CTRL-O: Language-Controllable Object-Centric Visual Representation Learning

Aniket Didolkar[1,2*] **Andrii Zadaianchuk**[3*†] **Rabiul Awal**[1,2*]
Maximilian Seitzer[4] **Efstratios Gavves**[3,5] **Aishwarya Agrawal**[1,2†]
[1]Mila - Quebec AI Institute, [2]Université de Montréal
[3]University of Amsterdam, The Netherlands [4]University of Tübingen
[5]Archimedes/Athena RC, Greece
https://ctrl-o-paper.github.io

Abstract

Object-centric representation learning aims to decompose visual scenes into fixed-size vectors called "slots" or "object files", where each slot captures a distinct object. Current state-of-the-art object-centric models have shown remarkable success in object discovery in diverse domains, including complex real-world scenes. However, these models suffer from a key limitation: they lack controllability. Specifically, current object-centric models learn representations based on their preconceived understanding of objects, without allowing user input to guide which objects are represented. Introducing controllability into object-centric models could unlock a range of useful capabilities, such as the ability to extract instance-specific representations from a scene. In this work, we propose a novel approach for user-directed control over slot representations by conditioning slots on language descriptions. The proposed **CONTROLLABLE OBJECT-CENTRIC REPRESENTATION LEARNING** *approach, which we term* CTRL-O*, achieves targeted object-language binding in complex real-world scenes without requiring mask supervision. Next, we apply these controllable slot representations on two downstream vision language tasks: text-to-image generation and visual question answering. The proposed approach enables instance-specific text-to-image generation and also achieves strong performance on visual question answering.*

1. Introduction

Object-centric representation learning aims to decompose a visual scene into its constituent entities or objects and represent each entity as a distinct vector called a *slot*. Slot-based representations are inherently compositional and support many complex downstream tasks such as dynamics modeling [13, 32, 47], control [6, 7, 17, 53], and reasoning [1, 30]. Moreover, studies in cognitive neuroscience [37, 44] have shown that human perception uses mechanisms akin to slot-based representations.

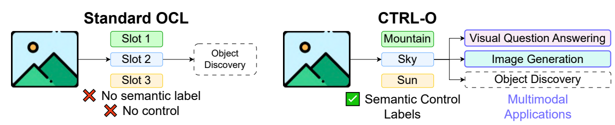

Figure 1. Left: Standard object-centric learning (OCL) assigns arbitrary slots with no control. Right: CTRL-O introduces language-based control, enabling specific object targeting and multimodal applications.

Existing unsupervised object-centric models [6, 10, 11, 29, 32, 42] can successfully decompose complex real-world visual scenes. However, they face a fundamental limitation: they lack control over the object representations. While these models expose control over the *number* of scene parts, they do not allow users to extract specific object representations within a scene (e.g., specified by user queries in the form of language or position markers). For instance, a user can specify that a scene should be decomposed into K slots, but they cannot direct a given slot to bind to a particular object of interest such as "a cat" or "a black purse".

This lack of control over the semantic content of the representation can be limiting, as this restricts the model always to extract a fixed decomposition of a scene based on its own preconceived understanding of objects and parts. Such rigidity can be problematic for applications that require representations at varying granularity, such as extracting the representation of a car wheel instead of the entire car, or vice versa.

Moreover, many downstream tasks may require or benefit from the knowledge of the semantic content in the slots. Due to the unsupervised nature of existing models, there is no way

[*]denotes equal contribution, order is determined by flipping a coin
[†]denotes equal advising contributions

to identify the content of a slot without manually checking its corresponding mask. For example, if the user needs representations of the cat and a dog in a particular image to answer a question, they would have to manually inspect masks of all discovered objects to pick the corresponding slots.

To address these limitations, we propose to inject controllability into object-centric representation learning. We achieve this by querying the model to represent specific objects in the image. Specifically, these queries condition the slot vectors to guide them to the objects described by the query. The queries can be in the form of natural language (such as object category names or referring expressions). The main challenge is to ensure that the slots conditioned on a specific query bind to the object referred to by that query. We term the challenge of binding slots to specific objects the *visual grounding problem* [16, 22, 36, 50]. We find that this problem is not trivial and introduce a novel controllable object-centric model — CTRL-O — to solve it. In our experiments, we demonstrate that the proposed approach can successfully bind slots to objects specified by user queries containing object categories or referring expressions in complex real-world scenes with limited supervision. In addition, we demonstrate the usefulness of the extracted controllable representations for two downstream tasks: visual question answering and instance-controllable image generation. Our contributions are as follows:

- We introduce CTRL-O, a novel method to learn controllable object-centric representations via user-defined inputs.
- We demonstrate that this approach supports extracting representations for complex reference expressions, enabling precise part specification within the representation.
- We validate the effectiveness of CTRL-O on two real-world downstream tasks: Instance-Controllable Image Generation and Visual Question Answering.

2. Related Works

Object-Centric Representation Learning Unsupervised object-centric representation learning (OCL) has gained a lot of interest in recent years [3, 5, 6, 10, 11, 15, 24, 29, 32, 42, 54]. OCL aims to extract individual representations for various entities in unstructured sensory inputs such as images. Slot Attention [29] introduces an attention-based mechanism to decompose images into object-centric representations. DINOSAUR [42] builds upon this by utilizing self-supervised DINO features [4, 35] to enhance unsupervised object discovery. While DINOSAUR can effectively identify objects in real-world data [27], it lacks mechanisms for top-down control over the representations. In contrast, CTRL-O provides controllable OCL by incorporating language-based control queries, allowing for flexible guidance with minimal supervision during training. Some works [24, 25] have explored conditioning mechanisms in object-centric models. SAVi [24] uses bounding boxes for the initial frame of a video for conditioning. CoSA [25] conditions on learned vector representations. These methods are often limited to specific forms of conditioning and are primarily evaluated on synthetic datasets, while our method can handle many forms of conditioning on real-world data. Finally, several recent works connect object-centric representations with language [12, 23, 46]. These works connect object representations with language post-hoc, assigning language labels to discovered slots. In contrast, CTRL-O integrates language and point conditioning directly into the learning process, allowing a user to control what representations should be extracted.

