GLASS: Guided Latent Slot Diffusion for Object-Centric Learning

Supplementary Material

Krishnakant Singh[1] Simone Schaub-Meyer[1,2] Stefan Roth[1,2]
[1]Department of Computer Science, TU Darmstadt [2]hessian.AI

In this appendix, we give additional details for purposes of reproducible research and show further results that provide more insights into our proposed GLASS method. Our code is available at https://github.com/visinf/glass/.

A. Datasets

As in previous work [32, 62, 73], we report all our results on the VOC [21] and COCO [44] datasets. Both these datasets serve as popular benchmarks for object discovery and have been used to evaluate various object-centric learning methods on real-world images [32, 62, 73].

VOC. The PASCAL VOC dataset [21] is a standard dataset used in object discovery. We use the "trainaug" variant for generating the images and their corresponding mask. "trainaug" is an unofficial split of datasets, consisting of 10,582 images, which include 1,464 images from the original VOC segmentation train set and 9,118 images from the SBD dataset [82]. GLASS and GLASS† are evaluated on the official VOC validation set of 1,449 images

COCO. The COCO dataset [44] consists of 118,287 images of complex multi-object scenes. Unlike the VOC dataset, where images often contain only a single object in the scene, COCO images contain at least two objects, often even a dozen. GLASS and GLASS† are evaluated on a validation set with 5,000 COCO images.

Module	Hyperparameter	Value
General	Batch size	32
	Precision	fp16
	Learning rate: Phase 1	2e-5
	Learning rate: Phase 2	4e-5
	Optimizer	Adam
	Learning rate scheduler	Constant
Encoder	Architecture	DINOv2
	Patch size	14
	Backbone	ViT-B
	Embedding dimensions	768
Decoder-1	Architecture	Stable Diffusion
	Model version	2.1
Decoder-2	Architecture	MLP
	No. of layers	3
	Hidden dimensions	1536

Table 10. **Training details for GLASS and GLASS†.**

B. Training Details

Training dataset. GLASS and GLASS† are trained on images generated with a Stable Diffusion v2.1 [56] model. For training the model on the COCO-generated dataset, we generate 100K images and their corresponding pseudo ground-truth, obtained using COCO images and following the process in Sec. 2 of the main paper. For training the model on the VOC-generated dataset, we generate approximately 10K images and their corresponding pseudo masks, obtained using VOC images following the process in Sec. 2 of the main paper.

Architecture details. GLASS for the COCO and VOC datasets is trained on a single NVIDIA A6000 Ada GPU. The training time for COCO models is typically 4 days, while for the VOC dataset, training is completed within two days. GLASS and its variants use DINOv2 [51] with ViT-B [41] and a patch size of 14 as its encoder model, and Stable Diffusion (SD) v2.1 [56] as well as a three-layer MLP network as its decoder models.

We train our model on the generated images in two phases. In phase 1, only the slot attention module and the MLP decoder are trained with an Adam [84] optimizer with a constant learning rate of 2e-5. In phase 2, the slot attention and MLP decoder module are trained with a learning rate of 1e-8 (essentially frozen). At the same time, we train the diffusion decoder with a learning rate of 4e-5 for the last 100K iterations.

Table 10 shows additional common details about the hyper-parameters and modules used in GLASS and GLASS†. When training on COCO-generated images, we train the model for 500K iterations, while when training on VOC-generated images, we train for 250K iterations. For the slot-attention module, the number of slot iterations in the GRU module is set to 5, and the number of slots is set to 7 for the COCO-generated and the VOC-generated dataset for GLASS. The slot size is set to 768; this configuration is akin to StableLSD [32]. The number of heads in the slot-attention module is set to 1, and a hidden size of 768 is used for the MLP. The final MLP layer in the slot-attention module projects the slots to a dimension of 1024.

