Interaction Information Optimization for Object-Centric Representation Learning

Riccardo Majellaro [1] **Jonathan Collu** [1] **Dimitrios Ieronymakis** [1]

Abstract

Being able to recognize and represent individual objects is a core component of human cognition. However, current representation learning approaches in the vision domain rarely represent scenes as sets of object-centric vectors. These structured representations could be largely beneficial to, for example, generalization capabilities, sample efficiency on downstream tasks, and modeling of object interactions in image generation and model-based reinforcement learning. In this work, we propose a method based on the information-theoretic concept of interaction information intending to improve performances and sample efficiency of object-centric models on the object discovery task. Our approach is derived by generalizing from 1 to N foreground objects the objective presented in the Inpainting Error Maximization [10] framework. Experiments on Slot Attention [7] over the Tetrominoes dataset show that our strategy can be effective, outperforming the baseline both considering and ignoring the background. The code to reproduce our experiments is available at https://github.com/riccardomajellaro/IIO-SlotAttention.

1. Introduction

The ability to perceive and decompose entities in complex visual scenes without supervision, although trivial for a cognitive being, remains an open challenge for computer vision. Solving this problem could have a significant impact on approaches such as reinforcement learning with graph neural networks [8, 6], or text-to-image generative models.

In recent breakthroughs such as [1, 4, 7], images are first decomposed into multiple vectors, each aiming to represent an object of the scene, then reconstructed back by merging together the inferred masks and textures of the entities from their representations. In particular, Slot Attention [7] is trained end-to-end by simply minimizing the reconstruction error which, despite being important for pushing the model to learn informative representations, does not explicitly guide each of them towards being related to a single object. Nevertheless, the Slot Attention mechanism enables competition between the slots for explaining parts of the image, allowing the model to reach the desired goal after an extensive training process. Extensions of this work [2, 11, 5] achieve better performances through variations in the training procedure and model architecture.

Other approaches addressing different problems may offer solutions that, if properly adapted to the context, could help enhance current unsupervised object-centric representation learning techniques. We see in [10] an example of this, where the authors propose to minimize the mutual information between foreground and background masks, with the goal of correctly segmenting an image into two partitions.

In this document, we propose a method that combines and extends the ideas of [7] and [10], aiming to improve training times and the quality of object-centric representations. We employ the Slot Attention architecture and train it using a combination of the reconstruction loss and a generalized formulation of the inpainting error maximization objective.

The structure of the document is organized as follows: Section 2 introduces the background knowledge required to understand the topic, Section 3 describes the related work and Section 4 describes the proposed method. In Section 5 we present the experiments carried out and the obtained results along with a discussion of these, while Section 6 corresponds to the conclusion.

2. Background

Slot Attention Locatello et al. introduce in [7] the Slot Attention module, responsible for producing through an iterative attention mechanism a set of K output vectors (*slots*) from a set of N input perceptual representations. In the paper, the latter are defined as feature vectors at the output

[1]Leiden Institute of Advanced Computer Science, Leiden University. Correspondence to: Riccardo Majellaro <r.majellaro@umail.leidenuniv.nl>, Jonathan Collu <j.collu@umail.leidenuniv.nl>.

Preprint

of a CNN backbone augmented with positional embeddings. The slots are initialized by independently sampling each from a Gaussian distribution with shared and learnable parameters $\mu, \sigma \in \mathbb{R}^{D_{\texttt{slots}}}$ ($D_{\texttt{slots}}$ is the dimension of a slot), which allows Slot Attention to generalize the number of slots at test time. The input feature vectors $\mathcal{X} \in \mathbb{R}^{N \times D_{\texttt{inputs}}}$ (with $D_{\texttt{inputs}}$ being the dimension of a feature vector) are first mapped to dimension D through the learnable linear transformations k (keys) and v (values). Then, at each of the T iterations, the slots $\mathcal{S} \in \mathbb{R}^{K \times D_{\texttt{slots}}}$ are refined through a series steps. First, a dot-product attention between keys and queries is computed:

$$\widetilde{\mathcal{A}} = \frac{k(\mathcal{X}) \cdot q(\mathcal{S})^T}{\sqrt{D}} \in \mathbb{R}^{N \times K}, \tag{1}$$

where q (queries) is a learnable linear transformation mapping the slots to dimension D. The attention coefficients are then normalized (softmax) over the slots (queries), in order to introduce competition between the slots for explaining parts of the input:

$$\mathcal{A}_{i,j} = \frac{\exp \widetilde{\mathcal{A}}_{i,j}}{\sum_{l=1}^{K} \exp \widetilde{\mathcal{A}}_{i,l}}. \tag{2}$$

