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

Supplementary Material

A. DINOSAUR Implementation Details

DINOSAUR uses a DINO [4] encoder to process the image into features. It relies on a feature reconstruction loss to supervise the object discovery process. Throughout the training, the DINO encoder is kept frozen. We adopt a similar approach, however we use a DINOv2 [35] encoder instead of a DINO encoder. Figure 6 illustrates the DINOSAUR architecture with a DINOv2 backbone. Additionally, we have added a learnable mapping network g, which is a 3-layer Transformer after the frozen DINOv2 encoder. SA module is applied on top of the mapping output as shown in Figure 2(a).

B. CTRL-O Implementation Details

Algorithm 1 Slot Attention with Language Conditioning

Input: inputs $\in \mathbb{R}^{N \times D_{\text{inputs}}}$, slots $\in \mathbb{R}^{K \times D_{\text{slot}}}$, language queries $\ell \in \mathbb{R}^{M \times D_{\text{lang}}}$

Layer params: k, q, v: linear projections for attention; p_ℓ: projection for language query; GRU; MLP; LayerNorm (x3)

1: inputs $\leftarrow$ LayerNorm(inputs)
2: $\ell_{\text{proj}} \leftarrow p_\ell(\ell)$ ▷ Project M language queries to slot dimension
3: $\{\text{slots}\}_{i=1}^{M} \leftarrow \ell_{\text{proj}}$ ▷ Condition first M slots on language query
4: **for** $t = 0 \ldots T-1$ **do**
5: $\quad \text{slots}_{\text{prev}} \leftarrow \text{slots}$
6: $\quad \text{slots} \leftarrow \text{LayerNorm}(\text{slots})$
7: $\quad \text{attn} \leftarrow \text{Softmax}(\frac{1}{\sqrt{D}} k(\text{inputs}) \cdot q(\text{slots})^\top \text{ axis} = \text{slots})$
8: $\quad \text{updates} \leftarrow \text{WeightedMean}(\text{weights} = \text{attn} + \epsilon, \text{values} = v(\text{inputs}))$
9: $\quad \text{slots} \leftarrow \text{GRU}(\text{state} = \text{slots}_{\text{prev}}, \text{inputs} = \text{updates})$
10: $\quad \text{slots} \leftarrow \text{slots} + \text{MLP}(\text{LayerNorm}(\text{slots}))$
11: **end for**
12: **return** slots

We present the modified Slot Attention with query-based initialization in Algorithm 1.

Control Contrastive Loss For conditioning, we mainly use language queries. However, we assume that each image in our dataset consists of multiple object annotations, each containing a center of mass annotation and a category or referring expression annotation. Therefore, we have two separate contrastive losses - one each for the language information and the point information, as shown in Figure 2(b).

Conditioning We run Slot Attention for a fixed number of slots K. However, in general, we may not have K queries per image. In such cases, we initialize a subset of the slots with the given queries, and the rest are free to bind to any of the other objects in the scene (see line 3 of Algorithm 1). When computing the contrastive loss, we only consider slots conditioned on some query.

C. Training CTRL-O with Language Queries

Needing center of mass annotations for the contrastive loss can be a limitation as these annotations may not be available in many datasets. Further, the main baseline that we consider for the referring expression segmentation task (Section 4.1) - Shatter-and-Gather [23] - does not require center of mass annotations. Therefore, for an apples-to-apples comparison, we implement a variant of CTRL-O which does not require center of mass annotations.

A visual depiction of this approach is presented in Figure 8. First, we remove additional center-of-mass information and leave only language queries in the contrastive loss. We find that simply removing the center-of-mass information leads to collapse of representations as the contrastive loss can be trivially satisfied by directly using the language embeddings on which the slots are conditioned on - we term this as *leakage*. To prevent leakage we propose to use CLIP [39] image features and language embeddings in the control contrastive loss. In particular, instead of taking the weighted average of DINO features (Figure 2(a)), we take the weighted average of patch-based CLIP features [12]. The slot conditioning still uses language embeddings from LLaMa-3-8B [2], however, CLIP language embeddings are used as targets in the contrastive loss. This way, CTRL-O learns to bind to the correct regions in the image specified by language queries without center-of-mass annotations.