Downstream tasks with object-centric representations There has been limited work exploring the applicability of object-centric models to downstream tasks. Slotformer [47] uses the learned slots for world modelling and video question answering. Zadaianchuk et al. [53], Yoon et al. [52] and Didolkar et al. [6] investigate the applicability of object-centric representations for learning RL policies in simple environments such as Atari [34]. One drawback of these works is that they mainly consider synthetic environments and toy tasks; thus, their applicability is limited. In contrast, in this paper, we consider downstream applications in complex real-world environments. There are only a few works that study applications of object-centric representations in real-world settings. Mamaghan et al [30] investigate the application of object-centric representations in Visual Question Answering. We consider them as a baseline for our experiments on VQA. Slot Diffusion [48] and Stable LSD [20] use object-centric representations for generating real-world images. However, both these approaches lack controllability; hence, it is difficult to specify any conditioning information or control the images that these approaches generate. In contrast, we demonstrate in Section 4.3, that CTRL-O, when used for image generation, provides fine-grained control over the image generation.

3. Method

In this section, we describe the proposed approach for injecting controllability into existing object-centric models. We present a visual depiction of our method in Fig. 2.

In our setup, the input consists of an image X and user-defined queries embedded into vectors $L = \{l_j \in \mathbb{R}^{D_{\text{emb}}}\}_{j=1}^{M}$. The expected object-centric representation of the image X is a set of slots $S = \{s_i \in \mathbb{R}^{D_{\text{slot}}}\}_{i=1}^{N}$. The first M slots (we assume that $M \leq N$) should represent the object identified by the corresponding queries, while the remaining slots represent the unspecified parts of the scene. This way, the obtained representation is a complete decomposition of the whole image X, while still containing the parts corresponding to the user-specified queries L.

We consider controllability in the form of *language*

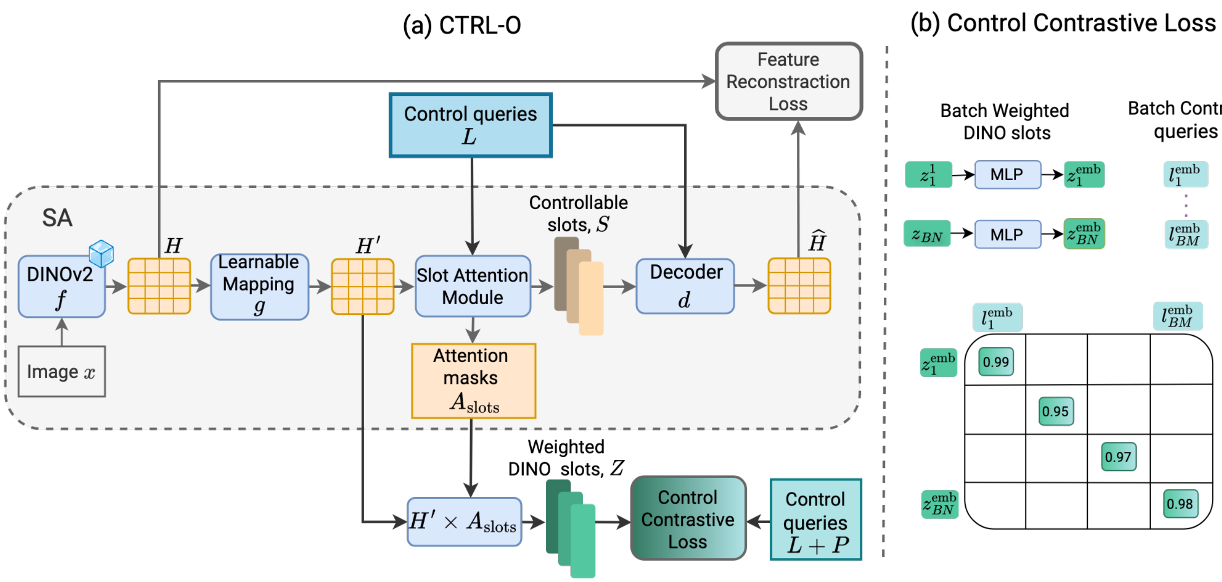

Figure 2. (a) Overview of CTRL-O architecture. An input image is processed by a frozen DINOv2 ViT model f, yielding patch features H. These features are then transformed into H' by a learnable transformer encoder g to align the feature space with the control queries. The control queries are introduced in the Slot Attention (SA) module, which guides the grouping of the encoded features into slots S. The initial slots in the SA module are conditioned with the control queries. Finally, an MLP decoder d, conditioned on control queries, reconstructs the DINOv2 features. (b) To ensure that slots utilize query information to represent specific objects, we apply a contrastive loss between control queries and the Slot Attention-modulated weighted DINO features A_{slot} (referred to as weighted DINO slots).

queries. We rely on the user to provide free-form text specifying object categories or object referring expressions whose visual representations are sought after. We encode this text into a fixed-sized vector embedding using LLM2Vec [2] with LLaMA-3-8BLLaMA-3-8B [33] to obtain these embeddings. These language embeddings comprise the queries we feed into the model.

3.1. CTRL-O Architecture

Background We base the proposed approach on DINOSAUR [42]. DINOSAUR uses the Slot Attention module [29] for object discovery. Slot Attention is an attention-based differentiable clustering procedure which, given a grid of features $H = \{h_k\}_{k=1}^{K} = f(X)$ obtained from an encoder f (we use DINOv2 [35]) applied to an image X, outputs a set of slots S such that each slot represents a distinct object in the image (see App. A for a detailed DINOSAUR description).

Query-based Slot Initialization We are solving the visual grounding problem: given the query corresponding to an object in the image, we want a slot to bind to exactly that object. One straightforward way to enforce grounding is to condition the slots directly on the query corresponding to each object. Specifically, we achieve this by adding the object query l_i to one of the slots (see Fig. 2, input to the Slot Attention Module). This approach is similar to SAVi [24], which conditions each slot on the objects' center of mass. In our experiments, we find that simply conditioning the slots on the queries does not lead to correct grounding; hence, a stronger signal is needed to ensure proper grounding.

