Invariant Slot Attention: Object Discovery with Slot-Centric Reference Frames

Ondrej Biza [*1] **Sjoerd van Steenkiste** [2] **Mehdi S. M. Sajjadi** [2] **Gamaleldin F. Elsayed** [†2]
Aravindh Mahendran [†2] **Thomas Kipf** [†2]

Abstract

Automatically discovering composable abstractions from raw perceptual data is a long-standing challenge in machine learning. Recent slot-based neural networks that learn about objects in a self-supervised manner have made exciting progress in this direction. However, they typically fall short at adequately capturing spatial symmetries present in the visual world, which leads to sample inefficiency, such as when entangling object appearance and pose. In this paper, we present a simple yet highly effective method for incorporating spatial symmetries via slot-centric reference frames. We incorporate equivariance to per-object pose transformations into the attention and generation mechanism of Slot Attention by translating, scaling, and rotating position encodings. These changes result in little computational overhead, are easy to implement, and can result in large gains in terms of data efficiency and overall improvements to object discovery. We evaluate our method on a wide range of synthetic object discovery benchmarks namely Tetrominoes, CLEVR-Tex, Objects Room and MultiShapeNet, and show promising improvements on the challenging real-world Waymo Open dataset.

+ JAX/FLAX source code:
  https://github.com/google-research/google-research/tree/master/invariant_slot_attention
+ Model checkpoints:
  https://huggingface.co/ondrejbiza/isa
+ Interactive demo:
  https://huggingface.co/spaces/ondrejbiza/isa

An earlier version appeared at the NeurIPS'22 NeurReps workshop as "Spatial Symmetry in Slot Attention". *Work done during an internship at Google Research. †Equal contribution. [1]Northeastern University, Boston, MA, USA. [2]Google Research. Correspondence to: Ondrej Biza <biza.o@northeastern.edu>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

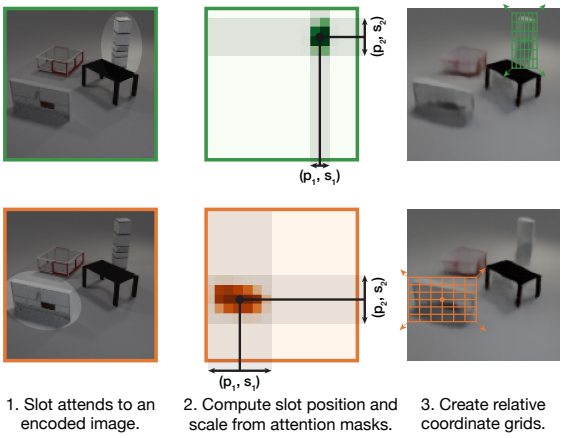


Figure 1. Left: Input image. Middle: Invariant Slot Attention masks. Right: Decoded images with relative coordinate grids (limited to 6×6 grids for ease of visualization).

1. Introduction

Humans have the intrinsic ability to form high level abstractions of objects in visual scenes regardless of their appearance (Barlow, 2009). In the neuroscience literature, it is hypothesized that humans place canonical 'reference frames' onto objects (Bottini & Doeller, 2020). Such reference frames entail a coordinate system around each object that may allow for forming robust object representations in a viewpoint-invariant manner (Hinton, 1981), and thus for making future predictions by manipulating those reference frames. In a scene with several objects and parts, Hawkins et al. (2019) posit that human grid cells represent a hierarchy of reference frames (e.g. the frame of a person's hand, the frame of the mug it is holding, the frame of the mug's handle) and hypothesize that there are special grid cells that model relative transformations between different levels in the scene.

In machine learning, it is also understood that such *invariant* object representations could be similarly beneficial. In order to learn per-object invariances, however, one needs robust models that can decompose visual scenes into objects. This topic, known as object-centric representation learning, has proven to be challenging in the absence of direct supervision (Greff et al., 2017). Yet, in the past few years, powered by more complex datasets, scalable architectures, and im-

proved compute infrastructure, substantial progress has been made in discovering object representations from raw perceptual data. This includes the use of (inverted) dot-product attention (Locatello et al., 2020), powerful decoders (Singh et al., 2021; 2022), optical flow (Kipf et al., 2022; Xie et al., 2022), depth maps (Elsayed et al., 2022), pre-trained features (Seitzer et al., 2023), etc. These advances have made an investigation into object-level invariances and their benefits for object representation learning more tractable.

Building upon these advances, and inspired by prior work such as AIR (Eslami et al., 2016) and Spatial Transformers (Jaderberg et al., 2015), invariance to object pose can be established by assigning a reference frame to each object. Any percept related to the object (such as spatial features in a feature map) can then be processed in a coordinate system relative to the object's reference frame, and any prediction about the object (such as a reconstruction of its appearance) retains the pose invariance property. Figure 1 shows an example of such a system in the setting of self-supervised object discovery. Figure 2 shows examples of learned reference frames.

Reasoning in relative reference frames is an example of a spatial symmetry; machine learning models can leverage these symmetries to improve sample-efficiency, generalization and prediction consistency (Bronstein et al., 2021). Much work has been done in leveraging *exact* spatial symmetries, for example in protein dynamics modelling (Han et al., 2022) or in constrained robotics environments (Wang et al., 2020a). Yet, these symmetries have been explored only to a limited extent in scene understanding, where an object's appearance might change due to lighting or occlusions, and the effect of 3D rotation on object appearance cannot be easily formalized (Park et al., 2022).

We focus our paper on the topic of unsupervised object discovery with slot-based models (Greff et al., 2019; 2020). Slot-based models compress a scene into a discrete number of latent variables–"slots". Slots can learn to represent individual objects in the scene without additional supervision. In these models slots are assumed to be symmetric under permutation, up to the choice of initialization, which can facilitate object discovery. Yet, other spatial symmetries have been explored only to a limited extent. While earlier methods (Jaderberg et al., 2015; Eslami et al., 2016) use explicit per-object poses in the decoder, their monolithic encoders still operate in terms of absolute scene coordinates and thus do not fully account for spatial symmetries.

In our work, we draw a connection between spatial symmetries and slot-based models via the use of *pose-relative position encoding*. We base our work on the Slot Attention (Locatello et al., 2020) architecture, which is a widely used slot-based model for object discovery. Slot Attention uses position encoding to map from the grid-structured image representation to its set of latent slots via an attention mechanism, and similarly uses position encoding to map back into the image space in the decoder. Our core insight is that we can achieve equivariance to spatial symmetry transformations (i.e., pose transformations) at a per-object level by transforming the respective position encodings relative to the each object's inferred pose. This presents an elegant yet computationally efficient solution for integrating spatial symmetries in object-centric architectures.

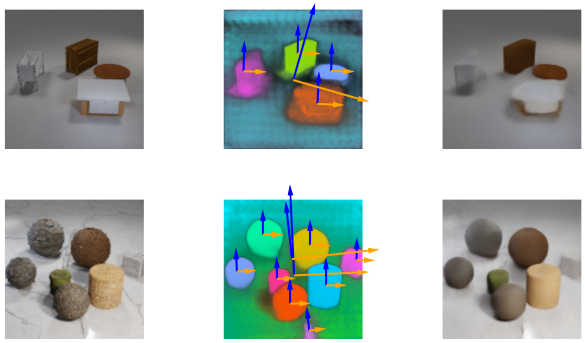


Figure 2. Examples of reference frames learned by Invariant Slot Attention. From left to right: input image, soft segmentation masks with a pair of arrows for each slot, decoded image.

In our experiments, we demonstrate that incorporating translation and scale symmetries into Slot Attention results in improved data efficiency, out-of-domain generalization, and often overall improvements in object discovery performance with little computational overhead. While our framework also allows for incorporation of approximate 2D rotation symmetry, we find that this frequently only leads to minor (or no) benefits, likely due to the nature of our datasets (images of 3D scenes). We evaluate our model, Invariant Slot Attention (ISA), on standard synthetic object discovery benchmarks, a challenging textured dataset, and in object discovery in a real-world autonomous driving dataset.

2. Related Work

Unsupervised Object Discovery. Object discovery using model inductive biases has been an active and growing area of research. Early works use an iterative inference setting: a model alternates between reconstructing an image and inferring the membership of pixels into groups forming object representation vectors (or *slots*) (Greff et al., 2015; 2016; 2017; van Steenkiste et al., 2018; Greff et al., 2019). These works do not implement spatial symmetries outside of the use of convolutional networks. Burgess et al. (2019); Engelcke et al. (2020) also fall in this category, wherein they auto-regressively predict per-slot attention masks, which are used to compute unstructured slot latent vectors.

In a separate line of work, Eslami et al. (2016); Kosiorek et al. (2018); Jiang et al. (2020) use a monolithic encoder to predict the position and scale of each object (z_{where}). A Spatial Transformer (ST) (Jaderberg et al., 2015) decodes object appearance invariant to z_{where}. To enable decoding of complex images, Monnier et al. (2021); Smirnov et al. (2021); Sauvalle & de La Fortelle (2022) combine ST decoders with background prediction models. During encoding, partial translation equivariance can be achieved by predicting object positions and scales relative to anchors in a grid (Crawford & Pineau, 2019; Lin et al., 2020; Jiang & Ahn, 2020).

Our work builds on the Slot Attention model (Section 3), which groups pixels into slots by iteratively computing cross-attention between inputs and slot latent vectors (Locatello et al., 2020). We take inspiration from capsules (Hinton et al., 2011; Sabour et al., 2017; Hinton et al., 2018; Kosiorek et al., 2019), which model the hierarchy of relative transformations between object parts forming objects (and objects forming a scene). Hinton (2021) points out that slots in Slot Attention can be thought of as *universal* capsules that learn to detect any object in any position and orientation. Unlike capsules, Slot Attention does not natively model the part-whole hierarchy. Parts of an object (spatial features from a CNN encoder) are anchored in the reference frame of the camera, instead of being represented relative to the position, orientation and scale of the object. Hence, the original Slot Attention model is not predisposed to spatially invariant reasoning about objects and their parts. Since Slot Attention has been the basis of several recent follow-up works (Emami et al., 2021; Singh et al., 2021; Sauvalle & de La Fortelle, 2022), video modeling (Kipf et al., 2022; Elsayed et al., 2022; Wu et al., 2022), multi-view scene understanding (Sajjadi et al., 2021), and panoptic video segmentation (Zhou et al., 2022), to name a few, making it spatially invariant has the potential for broader impact.

