts in sensory data *without* using such explicit supervision [3, 27, 43]. This suggest the existence of a self-supervised coding mechanism for disentanglement and compositional generalization, which is still lacking from current large-scale machine learning models, and is the focus of our work.

A key challenge for achieving disentanglement without explicit supervision is that it requires incorporating appropriate inductive biases [28]. Recently, significant progress has been made in formu-

NeurIPS 2024 Workshop on Compositional Learning: Perspectives, Methods, and Paths Forward.

lating suitable inductive biases for *object-centric* representation learning (OCRL) [5, 8, 13, 22, 29, 37, 38, 40, 41, 50]. Currently, however, scaling these OCRL methods to many practical problems of interest remains challenging. This is because these methods typically rely on inductive bias in the form of strong architectural priors. These priors enable disentanglement, but often hinder scalability.

In this work, we explore an alternative, more scalable approach for OCRL. Specifically, we propose to use a *more general* architecture for OCRL and add inductive biases to the model via additional *regularizers*. To formulate suitable regularizers, we take inspiration from recent theoretical results in Brady et al. [6]. These results showed that models which enforce two properties, (*i*) invertibility and (*ii*) compositionality, will provably learn disentangled representations of objects. We use a VAE loss [21] to enforce (i), and make the observation that a Transformer [45] offers an efficient means to scalably enforce (ii) via an inexpensive regularizer for a cross-attention mechanism. We then incorporate these two regularizers into a general purpose Transformer-based autoencoder.

We test this model's ability to disentangle objects on a Sprites dataset [47] and CLEVR6 [19]. We find that the model reliably learns disentangled representations of objects, improving performance over an unregularized Transformer. Furthermore, we find that this regularized Transformer generally achieves superior performance to existing OCRL models with more explicit object-centric priors such as Slot Attention [29] and Spatial Broadcast Decoders [48].

2 Background

Theory in Brady et al. [6].

Recent theoretical work in [6] showed how object-centric representations can be provably disentangled without supervision. These results assume a latent variable model for object-centric data where each object is represented by disjoint groups or *slots* of latents $\boldsymbol{z}_{B_k}$ s.t. $\boldsymbol{z} = (\boldsymbol{z}_{B_1}, ..., \boldsymbol{z}_{B_K}) \in \mathbb{R}^{d_z}$. These latents are rendered to an observation $\boldsymbol{x}$ by a generator $\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathcal{X} \subset \mathbb{R}^{d_x}$ which is assumed to satisfy two properties called *irreducibility* and *compositionality*. Informally, irreducibility states that pixels belonging to the same object share information, while compositionality states that each image pixel is *locally* a function of at most one latent slot, i.e., the Jacobian of $\boldsymbol{f}$ has a block-structure. More formally, for compositionality:

Definition 2.1 (Compositionality)**.** A function $\boldsymbol{f} : \mathbb{R}^{d_z} \to \mathcal{X} \subset \mathbb{R}^{d_x}$ satisfies *compositionality* if

$$\frac{\partial \boldsymbol{f}_n}{\partial \boldsymbol{z}_{B_k}}(\boldsymbol{z}) \neq 0 \implies \frac{\partial \boldsymbol{f}_n}{\partial \boldsymbol{z}_{B_j}}(\boldsymbol{z}) = 0, \quad \text{for any } k, j \in [K], k \neq j \text{ and any } n \in [d_x]. \tag{2.1}$$

If $\boldsymbol{f}$ satisfies these assumptions, Brady et al. [6] showed that a model $\hat{\boldsymbol{f}} : \mathbb{R}^{d_z} \to \mathbb{R}^{d_x}$ that is (i) invertible from $\mathbb{R}^{d_z}$ to $\mathcal{X}$ and also (ii) satisfies compositionality (Defn. 2.1), will learn a disentangled representation of objects. While providing these theoretical results, Brady et al. [6] did not show how (i) and (ii) can be implemented in a scalable manner. We explore this in § 3.

Prior Work in OCRL. Prior works in OCRL typically rely on architectural priors to learn object-centric representations [13, 29, 40, 41]. While such priors promote disentanglement, they are often too restrictive. For example, Spatial Broadcast Decoders [48] decode slots separately and only allow for weak interaction through a softmax function, which prevents modelling real-world data where objects exhibit more complex interactions [41]. While some works have shown success in disentangling objects using more powerful Transformer decoders [37, 41, 42], they rely on encoders that use Slot Attention [29] as an architectural component, which differs from current large-scale models, typically based on Transformers [1]. In contrast, we explore the more flexible approach of starting with a very general Transformer-based model and regularizing it towards a more constrained model.

3 Method

We now explore how the theoretical criteria (i) invertibility and (ii) compositionality outlined in § 2 can be enforced by a model in a scalable manner.

(i) Invertibility. Our theory requires invertibility between $\mathcal{X}_{\text{supp}} \subseteq \mathbb{R}^{d_x}$ and $\hat{\mathcal{Z}}_{\text{supp}} \subseteq \mathcal{Z} = \mathbb{R}^{d_z}$. For most settings of interest, the observed dimension d_x exceeds the ground-truth latent dimension d_z.

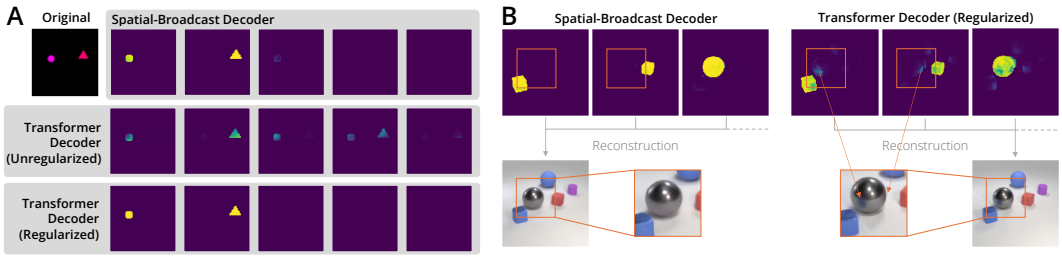

Figure 1: **(A) Sprites** Normalized slot-wise Jacobians for an unregularized ($\alpha = 0, \beta = 0$) and a regularized ($\alpha > 0, \beta > 0$) Transformer and a Spatial Broadcast Decoder (SBD). The unregularized model encodes objects across multiple slots, while the regularized model matches the disentanglement of the SBD. **(B) CLEVR6** Slot-wise Jacobians for a regularized Transformer and a SBD on objects in CLEVR6 which interact via reflections. As can be seen in reconstructions and Jacobians, the regularized Transformer models reflections, while mostly removing unnecessary interactions, while the SBD fails to model reflections due to its restricted architecture.