Pseudo labels. While generating each training image, we used Stable Diffusion's cross-attention and self-attention modules to extract the pseudo masks for the respective im-

Model	FG-ARI (%, ↑)	
	VOC	COCO
SA* [45] NeurIPS'20	12.3	21.4
SLATE* [64] ICLR'22	15.6	32.5
DINOSAUR-MLP [62] ICLR'23	**24.6**	**40.6**
DINOSAUR-Trans. [62] ICLR'23	23.1	35.2
SPOT [36] CVPR'24	20.9	36.5
SlotDiffusion* [73] NeurIPS'23	17.8	37.2
StableLSD [32] NeurIPS'23	8.7	28.9
GLASS† (ours)	21.3	32.5
GLASS (ours)	22.5	34.1

Table 11. **Comparison between OCL methods for the FG-ARI metric.** Here, our method is *only* behind DINOSAUR on the VOC dataset and performs close to SPOT on the COCO dataset. Please note that the FG-ARI metric is unreliable as it does not take into account the shape of the predicted mask and also ignores the background, making it unsuitable for object discovery as seen by the results in Fig. 10. * denotes numbers taken from [36].

age (as described in Sec. 4 of the main paper; we used the same setup as in [87]). We used a range-based thresholding to binarize these masks. Specifically, we assign a pixel to the background if its objectness score (the max value among all class scores of $\mathbf{M}_{\text{ref}}$, see Sec. 4) is below 0.4; conversely, if this objectness score is above 0.6, we assign the pixel to the foreground with the class label that has the max value. If the objectness score lies between 0.4 and 0.6, we assign a pseudo label of 255 (indicating that we are uncertain about the class label). This uncertain region helps in only calculating the semantic loss on regions with high certainty, avoiding uncertain regions.

B.1. Object-level property prediction

For the object-level property prediction task, we train a single-layer linear model for label prediction and a single-layer linear network for position prediction. We use early stopping with a patience level set to 5 and train for 50 epochs on the VOC dataset and 10 epochs on the COCO training sets. For matching the labels to the correct slots during training, we utilize the idea from [15] and use the mIoU criterion for matching labels to slots. Since the VOC training dataset does not have instance masks, we compute the IoU criterion using the semantic mask during training [62]. We use an AdamW [86] optimizer with a constant learning rate of 3e-4. For the position prediction task, we normalize the image coordinates to lie between 0 and 1 by dividing the image coordinates by the image size.

C. Comparison with FG-ARI Scores

Previous work in object-centric learning has regularly considered the FG-ARI metric for evaluation. The FG-ARI metric is a version of the adjusted Rand index (ARI) [83, 88], which measures the similarity between two different clusterings in a permutation-invariant way by taking into account the foreground regions in the image. However, the FG-ARI score is known to be an unreliable metric as discussed in [18, 32, 36, 37]; it does not take into account background pixels and does not account for the shape of predicted masks. Please see [36] for further discussion. For completeness, we nevertheless provide results for the FG-ARI metric in Tab. 11. While our FG-ARI scores are lower than some baselines, particularly on the COCO dataset, we believe that this should be mostly discounted due to the known deficiencies of this evaluation metric. That said, our method is *only* behind DINOSAUR on the VOC dataset and performs close to SPOT on the COCO dataset for the FG-ARI metric. Additionally, we refer to the comprehensive results for the mIoU and mBO metrics in Tab. 2 and the SO-PO-GO metrics in Tab. 3 of the main paper.

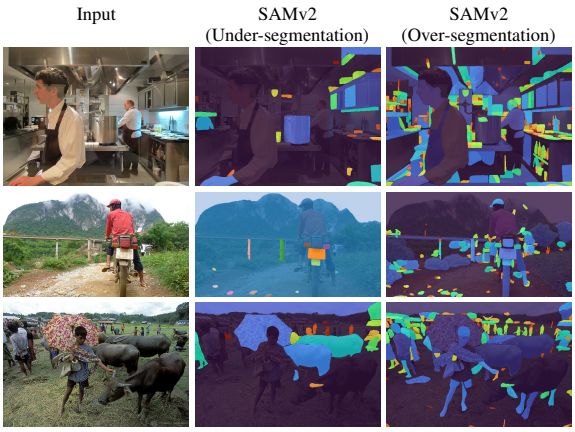

Figure 7. **Issue with SAMv2 masks as guiding signal.** Automatic segmentation using SAMv2 is very sensitive to the choice of hyperparameters, which makes it suffer from over- or under-segmentation issues. Hence, automatic SAMv2 masks are not ideal as guidance signal.

D. Guidance with SAMv2 Masks

Automatic segmentation using the SAMv2 model [55] requires careful tuning of many hyperparameters. Using default parameters results in severe under-segmentation issues (*cf*. Fig. 7, where the humans are segmented as backgrounds). If we use a denser sampling of points, this results in an over-segmentation of objects into their parts, see Fig. 7. These issues make plain SAMv2 segmentation masks unsuitable as guiding signals. We could overcome these issues with additional input prompts, such as bounding boxes, but this would make the annotation cost higher than simply using generated captions or image-level labels.