Relative Position Encoding. Follow-up work to the original Transformer model (Vaswani et al., 2017), replaces absolute position encodings with an encoding of the relative distance between a pair of words (Shaw et al., 2018). Since word or character distances are discrete, most works associate a learnable vector with each possible relative distance between words. Similarly, relative position encoding has been explored in the context of Vision Tranformers (Dosovitskiy et al., 2021; Cordonnier et al., 2020): both in terms of discrete (Bello et al., 2019; Parmar et al., 2019; Wang et al., 2020b; Srinivas et al., 2021) as well as continuous encodings between patches (Zhao et al., 2020). We refer to Dufter et al. (2022) for a recent review. These methods consider relative position encoding in an encoder-only setup, which is orthogonal to our pose-relative position encoding mechanism that operates on cross-attention and related mappings between visual tokens and object slots.

Ideas for relative position encoding in vision have further been explored in the context of the Detection Transformer (DETR) (Carion et al., 2020). Zhu et al. (2021) take a step towards disentangling object query positions in DETR for supervised detection tasks: attention is only computed over the local neighborhood of input tokens close to a reference point. Similarly, Meng et al. (2021); Gao et al. (2021) modulate DETR's decoder attention by predicting the supposed center (and possibly scale) of each detected object from object queries in each decoder layer. Finally, Wang et al. (2022); Liu et al. (2022); Zhang et al. (2022) show that separation of object positions and scales from object appearance leads to significant improvements. Unlike our method, their approach still operates on absolute object coordinates either as explicit input or output of learnable modules, i.e. it does not fully respect spatial symmetries. In ISA, we extract positions and scale from cross-attention masks, which is expected to ensure spatial equivariance in our architecture.

3. Background: Slot Attention

Our method for object discovery with slot-centric reference frames is based on the Slot Attention (Locatello et al., 2020) architecture, which presents a simple yet effective attention-based method for decomposing scenes into objects and for learning object representations from un-annotated image data. Slot Attention consists of an encoder, an attention mechanism, and a decoder.

The encoder (E_θ), e.g. a convolutional neural network (CNN), maps images into an intermediate representation of $N = H' \times W'$ tokens of dimension D_t. The attention mechanism in Slot Attention (SA) simulates a competition of latent slots ($\text{slots} \in \mathbb{R}^{K \times D_s}$) over input tokens ($\text{inputs} \in \mathbb{R}^{N \times D_t}$). Intuitively, slots win tokens if they are able to reconstruct the corresponding parts of the input image – these parts usually correspond to entire objects or coherent object parts.

The model computes a form of cross-attention (Luong et al., 2015; Vaswani et al., 2017) between slots, transformed into $\text{queries} = \mathcal{Q}(\text{slots})$, and inputs, transformed into keys and values:

$$\text{keys} = f\left(\mathcal{K}(\text{inputs}) + g(\text{abs_grid})\right) \tag{1}$$

$$\text{values} = f\left(\mathcal{V}(\text{inputs}) + g(\text{abs_grid})\right) \tag{2}$$

Here abs_grid encodes the absolute positions of input tokens in an image, $\mathcal{K}, \mathcal{V}, \mathcal{Q}, g$ are linear functions and f is an MLP. The position encodings are horizontal and vertical coordinate grids scaled to [-1, 1]. The dot product between queries and keys yields the attention weights, which are then normalized over queries to facilitate competition between slots. The output of cross-attention is computed in several iterations; in each iteration, a recurrent neural network updates the slot representations (App. Algorithm 1).

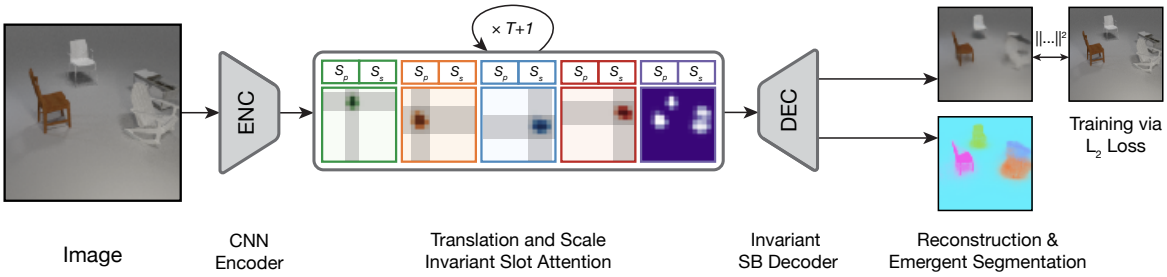


Figure 3. Invariant Slot Attention (ISA-TS) with an input image from MultiShapeNet (left), visualization of five intermediate attention masks (top; the purple masks corresponds to a background slot), and the decoded image and segmentation mask (right).

The decoder (D_ϕ) reconstructs the input image based on the slot latent vectors. We use the Spatial Broadcast (SB) decoder (Watters et al., 2019), which first repeats each slot $H' * W'$ times to create the initial spatial feature map, adds position encoding as in the encoder, and then maps it to (R,G,B,α) values for each pixel. The α mask, which represents the predicted segmentation mask, is normalized over slots. The final image is created by taking the weighted sum over K slots for each pixel.

4. Invariant Slot Attention

In Invariant Slot Attention (ISA), Figure 3, we introduce mechanisms for invariant encoding and decoding to the Slot Attention architecture, with the goal to disentangle slot appearance from slot position, orientation and scale.

In our formulation, slot latent vectors are *invariant* to object pose, while slot position, orientation and scale are *equivariant* to pose transformations. Note that slot scale refers to the spatial extent in pixel space and therefore accounts for both the object's size as well as distance to the camera. Our mechanism can be implemented in two easy steps:

Step 1: Either initialize slots with explicit reference frames in addition to their latent vectors that encode appearance, or, if available, compute slot reference frames from per-slot attention masks.

Step 2: Create relative position encodings, $\mathrm{rel_grid}$, in each iteration of Slot Attention and in the SB decoder using reference frames (slot poses) from step 1.

The following subsections explain the above steps in detail along with how $\mathrm{rel_grid}$ can be used instead of $\mathrm{abs_grid}$ in Slot Attention iterations and the SB decoder.

4.1. Translation and Scaling Invariant Slot Attention

To achieve invariance to translation and scaling, we instantiate 2D slot positions (S_p) and scales (S_s), which can be initially sampled randomly or learned (Appendix C). (S_p, S_s) define a reference frame for each slot. We use it to translate and scale the input position encoding for each slot $k \in \{1, ..., K\}$ separately:

$$\mathrm{rel_grid}^k = (\mathrm{abs_grid} - S_p^k) \;/\; S_s^k \tag{3}$$

We translate and scale by the *inverse* of slot positions and scales. Intuitively, we are undoing the estimated object pose so that we can process it in a canonical reference frame. To compute attention masks, we create per-slot keys and values with $\mathrm{rel_grid}$ instead of $\mathrm{abs_grid}$. That is, $\forall\, 1 \leq k \leq K$:

$$\mathrm{keys}^k = f\left(\mathcal{K}(\mathrm{inputs}) + g(\mathrm{rel_grid}^k)\right) \tag{4}$$

$$\mathrm{values}^k = f\left(\mathcal{V}(\mathrm{inputs}) + g(\mathrm{rel_grid}^k)\right) \tag{5}$$

Note that, compared to Equations (1) and (2), we now have $N * K$ keys and values, and K queries. The original Slot Attention has only N keys and values. We find the increase in computational cost negligible in the standard regime of detecting around 10 objects in an image.

To infer S_p and S_s, we use the obtained per-slot keys and values to compute attention weights, attn, from which we in turn extract new slot positions and scales:

$$S_p = \frac{\sum_{n=1}^{N} \mathrm{attn}_n * \mathrm{abs_grid}_n}{\sum_{n=1}^{N} \mathrm{attn}_n} \tag{6}$$

$$S_s = \sqrt{\frac{\sum_{n=1}^{N} (\mathrm{attn}_n + \epsilon) * (\mathrm{abs_grid}_n - S_p)^2}{\sum_{n=1}^{N} (\mathrm{attn}_n + \epsilon)}} \tag{7}$$

The intuition behind this process is that attention weights focus on the relevant object in the image, and we can use the center of mass of the attention mask to infer the objects' position, and its spread to infer its scale.

The same process is repeated in each Slot Attention iteration. Afterward, we run one additional iteration to compute slot statistics without updating the slot latent vectors.

In the Invariant Spatial Broadcast decoder, we create pose-relative position encoding using the same process as in the encoder. We use the final per-slot reference frames (S_p, S_s) computed in Slot Attention and we compute $\mathrm{rel_grid}$ as in eqn. (3). $\mathrm{rel_grid}$ is then projected to D_s, the slot dimension,

using a linear function h and added to spatially-broadcasted slots $\mathrm{SB} \in \mathbb{R}^{K \times H' \times W' \times D_s}$:

$$(R, G, B, \alpha) = D_\phi(\mathrm{SB} + h(\mathrm{rel_grid})) \tag{8}$$

where $\mathrm{rel_grid} \in \mathbb{R}^{K \times H' \times W' \times 2}$. All changes described in this section can be implemented in a few lines of code (Appendix Algorithm 1). Note that we backpropagate through both the computed slot positions and scales, and the relative coordinate grids.

Although we demonstrate our idea with image-based Slot Attention and 2D spatial symmetries, the same principle could apply in videos, by processing each frame with the proposed method, (Kipf et al., 2022; Elsayed et al., 2022), or in 3D with slot-based Neural Radiance Fields (Mildenhall et al., 2020) or Object Scene Representation Transformers (Sajjadi et al., 2022), by equipping slots with positions extracted from multiple views, (Stelzner et al., 2021; Sajjadi et al., 2021; Yu et al., 2022).

4.2. Invariance to Rotations

The orientation of an object is much more ambiguous than its position and scale. To study this symmetry, we use a simple principal component heuristic (Yi & Marshall, 2000) to estimate slot poses. We estimate the orientation of an object using the axis with the highest variation (the first principal component) of the attention mask. We then construct a rotation matrix S_r with the principal and an orthogonal axis, with rotations limited to $\pi/4$. Further details are provided in Appendix C.2. We then compute the relative position encoding by inverse translation, rotation and scaling:

$$\mathrm{rel_grid}^k = (S_r^k)^{-1}(\mathrm{abs_grid} - S_p^k) \;/\; S_s^k \,. \tag{9}$$

5. Experiments

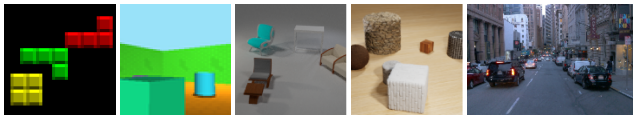


Figure 4. **Datasets** (left to right): Tetrominoes, Objects Room, MultiShapeNet, CLEVRTex, and Waymo Open.