Thus, we generally cannot use models which are invertible by construction such as normalizing flows [30]. An alternative is to use an *autoencoder* in which $\hat{\boldsymbol{f}}^{-1}$ and $\hat{\boldsymbol{f}}$ are parameterized separately by an *encoder* $\hat{\boldsymbol{g}} : \mathbb{R}^{d_x} \to \mathbb{R}^{d_{\hat{z}}}$ and a *decoder* $\hat{\boldsymbol{f}} : \mathbb{R}^{d_{\hat{z}}} \to \mathbb{R}^{d_x}$, which are trained to invert each other (on $\hat{\mathcal{Z}}_{\text{supp}}$ and $\mathcal{X}_{\text{supp}}$) by minimizing a reconstruction loss $\mathcal{L}_{\text{rec}} := \mathbb{E}\|\boldsymbol{x} - \hat{\boldsymbol{f}}(\hat{\boldsymbol{g}}(\boldsymbol{x}))\|^2$. Minimizing $\mathcal{L}_{\text{rec}}$ alone, however, does not suffice unless the inferred latent dimension $d_{\hat{z}}$ equals the ground-truth d_z. Yet, in practice d_z is unknown. Moreover, choosing $d_{\hat{z}} > d_z$ is important for scalability [37]. A viable alternative is thus to employ a soft constraint where $d_{\hat{z}} > d_z$, but the model is encouraged to encode $\boldsymbol{x}$ using minimal latent dimensions. To achieve this, we leverage the well known VAE loss [21], which couples $\mathcal{L}_{\text{rec}}$ with a KL-divergence loss $\mathcal{L}_{\text{KL}}$ between a factorized posterior $\boldsymbol{q}(\hat{\boldsymbol{z}}|\boldsymbol{x})$ and prior distribution $\boldsymbol{p}(\hat{\boldsymbol{z}})$, i.e., $\mathcal{L}_{\text{KL}} := \sum_{i\in[d_{\hat{z}}]} D_{\text{KL}}\left(\boldsymbol{q}(\hat{z}_i|\boldsymbol{x})\|\boldsymbol{p}(\hat{z}_i)\right)$. This loss encourages each $\hat{z}_i$ to be insensitive to changes in $\boldsymbol{x}$ such that unnecessary dimensions should contain no information about $\boldsymbol{x}$ [34].

Transformers for Enforcing (ii) Compositionality. We make the observation that the *Transformer* architecture [45] provides an efficient means to approximately regularize interactions. In a Transformer, slots are only permitted to interact via an *attention mechanism*. We will focus on a *cross-attention* mechanism, which maps a latent $\hat{\boldsymbol{z}}$ to an output $\hat{x}_l$ (e.g., a pixel) via:

$$\boldsymbol{K} = \boldsymbol{W}^K[\hat{\boldsymbol{z}}_{B_1} \cdots \hat{\boldsymbol{z}}_{B_K}], \qquad \boldsymbol{V} = \boldsymbol{W}^V[\hat{\boldsymbol{z}}_{B_1} \cdots \hat{\boldsymbol{z}}_{B_K}], \quad \boldsymbol{Q} = \boldsymbol{W}^Q[\boldsymbol{o}_1 \cdots \boldsymbol{o}_{d_x}], \tag{3.1}$$

$$\boldsymbol{A}_{l,k} = \frac{\exp\left(\boldsymbol{Q}_{:,l}^\top \boldsymbol{K}_{:,k}\right)}{\sum_{i\in[K]} \exp\left(\boldsymbol{Q}_{:,l}^\top \boldsymbol{K}_{:,i}\right)}, \quad \bar{\boldsymbol{x}}_l = \boldsymbol{A}_{l,:}\boldsymbol{V}^\top, \qquad \hat{x}_l = \psi(\bar{\boldsymbol{x}}_l)\,. \tag{3.2}$$

In Eq. (3.1), all slots are assumed to have equal size, and key $\boldsymbol{K}_{:,k}$ and value $\boldsymbol{V}_{:,k}$ vectors are computed for each slot $k \in [K]$. Query vectors are computed for output dimensions $l \in [d_x]$ (e.g., pixel coordinates) and each l is assigned a fixed vector $\boldsymbol{o}_l$. In Eq. (3.2), queries and keys are used to compute attention weights $\boldsymbol{A}_{l,k}$. These weights determine the slots pixel l "attends" to when generating pixel token $\bar{\boldsymbol{x}}_l$, which is mapped to a pixel $\hat{x}_l$ by nonlinear function ψ; see Appx. A for further details.

Within cross-attention, interactions across slots occur if the query vector for a pixel l attends to multiple slots, i.e., if $\boldsymbol{A}_{l,k}$ is non-zero for more than one k. Conversely, if $\boldsymbol{A}_{l,k}$ is non-zero for only one k, then, intuitively, no interactions should occur. This intuition can be corroborated formally by computing the Jacobian of cross-attention w.r.t. each slot (see Appx. A.1). Thus, we can encourage a model to satisfy compositionality by regularizing $\boldsymbol{A}$ towards having only one non-zero entry for each row $\boldsymbol{A}_{l,:}$. To this end, we propose to minimize the sum of all pairwise products $\boldsymbol{A}_{l,j}\boldsymbol{A}_{l,k}$, where $j \neq k$ (see Fig. 2). This quantity is non-negative and will only be zero when each row of $\boldsymbol{A}$ has exactly one non-zero entry. This resembles the *compositional contrast* of Brady et al. [6], but computed on $\boldsymbol{A}$, which can be efficiently optimized, as opposed to the Jacobian of $\hat{\boldsymbol{f}}$ which is intractable to optimize. We refer to this regularizer as $\mathcal{L}_{\text{comp}}$, see Eq. (A.9).