Decoder Conditioning Similar to DINOSAUR, we use a broadcast MLP decoder, separately decoding each slot into patch features. We empirically find that conditioning the decoder on the corresponding control queries improves language grounding (concrete evaluation presented in Table 1. To implement this, we concatenate the resulting slots with the control queries and pass them through an MLP whose output is fed into the patch decoder as shown in Fig. 2 (a).

3.2. Control Contrastive Loss to Enforce Grounding

To enforce grounding, we introduce a contrastive loss, as illustrated in Fig. 2 (b). The intuition behind this objective is that if a slot s_i is conditioned on a query l_i corresponding to the object o_i, then we want the encoder features corresponding to the slot s_i to be close in embedding space to the query l_i. To obtain the features corresponding to slot s_i, we spatially aggregate the features output by the mapping network (learnable mapping g in Fig. 2 (a)) by weighting them with the attention scores of slot s_i, obtained from the last iteration of slot attention: $z_i = \sum_{k=1}^{K} a_{ik} h_k$, where a_{ik} denotes the attention score of slot s_i on feature h_k. We further process z_i using an MLP to output z_i^{emb}, which is used in the contrastive loss. Note that we do not directly

Figure 3. **Referring Expression Controllability on Visual Genome**. Visualization CTRL-O with free-form queries. The original image (left) and predicted segmentation masks are shown, with conditioning phrases presented above the corresponding segmented image; unconditioned slots have no phrase.

use the slots for the contrastive loss because the loss can be trivially satisfied by the slots as they are conditioned on the control queries, which are the targets for the contrastive loss. For the contrastive loss, we use (z_i^{emb}, l_i) as positive pairs which should be similar, and (z_i^{emb}, l_t) as negative pairs which should be dissimilar, with $t \neq i$. For the negatives, we consider all conditioning queries across the entire batch. Let there be T conditioning queries in the batch. The loss for a single sample is then formalized as:

$$\mathcal{L}_{CC}^{l} = -\sum_{i=1}^{M} \log \frac{\exp(z_i^{\text{emb}} \cdot l_i / \tau)}{\sum_{t=1}^{T} \exp(z_i^{\text{emb}} \cdot l_t / \tau)} \tag{1}$$

Here, τ is the temperature, which is set to 0.1. We assume two training regimes: when only language queries l_i are available and when both language queries l_i, and center-of-mass point queries p_i are provided during training. In the first regime, we define control contrastive loss $\mathcal{L}_{CC}$ as simply $\mathcal{L}_{CC}^{l}$. In the second regime, control contrastive loss $\mathcal{L}_{CC}$ is defined as the sum of two losses with language and point queries: $\mathcal{L}_{CC} = \mathcal{L}_{CC}^{l} + \mathcal{L}_{CC}^{p}$. We incorporate control contrastive loss $\mathcal{L}_{CC}$ in addition to the feature reconstruction loss from DINOSAUR. For additional implementation details, see App. B.

4. Experiments

In this section, we first show that CTRL-O learns to bind to the right regions in the image given complex natural language queries. Next, we tackle two downstream tasks — instance-specific image generation and visual question answering — using a pretrained CTRL-O model.

Slot Init.	GT Masks	CL	DC	Binding Hits	FG-ARI	mBO
✓	✓	✗	✗	**71.2**	**69.8**	**35.4**
✓	✗	✗	✗	8.1	34.52	22.42
✓	✗	✗	✓	10.11	43.83	25.76
✓	✗	✓	✗	56.3	44.8	27.3
✓	✗	✓	✓	61.3	47.5	27.2

Table 1. **CTRL-O Model Component Ablation for Grounding**. Importance of various components for achieving strong grounding. We use COCO *train* set for training and *val* set for evaluation. CL = Contrastive Loss, DC = Decoder Conditioning.

Approach	Model	FG-ARI	mBO
Unsup.	DINOSAUR (MLP Dec.) [42]	40.5	27.7
	DINOSAUR (TF. Dec.) [42]	34.1	31.6
	Stable-LSD [20]	35.0	30.4
	SlotDiffusion [48]	37.3	31.4
Weak Sup.	Stable-LSD (Bbox Supervision) [43]	-	30.3
	CTRL-O (Trained on COCO)	47.5	27.2

Table 2. **Object Discovery Performance.** Comparison of CTRL-O with unsupervised and weakly-supervised object-centric approaches on the COCO dataset.

4.1. Grounding Object-Centric Models

We study CTRL-O across two axes: 1) **Object Discovery** — How well can CTRL-O discover and represent each object separately in a scene?, and 2) **Grounding** — Can the slot conditioned on some language query l_j bind to the region specified by that language query?

Metrics To evaluate object discovery, we use standard metrics such as adjusted rand index (ARI) [19] and mean best overlap (mBO) [38]. To measure grounding, we introduce a new metric called *Binding Hits* which measures the grounding accuracy of the conditioned slots. Refer to App. F for more details regarding these metrics.

Datasets and Training Details We use COCO [27] and Visual Genome (VG) [26] as our main datasets of study. COCO contains category annotations spanning 91 different categories while VG contains region descriptions. COCO contains object annotations along with corresponding segmentation masks; we use it for quantitative evaluation of CTRL-O on object discovery and grounding. VG does not contain segmentation masks; hence, we only evaluate on it qualitatively.