We evaluate Invariant Slot Attention (ISA) on six unsupervised object discovery datasets illustrated in fig. 4, which include both synthetic and real-world datasets. We measure the Adjusted Rand Index for foreground object segments (FG-ARI) and the mean squared error of decoded images (MSE) summed over all pixels and averaged over images, please see Appendix E for details.

5.1. Proof of concept: Tetrominoes

Starting with a proof of concept, we study translation invariance on the Tetrominoes dataset (Kabra et al., 2019) with simple geometric shapes over a black background. We compare Slot Attention (SA) against Translation Invariant Slot Attention (ISA-T). We conduct experiments in two settings: (1) where only a few (64 to 1024) training samples are available to the model, and (2) where training samples are biased to only have objects appear in the left side of the image and test samples may have objects in all positions. The former tests sample efficiency and the latter tests out-of-distribution generalization at test time.

Results are shown in Figure 5. It can be seen that using pose-relative position encoding substantially improves sample efficiency. In experiment (1), ISA-T requires around $2\times$ to $4\times$ fewer samples to achieve performance equivalent to SA. Note that we used translational data augmentation in this experiment and picked optimal hyperparameters that favoured the SA baseline. This demonstrates that sample efficiency gains achieved through our method are not superseded by those from standard data augmentation. We argue that this may be due to how ISA-T models translation of *individual objects*, which cannot be captured by simply translating the entire image. In experiment (2), ISA-T again outperforms the SA baseline, demonstrating stronger out-of-distribution generalization. No data augmentation was used in this experiment in order to carefully control the distribution shift.

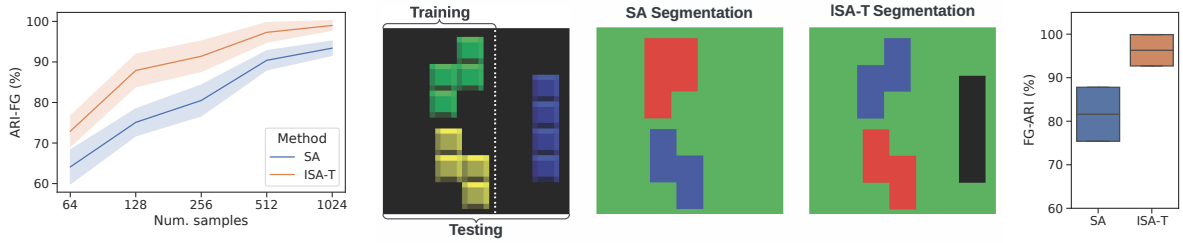
5.2. Evaluating translation and scaling invariance

We evaluate translation invariant SA, ISA-T, and translation and scaling invariant SA, ISA-TS, on four synthetic multi-object datasets for unsupervised scene decomposition.

In Objects Room (Burgess et al., 2019), as shown in Figure 6, both invariant models achieve higher ARI scores across all validation sets when compared to SA. We treat floors, walls and ceilings as objects in these experiments and hence report ARI instead of FG-ARI. We also note a substantial reduction in uncertainity across seeds for ISA.

In MultiShapeNet (Stelzner et al., 2021), we find both ISA-T and ISA-TS to greatly improve performance compared to SA (Figure 6-bottom, *All Objects*). Upon closer qualitative examination, we found that ISA-TS was more prone to over-segmentation; such as segmenting the head of a chair and its legs as separate objects. We controlled for this behavior by creating a new training and evaluation split consisting of all MultiShapeNet images with exactly four objects (Figure 6-bottom, *Four Objects*). Here we set the number of slots to five making over-segmentation less likely. In this controlled setting, ISA-TS outperforms ISA-T and SA.

We also evaluate on the CLEVR dataset (Johnson et al., 2017; Kabra et al., 2019), but both ISA variants and the baseline Slot Attention model achieve near-perfect segmentation and decoding, see appendix table 5.



(a) Sample efficiency.

(b) Test-time OOD generalization. From left-to-right: The setup, non-invariant baseline output, ISA-T (our) output, and FG-ARI for both methods (10 seeds per method).

Figure 5. Comparing between Slot Attention (SA) and Translation Invariant Slot Attention (ISA-T) on Tetrominoes. ISA-T (a) reaches the same performance as SA with 2 to 4 times fewer samples, and (b) is able to generalize to OOD object positions.

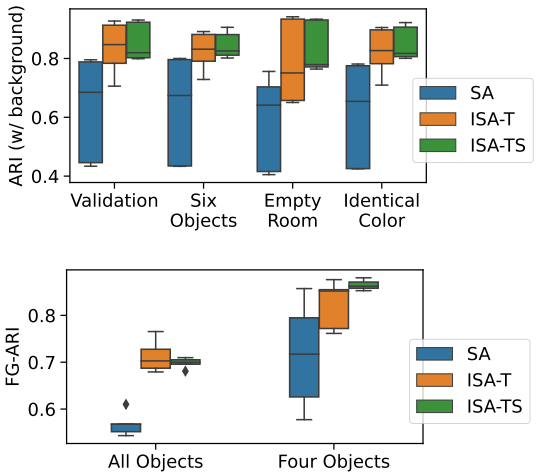


Figure 6. **Top: Objects Room** ARI with background segments included: Higher is better. We use 10 seeds per experiment. For exact numbers see Table 6. **Bottom: MultiShapeNet** FG-ARI results using 5 random seeds. For exact numbers see Table 7.

Table 1. CLEVRTex FG-ARI(%) results on the test set, CAMO set (objects and backgrounds blend together) and OOD set (novel textures). Prior results taken from (Karazija et al., 2021) use 3 random seeds, we use 10 random seeds. FG-ARI is reported in %. For MSE please see the Table 8. (CNN) refers to models using a 4-layer CNN backbone, while (ResNet) models use a ResNet-34.

Method	Main	CAMO	OOD
SPACE	17.5 ±4.1	10.6 ±2.1	12.7 ±3.4
DTI	79.9 ±1.4	72.9 ±1.9	73.7 ±1.0
AST-Seg-B3-CT	94.8 ±0.5	87.3 ±3.8	83.1 ±0.8
SA (CNN)	54.5 ±1.6	53.0 ±1.6	54.2 ±2.6
ISA-T (CNN)	66.8 ±5.7	65.0 ±4.9	65.1 ±4.8
ISA-TS (CNN)	78.8 ±3.9	72.9 ±3.5	73.2 ±3.1
SA (ResNet)	91.3 ±2.7	84.9 ±2.9	81.4 ±1.4
ISA-T (ResNet)	87.4 ±6.6	79.0 ±5.9	78.6 ±4.9
ISA-TS (ResNet)	92.9 ±0.4	86.2 ±0.8	84.4 ±0.8

CLEVRTex (Karazija et al., 2021) is perhaps the most complex synthetic dataset in the relevant literature, where plain Slot Attention was previously reported to fail; we report the results of 12 different Invariant Slot Attention variants and ablations in the appendix in Table 8 and summarize key findings in Table 1. Following the trend on previous datasets, ISA-T and ISA-TS each lead to 10%+ absolute improvement in FG-ARI (Table 1, CNN).

To our surprise, plain Slot Attention can be made a competitive baseline with three simple changes: a stronger backbone (ResNet-34 *without pre-training*) (He et al., 2016), a larger feature map resolution (16^2) with the same resolution in the encoder output and decoder input (i.e. the spatial broadcast resolution in the decoder is set to 16^2 as well), and learnable initial slots (please see Section C.1).

Even in this improved setting (Table 1, ResNet), we measure between 1% and 3% FG-ARI improvement between SA and ISA-TS depending on the validation set. ISA-TS outperforms the state-of-the-art (AST-Seg-B3-CT) on the out-of-distribution (OOD) test set, where *novel textures and shapes* are used (Sauvalle & de La Fortelle, 2022). Our results suggest that ImageNet and background model pre-training, which are used in AST-Seg-B3-CT, are not necessary to solve CLEVRTex. ISA-T, however, performs poorly in this setting. We note how its FG-ARI is being pulled down by a poorly performing seed, although that alone did not explain the overall drop.

In summary, we find that both translation and scaling symmetries can lead to large benefits in Slot Attention's ability to discover and segment objects. We hypothesize that this is in part because of the added inductive bias that guides the unsupervised discovery of objects, and partly due to weight sharing across positions and scales when decoding objects.

Table 2. **Rotation invariance**: Comparing ISA-TS against ISA-TSR in various benchmarks. Objects Room results are ARIs whereas all others are FG-ARIs. Remaining benchmarks evaluations are in the appendix Table 4.

	(FG-)ARI ↑	
Dataset	ISA-TS	ISA-TSR
Objects Room (w/ bg) Val.	85.5 ±6.6	84.3 ±4.6
CLEVR	98.9 ±0.2	98.0 ±0.9
MultiShapeNet		
- All Data	69.8 ±1.1	77.7 ±5.5
- Four Objects	86.5 ±1.1	80.7 ±6.4
CLEVRTex (CNN)	78.8 ±3.9	79.6 ±5.5
CLEVRTex (ResNet)	92.9 ±0.4	93.3 ±0.7

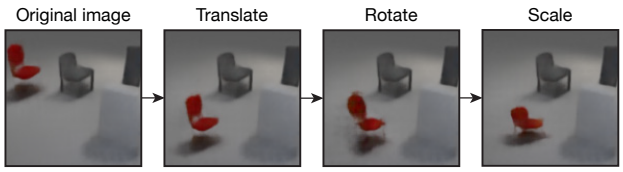


Figure 7. Controlling slots with ISA-TSR on MultiShapeNet.