Model. Combining these different objectives leads us to the following weighted three-part-loss:

$$\mathcal{L}_{\text{disent}}(\hat{\boldsymbol{f}}, \hat{\boldsymbol{g}}, \boldsymbol{x}) = \mathcal{L}_{\text{rec}} + \alpha\mathcal{L}_{\text{comp}} + \beta\mathcal{L}_{\text{KL}}. \tag{3.3}$$

Table 1: **Empirical Results.** We show the mean ± std. dev. for J-ARI and JIS (in %) over 3 seeds for different choices of encoders and decoders and weights of the loss terms in Eq. (3.3) on Sprites and CLEVR6.

Model			Sprites		CLEVR6	
Encoder	**Decoder**	**Loss**	**J-ARI** (↑)	**JIS** (↑)	**J-ARI** (↑)	**JIS** (↑)
Slot Attention	Spatial-Broadcast	$\alpha = 0, \beta = 0$	89.2 ± 1.2	91.4 ± 0.6	$\mathbf{97.0 \pm 0.1}$	$\mathbf{95.3 \pm 0.6}$
Slot Attention	Transformer	$\alpha = 0, \beta = 0$	90.1 ± 1.2	73.6 ± 1.2	95.4 ± 0.8	63.1 ± 0.8
Transformer	Transformer	$\alpha = 0, \beta = 0$	80.5 ± 3.4	57.0 ± 6.5	92.7 ± 2.7	54.8 ± 2.9
Transformer	Transformer	$\alpha > 0, \beta = 0$	82.8 ± 2.9	73.8 ± 3.3	79.2 ± 10.4	51.6 ± 4.8
Transformer	Transformer	$\alpha = 0, \beta > 0$	92.6 ± 1.6	92.8 ± 0.7	96.6 ± 0.3	80.3 ± 0.3
Transformer	Transformer	$\alpha > 0, \beta > 0$ **(Ours)**	$\mathbf{93.6 \pm 0.5}$	$\mathbf{95.0 \pm 1.7}$	96.5 ± 0.3	83.8 ± 1.0

We apply this loss to a flexible Transformer-based autoencoder, similar to the models of Jabri et al. [17], Jaegle et al. [18], Sajjadi et al. [38]. For the encoder $\hat{\boldsymbol{g}}$, we first map data $\boldsymbol{x}$ to features using the CNN of Locatello et al. [29]. These features are processed by a Transformer, which has both self- and cross-attention at every layer, yielding a representation $\hat{\boldsymbol{z}}$. Our decoder $\hat{\boldsymbol{f}}$ then maps $\hat{\boldsymbol{z}}$ to an output $\hat{\boldsymbol{x}}$ using a cross-attention Transformer regularized with $\mathcal{L}_{\text{comp}}$, see Appx. C for details.

4 Experiments

We now test the ability of our attention-regularized Transformer-VAE (§ 3) to learn object-centric representations. We discuss experimental details below (though see Appx. C for additional details).

Data. We consider two multi-object datasets in our experiments. The first, which we refer to as Sprites [6, 47, 49], consist of images with 2–4 objects set against a black background. The second is the CLEVR6 dataset [19], consisting of images with 2–6 objects. In Sprites, objects do not have reflections and rarely occlude such that slots have essentially have no interaction. In CLEVR6, however, objects can cast shadows and reflect upon each other, introducing more complex interactions.

Metrics. A common metric for object disentanglement is the Adjusted-Rand Index [ARI; 16]. The ARI measures the similarity between the set of pixels encoded by a model slot, and the set of ground-truth pixels for a given object in a scene, yielding an optimal score if each slot corresponds to exactly one object. To assign a pixel to a unique model slot, prior works typically choose the slot with the largest attention score (from, e.g., Slot Attention) for that pixel [40]. However, using attention scores can make model comparisons challenging and is also somewhat unprincipled (see Appx. C.2). We thus consider an alternative and compute the ARI using the Jacobian of a decoder (J-ARI). Specifically, we assign a pixel l to the slot with the largest L_1 norm for the slot-wise Jacobian $D_{B_k} \hat{f}_l(\hat{\boldsymbol{z}})$ (see Fig. 1 for a visualization of these Jacobians).

While J-ARI indicates which slots are most responsible for encoding each object, it does not indicate if additional slots affect the same object, i.e., $\|D_{B_k} \hat{f}_l(\hat{\boldsymbol{z}})\|_1 \neq 0$ for more than one k. To measure this, we also introduce the Jacobian Interaction Score (JIS). JIS is computed by taking the maximum of $\|D_{B_k} \hat{f}_l(\hat{\boldsymbol{z}})\|_1$ across slots after normalization, averaged over all pixels. If each pixel is affected by only one slot, JIS is 1. For datasets where objects essentially do not interact like Sprites, JIS should be close to 1, whereas for CLEVR6, it should be as high as possible while maintaining invertibility.

4.1 Results

$\mathcal{L}_{\text{disent}}$ Enables Object Disentanglement. In Tab. 1, we compare the J-ARI and JIS of our regularized Transformer-based model ($\alpha > 0, \beta > 0$) trained with $\mathcal{L}_{\text{disent}}$ (Eq. (3.3)) to the same model trained without regularization ($\alpha = 0, \beta = 0$), i.e., with only $\mathcal{L}_{\text{rec}}$. On Sprites, the regularized model achieves notably higher scores for both J-ARI and JIS. This is corroborated by visualizing the slot-wise Jacobians in Fig. 1A, where we see the regularized model cleanly disentangles objects, whereas the unregularized model often encodes objects across multiple slots. Similarly, on CLEVR6, the regularized model achieves superior disentanglement, as indicated by the higher values for both metrics.