It follows an aggregation of the input values for each slot (called *updates*) by a convex combination using, as weights, $\mathcal{A}$ normalized over the keys:

$$\mathcal{U} = \mathcal{W}^T \cdot v(\mathcal{X}) \in \mathbb{R}^{K \times D} \quad , \quad \mathcal{W}_{i,j} = \frac{\mathcal{A}_{i,j}}{\sum_{l=1}^{N} \mathcal{A}_{l,j}} \tag{3}$$

Finally, the updates $\mathcal{U}$ are fed to a Gated Recurrent Unit (GRU) followed by a MLP (ReLU as activation function) with a residual connection. After the T iterations, each of the K final slots is individually decoded into an image and a mask. The decoded masks are then normalized across the slots and used as weights for combining the K decoded images into the final reconstructed image. The training objective is the minimization of the reconstruction loss, defined as the mean squared error between the original image and the reconstructed one.

Inpainting Error Maximization Savarese et al. propose in [10] a method for unsupervised segmentation by partitioning pixels into two sets, foreground and background. The approach relies on the information-theoretic concept of mutual information between two random variables F and B:

$$I(F, B) = H(F) - H(F|B) = H(B) - H(B|F) \ , \tag{4}$$

with $H(F)$ being the entropy of F and $H(F|B)$ the conditional entropy of F given B. When F and B are independent, $H(F|B) = H(F)$ and $H(B|F) = H(B)$, hence $I(F, B) = 0$. Therefore, given an image $X \in \mathbb{R}^{C \times H \times W}$ (C, H and W represent respectively channels, width and height of the image) the aim is to extract a binary mask $\mathcal{M} \in \{0, 1\}^{H \times W}$ and its complementary $\overline{\mathcal{M}}$ such that $I(F_{\mathcal{M}}, B_{\mathcal{M}}) = 0$, where $F_{\mathcal{M}} = X \odot \mathcal{M}$ and $B_{\mathcal{M}} = X \odot \overline{\mathcal{M}}$ are the foreground and background pixel partitions. A problem with this formulation is the presence of a trivial solution that minimizes the mutual information by simply collapsing the two masks, in other words one is empty (all zeroes) and the other one is full (all ones). To partially avoid this problem, the authors replace the formulation with a normalized variant of it:

$$\begin{aligned} C(F_{\mathcal{M}}, B_{\mathcal{M}}) &= \frac{I(F_{\mathcal{M}}, B_{\mathcal{M}})}{H(F_{\mathcal{M}})} + \frac{I(B_{\mathcal{M}}, F_{\mathcal{M}})}{H(B_{\mathcal{M}})} \\ &= 2 - \left(\frac{H(F_{\mathcal{M}}|B_{\mathcal{M}})}{H(F_{\mathcal{M}})} + \frac{H(B_{\mathcal{M}}|F_{\mathcal{M}})}{H(B_{\mathcal{M}})} \right). \end{aligned} \tag{5}$$

The entropy $H(F_{\mathcal{M}})$ is defined as the "entry-wise" matrix L_1 norm i.e., in this case, equal to the number of pixels considered by the binary mask. The conditional entropy is instead represented as

$$\begin{aligned} H(F_{\mathcal{M}}|B_{\mathcal{M}}) &= ||F_{\mathcal{M}} - \mathcal{M} \odot \psi_{\mathsf{K}}(B_{\mathcal{M}}, \overline{\mathcal{M}})||_1 \\ &= ||\mathcal{M} \odot (X - \psi_{\mathsf{K}}(X \odot \overline{\mathcal{M}}, \overline{\mathcal{M}})||_1, \end{aligned} \tag{6}$$

with ψ_{K} being the inpainting module. Precisely,

$$\psi_{\mathsf{K}}(X, \mathcal{M}) = \frac{\mathsf{K} * X}{\mathsf{K} * \mathcal{M}}, \tag{7}$$

where $*$ indicates the convolution operator and K is a Gaussian filter defined as

$$\mathsf{K}_{i,j} \propto \frac{1}{2\pi\sigma^2} \exp\left(-\frac{i^2 + j^2}{2\sigma^2} \right). \tag{8}$$