5.3. Invariance to rotations

We find rotational invariance ISA-TSR leads to mixed results across datasets (see Table 2. On one hand, ISA-TSR further pushes the state-of-the-art on CLEVRTex, increasing FG-ARI by around 0.5%. On the other hand, we find that it decreases performance on Objects Room and CLEVR. Despite that, it does allow us to control the rotation of the decoded objects, which appears reasonably well captured (Figure 7). We hypothesize that our heuristic rotation estimator, that uses principal axes, is too susceptible to symmetric objects which cause ambiguity in the detected rotation. As a mitigation, future work could enable the model to predict corrections to the rotational heuristic.

5.4. Real-world evaluation: Waymo Open

To investigate the utility of object pose invariance as an inductive bias on real-world visual data, we evaluate ISA for unsupervised instance segmentation on Waymo Open v1.4 (Sun et al., 2020; Mei et al., 2022), a dataset of videos collected from cameras on Waymo cars. The dataset has been previously used in unsupervised learning (e.g. Elsayed et al. (2022)); however, we are the first to report positive single-frame RGB-only results for object discovery, i.e. without the use of optical flow, depth features, or temporal information. Waymo Open is highly challenging since both the foreground objects and backgrounds are highly varied and move as the car moves. In comparison, prior methods tested

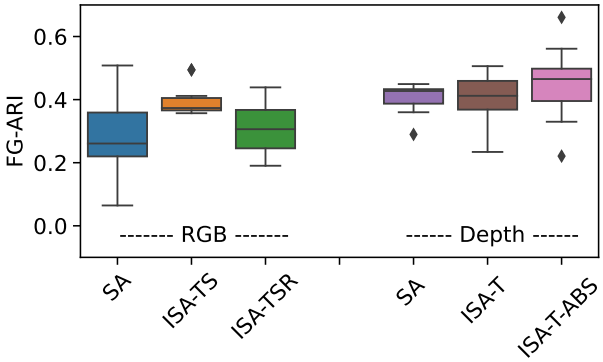


Figure 8. **Waymo Open** (ResNet) across 10 seeds.

on real-world datasets exploit static backgrounds (Kosiorek et al., 2018; Sauvalle & de La Fortelle, 2022).

We find ISA-TS to increase FG-ARI by 12% compared to SA (Figure 8). Adding rotational symmetry decreases FG-ARI by about 5% compared to ISA-TS; we hypothesize that rotations of complex objects and backgrounds are too ambiguous to detect by our heuristic. Slot positions and scales in ISA-TS allow controlling decoded outputs (Figure 9).

Qualitatively, we find the computed FG-ARI and the perceived quality of predicted segmentation masks to not always be in agreement – all Slot Attention models focus on landmarks, such as large buildings, trees, traffic signs and lanes of a road (Figure 10), whereas Waymo Open, naturally, only scores the detection of other cars and pedestrians.

Following Elsayed et al. (2022), we test if predicting depth instead of RGB improves emergent object segmentation. We find depth targets (the model still receives only RGB images as input) indeed increase the FG-ARI of Slot Attention, as cars and pedestrians do not blend in the clutter of complex backgrounds. But, depth images do not exhibit the same symmetries as RGB images. For example, as a car moves further away from the camera, its depth value changes whereas its color does not. Indeed, we find that ISA-T reaches the same performance as the baseline, unless we explicitly *break the symmetry* in the model.

5.5. Symmetry breaking

Invariant Slot Attention ensures that the slot latent vectors are invariant to the absolute positions of the features they attend to[1], as long as the individual feature vectors do not already have position information encoded in them. However, we find that the visual backbone can leak absolute position information, which they can detect, for example by looking at the zero-padded edges added in their convolutions. This

[1]This holds true only with random initial slot statistics, since learnable initial slot statistics already break the symmetry.

Figure 9. Controlling slots with ISA-TS (ResNet) on Waymo Open RGB.

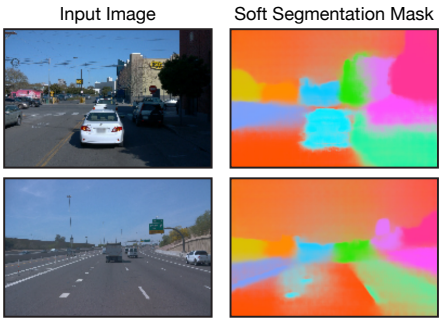


Figure 10. Waymo Open RGB predicted soft segmentation masks of ISA-TS (ResNet). Soft segmentation masks blend pixel colors based on normalized per-slot α masks.

effect is limited to image boundary regions in the default 4-layer CNN backbone used in Slot Attention (Locatello et al., 2020), which we adopted for ISA, but it is clearly pronounced when using a more expressive ResNet-34 encoder.

Instead of Invariant Slot Attention having to "cheat" to break equivariance, we can explicitly append slot positions/rotations/scales to the slot latent vectors right before they are passed to the gated recurrent unit (GRU) (Cho et al., 2014) in each Slot Attention iteration. The intuition for this design decision is that we allow the GRU to model the movement of per-slot attention masks over the iterations. We test this version of the model on CLEVRTex and find it to significantly improve ISA when using the default, shallow CNN backbone. We denote this model version using "-Append". Specifically, ISA-TSR-Append (85.4% FG-ARI) outperforms ISA-TSR (79.6% FG-ARI) and successfully segments various CLEVRTex scenes (Figure 14). When using a ResNet-34 backbone however, there is no significant difference between the two model variants, likely since the ResNet itself already leaks absolute position information into the model.

On Waymo Open, we further investigate breaking the symmetry in both encoder as well as the decoder by using both pose-relative as well as absolute position encoding. This leads to around 5% FG-ARI improvement over both SA and T-SA (ResNet), Table 10.

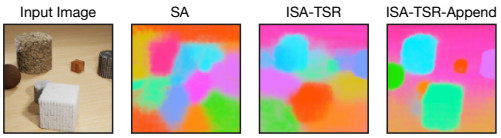


Figure 11. CLEVRTex predicted segmentation masks of three CNN Slot Attention variants. The absolute difference in FG-ARI between the leftmost and rightmost model is 30%.

Table 3. **Model ablations**: We report FG-ARI (in %) across Tetrominoes, CLEVRTex and Waymo Open for three different ISA model variants and two ablations: "dec. only" uses pose-relative position encoding only in the decoder and "stop grad." does not backpropagate gradients through the estimated pose parameters.

Model	Dataset	FG-ARI
ISA-T	Tetrominoes	96.3 ± 3.6
- dec. only		92.3 ± 2.4
ISA-TSR (ResNet)	CLEVRTex	93.3 ± 0.7
- dec. only		93.4 ± 1.0
- stop grad.		73.4 ± 10.5
ISA-TS (ResNet)	Waymo Open	39.8 ± 5.3
- dec. only		32.4 ± 7.1

5.6. Ablations

We conduct two ablation studies to examine whether spatial symmetries are necessary in the Slot Attention encoder module ("dec. only") and whether gradients need to flow through the computation of reference frame parameters (S_p, S_s, S_r) ("stop grad."). Ablation results are summarized in Table 3.

"Dec. only" refers to using $\mathrm{abs_grid}$ instead of $\mathrm{rel_grid}$ in the attention mechanism but still using $\mathrm{rel_grid}$ in the SB decoder. That is, the encoder is now not explicitly incorporating spatial symmetries whereas the decoder is still doing so. The slot reference frames required for doing so in the decoder are computed using the attention maps, attn, from the last iteration of slot attention. We find that this ablation has a significant negative impact on segmentation performance on Tetrominoes, in which objects are symmetric under translation within image boundaries (excl. occlusion). On CLEVRTex, we find its FG-ARI to be comparable to the

unablated version (ISA-TSR). We find, however, that this ablation can have a negative effect on reconstruction quality: whereas ISA-TSR (ResNet) achieves a reconstruction MSE of 185.9 ± 6.4 on CLEVTex (Main), the "Dec. only" ablation results in an increased error of 200.4 ± 6.6. Since "dec. only" does not decompose appearance and pose during encoding, we hypothesize that this entangled slot representation is detrimental to generalizing across object positions and textures. Similar to Tetrominoes, we find a significant reduction in FG-ARI for this ablation on Waymo Open.

The "stop grad." results in Table 3, show that is it crucial to allow the gradients to flow through the pose estimation mechanism, which suggest that the backwards pathway alters cross-attention weights in Slot Attention to obtain more useful reference frames.

6. Conclusion

We have introduced Invariant Slot Attention (ISA), a method for unsupervised scene decomposition and object discovery using slot-centric reference frames, which enables learning object representations that are invariant to geometric transformation including translation, scale and rotation. Our method incorporates spatial symmetries on a per-object basis with little computational overhead, which is achieved via simple changes to the positional encoding used both in the attention mechanism and the decoder of Slot Attention.

Even though our investigation of data efficiency and generalization benefits focused on unsupervised scene understanding, *per-object* reference frames are likely worth broader consideration in future work, as data augmentation alone is unable to capture these symmetries effectively. One limitation of such an approach is that *exact* translation, scale and rotation symmetries of objects are rarely present in images of 3D scenes, and it is unclear how backgrounds should be handled. Despite this, we see evidence that per-object symmetries, especially in combination with ways for the model to *break the symmetry* by additionally using absolute, global reference frames, frequently improve model performance.

Acknowledgements

We would like to thank Mike Mozer for helpful discussions and feedback on the manuscript, and Klaus Greff for assistance with Kubric-generated datasets (Greff et al., 2022). We would also like to thank the anonymous reviewers of the NeurIPS'22 NeurReps workshop and ICML'23.

References

Barlow, H. Grandmother cells, symmetry, and invariance: How the term arose and what the facts suggest. In Gazzaniga, M. S. (ed.), *The Cognitive Neurosciences*, pp. 309–320. The MIT Press, fourth edition, 2009.

Bello, I., Zoph, B., Le, Q., Vaswani, A., and Shlens, J. Attention Augmented Convolutional Networks. In *ICCV*, 2019.

Bottini, R. and Doeller, C. F. Knowledge across reference frames: Cognitive maps and image spaces. *Trends in Cognitive Sciences*, 24(8):606–619, 2020.

Bronstein, M. M., Bruna, J., Cohen, T., and Velickovic, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021.

Burgess, C. P., Matthey, L., Watters, N., Kabra, R., Higgins, I., Botvinick, M. M., and Lerchner, A. Monet: Unsupervised scene decomposition and representation. *CoRR*, abs/1901.11390, 2019.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-End Object Detection with Transformers. In *ECCV*, 2020.

Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. On the relationship between self-attention and convolutional layers. In *ICLR*, 2020.