Comparison to Existing Object-Centric Autoencoders. In Tab. 1, we also compare our model to existing models using encoders with Slot Attention and Spatial Broadcast Decoders (SBDs). On Sprites, our model achieves higher J-ARI and JIS than these models, despite using a weaker

architectural prior. On CLEVR6, our model outperforms Slot Attention with a Transformer decoder in terms of J-ARI and JIS. Models using a SBD, however, achieve a higher and nearly perfect JIS, i.e., the learned slots essentially never affect the same pixel. In Fig 1B, we see this comes at the cost of SBDs failing to model reflections between objects, while our model captures this interaction. This highlights that regularizing a flexible architecture with $\mathcal{L}_{\text{disent}}$ can enable a better balance between restricting interactions and model expressivity.

Ablation Over Losses. Lastly, in Tab. 1, we ablate the impact of the regularizers in $\mathcal{L}_{\text{disent}}$. Training without $\mathcal{L}_{\text{KL}}$ ($\alpha > 0, \beta = 0$) can in some cases give improvements in J-ARI and JIS over an unregularized model ($\alpha = 0, \beta = 0$). However, across datasets this loss yields worse disentanglement than $\mathcal{L}_{\text{disent}}$ ($\alpha > 0, \beta > 0$). This highlights that penalizing latent capacity via $\mathcal{L}_{\text{KL}}$ is important for object disentanglement. Training without $\mathcal{L}_{\text{comp}}$ ($\alpha = 0, \beta > 0$) generally yields a drop across both metrics compared to $\mathcal{L}_{\text{disent}}$, though on CLEVR6 this loss achieves a comparable J-ARI. We found that training with $\mathcal{L}_{\text{KL}}$ can, in some cases, implicitly minimize $\mathcal{L}_{\text{comp}}$, explaining this result (Fig. 3).

References

[1] R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. [Cited on p. 2.]

[2] J. L. Ba. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. [Cited on p. 9.]

[3] T. E. J. Behrens, T. H. Muller, J. C. R. Whittington, S. Mark, A. Baram, K. L. Stachenfeld, and Z. Kurth-Nelson. What is a cognitive map? organizing knowledge for flexible behavior. *Neuron*, 100:490–509, 2018. [Cited on p. 1.]

[4] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013. [Cited on p. 1.]

[5] O. Biza, S. Van Steenkiste, M. S. Sajjadi, G. F. Elsayed, A. Mahendran, and T. Kipf. Invariant slot attention: Object discovery with slot-centric reference frames. *arXiv preprint arXiv:2302.04973*, 2023. [Cited on p. 2.]

[6] J. Brady, R. S. Zimmermann, Y. Sharma, B. Schölkopf, J. von Kügelgen, and W. Brendel. Provably learning object-centric representations. In *International Conference on Machine Learning*, pages 3038–3062. PMLR, 2023. [Cited on p. 1, 2, 3, 4, and 10.]

[7] T. Brooks, A. Holynski, and A. A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. [Cited on p. 1.]

[8] C. P. Burgess, L. Matthey, N. Watters, R. Kabra, I. Higgins, M. Botvinick, and A. Lerchner. Monet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019. [Cited on p. 2.]

[9] M. Chang, T. Griffiths, and S. Levine. Object representations as fixed points: Training iterative refinement algorithms with implicit differentiation. *Advances in Neural Information Processing Systems*, 35:32694–32708, 2022. [Cited on p. 11.]

[10] J. A. Fodor and Z. W. Pylyshyn. Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28(1):3–71, 1988. [Cited on p. 1.]

[11] A. Goyal and Y. Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068, 2022. [Cited on p. 1.]

[12] A. Goyal, A. Lamb, J. Hoffmann, S. Sodhani, S. Levine, Y. Bengio, and B. Schölkopf. Recurrent independent mechanisms. In *International Conference on Learning Representations*, 2021. [Cited on p. 11.]

[13] K. Greff, R. L. Kaufman, R. Kabra, N. Watters, C. Burgess, D. Zoran, L. Matthey, M. Botvinick, and A. Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning*, pages 2424–2433. PMLR, 2019. [Cited on p. 2.]

[14] K. Greff, S. Van Steenkiste, and J. Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020. [Cited on p. 1.]

[15] I. Higgins, D. Amos, D. Pfau, S. Racanière, L. Matthey, D. J. Rezende, and A. Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018. [Cited on p. 1.]

[16] L. J. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985. [Cited on p. 4.]

[17] A. Jabri, D. J. Fleet, and T. Chen. Scalable adaptive computation for iterative generation. In *International Conference on Machine Learning*, pages 14569–14589. PMLR, 2023. [Cited on p. 4.]

[18] A. Jaegle, S. Borgeaud, J.-B. Alayrac, C. Doersch, C. Ionescu, D. Ding, S. Koppula, D. Zoran, A. Brock, E. Shelhamer, et al. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. [Cited on p. 4.]

[19] J. Johnson, B. Hariharan, L. Van Der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. 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. [Cited on p. 2, 4, and 11.]

[20] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. [Cited on p. 11.]

[21] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014. [Cited on p. 2 and 3.]

[22] T. Kipf, G. F. Elsayed, A. Mahendran, A. Stone, S. Sabour, G. Heigold, R. Jonschkowski, A. Dosovitskiy, and K. Greff. Conditional object-centric learning from video. *arXiv preprint arXiv:2111.12594*, 2021. [Cited on p. 2.]

[23] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. [Cited on p. 1.]

[24] A. Kori, F. Locatello, A. Santhirasekaram, F. Toni, B. Glocker, and F. D. S. Ribeiro. Identifiable object-centric representation learning via probabilistic slot attention. *arXiv preprint arXiv:2406.07141*, 2024. [Cited on p. 10.]