K is additionally normalized so that its elements sum up to one. In practice, before summing up the absolute differences in 6, an average over the channels is taken, leaving a mean absolute distance term per pixel. For this reason, each conditional entropy ranges from 0, when the inpainting perfectly corresponds to the original image in the considered mask area, to $H(F_{\mathcal{M}}) = ||\mathcal{M}||_1$, in the case of maximum inpainting error. The search for a binary mask $\mathcal{M}$ can finally be defined by the following objective function:

$$\begin{aligned} \max_{\mathcal{M} \in \{0,1\}^{H \times W}} & \frac{||\mathcal{M} \odot (X - \psi_{\mathsf{K}}(X \odot \overline{\mathcal{M}}, \overline{\mathcal{M}}))||_1}{||\mathcal{M}||_1} \\ & + \frac{||\overline{\mathcal{M}} \odot (X - \psi_{\mathsf{K}}(X \odot \mathcal{M}, \mathcal{M}))||_1}{||\overline{\mathcal{M}}||_1}. \end{aligned} \tag{9}$$

3. Related Work

Object-centric representation learning is recently arousing interest among Computer vision and deep learning researchers. As a result, proposals bringing precious improvements in the field are rising in number and heterogeneity of ideas. Different unsupervised approaches, including

[3, 4, 1, 7], perform object discovery by decomposing input scenes into multiple latent variables, each representing a single object. Within these proposals, Slot Attention [7] emerges for being faster to train and more memory efficient while matching or even outperforming the other methods. More recent techniques obtain SOTA performances by extending Slot Attention in various ways. In [2] an implicit differentiation is introduced to avoid differentiating through the unrolled refinement process, while Singh et al. [11] experiment with a transformer-based variant of the Slot Attention architecture. The work presented in [5], instead, replaces the initial random slots with learned ones and applies a bi-level optimization strategy. In a different line of work, approaches based on information-theoretic concepts attempt to tackle the task of unsupervised segmentation. For instance, [10, 12, 13] minimize mutual information to partition image pixels in background and foreground sets. In [12], the authors train an inpainter to reconstruct a masked flow adversarially with a generator that produces masks with the aim of leading the inpainter to mispredict the original optical flow. In [10], instead, the inpainter is fixed and thus no training is required. Here, the discrete foreground mask is iteratively optimized to share as little information as possible (mutual information minimization), resulting in a more accurate segmentation. Our contribution is to extend the loss formulation provided by [10] to work with multiple masks, and minimize it in combination with the reconstruction loss to train Slot Attention. In some cases, this strategy can efficiently push Slot Attention to infer masks that better separate information and overcome some of its limitations.

4. Methods

4.1. Interaction Information Optimization

When dealing with multiple objects (two or more excluding the background), two masks are no longer sufficient and minimizing equations 4 or 5 loses significance. The generalization to n variables of the mutual information (eq. 4) is the interaction information, defined as follows:

$$I(Z_1, ..., Z_n) = I(Z_1, ..., Z_{n-1}) - I(Z_1, ..., Z_{n-1}|Z_n) \ , \tag{10}$$

where

$$\begin{aligned} I(Z_1, ..., Z_{n-1}|Z_n) = I(Z_1, ..., Z_{n-2}|Z_n) \\ - I(Z_1, ..., Z_{n-2}|Z_{n-1}, Z_n) \end{aligned} \tag{11}$$

is the conditional interaction information, as reported in 1.15 of [9]. We consider the interaction information conditioned by more than one variable as

$$\begin{aligned} I(Z_1, ..., Z_{n-i}|Z_{n-i+1}, ..., Z_n) = \\ I(Z_1, ..., Z_{n-i-1}|Z_{n-i+1}, ..., Z_n) \\ - I(Z_1, ..., Z_{n-i-1}|Z_{n-i}, Z_{n-i+1}, ..., Z_n) \ , \end{aligned} \tag{12}$$

which corresponds to 11 in the case $i = 1$. Additionally, as proved in Appendix A, $I(Z_1, Z_2|Z_3, ..., Z_n)$ can be represented in terms of conditional entropy as

$$\begin{aligned} I(Z_1, Z_2|Z_3, ..., Z_n) = H(Z_1|Z_3, ..., Z_n) \\ - H(Z_1|Z_2, Z_3, ..., Z_n) \ , \end{aligned} \tag{13}$$

which, when $n = 3$, corresponds to the definition of conditional mutual information $I(Z_1, Z_2|Z_3) = H(Z_1|Z_3) - H(Z_1|Z_2, Z_3)$. This, along with equations 12 and 4, allows us to define equation 10 in terms of entropy and conditional entropy. For instance, with $n = 4$, the expansion is:

$$\begin{aligned} I(Z_1, Z_2, Z_3, Z_4) = H(Z_1) - H(Z_1|Z_2, Z_3, Z_4) \\ - H(Z_1|Z_2) - H(Z_1|Z_3) - H(Z_1|Z_4) \\ + H(Z_1|Z_2, Z_3) + H(Z_1|Z_2, Z_4) + H(Z_1|Z_3, Z_4) \ . \end{aligned}$$