Crawford, E. and Pineau, J. Spatially invariant unsupervised object detection with convolutional neural networks. In *AAAI*, 2019.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.

Dufter, P., Schmitt, M., and Schütze, H. Position Information in Transformers: An Overview. *Comput. Linguistics*, 48(3):733–763, 2022.

Elsayed, G. F., Mahendran, A., van Steenkiste, S., Greff, K., Mozer, M. C., and Kipf, T. SAVi++: Towards end-to-end object-centric learning from real-world videos. *NeurIPS*, 2022.

Emami, P., He, P., Ranka, S., and Rangarajan, A. Efficient iterative amortized inference for learning symmetric and disentangled multi-object representations. In *ICML*, 2021.

Engelcke, M., Kosiorek, A. R., Jones, O. P., and Posner, I. GENESIS: generative scene inference and sampling with object-centric latent representations. In *ICLR*, 2020.

Eslami, S. M. A., Heess, N., Weber, T., Tassa, Y., Szepesvari, D., Kavukcuoglu, K., and Hinton, G. E. Attend, infer, repeat: Fast scene understanding with generative models. In *NeurIPS*, 2016.

Gao, P., Zheng, M., Wang, X., Dai, J., and Li, H. Fast Convergence of DETR with Spatially Modulated Co-Attention. In *ICCV*, 2021.

Greff, K., Srivastava, R. K., and Schmidhuber, J. Binding via reconstruction clustering. *CoRR*, abs/1511.06418, 2015.

Greff, K., Rasmus, A., Berglund, M., Hao, T. H., Valpola, H., and Schmidhuber, J. Tagger: Deep unsupervised perceptual grouping. In Lee, D. D., Sugiyama, M., von Luxburg, U., Guyon, I., and Garnett, R. (eds.), *NeurIPS*, 2016.

Greff, K., van Steenkiste, S., and Schmidhuber, J. Neural expectation maximization. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *NeurIPS*, 2017.

Greff, K., Kaufman, R. L., Kabra, R., Watters, N., Burgess, C., Zoran, D., Matthey, L., Botvinick, M. M., and Lerchner, A. Multi-object representation learning with iterative variational inference. In *ICML*, 2019.

Greff, K., Van Steenkiste, S., and Schmidhuber, J. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., Fleet, D. J., Gnanapragasam, D., Golemo, F., Herrmann, C., Kipf, T., Kundu, A., Lagun, D., Laradji, I. H., Liu, H. D., Meyer, H., Miao, Y., Nowrouzezahrai, D., Öztireli, A. C., Pot, E., Radwan, N., Rebain, D., Sabour, S., Sajjadi, M. S. M., Sela, M., Sitzmann, V., Stone, A., Sun, D., Vora, S., Wang, Z., Wu, T., Yi, K. M., Zhong, F., and Tagliasacchi, A. Kubric: A scalable dataset generator. In *CVPR*, 2022.

Han, J., Rong, Y., Xu, T., Sun, F., and Huang, W. Equivariant graph hierarchy-based neural networks. *CoRR*, abs/2202.10643, 2022.

Hawkins, J., Lewis, M., Klukas, M., Purdy, S., and Ahmad, S. A framework for intelligence and cortical function based on grid cells in the neocortex. *Frontiers in Neural Circuits*, 12, 2019.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hinton, G. E. A parallel computation that assigns canonical object-based frames of reference. In Hayes, P. J. (ed.), *Proceedings of the 7th International Joint Conference on Artificial Intelligence, IJCAI '81*, 1981.

Hinton, G. E. How to represent part-whole hierarchies in a neural network. *CoRR*, abs/2102.12627, 2021.

Hinton, G. E., Krizhevsky, A., and Wang, S. D. Transforming auto-encoders. In Honkela, T., Duch, W., Girolami, M. A., and Kaski, S. (eds.), *International Conference on Artificial Neural Networks*, 2011.

Hinton, G. E., Sabour, S., and Frosst, N. Matrix capsules with EM routing. In *ICLR*, 2018.

Hubert, L. and Arabie, P. Comparing partitions. *Journal of Classification*, 1985.

Ioffe, S. and Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. Spatial transformer networks. In *NeurIPS*, 2015.

Jiang, J. and Ahn, S. Generative neurosymbolic machines. In *NeurIPS*, 2020.

Jiang, J., Janghorbani, S., de Melo, G., and Ahn, S. SCALOR: generative world models with scalable object representations. In *ICLR*, 2020.

Johnson, J., Hariharan, B., van der Maaten, L., Fei-Fei, L., Zitnick, C. L., and Girshick, R. B. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

Kabra, R., Burgess, C., Matthey, L., Kaufman, R. L., Greff, K., Reynolds, M., and Lerchner, A. Multi-object datasets. https://github.com/deepmind/multi-object-datasets/, 2019.

Karazija, L., Laina, I., and Rupprecht, C. Clevrtex: A texture-rich benchmark for unsupervised multi-object segmentation. In *NeurIPS Track on Datasets and Benchmarks 1*, 2021.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y. (eds.), *ICLR*, 2015.

Kipf, T., Elsayed, G. F., Mahendran, A., Stone, A., Sabour, S., Heigold, G., Jonschkowski, R., Dosovitskiy, A., and Greff, K. Conditional object-centric learning from video. In *ICLR*, 2022.

Kosiorek, A. R., Kim, H., Teh, Y. W., and Posner, I. Sequential attend, infer, repeat: Generative modelling of moving objects. In *NeurIPS*, 2018.

Kosiorek, A. R., Sabour, S., Teh, Y. W., and Hinton, G. E. Stacked capsule autoencoders. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *NeurIPS*, pp. 15486–15496, 2019.

Lin, Z., Wu, Y., Peri, S. V., Sun, W., Singh, G., Deng, F., Jiang, J., and Ahn, S. SPACE: unsupervised object-oriented scene representation via spatial attention and decomposition. In *ICLR*, 2020.

Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., and Zhang, L. DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR. In *ICLR*, 2022.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. In *NeurIPS*, 2020.

Loshchilov, I. and Hutter, F. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017.

Luong, T., Pham, H., and Manning, C. D. Effective approaches to attention-based neural machine translation. In *EMNLP*, 2015.

Mei, J., Zhu, A. Z., Yan, X., Yan, H., Qiao, S., Chen, L., and Kretzschmar, H. Waymo open dataset: Panoramic video panoptic segmentation. In *ECCV*, 2022.

Meng, D., Chen, X., Fan, Z., Zeng, G., Li, H., Yuan, Y., Sun, L., and Wang, J. Conditional DETR for Fast Training Convergence. In *ICCV*, 2021.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., and Ng, R. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Monnier, T., Vincent, E., Ponce, J., and Aubry, M. Unsupervised layered image decomposition into object prototypes. In *ICCV*, 2021.

Nair, V. and Hinton, G. E. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

Park, J. Y., Biza, O., Zhao, L., van de Meent, J., and Walters, R. Learning symmetric embeddings for equivariant world models. In *ICML*, 2022.

Parmar, N., Ramachandran, P., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-Alone Self-Attention in Vision Models. In *NeurIPS*, 2019.

Ranftl, R., Bochkovskiy, A., and Koltun, V. Vision transformers for dense prediction. In *ICCV*, 2021.

Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *NeurIPS*, 2017.

Sajjadi, M. S. M., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lucic, M., Duckworth, D., Dosovitskiy, A., Uszkoreit, J., Funkhouser, T. A., and Tagliasacchi, A. Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. *CoRR*, abs/2111.13152, 2021.

Sajjadi, M. S. M., Duckworth, D., Mahendran, A., van Steenkiste, S., Pavetic, F., Lucic, M., Guibas, L. J., Greff, K., and Kipf, T. Object scene representation transformer. *CoRR*, abs/2206.06922, 2022.

Sauvalle, B. and de La Fortelle, A. Unsupervised multi-object segmentation using attention and soft-argmax. *CoRR*, abs/2205.13271, 2022.

Seitzer, M., Horn, M., Zadaianchuk, A., Zietlow, D., Xiao, T., Simon-Gabriel, C.-J., He, T., Zhang, Z., Schölkopf, B., Brox, T., and Locatello, F. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023.

Shaw, P., Uszkoreit, J., and Vaswani, A. Self-Attention with Relative Position Representations. In *NAACL-HLT*, 2018.

Singh, G., Deng, F., and Ahn, S. Illiterate DALL-E learns to compose. *CoRR*, abs/2110.11405, 2021.

Singh, G., Wu, Y., and Ahn, S. Simple unsupervised object-centric learning for complex and naturalistic videos. *CoRR*, abs/2205.14065, 2022.

Smirnov, D., Gharbi, M., Fisher, M., Guizilini, V., Efros, A. A., and Solomon, J. M. Marionette: Self-supervised sprite learning. In *NeurIPS*, 2021.

Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., and Vaswani, A. Bottleneck Transformers for Visual Recognition. In *CVPR*, 2021.

Stelzner, K., Kersting, K., and Kosiorek, A. R. Decomposing 3d scenes into objects via unsupervised volume segmentation. *CoRR*, abs/2104.01148, 2021.

Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhang, Y., Shlens, J., Chen, Z., and Anguelov, D. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

van Steenkiste, S., Chang, M., Greff, K., and Schmidhuber, J. Relational neural expectation maximization: Unsupervised discovery of objects and their interactions. In *ICLR*, 2018.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is All you Need. In *NeurIPS*, 2017.

Wang, D., Kohler, C., and Jr., R. P. Policy learning in SE(3) action spaces. In *Robot Learning, CoRL*, 2020a.

Wang, H., Zhu, Y., Green, B., Adam, H., Yuille, A. L., and Chen, L.-C. Axial-DeepLab: Stand-Alone Axial-Attention for Panoptic Segmentation. In *ECCV*, 2020b.

Wang, Y., Zhang, X., Yang, T., and Sun, J. Anchor DETR: Query Design for Transformer-Based Detector. In *AAAI*, 2022.

Watters, N., Matthey, L., Burgess, C. P., and Lerchner, A. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *CoRR*, abs/1901.07017, 2019.

Wu, Y. and He, K. Group normalization. In *ECCV*, 2018.

Wu, Z., Dvornik, N., Greff, K., Kipf, T., and Garg, A. Slotformer: Unsupervised visual dynamics simulation with object-centric models. *CoRR*, abs/2210.05861, 2022.