[25] B. M. Lake, T. D. Ullman, J. B. Tenenbaum, and S. J. Gershman. Building machines that learn and think like people. *Behavioral and brain sciences*, 40:e253, 2017. [Cited on p. 1.]

[26] A. Lamb, D. He, A. Goyal, G. Ke, C.-F. Liao, M. Ravanelli, and Y. Bengio. Transformers with competitive ensembles of independent mechanisms. *arXiv preprint arXiv:2103.00336*, 2021. [Cited on p. 11.]

[27] Y. LeCun. A path towards autonomous machine intelligence version 0.9.2, 2022-06-27. *OpenReview*, pages 1–62, 2022. [Cited on p. 1.]

[28] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *International Conference on Machine Learning*, pages 4114–4124. PMLR, 2019. [Cited on p. 1.]

[29] F. Locatello, D. Weissenborn, T. Unterthiner, A. Mahendran, G. Heigold, J. Uszkoreit, A. Dosovitskiy, and T. Kipf. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems*, volume 33, pages 11525–11538, 2020. [Cited on p. 2, 4, 11, and 12.]

[30] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. [Cited on p. 3.]

[31] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. Pmlr, 2021. [Cited on p. 1.]

[32] A. Ramesh, P. Dhariwal, A. Nichol, C. Chu, and M. Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1(2):3, 2022. [Cited on p. 1.]

[33] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024. [Cited on p. 1.]

[34] M. Rolinek, D. Zietlow, and G. Martius. Variational autoencoders pursue pca directions (by accident). In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12406–12415, 2019. [Cited on p. 3.]

[35] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22500–22510, 2023. [Cited on p. 1.]

[36] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35: 36479–36494, 2022. [Cited on p. 1.]

[37] M. S. Sajjadi, D. Duckworth, A. Mahendran, S. Van Steenkiste, F. Pavetic, M. Lucic, L. J. Guibas, K. Greff, and T. Kipf. Object scene representation transformer. In *Advances in Neural Information Processing Systems*, volume 35, pages 9512–9524, 2022. [Cited on p. 2 and 3.]

[38] M. S. Sajjadi, H. Meyer, E. Pot, U. Bergmann, K. Greff, N. Radwan, S. Vora, M. Lučić, D. Duckworth, A. Dosovitskiy, et al. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6229–6238, 2022. [Cited on p. 2 and 4.]

[39] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. [Cited on p. 1.]

[40] M. Seitzer, M. Horn, A. Zadaianchuk, D. Zietlow, T. Xiao, C.-J. Simon-Gabriel, T. He, Z. Zhang, B. Schölkopf, T. Brox, and F. Locatello. Bridging the gap to real-world object-centric learning. In *The Eleventh International Conference on Learning Representations*, 2023. [Cited on p. 2, 4, and 12.]

[41] G. Singh, F. Deng, and S. Ahn. Illiterate DALL-E learns to compose. In *International Conference on Learning Representations*, 2022. [Cited on p. 2 and 11.]

[42] G. Singh, Y.-F. Wu, and S. Ahn. Simple unsupervised object-centric learning for complex and naturalistic videos. *Advances in Neural Information Processing Systems*, 35:18181–18196, 2022. [Cited on p. 2.]

[43] E. C. Tolman. Cognitive maps in rats and men. *Psychological review*, 55 4:189–208, 1948. [Cited on p. 1.]

[44] A. Vani, B. Nguyen, S. Lavoie, R. Krishna, and A. Courville. Sparo: Selective attention for robust and compositional transformer encodings for vision. *arXiv preprint arXiv:2404.15721*, 2024. [Cited on p. 11.]

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. [Cited on p. 2, 3, and 9.]

[46] Y. Wang, L. Liu, and J. Dauwels. Slot-vae: Object-centric scene generation with slot attention. In *International Conference on Machine Learning*, pages 36020–36035. PMLR, 2023. [Cited on p. 10.]

[47] N. Watters, L. Matthey, S. Borgeaud, R. Kabra, and A. Lerchner. Spriteworld: A flexible, configurable reinforcement learning environment, 2019. [Cited on p. 2, 4, and 11.]

[48] N. Watters, L. Matthey, C. P. Burgess, and A. Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in VAEs. *arXiv preprint arXiv:1901.07017*, 2019. [Cited on p. 2 and 11.]

[49] T. Wiedemer, P. Mayilvahanan, M. Bethge, and W. Brendel. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems*, volume 36, 2024. [Cited on p. 4.]

[50] Y.-F. Wu, K. Greff, G. F. Elsayed, M. C. Mozer, T. Kipf, and S. van Steenkiste. Inverted-attention transformers can learn object representations: Insights from slot attention. In *Causal Representation Learning Workshop at NeurIPS 2023*. [Cited on p. 2.]

Appendices

A Transformers for Compositionality

Each layer of a Transformer [45] consist of two main components: an MLP sub-layer and an attention mechanism. Notably, in the MLP sub-layer, MLPs are applied separately to each slot or pixel query and their outputs are then concatenated. Further, additional layer normalization operations [2] are typically used in Transformers but are also separately applied to each slot or pixel query. Thus, the only opportunity for interaction between slots in a Transformer occurs through the attention mechanism. Our focus in this work is on the cross-attention mechanism, opposed to the alternative self-attention. As noted in § 3, cross-attention takes the form:

$$\boldsymbol{K} = \boldsymbol{W}^K[\hat{\boldsymbol{z}}_{B_k}]_{k\in[K]}, \qquad \boldsymbol{V} = \boldsymbol{W}^V[\hat{\boldsymbol{z}}_{B_k}]_{k\in[K]}, \qquad \boldsymbol{Q} = \boldsymbol{W}^Q[\boldsymbol{o}_d]_{d\in[d_x]}, \tag{A.1}$$