Note that all the terms are written with respect to the first variable, Z_1 in this case, therefore we refer to this formulation as "I centered on Z_1". Consider now an image $X \in \mathbb{R}^{C \times H \times W}$ and a function ϕ_θ, parametrized by θ, that maps X to a set of K normalized continuous masks $\mathcal{M} = \phi_\theta(X)$, with each mask $m_i \in \mathcal{M}$ for $i = 1, ..., K$ being in $[0, 1]^{H \times W}$. By modeling $H(X \odot m_1|X \odot m_2, .., X \odot m_K)$ as $H(X \odot m_1|X \odot \sum_{i=2}^{K} m_i)$, it is possible to compute the conditional entropy on the right side of equation 13 as in 6, thus

$$||m_1 \odot (X - \psi_{\mathsf{K}}(X \odot \sum_{i=2}^{K} m_i, \sum_{i=2}^{K} m_i)||_1 \ .$$

The entropy $H(X \odot m_1)$ is again modeled as $||m_1||_1$. In this way, we can finally compute the interaction information $I(X \odot m_1, ..., X \odot m_K)$ between K masks of an image X, which is composed of 2^{K-1} (conditional) entropy terms, half of them with positive sign and half with negative sign, each ranging from 0 to $H(X \odot m_1) = ||m_1||_1$. Our goal is now to maximize the inpainting error (conditional entropy) terms in I, leading in the optimal case to 2^{K-2} positive terms and 2^{K-2} negative terms of magnitude $||m_1||_1$, and therefore to $I = 0$. In Appendix B we present more details regarding the expansion of I. Moreover, we empirically found out that the optimization of K interaction information terms, each centered on one of the K variables and normalized by the entropy of that variable, is a more robust objective than the optimization of a single I term. The loss is finally formulated as follows:

$$\mathcal{L}(\theta; X) = - \sum_{m \in \mathcal{M}} \sum_{i=1}^{|\mathcal{M}|-1} \sum_{\substack{C \in \\ \mathcal{C}_i(\mathcal{M} \setminus m)}} \frac{H\left(X \odot m | X \odot \sum\limits_{c \in C} c\right)}{||m||_1} \ , \tag{14}$$

where $\mathcal{C}_i(\mathcal{M})$ represents the set of combinations of i element from set $\mathcal{M}$ (without repetitions). In our experiments, we employ Slot Attention for mapping images to masks, and we optimize its parameters by minimizing the loss $\mathcal{L}$.

4.2. Training procedure

Since our loss does not take the reconstruction quality into account, optimizing that alone would not ensure that the learned representations contain information about the textures of the objects. Moreover, when different objects share the same or very similar colors, our loss function tends to group them in a single slot, while when a single object includes very different colors it gets easily divided into separate slots. For these reasons, it is crucial to optimize both reconstruction and interaction information. However, we experimented that a simple weighted sum of the two loss terms leads to highly unstable training. Thus, our strategy consists of, first, pre-training Slot Attention by minimizing only the reconstruction loss until the objects are decoded with correct textures and reasonably separated amongst the slots. There is no need to reach a perfect segmentation during this phase; what we aim for is that every single slot has its own object to pay attention to. After the pre-training, we fine-tune the model by optimizing the interaction information for a small number of epochs, which is usually sufficient to reach a near-optimal segmentation. In order to avoid that the optimization of the interaction information degrades the results obtained by the first phase, during the fine-tuning we introduce a hinge term beside our loss, which makes sure that the reconstruction error remains lower than a fixed margin. Specifically, when the reconstruction loss is lower than a threshold, its value is set to zero and only the inpainting error is considered, while when it is larger its value is scaled up to match the order of the other loss term. The scaling factor is important since the reconstruction loss is usually different orders of magnitude smaller with respect to our loss, and a good balance between the functions to be optimized is necessary.

5. Experiments

5.1. Experimental Setup

ARI score The adjusted rand index (ARI) score is an evaluation metric used to measure the similarity between two clusters, and it is defined as:

$$\text{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_i \binom{b_j}{2}\right] / \sum_i \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_i \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_i \binom{b_j}{2}\right] / \sum_i \binom{n}{2}} .$$