Methods	Sup.	Image-text pretraining dataset	Fine tuning	RefCOCO *val*	RefCOCO *testA*	RefCOCO *testB*	RefCOCO+ *val*	RefCOCO+ *testA*	RefCOCO+ *testB*	Gref *val*
GroupViT [49]	$\mathcal{T}$	CC12M+YFCC	✗	7.99	6.16	10.51	8.49	6.79	10.59	10.68
			✓	10.82	11.11	11.29	11.14	10.78	11.84	12.77
MaskCLIP [55]	$\mathcal{T}$	WIT	✗	11.52	11.85	12.06	11.87	12.01	12.57	12.74
			✓	19.45	18.69	21.37	19.97	18.93	21.48	21.11
Shatter & Gather [23]	$\mathcal{T}$	VG	✗	21.80	19.00	24.96	22.20	19.86	24.85	25.89
CTRL-O	$\mathcal{T}$	VG	✗	21.80	20.10	21.57	21.90	21.54	21.36	25.32
CTRL-O	$\mathcal{T}+\mathcal{P}$	VG	✗	28.2	33.13	27.05	25.87	30.58	22.58	30.50

Table 3. **Referring expression segmentation** Comparison with weakly-supervised reference expression segmentation methods (Shatter & Gather) and open-vocabulary segmentation methods (GroupViT and MaskCLIP). The results on three datasets are reported in mIoU (%). Fine-tuning ✓means that the model is trained with the image-text pairs of the target benchmark; otherwise, the model is trained on the image-text pretraining dataset, and applied to the reference datasets zero-shot.

As several images in COCO contain multiple instances of the same object category, conditioning multiple slots on the same category name can be ambiguous for the model. Also, such conditioning poses problems for reliably computing the Binding Hits metric. Therefore, to disambiguate multiple instances of the same object category, we condition the slots on both category names and center of mass coordinates. We embed the language query using Meta-LLaMA-8B and the center of mass coordinates using a 2-layered MLP and concatenate them into a conditioning vector.

CTRL-O is trained for 300k steps on VG and COCO datasets with a batch size of 128 and Adam with 0.0004 learning rate. For VQA downstream task (Sec. 4.4), we use a batch size of 32 and AdamW with $5e-4$ learning rate.

Object Discovery (Table 2) We compare CTRL-O to various unsupervised and one weakly-supervised object discovery method. All the methods considered in Table 2 apply Slot Attention to the features of a pretrained encoder to extract slots. Following DINOSAUR, this has become the standard in unsupervised object discovery. The weakly-supervised approach, Stable LSD (w/ bbox supervision) [43], uses bounding boxes to supervise the Slot Attention alpha masks. CTRL-O conditions the slots on language and center of mass queries and also uses the same information for contrastive loss. Therefore, CTRL-O can be classified as a weakly-supervised approach. Note that we do not use any of the guidance information to directly supervise the Slot Attention masks; thus, our form of supervision is weaker as compared to Stable LSD (w/ bbox supervision). In Table 2, we show that CTRL-O outperforms all unsupervised approaches in terms of ARI but lags behind in terms mBO. The lower performance in terms of mBO can be attributed to the MLP decoder of the underlying DINOSAUR model, which also obtains a lower mBO. We find that a transformer decoder [42] or a diffusion decoder [20, 43, 48] results in sharper masks as compared to the MLP decoder. This experiment verifies that CTRL-O can discover objects in complex natural scenes. Next, we evaluate whether it can bind to the region specified by the control queries.

Grounding Object Categories (Table 1) Controllability is a new paradigm for object-centric models that has not been explored before. Hence, there are no direct baselines with which we can compare. Instead, we try to demonstrate the difficulty of the grounding problem and ablate over the components introduced in Sec. 3 to understand their importance in achieving good grounding. We use the COCO dataset for this evaluation. Table 1 presents the results for various ablations. To obtain an upper bound for grounding performance, we train a (fully supervised) CTRL-O model by directly predicting the ground truth masks (first row in Table 1). While such a supervised baseline achieves strong segmentation performance (as indicated by ARI and mBO), it still cannot achieve perfect grounding (close to 100% Binding Hits), highlighting the difficulty of the grounding problem. Out of the components introduced in Sec. 3, the control contrastive loss is the most crucial component for achieving good grounding accuracy, followed by decoder conditioning. Without the contrastive loss, the model has no incentive to utilize the queries; hence, no binding is emerging.

Grounding Referring Expressions (Figure 3) For COCO, we achieved controllability through both center of mass and category information. However, this approach is limited: COCO has a fixed number of categories, which affects generalizability. To overcome this issue, we can rely on *referring expressions*, where a user can refer to the target object using free-form natural language queries. To incorporate this ability, we use the VG dataset [26]. Since VG does not provide segmentation masks, we only evaluate CTRL-O qualitatively in this dataset.

We present the qualitative evaluation on VG in Fig. 3.

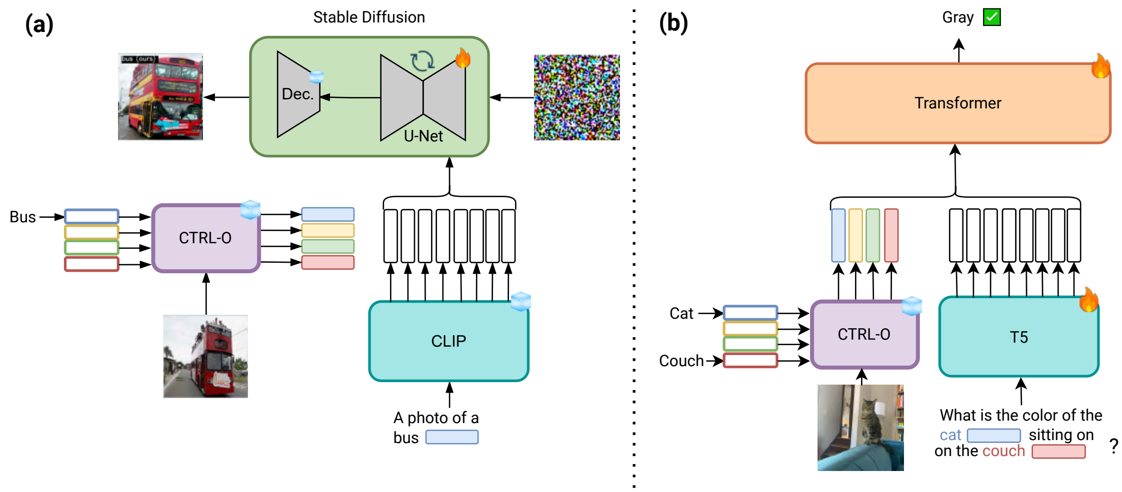

Figure 4. (a) **Instance Specific Image Generation**: We query an image to extract instance slot representations, which are then input into a Stable Diffusion model with the caption to generate the image. (b) **Visual Question Answering**: Slots are extracted from noun chunks or referring expressions in the question, then embedded into the text and input into the language model. = frozen; = trainable.