Xie, J., Xie, W., and Zisserman, A. Segmenting moving objects via an object-centric layered representation. In *NeurIPS*, 2022.

Yi, W. and Marshall, S. Principal component analysis in application to object orientation. *Geo-spatial Information Science*, 2000.

Yu, H., Guibas, L. J., and Wu, J. Unsupervised discovery of object radiance fields. In *ICLR*, 2022.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L. M., and Shum, H.-Y. DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection, 2022.

Zhao, H., Jia, J., and Koltun, V. Exploring Self-Attention for Image Recognition. In *CVPR*, 2020.

Zhou, Y., Zhang, H., Lee, H., Sun, S., Li, P., Zhu, Y., Yoo, B., Qi, X., and Han, J.-J. Slot-VPS: Object-centric representation learning for video panoptic segmentation. In *CVPR*, 2022.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. Deformable DETR: Deformable Transformers for End-to-End Object Detection. In *ICLR*, 2021.

A. Limitations

Although invariant slot attention has many benefits, we have identified the following areas of improvement and possible future work.

ISA-TSR incorporates spatial symmetries in 2D pixel space whereas the world is 3D. Without prior knowledge of the object's underlying 3D geometry it is difficult to model, for example, out of plane rotation, shear, and non-rigid deformations. Incorporating spatial symmetries in 3D extension of our method could help address this limitation.

Complex scenes are composed of not just objects but also "stuff" such as sky, road, and ocean. Modeling position, scale, and rotation of the sky, for example, is counter intuitive. Instead these could be modelled as background. A highly promising future direction is to combine Invariant Slot Attention with a background model – do the foreground and background models require different symmetries?

A limitation, not of our method, but of our experiments is that we focus solely on object discovery whereas our architectural contribution is more general. Our position and scale invariant cross-attention approach could be, for example, ported to DETR for supervised object detection and panoptic segmentation by replacing Slot Attention slot queries with DETR object queries. As in recent versions of DETR, such as Zhang et al. (2022), we use learnable initial slot positions and scales in some of our experiments. Zhang et al. (2022) predict the positions and scales of object queries (analogous to slots in slot attention), which are then refined throughout their decoder. We instead extract positions, rotations and scales from the attention masks to ensure equivariance of the computation.

Lastly, our experiments are still bottlenecked by the current capabilities of object discovery methods. As new methods for unsupervised / weakly-supervised instance level scene decomposition in in-the-wild data emerge, future work should investigate whether the explicit modeling of spatial symmetries continues to play a key role.

B. Pseudocode

The pseudo-code of ISA-TSR is provided in Algorithm 1. This follows the discussion in Section 4.1. The only detail that we'd like to add here is a scaling parameter δ which scales `rel_grid` in line 6. We set $\delta = 5$ in all our experiments. We do this for numerical and stability reasons. For example, this results in `rel_grid` ranging from -2 to 2 instead of -10 to 10 for an object with $S_s = 0.1$.

C. Model details

C.1. Architecture details

The model consists of the same three components as Slot Attention: an encoder E_θ, the attention module, and the decoder D_ϕ. In our experiments, two types of visual backbones were used: 1) a shallow CNN encoder similar to that of (Locatello et al., 2020) and 2) a ResNet-34 (He et al., 2016), details of which are explained below. In all cases the spatial broadcast operation in the SB decoder matched the number of tokens output by the encoder.

Encoder details For the Tetrominoes dataset, the shallow CNN encoder consists of 4 convolutional layers (64 channels, kernel size 5×5, stride 1, 'SAME' padding) and ReLU activations (Nair & Hinton, 2010). The output feature map size is 35×35 which is the same as the input size.

In the Objects Room dataset, the shallow CNN encoder is the same as that in Tetrominoes except that the first two layers have stride 2 and that input images are of size 64×64 resulting in feature maps of size 16×16.

For all other synthetic datasets (CLEVR, CLEVRTex, MultishapeNet), the shallow CNN encoder consists of 4 convolutional layers (64 channels, kernel size 5×5, 'SAME' padding) and ReLU activations. The stride is 2 for the first three layers and 1 for the remaining one layer. Input images are of size 128×128. We thus obtain feature maps of size 16×16.

ResNet-34 (He et al., 2016) is the standard ResNet model with 34 convolutional layers. We used group normalization (Wu & He, 2018) instead of batch normalization (Ioffe & Szegedy, 2015) to avoid a dependence on training batch size. All experiments that used this encoder, as opposed to the shallow CNN, have been marked with '(ResNet)' in the main manuscript. This includes several experiments on the CLEVRTex dataset and all the experiments on the WaymoOpen dataset. The inspiration to use a bigger backbone on more complex data comes from the SAVi++ method (Elsayed et al.,

Algorithm 1 Translation, Rotation, and Scaling Invariant Slot Attention.

Inputs: inputs $\in \mathbb{R}^{N \times D_{inputs}}$, abs_grid $\in \mathbb{R}^{N \times 2}$, slots $\in \mathbb{R}^{K \times D_{slots}}$, Slot positions, $S_p \in \mathbb{R}^{K \times 2}$, Slot rotations, $S_r \in \mathbb{R}^{K \times 2 \times 2}$, Slot scales, $S_s \in \mathbb{R}^{K \times 2}$, T iterations, small ϵ.
Data: Encoders f, g, k, v, q, parameters of LayerNorms, MLP and GRU, δ.
Outputs: slots $\in \mathbb{R}^{K \times D_{slot}}$, $S_p \in \mathbb{R}^{K \times 2}$, $S_r \in \mathbb{R}^{K \times 2 \times 2}$, $S_s \in \mathbb{R}^{K \times 2}$.

inputs = LayerNorm(inputs)
for $t = 1$ **to** $T + 1$ **do**
  slots_prev = slots
  slots = LayerNorm(slots)

  # Computes relative grids per slot, and associated key, value embeddings.
  for $k = 1$ **to** K **do**
    $\text{rel_grid}^k = [(S_r^k)^{-1}(\text{abs_grid} - S_p^k)] \ / \ \left(S_s^k \times \delta\right)$
    $\text{keys}^k = f\left(k(\text{inputs}) + g(\text{rel_grid}^k)\right)$
    $\text{values}^k = f\left(v(\text{inputs}) + g(\text{rel_grid}^k)\right)$
  end for

  # Inverted dot production attention.
  for $k = 1$ **to** K **do**
    $\text{attn}^k = \frac{1}{\sqrt{K}} \text{keys}^k * q(\text{slots}^k)^T$
  end for
  attn = softmax(attn, axis = "slots")
  updates = WeightedMean(weights = attn, values = values)
  attn /= Sum(attn, axis = "inputs")

  # Updates S_p, S_s and slots.
  for $k = 1$ **to** K **do**
    $S_p^k = \text{WeightedMean}(\text{weights} = \text{attn}^k, \text{values} = \text{abs_grid})$
    $S_r^k = \text{Symmetrize}(\text{WeightedPCAAnalytical}(\text{inputs} = \text{abs_grid} - S_p^k, \text{weights} = \text{attn}^k)$
    $S_s^k = \sqrt{\text{WeightedMean}(\text{weights} = \text{attn}^k + \epsilon, \text{values} = [(S_r^k)^{-1}(\text{abs_grid} - S_p^k)]^2)}$
  end for
  if $t < T + 1$ **then**
    slots = GRU(state = slots_prev, inputs = updates)
    slots += MLP(LayerNorm(slots))
  end if
end for

2022). Input images are 128×128 for CLEVRTex and 128×192 for Waymo Open. We reduce the stride in the ResNet root block (3×3 convolutional layer in the root block with stride 1 with no max pooling operation after it, as is conventional for ResNets on small input images). Thus the output feature map size is 16×16 for CLEVRTex and 16×24 for Waymo Open.

Attention module details The attention module consists of functions $\mathcal{Q}, \mathcal{K}, \mathcal{V}, \text{ and } g$ which are linear projections (without bias, except for g which has a bias term) and function f which is a MLP. The linear projections have output dimension 64, the MLP has a hidden size 128, output dimension 64, and applies layer-norm on its inputs ("pre-norm"). We used three iterations of slot attention as in the original publication (Locatello et al., 2020) which are mediated by a Gated Recurrent Unit and a MLP with hidden size 128, pre-normalization, output size 64, and a residual connection.

All experiments use 11 slots except in Tetrominoes where we use 4 slots and MultiShapeNet where we use 5 slots in both splits. Slots in slot attention are iteratively updated but must be initialized. We used a learned initialization in all cases. The learnable weights are themselves initialized using a standard normal distribution $\mathcal{N}(0, 1)$. Reference frames associated with these slots are also initialized with learnable embeddings in all experiments except in Tetrominoes and in CLEVR where they are randomly sampled in every forward pass. In those experiments where they are learnable embeddings, S_p are initialized using $\mathcal{U}[-1, 1]$, S_s are initialized using $\mathcal{N}(0.1, 0.01)$, S_r are initialized using $\frac{\pi}{4}\tanh(\mathcal{N}(0., 0.1))$. In those experiments where they are randomly sampled in every forward pass, S_p are sampled from $\mathcal{U}[-1, 1]$, S_s are sampled from $\mathcal{N}(0.1, 0.1)$, and S_r are sampled from $\mathcal{U}\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$.

Spatial broadcast decoder details In Tetrominoes, we use an MLP that decodes each pixel independently given the slot vector and pixel coordinates. It has five layers, with hidden size 256, interleaved with ReLU activations.

For the Objects Room dataset, we use a CNN decoder with 5 transpose convolutional layers (kernel size 5×5, stride 2 for the first two layers and stride 1 for the remaining, 'SAME' padding, 64 channels) interleaved by ReLU activations. Input feature maps are spatially broadcasted position encoded slots at a resolution of 16×16. The output image is of size 64×64.

All other datasets use the same setup as the Objects Room dataset, except that outputs are of size 128×128 (except Waymo Open at 128×192) and so while input feature maps are at 16×16 (16×24 for Waymo Open) there is one extra convolution transpose layer with stride 2.

A final dense layer (also referred to as a 1×1 conv.) projects the decoder output to the 3 channel RGB predictions.

Function h is applied to `rel_grid` and is a linear projection to 64 dimensions with a bias term.