$$\boldsymbol{A}_{d,k} = \frac{\exp\left(\boldsymbol{Q}_{:,d}^\top \boldsymbol{K}_{:,k}\right)}{\sum_{l\in[K]}\exp\left(\boldsymbol{Q}_{:,d}^\top \boldsymbol{K}_{:,l}\right)}, \qquad \bar{\boldsymbol{x}}_d = \boldsymbol{A}_{d,:}\boldsymbol{V}^\top, \qquad \hat{x}_d = \psi(\bar{\boldsymbol{x}}_d)\,. \tag{A.2}$$

where $\boldsymbol{K}_{:,k}, \boldsymbol{V}_{:,k} \in \mathbb{R}^{d_q}$, $\boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{d_q \times |B_k|}$ for query dimension d_q. Further, $\boldsymbol{o}_d \in \mathbb{R}^{d_o}$, $\boldsymbol{Q}_{:,l} \in \mathbb{R}^{d_q}$, $\boldsymbol{W}^Q \in \mathbb{R}^{d_q \times d_o}$, where d_o is the dimension of a pixel coordinate vector, and $\psi : \mathbb{R}^{d_q} \to \mathbb{R}$.

Additional Details. In Eq. (A.2), we do not include the scaling factor $d_q^{-\frac{1}{2}}$ for $\boldsymbol{A}_{d,k}$, that is typically used as it does not affect our arguments below. We do, however, include it in our experiments. Further, when $\boldsymbol{x}$ is an RGB image, $\hat{x}_d$ will not be a scalar but will instead be a vector in $\mathbb{R}^3$ since each pixel has 3 color channels. Additionally, in our experiments, multi-head attention is used. In this case, slot keys and values and pixel queries are partitioned into h sub-vectors. Eqs. (A.1) and (A.2) are then applied separately to each resulting sub-latent, and the resulting outputs are concatenated. When using multiple layers of cross-attention, as we do in our experiments, ψ is only applied at the last layer and vectors $\boldsymbol{o}_l$ for a subsequent layer are defined as the vectors $\bar{\boldsymbol{x}}_d$ from the prior layer. Eqs. (A.1) and (A.2) is then repeated. We discuss how these additional caveats are dealt with empirically when implementing $\mathcal{L}_{\text{comp}}$ below in Appx. A.2, however, they do not affect our formal argument regarding regularizing interactions in Appx. A.1.

A.1 Jacobian of Cross-Attention Mechanism

Our goal is to show that if $\boldsymbol{A}_{d,k}$ in equation is 0, then partial derivative of Eqs. (A.1) and (A.2) w.r.t slot $\hat{\boldsymbol{z}}_{B_k}$, i.e, $\frac{\partial \hat{x}_d}{\partial \hat{z}_{B_k}}$ will also be zero. This would then imply that if $\boldsymbol{A}_{d,:}$ is non-zero for at most one slot k for every $d \in [d_x]$, and every $\hat{\boldsymbol{z}} \in \hat{\mathcal{Z}}_{\text{supp}}$, then the model is compositional in the sense of Defn. 2.1, since all such derivative products $\frac{\partial \hat{x}_d}{\partial \hat{\boldsymbol{z}}_{B_k}} \frac{\partial \hat{x}_d}{\partial \hat{\boldsymbol{z}}_{B_l}}$ for $l \neq k$ are zero. To this end, we are interested in computing the derivative:

$$\frac{\partial \hat{x}_d}{\partial (\hat{\boldsymbol{z}}_{B_m})_r} = \partial_i \psi(\bar{\boldsymbol{x}}) \frac{\partial (\bar{\boldsymbol{x}}_d)_i}{\partial (\hat{\boldsymbol{z}}_{B_m})_r} \tag{A.3}$$

where we here and from now on use the convention that we sum over every index that appears only on one side. To evaluate this we decompose the terms

$$(\bar{\boldsymbol{x}}_d)_i = \boldsymbol{A}_{d,k}\boldsymbol{V}_{i,k} = \boldsymbol{A}_{d,k}\boldsymbol{W}^V_{i,j}(\hat{\boldsymbol{z}}_{B_k})_j. \tag{A.4}$$

We set $\boldsymbol{M}_{d,:} = \boldsymbol{o}_d^\top (\boldsymbol{W}^Q)^\top \boldsymbol{W}^K$ so that

$$\boldsymbol{Q}_{:,d}^\top \boldsymbol{K}_{:,k} = \boldsymbol{M}_{d,i}(\hat{\boldsymbol{z}}_{B_k})_i. \tag{A.5}$$

This implies that

$$\frac{\partial}{\partial (\hat{\boldsymbol{z}}_{B_m})_i} \exp(\boldsymbol{Q}_{:,d}^\top \boldsymbol{K}_{:,k}) = \boldsymbol{M}_{d,i}\delta_{km} \exp(\boldsymbol{Q}_{:,d}^\top \boldsymbol{K}_{:,k}) \tag{A.6}$$

where δ is the Kronecker-Delta (and here no summation over k or d is done). This implies using the product rule and the chain rule that

$$\frac{\partial \boldsymbol{A}_{d,k}}{\partial(\hat{\boldsymbol{z}}_{B_m})_i} = \boldsymbol{M}_{d,i}\delta_{k,m}\boldsymbol{A}_{d,k} - \boldsymbol{M}_{d,i}\boldsymbol{A}_{d,k}\boldsymbol{A}_{d,m}. \tag{A.7}$$