We visualize the attention regions of each slot for a random sample of examples. We observe that slots conditioned on phrases bind to the object referred to in the phrase, while the unconditioned slots bind to the other objects not used for conditioning. The last two rows of the visualization demonstrate that CTRL-O can decompose the same scene at different levels of granularity, such as, just the leg (corresponding to the query "The brown pants worn by the skier"), just the bag ("The orange bag on the skier's back"), or the entire skier ("Man going down ski slope").

Referring Expression Segmentation Evaluation (Table 3) We evaluate CTRL-O trained on VG on referring expression segmentation on the RefCOCO, RefCOCO+, and Gref datasets. This is a zero-shot evaluation since these datasets were not used for training. We compare to various referring expression segmentation baselines, using mIoU between the predicted and ground truth mask as the metric. The most relevant baseline is Shatter & Gather (SaG) [23], which also employs slot attention to extract slots from the image, followed by cross attention to associate the referring expression with a slot. One limitation of SaG is that the referring expression does not directly influence the slot extraction process. This can be problematic for cases where the referring expression refers to an object not extracted by Slot Attention. This is not the case for CTRL-O, as the referring expression directly influences the slot extraction. Moreover, all the baselines mentioned in Table 3 can only process a single referring expression per forward pass, while CTRL-O can process multiple referring expressions in parallel by conditioning multiple slots on the corresponding expressions. We find that CTRL-O outperforms all the baselines. However, CTRL-O by default uses language queries and center of mass annotations ($\mathcal{T}+\mathcal{P}$) while the baselines use only language queries as weak supervision. To ensure fairness, we also evaluate CTRL-O with only language queries ($\mathcal{T}$), which achieves competitive performance with SaG. Implementation details for this variant are in App. C.

4.2. Unlocking New OCL Capabilities with CTRL-O

In this section, we show that CTRL-O can be used for *instance controllable image generation* (Fig. 4(a)) — a use case where existing object-centric methods fail. We also demonstrate how CTRL-O can be used in a novel way to improve over existing object-centric methods in Visual Question Answering (VQA) (Fig. 4(b)). In these experiments, we use CTRL-O pre-trained on both VG and COCO with language-only conditioning (referring expressions in VG and Object Categories in COCO).

4.3. Instance Controllable Image Generation

In this section, our goal is to demonstrate that object-centric representations obtained from CTRL-O can be used for controllable image generation. Specifically, we aim for control in the space of instances where specific instances can be extracted from images and used as a conditioning for generating images containing those instances. Prior works, such as SlotDiffusion [48] and Stable-LSD [20], use Stable Diffusion (SD) [40] as a decoder to reconstruct images from slot vectors. These models condition the U-Net in SD on slots obtained from Slot Attention, enabling slot-conditional generation. However, these approaches are fundamentally limited: 1) To control the instances in the generated image, the user needs to manually reconstruct the masks corresponding to all the slots and find target slots that correspond to instances of interest. 2) They fix the diffusion model to a fixed number of slots, as in Slot Attention, limiting flexibility. Users cannot condition on a subset of objects while leaving the image layout flexible, and text inputs are unsupported.

CTRL-O addresses the above limitations: 1) Language-based control is inherent to CTRL-O. Hence, to control the instances present in the generated image, a user needs only to query CTRL-O to extract the corresponding instance representations from a given image as shown in Fig. 4(a). 2) We maintain the text interface of SD and only add the slots to control the visual identity of specific instances. This is similar to [51] where a user can specify an image containing an object for instance controlled generation. Instead of specifying images, we use slots obtained from CTRL-O for this task. For example, if we want to generate a specific bus as in Fig. 4(a), we use CTRL-O to extract its representation from an image, allowing us to prompt the diffusion model with "A photo of a bus. S_{bus}", where S_{bus} is obtained from CTRL-O.

CTRL-O-SD We present our image generation pipeline in Fig. 4(a), where CTRL-O is kept frozen and SD [40] serves as the generative model. We finetune the U-Net in SD, using the COCO dataset, while keeping other components fixed. Captions are used as conditioning inputs, and category names are extracted to obtain corresponding slots via CTRL-O. These slots are projected into CLIP embeddings and appended to the caption, as shown in Fig. 4(a). Unlike Stable LSD, where the slot attention module is trained alongside the diffusion model, we treat generation as a downstream task for CTRL-O.

Method	FID (↓)
LSD [20]	26.20
CTRL-O	25.20

Table 4. **Image generation quality**. CTRL-O achieves a lower FID.

Method	CLIP-I (↑)
SD [40]	0.71
CTRL-O	0.78

Table 5. **Instance controllable generation**. CTRL-O generates instances closer to the query image.

We compare CTRL-O-SD to Stable LSD, an existing object-centric method, using Fréchet Inception Distance (FID) to measure generative quality. Since Stable LSD lacks controllability, we focus only on image generation quality. As shown in Table 4, CTRL-O-SD achieves a lower FID, indicating superior image generation.

Next, we assess controllability on the COCO validation set, using SD for comparison. For SD, captions are used as conditioning; for CTRL-O-SD, captions and slots extracted from ground truth images are conditioned on object categories. We use the CLIP-I Score, which measures cosine similarity between the CLIP vision embedding of generated and ground truth images. Table 5 shows that CTRL-O-SD outperforms SD, demonstrating better instance controllability.

We also visualize the generated images from CTRL-O-SD and SD in Figure 5. We can see that the images generated using CTRL-O-SD contain instances that are closer to the query image. Additionally, CTRL-O-SD can compose slots corresponding to two different categories "A bench" and "A pizza" from two different images, to produce an image specified by "A pizza on a bench. S_{Bench} S_{Pizza}". We also highlight failure modes in App. H.4, where CTRL-O struggles with object binding and the diffusion model faces issues like object deformation and repetition.