D. Choice of Decoder for CTRL-O

In this subsection, we additionally study the compatibility of CTRL-O with different previously proposed decoders. In particular, we investigate the compatibility and scalability of our method with two different decoder architectures (MLP and Transformer). In Table 2, we compare our method with other OCL methods, showing that while our method strongly

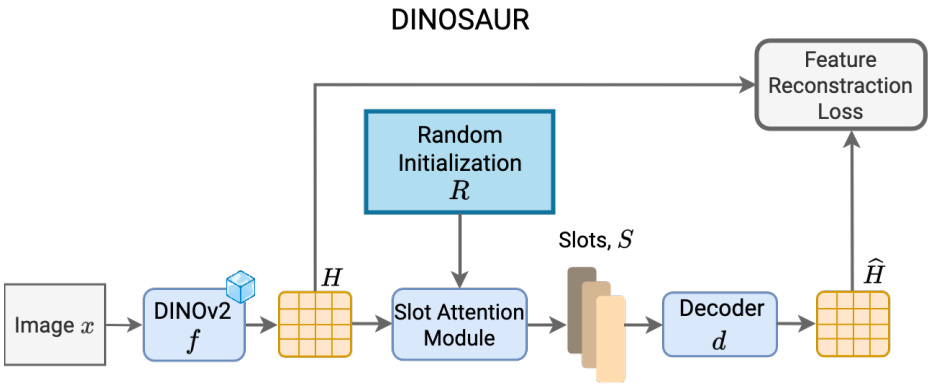

Figure 6. Overview of DINOSAUR architecture. The image is processed into a set of patch features H by a frozen DINO ViT model. The Slot Attention module groups the encoded features into a set of slots initialized by random queries sampled from the same Gaussian distribution with learnable parameters. By contrast, CTRL-O is initialized by the combination of control queries for conditioned slots and random queries for unconditioned slots. DINOSAUR is trained by reconstructing the DINO features from the slots using MLP decoder [42].

outperforms other methods in FG-ARI, its mask quality is lower than methods with stronger (pretrained) diffusion and Transformer decoders that have less inductive bias towards scene decomposition. Thus, it is important to investigate how our method performs with different decoders and whether we can scale MLP decoders for better mask quality. Object discovery with the Transformer decoder was shown to be sensitive to hyperparameters and can entirely fail (see App. D.4 and D.5 of DINOSAUR paper [42]). Subsequently, we also find that CTRL-O with Transformer decoder achieves 10.2 mBO. Through thorough investigation, we conclude that *Transformer decoder is not compatible with contrastive loss, which is needed for language controllability in CTRL-O but not in the baselines*. Thus, to improve masks quality we propose to scale the MLP decoder itself; scaling MLP dim to 4096 led to improved 28.0 mBO and 47.9 FG-ARI.

E. Referring Expression Visualization

In Figure 7, we compare the visualizations obtained from CTRL-O ($\mathcal{L}+\mathcal{P}$ setting), CTRL-O ($\mathcal{L}$ setting), and Shatter-and-Gather (SaG). Note that the queries listed on the top of each column are free-form queries created by a user and may not be similar to those typically found in the visual genome dataset. One potential issue with Shatter-and-Gather is that the language queries do not influence the slot extraction process - Slot attention first extracts a fixed number of slots, after which the query binds to the most relevant slot post-hoc. This can be limiting, as in some cases, the region referred to by the query may not be extracted into a single slot. In such cases, the language query may not bind to any slot. In Figure 7, we find that this is exactly what happens in several cases for Shatter-and-Gather. For example, in the first column, for the query "The orange bag on the skier's bag", SaG binds to the skier's shoes. In the 5th column, SaG fails to bind to any region for the query "the lamp". In contrast, both variants of CTRL-O frequently bind to the correct regions specified by the queries. Secondly, in CTRL-O the language queries directly influence slot extraction which allows it to explicitly extract the referred regions from the image and bind to them.

A particularly interesting case is the last row for CTRL-O ($\mathcal{L}$), where it learns to bind correctly even though queries are less specific and more subjective - "the ancient building" and "the new building". This emphasizes the generalizability of CTRL-O to complex language queries.

F. Object Discovery and Binding Metrics

FG-ARI The *adjusted rand index* (ARI) measures the similarity between two clusterings [19]. We use the instance/object masks as the targets. We only compute this metric for pixels in the foreground (hence, FG-ARI). Unlabeled pixels are treated as background.

mBO To compute the mBO [38], each ground truth mask (excluding the background mask) is assigned to the predicted mask with the largest overlap in terms of IoU.The mBO is computed as the average IoU of these mask pairs.