Consider a set of n elements and two clusters of these, namely the ground truth one $\mathcal{Y} = \{Y_1, ..., Y_r\}$ and the predicted one $\mathcal{X} = \{X_1, ..., X_s\}$. Let n_{ij} be the number of elements in common between X_i and Y_j, a_i the sum of all n_{ij} for a fixed i, and b_j is the sum of all n_{ij} for a fixed j. The term $\sum_{ij}\binom{n_{ij}}{2}$ is the rand index (RI) and measures the ratio between the number of correctly clustered elements and the total number of pairs. The term $\left[\sum_i \binom{a_i}{2}\sum_i \binom{b_j}{2}\right] / \sum_i \binom{n}{2}$ is the expected RI and is introduced to have a more precise measure of the similarity between two clusters. In fact, the plain RI can assume real values between 0 and 1 (the closer is the RI to 1 the more similar the two clusters are), while ARI can assume values between -0.5 and 1. An ARI value of 1 represents a perfect clustering and, as it tends to 0, it gets closer to mirroring a random one. A negative ARI indicates, instead, a worse than random performance. This metric is the one used in [7] to compute the similarity between the ground truth masks and the predicted ones. In their experiments, the authors of the above-mentioned paper did not consider the background in the ARI score computation, while in the next sections, we present our results both including and excluding it.

Datasets We investigated the effect of the interaction information optimization on Slot Attention using one of the three datasets considered in the original Slot Attention paper. The dataset is named Tetrominoes and is composed of 35×35 images, each containing three Tetris-like shapes (sampled from 17 unique shapes and orientations and six possible colors) on a black background. We trained over 60K samples and evaluated over 1K. The dataset is available at https://github.com/deepmind/multi_object_datasets.

Experiments design We pre-trained Slot Attention (following the hyperparameters configurations suggested in [7]) for 50 epochs on Tetrominoes. At this point, we fine-tuned the model for 5 epochs by optimizing the interaction information to infer more precise object borders and confine the information related to the background in a single mask (in the case of Tetrominoes). For our experiments, the hinge margin was set to $4\times^{-4}$. The scaling factor of the hinge loss was set to 10^3 in order to increase the importance of the reconstruction error when over the threshold.

Baseline We compared our method with the plain Slot attention trained for 55 and 200 epochs on Tetrominoes. It is relevant to highlight that in [7], Slot Attention has been trained for longer, however since the main purpose of this work is to make Slot Attention's training more efficient, we considered comparing our method with Slot Attention trained for the same number of epochs (55) and for 200 epochs.

5.2. Quantitative Results

Tetrominoes As observable from Table 1, if we do not consider the background, Slot Attention can predict close-

ARI score on Tetrominoes		
Experiment	w/background	no/background
SA (55)	47.40 ± 1.00	96.00 ± 0.34
SA (200)	46.38 ± 0.79	$\mathbf{98.56 \pm 0.32}$
SA (50) + IIO (5)	$\mathbf{96.53 \pm 0.41}$	97.77 ± 0.31

Table 1. ARI score obtained on Tetrominoes both considering and not considering the background. The first row reports Slot Attention after 55 epochs of training, while the second row presents Slot Attention after 200 epochs. The third row shows Slot Attention pre-trained for 50 epochs and fine-tuned with the interaction information for 5 epochs. Since the slots are initialized randomly and specialized iteratively, the results (reported in the format: $mean \pm std$) are averaged over 5 repetitions.

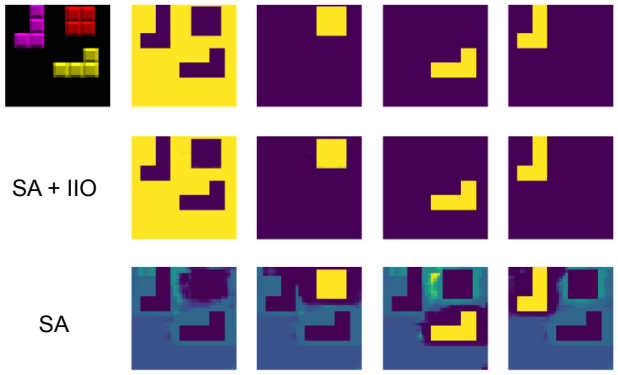

Figure 1. Comparison between the masks produced by Slot Attention on Tetrominoes with and without our interaction information optimization. The first row reports the ground truth masks, the second row SA pre-trained for 50 epochs and fine-tuned with IIO for 5 epochs, while the third row SA trained for 200 epochs without IIO fine-tuning.