4.4. Visual Questions Answering

Model	No-Coupling	Coupling
DINOv2 (22M)	58.26	58.12
CLIP (191M)	58.43	58.64
DINOSAUR (37M)	58.32	57.66
CTRL-O (39M)	59.18	60.25

Table 6. **Performance on VQAv2.** CTRL-O achieves the highest accuracy in both settings.

We consider VQA task on the VQAv2 dataset [14] as a classification problem, using accuracy as the metric. Language-guided object-centric representations from CTRL-O can potentially provide a much stronger coupling between the vision and language inputs. To achieve this, we propose to directly insert the slots into the question before feeding it into the language model (see Fig. 4(b)). Given a question, we first use spaCy to extract noun chunks (e.g., Cat, Couch) for the image, which are then used to condition the slots in CTRL-O to extract the corresponding slots. These slots are then inserted into the question at the appropriate positions as shown in Fig. 4(b). For example, the question "What is the color of the cat sitting on the couch?" becomes "What is the color of the cat S_{cat} sitting on the couch S_{couch}?". We use a learnable linear projection to map the slots to the same dimension as the T5 embeddings. We feed this question, with the inserted slots, into the language model. Therefore, the language and vision inputs are strongly coupled from the input stage, allowing for more interaction between visual and language components. We refer to this technique as *coupling*.

Setup. Our full pipeline is shown in Fig. 4(b), inspired by [30]. We use T5 as the language embedding model, and the vision model is CTRL-O. For the baselines, we use CLIP [39], DINOv2 [35], and DINOSAUR [42] features as representations. The output network is a transformer with 2 layers and 64 heads. For all methods, we feed visual representations (patches for CLIP and DINOv2 and slots for CTRL-O and DINOSAUR) and the language embeddings from T5 into the output network. The output network is trained from scratch, while T5 undergoes fine-tuning.

We introduce two variants of VQA training with CTRL-O: 1) CTRL-O that directly feeds the slots into the

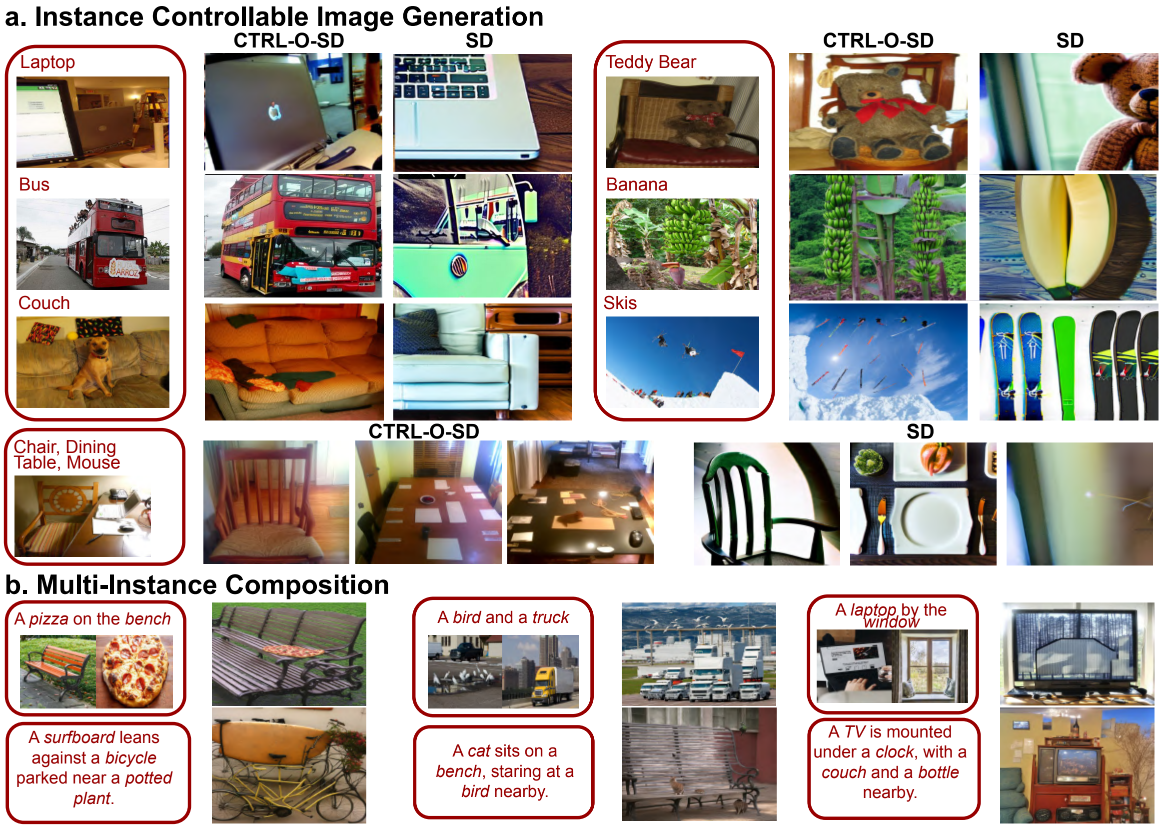

Figure 5. **a. Instance Controllable Image Generation.** Comparison between CTRL-O-SD and the baseline Stable Diffusion (SD). For a given query image (marked query), we extract a slot representation of a specific instance I_q (e.g., laptop, bus, banana). In CTRL-O-SD, the input is "A photo of I_q. S_{I_q}" to guide instance generation, while for SD, only "A photo of I_q" is used. Our approach produces images that more closely match the visual identity of the conditioned instance. **b. Multi-Instance Composition**. We extract instances from multiple images (e.g., "bench" and "pizza") and compose them into a single image, as seen with "the pizza on the bench".