Plugging this together we get

$$\begin{aligned}
\frac{\partial \hat{x}_d}{\partial(\hat{\boldsymbol{z}}_{B_m})_r} &= \partial_i\psi(\bar{\boldsymbol{x}})\frac{\partial(\bar{\boldsymbol{x}}_d)_i}{\partial(\hat{\boldsymbol{z}}_{B_m})_r} \\
&= \boldsymbol{A}_{d,m}\boldsymbol{W}^V_{i,r}\partial_i\psi(\bar{\boldsymbol{x}}) + \partial_i\psi(\bar{\boldsymbol{x}})\boldsymbol{W}^V_{i,j}(\hat{\boldsymbol{z}}_{B_k})_j\frac{\partial \boldsymbol{A}_{d,k}}{\partial(\hat{\boldsymbol{z}}_{B_m})_r} \\
&= \boldsymbol{A}_{d,m}\boldsymbol{W}^V_{i,r}\partial_i\psi(\bar{\boldsymbol{x}}) + \partial_i\psi(\bar{\boldsymbol{x}})\boldsymbol{W}^V_{i,j}(\hat{\boldsymbol{z}}_{B_k})_j(\boldsymbol{M}_{d,r}\delta_{k,m}\boldsymbol{A}_{d,k} - \boldsymbol{M}_{d,r}\boldsymbol{A}_{d,k}\boldsymbol{A}_{d,m}) \\
&= \boldsymbol{A}_{d,m}\partial_i\psi(\bar{\boldsymbol{x}})(\boldsymbol{W}^V_{i,r} + \boldsymbol{W}^V_{i,j}(\hat{\boldsymbol{z}}_{B_m})_j\boldsymbol{M}_{d,r}) - \partial_i\psi(\bar{\boldsymbol{x}})\boldsymbol{W}^V_{i,j}(\hat{\boldsymbol{z}}_{B_k})_j\boldsymbol{M}_{d,r}\boldsymbol{A}_{d,k}\boldsymbol{A}_{d,m}
\end{aligned} \tag{A.8}$$

From this, we can see that if $\boldsymbol{A}_{d,m} = 0$, then the partial derivative $\frac{\partial \hat{x}_d}{\partial \hat{\boldsymbol{z}}_{B_m}}$, will indeed be zero as $\boldsymbol{A}_{d,m}$ scales both terms in the last line of Eq. (A.8).

A.2 Compositionality Regularizer

Based on Appx. A.1, we propose to regularize for compositionality in a Transformer by minimizing the sum of all pairwise products $\boldsymbol{A}_{l,j}\boldsymbol{A}_{l,k}$, where $j \neq k$. More specifically, we minimize the following loss:

$$\mathcal{L}_{\text{comp}} := \mathbb{E}\sum_{l\in[d_x]}\sum_{j\in[K]}\sum_{k=j+1}^{K}\boldsymbol{A}_{l,j}(\hat{\boldsymbol{z}})\boldsymbol{A}_{l,k}(\hat{\boldsymbol{z}}) \tag{A.9}$$

where $\boldsymbol{A}_{l,k}(\hat{\boldsymbol{z}})$ is used to indicate the input dependence of attention weights on latents $\hat{\boldsymbol{z}}$. $\mathcal{L}_{\text{comp}}$ is a non-negative quantity which will be zero if and only if a matrix has at most one non-zero for each row [6].

Code to compute $\mathcal{L}_{\text{comp}}$ for a batch of attention matrices can be seen in Fig. 2. We note that when using multiple attention heads, we first sum the attention matrices over all heads to ensure consistent pixel assignments across different heads. When using multiple layers, we also sum the attention matrices over each layer, for the same reason. $\mathcal{L}_{\text{comp}}$ is then computed on the resulting attention matrix.

```python
def L_comp(attn):
    batch_size, num_slots, num_pixels = attn.shape
    interaction = 0
    for i in range(num_slots):
        for j in range(i, num_slots - 1):
            interaction += attn[:, i] * attn[:, j + 1]
    return interaction.mean()
```

Figure 2: PyTorch code to compute $\mathcal{L}_{\text{comp}}$.

Computational Efficiency. We note that regularizing with $\mathcal{L}_{\text{comp}}$ adds minimal additional computational overhead since attention weights are already computed at each forward pass through the model, and, moreover can be easily optimized using gradient descent. This is in contrast to Brady et al. [6] which required computing the Jacobian of the decoder $\hat{\boldsymbol{f}}$ at each forward pass and then optimizing it using gradient descent. This results in second-order optimization which is computationally intractable for high-dimensional data such as images [6].

B Extended Discussion

VAE Losses in Object-Centric Models. Prior work in Wang et al. [46] also apply a VAE loss to an unsupervised object-centric learning setting. However, while we minimize $\mathcal{L}_{\text{KL}}$ directly on inferred slots in $\hat{\boldsymbol{z}}$ given by our Transformer encoder, Wang et al. [46] minimize $\mathcal{L}_{\text{KL}}$ on an intermediate representation which is then further processed to yield $\hat{\boldsymbol{z}}$. Furthermore, the focus of Wang et al. [46] is on scene generation an not penalizing the capacity of $\hat{\boldsymbol{z}}$. Additionally, Kori et al. [24] explore a loss for object-centric learning resembling a VAE loss, though their aim is to enforce a certain probabilistic structure on $\hat{\boldsymbol{z}}$ implied by their theoretical disentanglement result, opposed to penalize latent capacity.

Relation Between a Transformer Regularized with $\mathcal{L}_{\text{comp}}$ and Prior Works. Goyal et al. [12] proposed RIMs which is a Transformer-style architecture aimed at enforcing a "modular" structure. Contrary to our work, Goyal et al. [12] do not regularize for modularity, but posit that it may emerge from "competition" induced by an attention mechanism. Similarly, Lamb et al. [26] propose an alternative Transformer architecture aimed at enforcing modularity, which also tries to enforce competition using a mechanism similar to Goyal et al. [12]. More recently, Vani et al. [44] proposed a Transformer component aimed at yielding disentanglement by processing a Transformer embedding into different slots using separate attention heads for each slot. While these works are similar to ours in that they aim to learn disentangled representations of concepts using a Transformer-style architecture, they are based on architectural changes to a Transformer, whereas we use a standard cross-attention Transformer decoder and regularize it explicitly towards having a modular structure using $\mathcal{L}_{\text{comp}}$.