Binding Hits This metric measures controllable grounding. For binding hits, consider that a slot s_i is conditioned on a query L_i identifying an object o_i with ground-truth mask m_i. The broadcast decoder of slot attention outputs a mask per slot. If the overlap between the predicted mask for slot s_i, denoted as $\hat{m}_i$, and the ground truth mask m_i is the highest among all pairs of predicted and ground truth masks, it is considered as a hit (1) else it is considered as a miss (0). Binding Hits metric is measured as the average number of hits across the dataset.

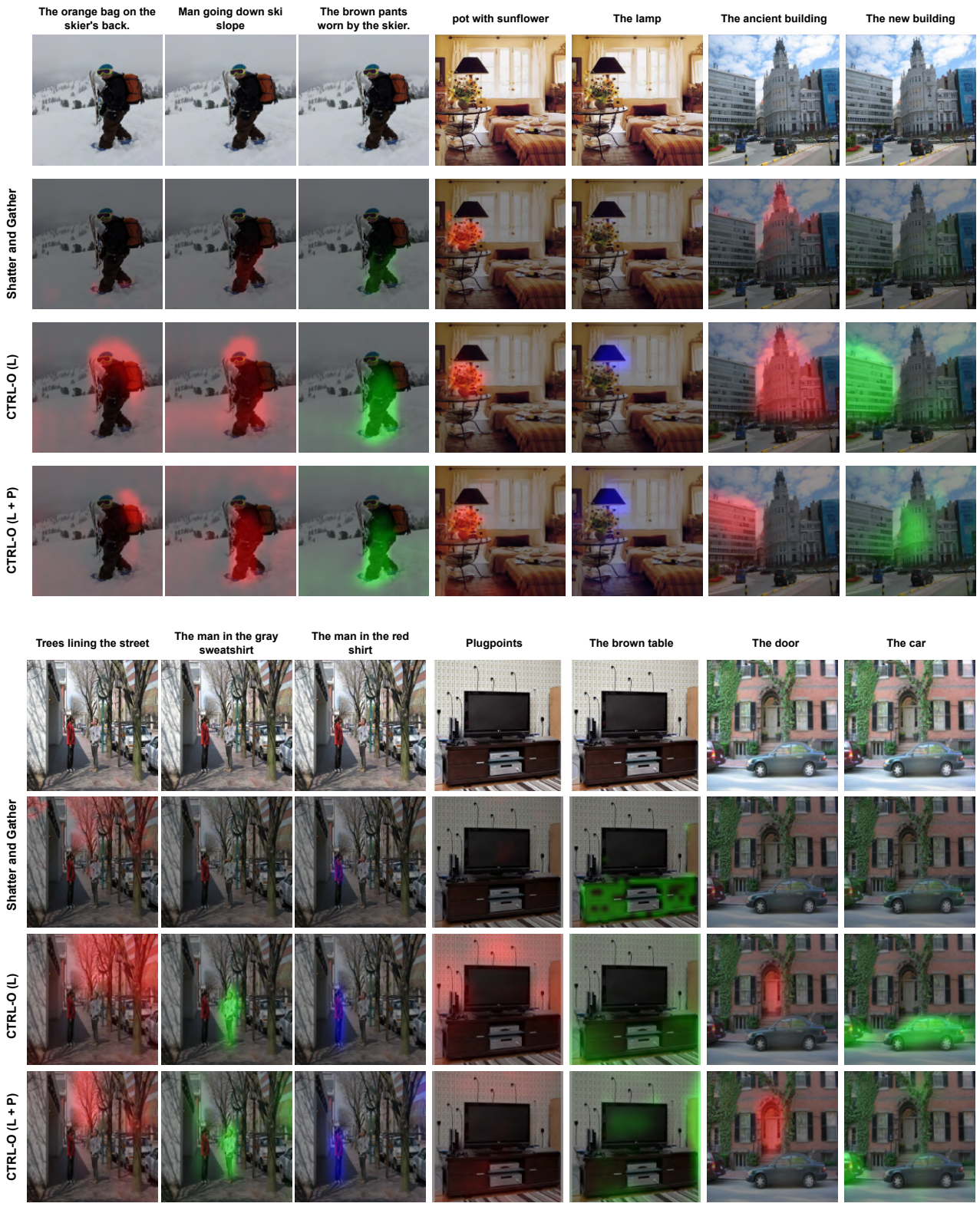

Figure 7. **Visualization Comparison** In this figure we visualize and compare the masks obtained using CTRL-O ($\mathcal{L} + \mathcal{P}$), CTRL-O ($\mathcal{L}$), and SaG when queried with free-form language queries. We can see that both the variants based on CTRL-O are significantly better at binding to the correct region descriptions as compared to SaG. This difference can be attributed CTRL-O using the language guidance to directly influence the slot extraction process while SaG considers the langauge to slot binding as a post-processing step after the slots have been extracted.

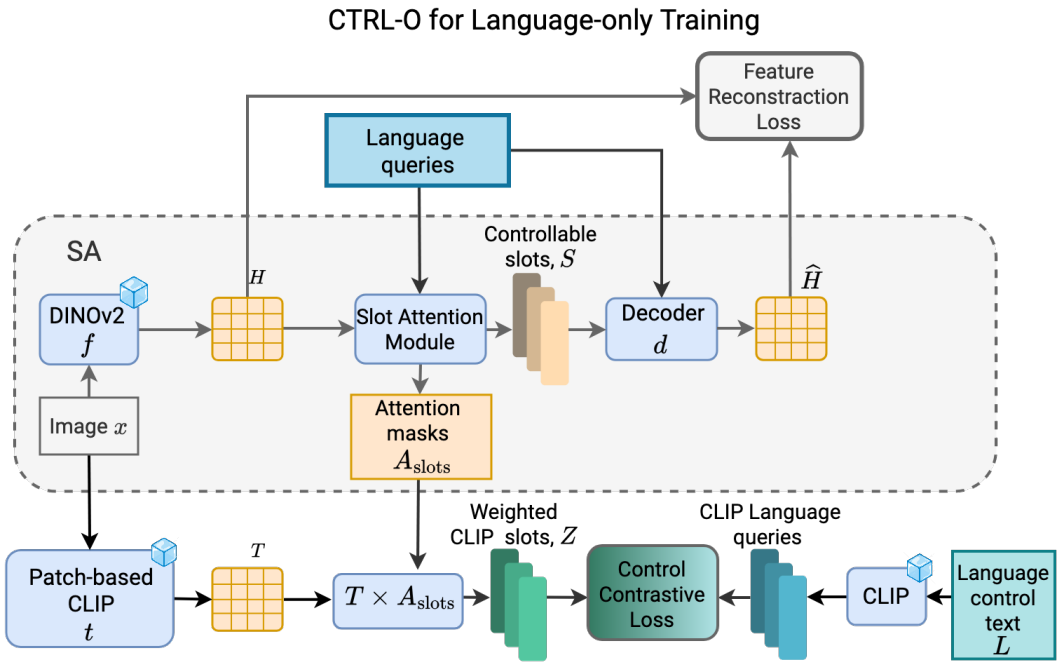

Figure 8. **Language-Only CTRL-O** Overview of language-only training. In this setting, we use the frozen CLIP model to compute both weighted CLIP slots and CLIP Language queries that we use in control contrastive loss. We average features from CLIP using attention weights from the Slot Attention module.

G. Additional Details of VQA Experiments

Evaluation Metric. We evaluate VQA models using classification accuracy across 3000 classes, using the top-frequent answers, which covers more than 90% of the question in the dataset.

Discussion on Coupling in CTRL-O VQA model The standard approach for solving VQA tasks with pretrained vision and language backbones is to feed the output representations of the vision model and the language model into a single neural network - usually a Transformer [45] - which then outputs a distribution over the answer categories [8, 30, 31]. To solve VQA, it is crucial to have strong interaction between the visual and language inputs. However, in pre-existing approaches, this interaction only happens in the output network (the Transformer that processes the language and vision outputs), which can be limiting.

To address this, we introduce an approach called *coupling*. Coupling, with the help of CTRL-O, directly inserts the visual representations into the language query, thus enabling strong vision and language interaction from the input stage. The proposed approach is presented in Figure 4(b).