Transformer output network without embedding them in the language, similar to baseline methods; 2) CTRL-O (with coupling) that inserts the corresponding slot representations into the appropriate place in the question as shown in Fig. 4(b). To ensure a fair comparison, we extend coupling experiments to all baselines. Since these models lack explicit object binding, we insert their aggregated features (CLS token for DINOv2/CLIP and slot mean for DINOSAUR) into the same positions as CTRL-O 's control slots. We report VQA classification accuracy (see details in App. G).

Results (Table 6) Both CTRL-O variants outperform all baselines, demonstrating the effectiveness of its object-centric representations. Notably, coupling primarily benefits CTRL-O, as its representations are explicitly aligned with language. In contrast, baselines insert full image representations that do not explicitly correspond to the preceding text, limiting coupling's usefulness. We note that while our method improves performance, it remains below the latest state-of-the-art models (>80%) [28], which leverage large language models (LLMs) and web-scale multimodal data.

5. Conclusion

We introduced CTRL-O, a controllable object-centric model that can be queried to extract representations of specific objects in a scene. We experimentally showed that representations of specific objects can be extracted in complex real-world scenes based on a range of user queries such as object category names or referring expressions. This capability expands the applicability of object-centric models to various real-world applications, such as instance-controllable image generation and visual question answering. Therefore, our work takes a step towards improving the applicability of object-centric representations to complex real-world downstream tasks. Future work could further enhance object grounding, for example, by considering multi-modal [21], 3D- and motion-aware dense features [9, 41], making object-centric representations more useful for diverse downstream tasks. We hope that learning controllable object-centric representations becomes the standard in OCL and leads to broader adoption of object-centric models to various domains and downstream tasks.

Acknowledgments

The authors would like to thank Kanishk Jain, Ankur Sikarwar, and Le Zhang for reviewing and providing feedback on an earlier version of the paper and Amir M. Karimi and Samuele Papa for providing the code for the VQA experiments in Section 4.4. This research was enabled in part by compute resources provided by Mila (mila.quebec). AD would like to thank Nanda Harishankar Krishna, Moksh Jain, and Rohan Banerjee for help with Fig. 4. Additionally, AD would like to thank Anirudh Goyal and Mike Mozer for helpful discussions regarding the research direction explored in this paper. AD also acknowledges the support of a scholarship from UNIQUE (https://www.unique.quebec). AZ is funded by the European Union (ERC, EVA, 950086). During this project, Aishwarya Agrawal was supported by the Canada CIFAR AI Chair award.

Contributions

AD and RA initially started a collaboration on object-centric models for vision-language tasks. AD came up with the idea of controllable slots conditioned on language. AZ proposed the contrastive objective for the model. AD and AZ developed the code for the model and conducted experiments on object grounding. AD ran all the baselines in Section 4.1. RA developed the code and ran experiments for instance-controllable image generation. AZ and RA developed the initial code for the VQA experiment, which AD and RA further extended for the final results. AZ, RA, and AD ran the baselines for Section 4.4. AA and AZ provided supervision and guidance throughout the project. MS and EG took part in discussions. AD, RA, AZ, and AA wrote the paper with help from MS.

References

[1] Rim Assouel, Pau Rodriguez, Perouz Taslakian, David Vazquez, and Yoshua Bengio. Object-centric compositional imagination for visual abstract reasoning. In *ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality*, 2022. 1

[2] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961*, 2024. 3, 1

[3] Ondrej Biza, Sjoerd Van Steenkiste, Mehdi SM Sajjadi, Gamaleldin F Elsayed, Aravindh Mahendran, and Thomas Kipf. Invariant slot attention: Object discovery with slot-centric reference frames. *arXiv preprint arXiv:2302.04973*, 2023. 2

[4] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging Properties in Self-Supervised Vision Transformers. *ICCV*, 2021. 2, 1

[5] Aniket Didolkar, Andrii Zadaianchuk, Anirudh Goyal, Mike Mozer, Yoshua Bengio, Georg Martius, and Maximilian Seitzer. Zero-shot object-centric representation learning. *arXiv preprint arXiv:2408.09162*, 2024. 2

[6] Aniket Rajiv Didolkar, Anirudh Goyal, and Yoshua Bengio. Cycle consistency driven object discovery. In *ICLR*, 2024. 1, 2

[7] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. PaLM-E: An embodied multimodal language model. In *ICML*, 2023. 1

[8] Constantin Eichenberg, Sidney Black, Samuel Weinbach, Letitia Parcalabescu, and Anette Frank. Magma–multimodal augmentation of generative models through adapter-based finetuning. *arXiv preprint arXiv:2112.05253*, 2021. 4

[9] Mohamed El Banani, Amit Raj, Kevis-Kokitsi Maninis, Abhishek Kar, Yuanzhen Li, Michael Rubinstein, Deqing Sun, Leonidas Guibas, Justin Johnson, and Varun Jampani. Probing the 3d awareness of visual foundation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21795–21806, 2024. 8

[10] Martin Engelcke, Adam R. Kosiorek, Oiwi Parker Jones, and Ingmar Posner. GENESIS: Generative Scene Inference and Sampling with Object-Centric Latent Representations. In *ICLR*, 2020. 1, 2

[11] S. M. Ali Eslami, Nicolas Heess, Theophane Weber, Yuval Tassa, David Szepesvari, Koray Kavukcuoglu, and Geoffrey E. Hinton. Attend, Infer, Repeat: Fast Scene Understanding with Generative Models. In *NeurIPS*, 2016. 1, 2

[12] Ke Fan, Zechen Bai, Tianjun Xiao, Dominik Zietlow, Max Horn, Zixu Zhao, Carl-Johann Simon-Gabriel, Mike Zheng Shou, Francesco Locatello, Bernt Schiele, Thomas Brox, Zheng Zhang, Yanwei Fu, and Tong He. Unsupervised open-vocabulary object localization in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13747–13755, 2023. 2, 1

[13] Anirudh Goyal, Aniket Didolkar, Nan Rosemary Ke, Charles Blundell, Philippe Beaudoin, Nicolas Heess, Michael C Mozer, and Yoshua Bengio. Neural production systems. *Advances in Neural Information Processing Systems*, 34:25673–25687, 2021. 1