C Experimental Details

C.1 Data, Model, and Training Details

Data. The Sprites dataset used in § 4 was generated using the Spriteworld renderer [47] and consist of 100,000 images of size $64 \times 64 \times 3$ each with between 2 and 4 objects. The CLEVR6 dataset [19, 29] consist of 53,483 images of size $128 \times 128 \times 3$ each with between 2 and 6 objects. For Sprites, we use 5,000 images for validation, 5,000 for testing, and the rest for training, while for CLEVR6, we use 2,000 images for validation and 2,000 for testing.

Encoders. All models use encoders which first process images using the same CNN of Locatello et al. [29]. When using a Transformer encoder, these CNN features are fed to a 5 layer Transformer which uses both self- and cross-attention with 4 attention heads. When using a Slot Attention encoder, we use 3 Slot Attention iterations, and use the improved implicit differentiation proposed in Chang et al. [9]. Both the Transformer and Slot Attention encoders use learned query vectors opposed to randomly sample queries. On Sprites, all models use 5 slots, each with 32 dimensions, while on CLEVR6, all models use 7 slots, each with 64 dimensions. When using a VAE loss, this slot dimension doubles since we must model the mean and variance of each latent dimension.

Decoders. When using a Spatial Broadcast decoder [48], we use the same architecture as [29] across all experiments, using a channel dimension of 32 for both datasets. When using a Transformer decoder, we first upscale slots to 516 dimensions by processing them separately using a 2 layer MLP, with a hidden dimension of 2064. We then apply a 2 layer cross-attention Transformer to these features which uses 12 attention heads. To obtain the vectors $\boldsymbol{o}_l$ in Eq. (3.1), we apply a 2D positional encoding to each pixel coordinate. This vector is then mapped by a 2 layer MLP with a hidden dimension of 360 to yield $\boldsymbol{o}_l$, which has dimension 180. The function ψ in Eq. (3.2) is implemented by a 3 layer MLP with a hidden dimension of 180, which outputs a 3 dimensional pixel $\hat{\boldsymbol{x}}_l$ for each pixel l. We additionally note that this architecture does not rely on auto-regressive masking as in Singh et al. [41].

Training Details. We train all models on Spriteworld across 3 random seeds using batches of 64 for 500,000 iterations. For CLEVR6, we use batches of 32 and train for 400,000 iterations. In all cases, we use the Adam optimizer [20] with a learning rate of 5×10^{-4} which we warm-up for the first 30,000 training iterations and then decay by a factor of 10 throughout training. When training with $\beta\mathcal{L}_{\text{KL}}$ and $\alpha\mathcal{L}_{\text{comp}}$, we use hyperparameter weights of 0.05, which we found to work well across both datasets. We found much larger values could lead to training instability and, in some cases, insufficient optimization of $\mathcal{L}_{\text{rec}}$, while smaller values often led to insufficient optimization of the regularizers. We warm-up the value of α for the first 30,000 training iterations. Additionally, when training with α or β, we drop the value of the learning rate after 30,000 training iterations to 1×10^{-4}, which improved training stability. Lastly, on Sprites, we weight $\mathcal{L}_{\text{rec}}$ by a factor of 5, when training with α or β.

C.2 Metrics and Evaluation

Computing ARI with Attention Scores. To compute the Adjusted Rand Index (ARI), each pixel must first be assigned to a unique model slot. To this end, prior works typically choose the slot with the largest attention score from either Slot Attention or the alpha mask of a Spatial Broadcast

decoder [29, 40]. This approach can be problematic since the attention scores used are model-dependent, making a direct comparison of ARI across models challenging. Further, the relationship between attention scores and the pixels encoded in a model slot is somewhat indirect. As noted in § 4, we consider an alternative and compute the ARI using the Jacobian of a decoder (J-ARI). Specifically, we assign a pixel l to the slot with the largest L_1 norm for the slot-wise Jacobian $D_{B_k}\hat{f}_l(\hat{\boldsymbol{z}})$. This can be done for any autoencoder and provides a more principled metric for object disentanglement since a decoder's Jacobian directly describes the pixels each slot encodes (assuming $\hat{\boldsymbol{f}}, \hat{\boldsymbol{g}}$ invert each other).

Evaluation. We select models for testing which had the highest average values for J-ARI and JIS (each of which take values from 0 to 1) on the validation set. These models were then evaluated on the test set yielding the scores reported in Tab. 1.

C.3 Additional Figures

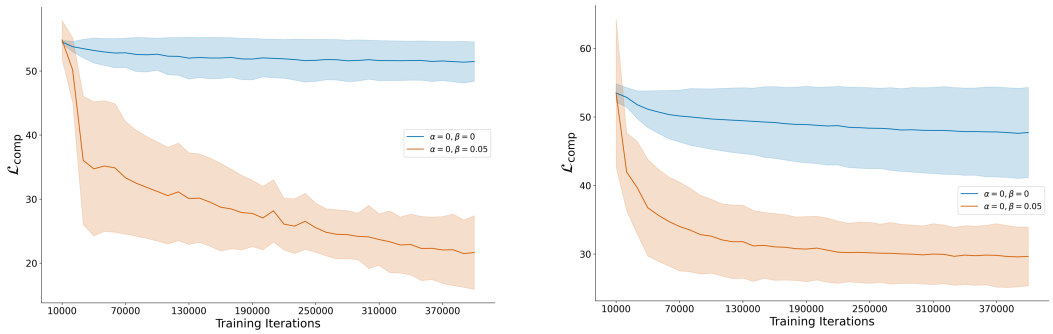

Figure 3: **Analysis of $\mathcal{L}_{\text{comp}}$ when using a VAE loss.** We plot $\mathcal{L}_{\text{comp}}$ for the first 400,000 training iterations for a Transformer autoencoder trained without regularization ($\alpha = 0, \beta = 0$) and with a VAE loss which does not explicitly optimize $\mathcal{L}_{\text{comp}}$, ($\alpha = 0, \beta = 0.05$). We find on Sprites (left) and CLEVR6 (right), the VAE loss achieves much lower $\mathcal{L}_{\text{comp}}$ than the unregularized model. This provides an explanation for the solid object-disentanglement often achieved by the VAE loss in Tab. 1