C.2. Rotation estimation

Our heuristic assumes that the orientation of a slot is given by the axis with the highest variation (the first principal component). E.g., the axis of an outline of a car would be horizontal, whereas the axis of an outline of a person would be vertical. We can compute this axis v_1^k (and a second orthogonal axis v_2^k) for each slot k by using weighted Principal Component Analysis (PCA) with abs_grid as the input and attn as the weights:

$$v_1^k, v_2^k = \text{WeightedPCA}(\text{abs_grid}, \text{attn}^k) \tag{10}$$

Importantly, we can compute the axes *analytically* by computing the eigenvalues of a 2×2 weighted covariance matrix. This facilitates stable gradients when we backpropagate through the rotation detection. We further 'post-process' the axes to (1) make sure the coordinate grid is always left-handed (so that we do not accidentally mirror objects) and (2) limit rotations to $[0, 45°]$. The latter partially accounts for the ambiguity in the detected rotation – our heuristic attempts to align the object either horizontally or vertically with a gentle rotation without flipping it upside down, etc.:

$$\tilde{v}_1^k, \tilde{v}_2^k = \text{post-process}\left(v_1^k, v_2^k\right) , \tag{11}$$

$$S_r^k = \begin{bmatrix} | & | \\ \tilde{v}_1^k & \tilde{v}_2^k \\ | & | \end{bmatrix} . \tag{12}$$

$\tilde{v}_1$ and $\tilde{v}_2$ form the columns of a rotation matrix S_r^k.

D. Optimization details

The model is trained using Adam (Kingma & Ba, 2015) with a learning rate of 4×10^{-4} on all datasets except for Waymo Open Depths, where we use 2×10^{-4} following (Elsayed et al., 2022). We use a learning rate warm-up going from 0 for 50k steps. Afterwards, the learning rate decays using cosine decay to 0 (Loshchilov & Hutter, 2017). We train for 500k training steps, except for Waymo, 300k, and Tetrominoes, 50k. In Tetrominoes, we use 5k steps warm-up. The batch size is 64.

E. Evaluation metrics

In all datasets, except for Objects Room where we report ARI, we report the foreground adjusted rand index (FG-ARI). This metric measures the foreground object decomposition and is based on the ARI metric (Hubert & Arabie, 1985) popular in clustering literature. It measures, for each pair of pixels, whether their grouping is the same in the predicted segmentation maps and in the ground truth. It is permutation invariant and therefore suitable for class agnostic instance segmentation without the need for explicit matching between predicted and ground truth segments. FG-ARI has been reported in several recent works in the community such as (Singh et al., 2022; Kipf et al., 2022; Locatello et al., 2020).

We also report image reconstruction performance in terms of mean squared error (MSE) in some datasets. These are reported mainly for completeness and the model has not been optimized to produce high quality reconstructions.

F. Dataset details

F.1. Waymo Open

For depth prediction experiments, we use a pre-trained Dense Prediction Transformer (DPT, Ranftl et al. (2021)) to predict disparity masks for all images, which we then normalize to a 0-1 range *per image*. The DPT prediction are then used as *targets* for the Slot Attention decoder.

G. Additional experimental results

G.1. Invariance to rotations

All quantitative results comparing ISA-TS and ISA-TSR are reported in Table 4.

Table 4. **Rotation invariance**: Comparing ISA-TS against ISA-TSR in various benchmarks. Objects room results are ARIs whereas all others are FG-ARIs.

	(FG-)ARI ↑	
Dataset	ISA-TS	ISA-TSR
Objects room (w/ bg)		
Validation	85.5 ± 6.6	84.3 ± 4.6
Six Objects	84.5 ± 4.6	83.2 ± 2.7
Empty Room	83.6 ± 8.8	81.5 ± 7.1
Identical Colors	85.1 ± 5.9	83.8 ± 3.9
CLEVR	98.9 ± 0.2	98.0 ± 0.9
MultiShapeNet		
- All Data	69.8 ± 1.1	77.7 ± 5.5
- Four Objects	86.5 ± 1.1	80.7 ± 6.4
CLEVRTex (CNN)	78.8 ± 3.9	79.6 ± 5.5
CLEVRTex CAMO (CNN)	72.9 ± 3.5	73.8 ± 4.9
CLEVRTex OOD (CNN)	73.2 ± 3.1	74.9 ± 3.8
CLEVRTex (ResNet)	92.9 ± 0.4	93.3 ± 0.7
CLEVRTex CAMO (ResNet)	86.2 ± 0.8	87.0 ± 1.7
CLEVRTex OOD (ResNet)	84.4 ± 0.8	84.9 ± 1.2

Table 5. Results on the CLEVR dataset. Except for AST which used 3 seeds, all methods used 5 seeds. AST-Seg-B3-CT numbers are copied from Sauvalle & de La Fortelle (2022).

Method	↑FG-ARI	↓MSE
AST-Seg-B3-CT	98.3$_{\pm 0.1}$	16$_{\pm 1}$
SA	98.8$_{\pm 0.2}$	13$_{\pm 2}$
ISA-T	**99.0**$_{\pm 0.2}$	**11**$_{\pm 2}$
ISA-TS	98.9$_{\pm 0.2}$	**11**$_{\pm 1}$
ISA-TSR	98.0$_{\pm 0.9}$	12$_{\pm 2}$

Table 6. Result on Objects Room with background segments included, 10 random seeds.

Method	↑ARI (11 slots), background included.			
	Validation	Six Objects	Empty Room	Identical Colors
SA	63.0$_{\pm 17.9}$	62.8$_{\pm 18.4}$	58.4$_{\pm 16.4}$	61.2$_{\pm 17.8}$
ISA-T	83.5$_{\pm 9.2}$	82.5$_{\pm 6.7}$	78.7$_{\pm 14.3}$	82.4$_{\pm 8.2}$
ISA-TS	**85.5**$_{\pm 6.6}$	**84.5**$_{\pm 4.6}$	**83.6**$_{\pm 8.8}$	**85.1**$_{\pm 5.9}$
ISA-TSR	84.3$_{\pm 4.6}$	83.2$_{\pm 2.7}$	81.5$_{\pm 7.1}$	83.8$_{\pm 3.9}$

G.2. CLEVR dataset

A table of results for the CLEVR dataset are presented in Table 5.

G.3. Objects Room dataset

A table version of the bar plot in Figure 6-Top, is presented in Table 6.

G.4. MultiShapeNet

A table version of the bar plot in Figure 6-Bottom, along with the performance of ISA-TSR, is presented in Table 7. We further show learned segmentation masks, slot positions and scales in Figure 12.

Table 7. Results on MultiShapeNet-Easy, 5 random seeds.

Method	All Data		Four Objects	
	↑FG-ARI	↓MSE	↑FG-ARI	↓MSE
SA	56.8$_{\pm 2.6}$	33$_{\pm 5}$	71.5$_{\pm 11.6}$	50$_{\pm 2}$
ISA-T	71.2$_{\pm 3.5}$	29$_{\pm 3}$	82.3$_{\pm 5.3}$	43$_{\pm 4}$
ISA-TS	69.8$_{\pm 1.1}$	**27**$_{\pm 1}$	**86.5**$_{\pm 1.1}$	**41**$_{\pm 3}$
ISA-TSR	**77.7**$_{\pm 5.5}$	35$_{\pm 3}$	80.7$_{\pm 6.4}$	51$_{\pm 6}$

G.5. CLEVRTex results

Table 8 compiles all evaluations on the CLEVRTex alongside FG-ARI and MSE metrics and several baselines. We further report FG-mIoU results in Table 9. We further show image decodings, learned segmentation masks, slot positions, orientations and scales in Figures 13, 14 and 15.

G.6. Waymo Open results

Table 10 compiles all experiments pertaining to the Waymo Open dataset.

Table 8. CLEVRTex results on the test set, CAMO set (objects and backgrounds blend together) and OOD set (novel textures). Prior results taken from (Karazija et al., 2021) use 3 random seeds, we use 10 random seeds. FG-ARI is reported in %.

Method	**CLEVRTex**		**CLEVRTex CAMO**		**CLEVRTex OOD**	
	↑FG-ARI	↓MSE	↑FG-ARI	↓MSE	↑FG-ARI	↓MSE
SPACE	17.5±4.1	298±80	10.6±2.1	251±61	12.7±3.4	**387**±66
DTI	79.9±1.4	438±22	72.9±1.9	377±17	73.7±1.0	590±4
Gen-V2	31.2±12.4	315±106	29.6±12.8	278±75	29.0±11.2	539±147
eMORL	45.0±7.8	318±43	42.3±7.2	269±31	43.1±9.3	471±51
AST-Seg-B3-CT	**94.8**±0.5	**139**±7	87.3±3.8	**145**±6	83.1±0.8	832±24
SA	54.5±1.6	241.2±14.0	53.0±1.6	216.6±11.8	54.2±2.6	414.7±27.7
ISA-T (CNN)	66.8±5.7	229.8±20.4	65.0±4.9	213.2±16.3	65.1±4.8	458.6±25.2
ISA-TS (CNN)	78.8±3.9	223.8±6.5	72.9±3.5	210.9±6.2	73.2±3.1	480.6±32.6
ISA-TSR (CNN)	79.6±5.5	234.5±6.6	73.8±4.9	220.5±8.8	74.9±3.8	499.9±56.9
+ append	85.4±2.4	234.2±3.9	78.8±2.2	223.5±4.6	78.2±1.9	554.0±25.1
SA (ResNet)	91.3±2.7	206.7±20.8	84.9±2.9	224.4±21.9	81.4±1.4	629.1±29.9
ISA-T (ResNet)	87.4±6.6	197.1±27.0	79.0±5.9	217.6±24.5	78.6±4.9	548.1±21.3
ISA-TS (ResNet)	92.9±0.4	176.9±3.4	86.2±0.8	196.3±7.1	84.4±0.8	578.3±22.5
ISA-TSR (ResNet)	93.3±0.7	185.9±6.4	87.0±1.7	206.1±10.0	**84.9**±1.2	595.2±35.6
+ append	93.6±0.8	185.6±5.8	**87.5**±0.9	201.1±6.9	84.8±0.7	647.1±24.6
- dec. only	93.4±1.0	200.4±6.6	87.2±1.7	223.5±5.3	83.9±1.6	614.4±49.7
- stop grad	73.4±10.5	392.6±116.5	62.8±10.3	388.2±81.2	67.5±8.0	621.9±62.8

Table 9. CLEVRTex results on the test set, CAMO set (objects and backgrounds blend together) and OOD set (novel textures). FG-mIoU is reported in %. 10 seeds.