[14] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 7

[15] Klaus Greff, Raphaël Lopez Kaufman, Rishabh Kabra, Nick Watters, Christopher Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-Object Representation Learning with Iterative Variational Inference. In *ICML*, 2019. 2

[16] Klaus Greff, Sjoerd Van Steenkiste, and Jürgen Schmidhuber. On the Binding Problem in Artificial Neural Networks. *arXiv:2012.05208*, 2020. 2

[17] Dan Haramati, Tal Daniel, and Aviv Tamar. Entity-centric reinforcement learning for object manipulation from pixels. In *ICLR*, 2024. 1

[18] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, Günter Klambauer, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR*, abs/1706.08500, 2017. 5

[19] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 1985. 4, 2

[20] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. In *NeurIPS*, 2023. 2, 4, 5, 6, 7

[21] Cijo Jose, Théo Moutakanni, Dahyun Kang, Federico Baldassarre, Timothée Darcet, Hu Xu, Daniel Li, Marc Szafraniec, Michaël Ramamonjisoa, Maxime Oquab, et al. Dinov2 meets text: A unified framework for image-and pixel-level vision-language alignment. *arXiv preprint arXiv:2412.16334*, 2024. 8

[22] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. ReferItGame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 787–798, Doha, Qatar, 2014. Association for Computational Linguistics. 2

[23] Dongwon Kim, Namyup Kim, Cuiling Lan, and Suha Kwak. Shatter and gather: Learning referring image segmentation with text supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15547–15557, 2023. 2, 5, 6, 1

[24] Thomas Kipf, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Austin Stone, Sara Sabour, Georg Heigold, Rico Jonschkowski, Alexey Dosovitskiy, and Klaus Greff. Conditional Object-centric Learning from Video. In *ICLR*, 2022. 2, 3

[25] Avinash Kori, Francesco Locatello, Francesca Toni, and Ben Glocker. Unsupervised conditional slot attention for object centric learning. *arXiv preprint arXiv:2307.09437*, 2023. 2

[26] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Fei-Fei Li. Visual genome: Connecting language and vision using crowdsourced dense image annotations, 2016. 4, 5

[27] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *ECCV*, 2014. 2, 4

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023. 8

[29] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-Centric Learning with Slot Attention. In *NeurIPS*, 2020. 1, 2, 3

[30] Amir Mohammad Karimi Mamaghan, Samuele Papa, Karl Henrik Johansson, Stefan Bauer, and Andrea Dittadi. Exploring the effectiveness of object-centric representations in visual question answering: Comparative insights with foundation models. *arXiv:2407.15589*, 2024. 1, 2, 7, 4

[31] Oscar Mañas, Pau Rodriguez, Saba Ahmadi, Aida Nematzadeh, Yash Goyal, and Aishwarya Agrawal. Mapl: Parameter-efficient adaptation of unimodal pre-trained models for vision-language few-shot prompting. *arXiv preprint arXiv:2210.07179*, 2022. 4

[32] Anna Manasyan, Maximilian Seitzer, Filip Radovic, Georg Martius, and Andrii Zadaianchuk. Temporally consistent object-centric learning by contrasting slots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1, 2

[33] Meta. The llama 3 herd of models, 2024. 3

[34] Volodymyr Mnih. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013. 2

[35] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2023. 2, 3, 7, 1

[36] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, 2018. 2

[37] Steven Pinker. Visual cognition: An introduction. *Cognition*, 1984. 1

[38] Jordi Pont-Tuset, Pablo Arbeláez, Jonathan T. Barron, Ferran Marques, and Jitendra Malik. Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 4, 2

[39] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 7, 1

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 6, 7

[41] Mohammadreza Salehi, Efstratios Gavves, Cees G.M. Snoek, and Yuki M. Asano. Time does tell: Self-supervised time-tuning of dense image representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16536–16547, 2023. 8

[42] Maximilian Seitzer, Max Horn, Andrii Zadaianchuk, Dominik Zietlow, Tianjun Xiao, Carl-Johann Simon-Gabriel, Tong He, Zheng Zhang, Bernhard Schölkopf, Thomas Brox, and Francesco Locatello. Bridging the gap to real-world object-centric learning. In *ICLR*, 2023. 1, 2, 3, 4, 5, 7

[43] Krishnakant Singh, Simone Schaub-Meyer, and Stefan Roth. Guided latent slot diffusion for object-centric learning, 2024. 4, 5

[44] Elizabeth S. Spelke. Core knowledge. *The American psychologist*, 2000. 1

[45] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 4

[46] Ruocheng Wang, Jiayuan Mao, Samuel J Gershman, and Jiajun Wu. Language-mediated, object-centric representation learning. *arXiv preprint arXiv:2012.15814*, 2020. 2

[47] Ziyi Wu, Nikita Dvornik, Klaus Greff, Thomas Kipf, and Animesh Garg. SlotFormer: Unsupervised visual dynamics simulation with object-centric models. In *ICLR*, 2023. 1, 2

[48] Ziyi Wu, Jingyu Hu, Wuyue Lu, Igor Gilitschenski, and Animesh Garg. Slotdiffusion: Object-centric generative modeling with diffusion models. In *NeurIPS*, 2023. 2, 4, 5, 6

[49] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022. 5

[50] Kelvin Xu. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015. 2

[51] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18381–18391, 2022. 7

[52] Jaesik Yoon, Yi-Fu Wu, Heechul Bae, and Sungjin Ahn. An investigation into pre-training object-centric representations for reinforcement learning. In *ICML*, 2023. 2

[53] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Self-supervised Visual Reinforcement Learning with Object-centric Representations. In *ICLR*, 2020. 1, 2

[54] Andrii Zadaianchuk, Maximilian Seitzer, and Georg Martius. Object-centric learning for real-world videos by predicting temporal feature similarities. In *NeurIPS*, 2023. 2

[55] Chong Zhou, Chen Change Loy, and Bo Dai. Extract free dense labels from clip. In *European Conference on Computer Vision (ECCV)*, 2022. 5