to-perfect object segmentations already after 55 training epochs. When considering the background, instead, we find that the results are fairly poor. At 200 epochs, the score for the foreground objects slightly improves as the model learns to infer more precise borders around the entities, but gets even worse at partitioning the background. These results are supported by the fact that, as mentioned in [7], Slot Attention tends on average to spread the background information uniformly among the slots instead of confining it in a single one, especially on datasets where the background texture is fixed for all the samples. The authors state that this phenomenon does not affect foreground object segmentation and, although it is ulteriorly confirmed in our experiments, it still limits the quality of the representations. This behavior could reduce the benefits of employing object-centric representations in approaches such as model-based reinforcement learning. By focusing on our method results, we can observe that with only 5 fine-tuning epochs of a model pre-trained for 50, the foreground score is higher than the one achieved by Slot Attention after 55 epochs, and gets very close to the one obtained by training for almost four times longer (200 epochs). When taking the background into account, our method drastically outperforms both baselines. For this reason, the proposed method is particularly suited for situations where sample efficiency is crucial and an inaccurate representation of static objects, such as a fixed-color background, could have a large negative impact. These results are visualized and further interpreted with the help of qualitative examples in the next section.

5.3. Qualitative Results

Tetrominoes Figures 1, 2, and 3 show the qualitative results by displaying both the ground truth and the inferred masks. Coherently with the quantitative results, we can observe that the masks produced by Slot Attention after 200 training epochs are able to perfectly segment the objects, but not to recognize the background as a separate entity (presumably because it is static in this dataset). In fact, the model incorrectly learned to distribute the background area across all the masks. On the other hand, with our method, in just 55 epochs we obtain almost perfect masks for all the entities, background included. In a few cases such as Figure 3, a mask can contain small imperfections, justifying the slightly lower performances compared to Slot Attention trained for 200 epochs. Additional training, either pre-training or fine-tuning, can mitigate these minor defects.

6. Conclusion

In this work, we presented a generalized formulation of the inpainting error maximization (IEM) framework on more than two masks and combined it with Slot Attention. This strategy aims at enhancing training efficiency and object decomposition capabilities without degrading the information content obtained through the minimization of the reconstruction error. The experimental results showed the effectiveness of our method on Tetrominoes, where augmenting Slot Attention with our interaction information based objective led to improved data efficiency and significant gains over the baseline when considering the background. However, conducting additional experiments on different datasets is required to gain a deeper understanding of the possible limitations and advantages of the method we proposed.

References

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

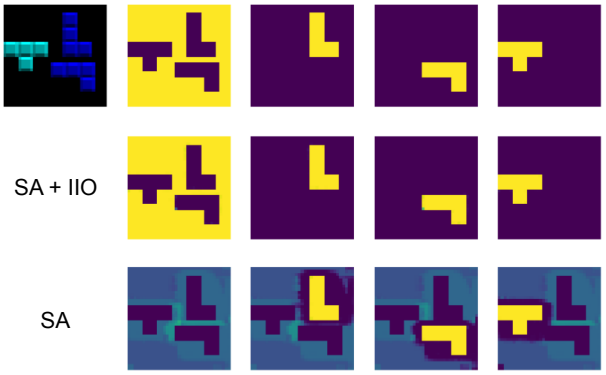

Figure 2. Comparison between the masks produced by Slot Attention on Tetrominoes with and without our interaction information optimization. The first row reports the ground truth masks, the second row SA pre-trained for 50 epochs and fine-tuned with IIO for 5 epochs, while the third row SA trained for 200 epochs without IIO fine-tuning.

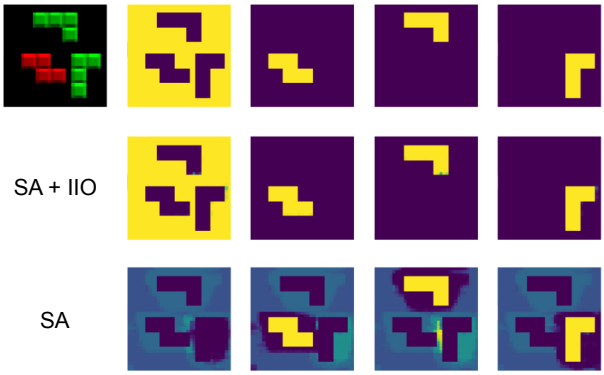

Figure 3. Comparison between the masks produced by Slot Attention on Tetrominoes with and without our interaction information optimization. The first row reports the ground truth masks, the second row SA pre-trained for 50 epochs and fine-tuned with IIO for 5 epochs, while the third row SA trained for 200 epochs without IIO fine-tuning.

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

[3] Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. Genesis: Generative scene inference and sampling with object-centric latent representations. *arXiv preprint arXiv:1907.13052*, 2019.

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

[5] Jia, B., Liu, Y., and Huang, S. Improving object-centric learning with query optimization. In *The Eleventh International Conference on Learning Representations*, 2022.

[6] Kipf, T., Van der Pol, E., and Welling, M. Contrastive learning of structured world models. *arXiv preprint arXiv:1911.12247*, 2019.