Method	**CLEVRTex**		**CLEVRTex CAMO**		**CLEVRTex OOD**	
	↑FG-mIoU	↓MSE	↑FG-mIoU	↓MSE	↑FG-mIoU	↓MSE
SA (ResNet)	69.7±2.2	206.7±20.8	63.9±2.1	224.4±21.9	63.5±1.6	629.1±29.9
ISA-T (ResNet)	66.4±3.2	197.1±27.0	58.7±2.7	217.6±24.5	61.7±2.8	**548.1**±21.3
ISA-TS (ResNet)	**72.4**±0.8	**176.9**±3.4	**65.6**±0.6	**196.3**±7.1	**66.7**±0.9	578.3±22.5
ISA-TSR (ResNet)	71.4±1.5	185.9±6.4	64.9±1.2	206.1±10.0	66.3±1.3	595.2±35.6

Table 10. Results on Waymo Open v1.4 with RGB and depth targets. All experiments used the ResNet-34 encoder and ran 10 random seeds.

Method	Target	FG-ARI	MSE
SA	RGB	27.6±12.7	584±35
ISA-TS	RGB	**39.8**±5.3	**523**±12
- dec. only	RGB	32.4±7.1	586±29
ISA-TSR	RGB	31.2±8.4	561±26
SA	Depth	40.6±5.0	**95**±9
ISA-T	Depth	40.6±7.9	106±11
ISA-T-ABS	Depth	**45.2**±12.2	96±9

Figure 12. Per-slot reference frames learned without supervision on MultiShapeNet. Left column: input image. Right column: reconstructed image. Middle column: we show the (x, y) position, (sx, sy) scale and orientation of individual slots overlaid over a predicted soft segmentation mask. Each pair of blue and orange arrows corresponds to one slot. The longer the arrows, the higher the scale of the slot is. We filter out slots that have a low predicted probability in the segmentation mask (i.e. they are probably inactive).

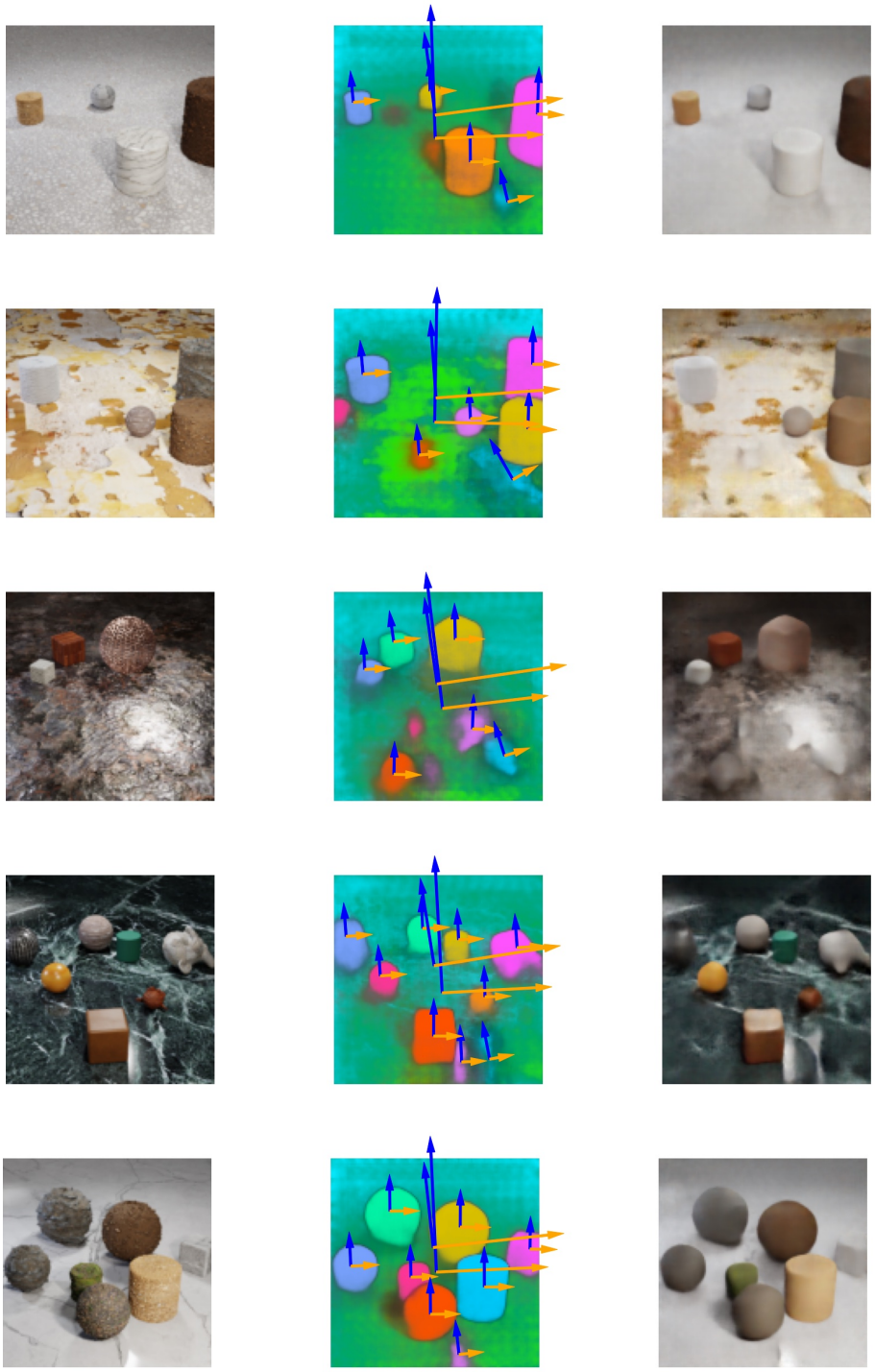


Figure 13. Per-slot reference frames learned without supervision on CLEVRTex. Left column: input image. Right column: reconstructed image. Middle column: we show the (x, y) position, (sx, sy) scale and orientation of individual slots overlaid over a predicted soft segmentation mask. Each pair of blue and orange arrows corresponds to one slot. The longer the arrows, the higher the scale of the slot is. We filter out slots that have a low predicted probability in the segmentation mask (i.e. they are probably inactive).

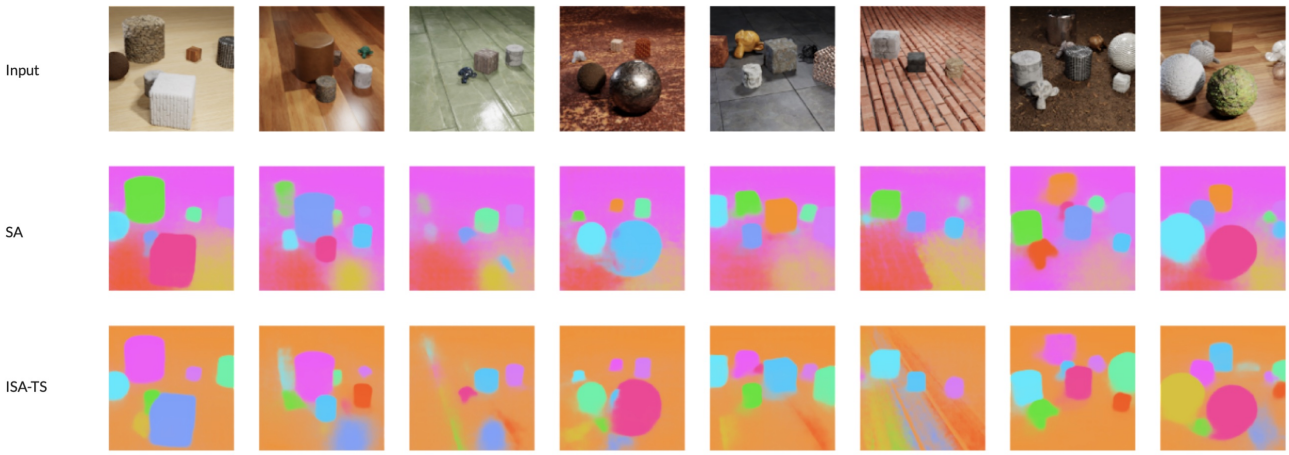


Figure 14. Comparison between Slot Attention and Translation and Scale Invariant Slot Attention in image segmentation.

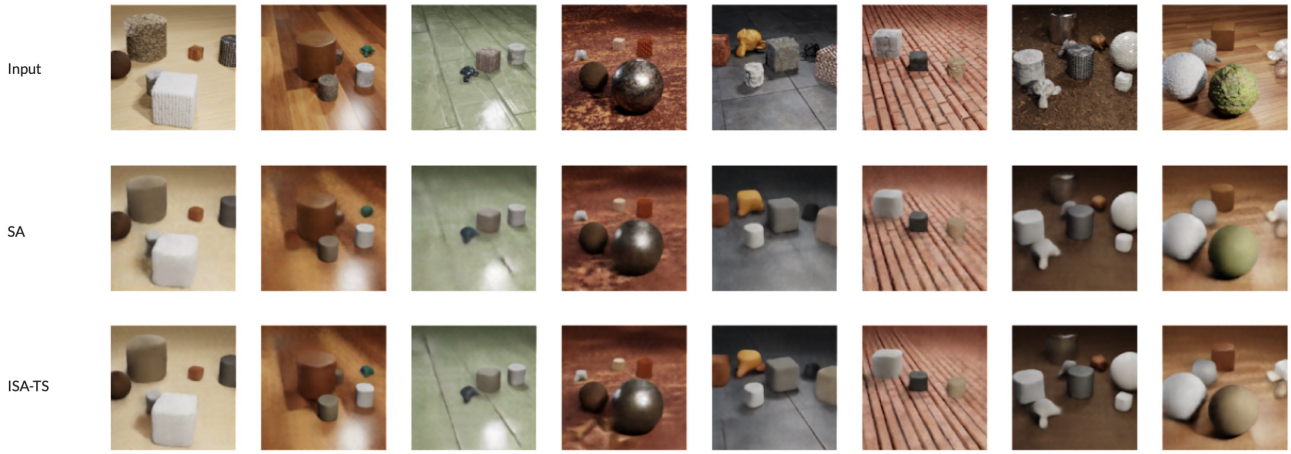


Figure 15. Comparison between Slot Attention and Translation and Scale Invariant Slot Attention in image decoding.