H. CTRL-O SD

H.1. Fine-Tuning Details

In CTRL-O-SD, we finetune a pretrained Stable Diffusion model initialized from the `stabilityai/stable-diffusion-2-1` checkpoint. As illustrated in Figure 4(a), CTRL-O extracts slots from a given image based on user-provided queries. These extracted slots are then incorporated into the caption, which is fed into Stable Diffusion. Notably, CTRL-O remains frozen during the fine-tuning process, distinguishing our approach from prior works like Slot Diffusion [48] and Stable LSD [20], where the object-centric model and the diffusion model are trained jointly.

Implementation Details We train the model on the COCO 2017 training set. For each image, we first extract COCO categories from its associated caption and use these categories to query CTRL-O, to generate the corresponding slots. These slots are subsequently appended to the caption, as shown in Figure 4(a). The resulting caption is then passed through the CLIP language encoder to condition Stable Diffusion. To integrate CTRL-O outputs into the CLIP language embedding space, we introduce a learnable linear layer that maps the extracted slots to the CLIP embedding space. During training, the only components updated are the

U-Net parameters of Stable Diffusion. We use random flips as a data augmentation strategy. Training is performed for 300 epochs with a learning rate of 2×10^{-5}, utilizing gradient accumulation with 2 steps. Additionally, we reproduce Stable LSD using the author-provided code and hyperparameters on the COCO dataset. The input resolution to the vision encoder for CTRL-O is 224×224, while Stable LSD uses 448×448.

H.2. Image Generation Metrics

Fréchet Inception Distance (FID) score We calculate the Fréchet Inception Distance (FID) score [18] to assess the quality of generated images in comparison to real images. The FID score computes the Fréchet distance between feature distributions of generated and real images, extracted via an Inception v3 model. Lower FID scores indicate a closer match to real images, corresponding to higher image fidelity and diversity.

CLIP-I Score We use CLIP-I Score to verify whether the generated images contain the same instances present in the query image. This should be the expected behavior of CTRL-O-SD when conditioned on a caption containing slots corresponding to specific instances. We compute this metric on the COCO validation set. We embed the generated image and the query image into the CLIP embedding space using the CLIP ViT Encoder (`openai/clip-vit-base-patch16`). We then compute the cosine similarity between the two embeddings. This similarity is averaged across all images to compute the final CLIP-I Score.

H.3. Image Reconstruction Visualization

In this section, we present a qualitative analysis of the reconstruction capabilities of the LSD and CTRL-O-SD models. The goal is to evaluate how effectively these models retain structural and semantic details. LSD provides the full 7-slot representation derived from the object-centric model to the generative model, providing comprehensive image context for reconstruction. In contrast, CTRL-O-SD provides the caption along with only a subset of slots corresponding to the categories in the caption to the generative model. To obtain these slots, we condition the slots in CTRL-O with the categories present in the caption and append the corresponding slots to the caption. This flexibility in CTRL-O-SD enables instance-specific image generation (see Fig. 5 for examples), which is not feasible with Stable LSD. As illustrated in Fig. 9, both models demonstrate comparable reconstruction quality.

H.4. Image Generation Failures

In this section we highlight some failure cases of CTRL-O-SD.

- **Incorrect Focus:** The model occasionally fails to accurately prioritize the main objects in the query, often diverting attention to irrelevant elements. For instance, when prompted to generate an image centered around a cell phone, the model might emphasize a person in the background instead. As we have seen from Table 1, CTRL-O does not achieve perfect binding. Hence, this failure can be caused by the slots not binding to the correct regions in the image.
- **Deformed Outputs:** The model sometimes generates distorted representations of people and animals, with unnatural proportions or malformed features. Such deformities highlight limitations in the model's ability to represent detailed anatomy accurately, indicating a need for refined control over complex shapes and structures. This failure may also be attributed to the failures of the underlying generative model rather than CTRL-O.
- **Object Duplication:** There are instances where the model replicates objects within a single scene, leading to unrealistic and cluttered outputs.

These failure modes suggest areas for further improvement for CTRL-O and CTRL-O-SD, particularly in object binding for CTRL-O and image generation quality for CTRL-O-SD.