[7] Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

[8] Lu, Y., Chen, Y., Zhao, D., and Li, D. Mgrl: Graph neural network based inference in a markov network with reinforcement learning for visual navigation. *Neurocomputing*, 421:140–150, 2021.

[9] Sakaguchi, M. Interaction information in multivariate probability distributions. In *Kodai Mathematical Seminar Reports*, volume 19, pp. 147–155. Department of Mathematics, Tokyo Institute of Technology, 1967.

[10] Savarese, P., Kim, S. S., Maire, M., Shakhnarovich, G., and McAllester, D. Information-theoretic segmentation by inpainting error maximization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4029–4039, 2021.

[11] Singh, G., Deng, F., and Ahn, S. Illiterate dall-e learns to compose. *arXiv preprint arXiv:2110.11405*, 2021.

[12] Yang, Y., Loquercio, A., Scaramuzza, D., and Soatto, S. Unsupervised moving object detection via contextual information separation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 879–888, 2019.

[13] Yang, Y., Lai, B., and Soatto, S. Time-supervised primary object segmentation. *ArXiv*, abs/2008.07012, 2020.

Appendix

A. Mutual information conditioned on multiple variables

We show here that $I(Z_1, Z_2|Z_3, \dots, Z_n) = H(Z_1|Z_3, \dots, Z_n) - H(Z_1|Z_2, Z_3, \dots, Z_n)$, where $Z_1, \dots, Z_n$ are continuous random variables with support sets $\mathcal{Z}_1, \dots, \mathcal{Z}_n$.

$$I(Z_1, Z_2|Z_3, \dots, Z_n) =$$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} p(z_3, \dots, z_n) \left(\int_{\mathcal{Z}_2} \int_{\mathcal{Z}_1} p(z_1, z_2|z_3, \dots, z_n) \log \frac{p(z_1, z_2|z_3, \dots, z_n)}{p(z_1|z_3, \dots, z_n) p(z_2|z_3, \dots, z_n)} \, dz_1 dz_2 \right) dz_3 \dots dz_n =$$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} p(z_3, \dots, z_n) \left(\int_{\mathcal{Z}_2} \int_{\mathcal{Z}_1} p(z_1, z_2|z_3, \dots, z_n) \log \frac{p(z_1, z_2|z_3, \dots, z_n)}{p(z_2|z_3, \dots, z_n)} \, dz_1 dz_2 \right) dz_3 \dots dz_n -$$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} p(z_3, \dots, z_n) \left(\int_{\mathcal{Z}_2} \int_{\mathcal{Z}_1} p(z_1, z_2|z_3, \dots, z_n) \log p(z_1|z_3, \dots, z_n) \, dz_1 dz_2 \right) dz_3 \dots dz_n .$$

Since $p(z_1, z_2|z_3, \dots, z_n) = \dfrac{p(z_1, z_2, z_3, \dots, z_n)}{p(z_3, \dots, z_n)} = \dfrac{p(z_1|z_2, z_3, \dots, z_n) p(z_2|z_3, \dots, z_n) p(z_3, \dots, z_n)}{p(z_3, \dots, z_n)} =$
$p(z_1|z_2, z_3, \dots, z_n) p(z_2|z_3, \dots, z_n)$, then $I(Z_1, Z_2|Z_3, \dots, Z_n) =$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} p(z_3, \dots, z_n) \left(\int_{\mathcal{Z}_2} p(z_2|z_3, \dots, z_n) \int_{\mathcal{Z}_1} p(z_1|z_2, z_3, \dots, z_n) \log p(z_1|z_2, z_3, \dots, z_n) \, dz_1 dz_2 \right) dz_3 \dots dz_n -$$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} p(z_3, \dots, z_n) \left(\int_{\mathcal{Z}_2} \int_{\mathcal{Z}_1} p(z_1, z_2|z_3, \dots, z_n) \log p(z_1|z_3, \dots, z_n) \, dz_1 dz_2 \right) dz_3 \dots dz_n =$$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} \int_{\mathcal{Z}_2} p(z_2, z_3, \dots, z_n) \left(\int_{\mathcal{Z}_1} p(z_1|z_2, z_3, \dots, z_n) \log p(z_1|z_2, z_3, \dots, z_n) \, dz_1 \right) dz_2 dz_3 \dots dz_n -$$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} p(z_3, \dots, z_n) \left(\int_{\mathcal{Z}_2} \int_{\mathcal{Z}_1} p(z_1, z_2|z_3, \dots, z_n) \log p(z_1|z_3, \dots, z_n) \, dz_1 dz_2 \right) dz_3 \dots dz_n .$$

By switching the order of integration in the double integral of the last term, and by knowing that

$p(z_1, |z_3, \dots, z_n) = \displaystyle\int_{\mathcal{Z}_2} p(z_1, z_2|z_3, \dots, z_n) \, dz_2$ (marginalization), then: $I(Z_1, Z_2|Z_3, \dots, Z_n) =$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_2} p(z_2, \dots, z_n) \left(\int_{\mathcal{Z}_1} p(z_1|z_2, \dots, z_n) \log p(z_1|z_2, \dots, z_n) \, dz_1 \right) dz_2 \dots dz_n -$$

$$\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} p(z_3, \dots, z_n) \left(\int_{\mathcal{Z}_1} p(z_1|z_3, \dots, z_n) \log p(z_1|z_3, \dots, z_n) \, dz_1 \right) dz_3 \dots dz_n .$$

Given that, by definition,

$$H(Z_1|Z_3, \dots, Z_n) = -\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_3} p(z_3, \dots, z_n) \left(\int_{\mathcal{Z}_1} p(z_1|z_3, \dots, z_n) \log p(z_1|z_3, \dots, z_n) \, dz_1 \right) dz_3 \dots dz_n$$

and $H(Z_1|Z_2, Z_3, \dots, Z_n) =$

$$-\int_{\mathcal{Z}_n} \dots \int_{\mathcal{Z}_2} p(z_2, \dots, z_n) \left(\int_{\mathcal{Z}_1} p(z_1|z_2, \dots, z_n) \log p(z_1|z_2, \dots, z_n) \, dz_1 \right) dz_2 \dots dz_n ,$$

then we finally obtain that $I(Z_1, Z_2|Z_3, \dots, Z_n) = H(Z_1|Z_3, \dots, Z_n) - H(Z_1|Z_2, Z_3, \dots, Z_n)$.

B. Expansion of $I(Z_1, ..., Z_n)$

We show here the expansion of $I(Z_1, ..., Z_n)$, centered on Z_1, by using equations 10, 12, 13 and 4.

t = 0 $\quad I(Z_1, ..., Z_n) =$

t = 1 $\quad I(Z_1, ..., Z_{n-1}) - I(Z_1, ..., Z_{n-1}|Z_n) =$

t = 2 $\quad I(Z_1, ..., Z_{n-2}) - I(Z_1, ..., Z_{n-2}|Z_{n-1}) - I(Z_1, ..., Z_{n-2}|Z_n) + I(Z_1, ..., Z_{n-2}|Z_{n-1}, Z_n) =$

$\vdots \qquad \vdots \qquad \vdots \qquad \vdots$

t = n-2 $\quad I(Z_1, Z_2) - I(Z_1, Z_2|Z_3) - I(Z_1, Z_2|Z_4) + I(Z_1, Z_2|Z_3, Z_4) - I(Z_1, Z_2|Z_5) + I(Z_1, Z_2|Z_3, Z_5)$
$+ I(Z_1, Z_2|Z_4, Z_5) - I(Z_1, Z_2|Z_3, Z_4, Z_5) \quad \cdots$

t = n-1 $\quad H(Z_1) - H(Z_1|Z_2) - H(Z_1|Z_3) + H(Z_1|Z_2, Z_3) - H(Z_1|Z_4) + H(Z_1|Z_2, Z_4) + H(Z_1|Z_3, Z_4)$
$- H(Z_1|Z_2, Z_3, Z_4) - H(Z_1|Z_5) + H(Z_1|Z_2, Z_5) + H(Z_1|Z_3, Z_5) - H(Z_1|Z_2, Z_3, Z_5)$
$+ H(Z_1|Z_4, Z_5) - H(Z_1|Z_2, Z_4, Z_5) - H(Z_1|Z_3, Z_4, Z_5) + H(Z_1|Z_2, Z_3, Z_4, Z_5) \quad \cdots$

At each step t every term in the previous step is expanded in two terms, as equations 10, 12, 13 and 4 all have one positive and one negative term on the right side. The number of terms at step t follows therefore the expression 2^t, and since the expansion stops at $t = n - 1$, the total number of terms in $I(Z_1, ..., Z_n)$ is 2^{n-1}, half of which positives (i.e. 2^{n-2}) and half negatives. Note that at the last step all the terms are entropy and conditional entropy functions, each with respect to (centered on) Z_1. Moreover, as conditional variables inside these conditional entropy terms, we find all the combinations (without repetitions) of variables in the set $\{Z_2, ..., Z_n\}$, considering combinations of size from 1 to $n - 1$.