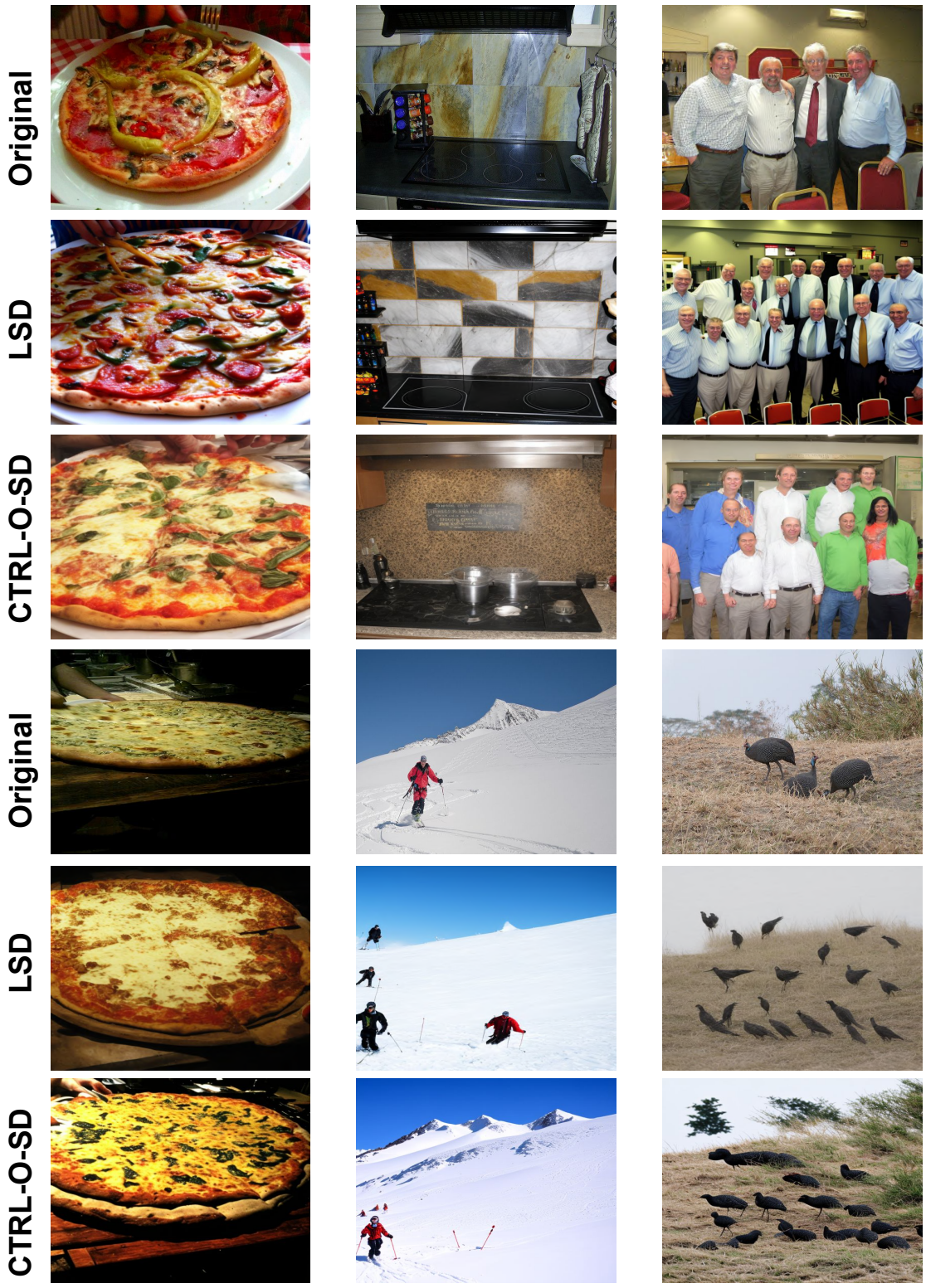

Figure 9. **Image Reconstruction** Qualitative comparisons of reconstruction outputs for LSD and CTRL-O-SD models. Each column corresponds to a different image. Rows correspond to original inputs, LSD generations, and CTRL-O-SD generations respectively. LSD generates outputs conditioned on full 7-slot representations derived from the original image, while CTRL-O-SD uses captions appended with a subset of slots for conditioning. The results show that both models achieve similar reconstruction quality.

Figure 10. **Failure Modes of CTRL-O SD.** Examples highlighting some failures in CTRL-O-SD such as incorrect focus, deformities in representations of people or animals, and object duplication. Each labeled box illustrates specific instances of these failures.