E. Compositional Generation with StableLSD

StableLSD [32] struggles with compositional generation, as shown in Fig. 8. StableLSD is not able to add or remove objects from the original image; moreover, the quality of

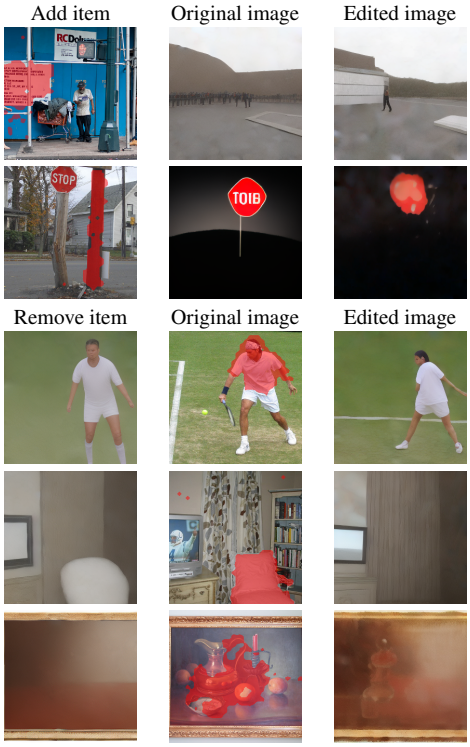

Figure 8. **Compositional generation with StableLSD [32].** The quality of the image reconstruction is rather poor for StableLSD. Moreover, the approach does not always remove the annotated item from the image *(row 3 and 5)*. Adding of an item to a new scene also results in failure as is the case with *(row 2)*. There is some compositionality, which is exhibited by *(rows 1 and 4)*, but the quality of image reconstruction is poor and the edited image is not faithful to the original image.

Figure 9. **Qualitative comparison for concept composition with Composable Diffusion [85].** GLASS is able to extract and compose concepts/slots from a source (Src.) and transfer them to a destination image (Dst.) image to generate an image containing concepts/slots from both images. We compare this to text-based compositional models [85], where we compose the labels from the two images, *(row-1)* "Pizza and Person", *(row-2)* "BOWL AND BOWL AND BANANA", *(row-3)*, "CUP AND CAKE AND SANDWICH", and *(row-4)* "CAT AND SINK" to generate an image containing both concepts. As seen, our results have a higher fidelity, more realism, and are more faithful to the concepts given.

reconstruction and faithfulness of the input image is rather poor for StableLSD compared to results of our method, as shown in Figs. 12 and 13.

F. Additional Results

Additional qualitative results for object discovery. Fig. 10 shows additional qualitative results for object discovery. GLASS decomposes the scene more cleanly and meaningfully than SotA OCL methods such as DINOSAUR [62], StableLSD [32], and SPOT [36]. Images segmented by GLASS have more precise boundaries, the background segmentation is much cleaner, and the segmented regions do not split or group objects.

Additional comparison to DatasetDiffusion. Going beyond Tab. 9 in the main paper, we additionally compare GLASS to an extended variant of DatasetDiffusion [50], which is trained with our 100K generated images for the COCO dataset and uses a DINOv2 backbone network. This extended variant of [50] achieves an mIoU_c of 42.8 %, while GLASS-Sem still achieves a higher score of 46.7 % on the COCO dataset. This suggests that our proposed architecture and training scheme contribute significantly to the gains over DatasetDiffusion [50].

Additional conditional generation results. Fig. 11 shows additional results for the conditional generation of images using StableLSD, GLASS†, and GLASS. Our method generates images that are more faithful to the input image and have higher fidelity than StableLSD.

Additional compositional generation results. Figs. 12 and 13 show results from GLASS for compositional generation. In particular, we can see that we can add objects from one scene to another. This is possible even when the context is quite different, for example, adding a baseball player to the bowl of food (Fig. 13 *(row-1)*). Also, we can remove objects completely from a scene. Please note that, to our knowledge, no other OCL method can perform these actions with this fidelity or faithfulness.

Comparison to text-based compositional models. We conduct a preliminary study of comparing GLASS for compositional generation against a text-based compositional generation method, namely Composable Diffusion [85], which

Figure 10. **Qualitative comparison for object discovery.** GLASS and GLASS† can decompose an image at the object level and do not split an object into its parts or group objects belonging to the same class. Also, our approach yields cleaner boundaries for the foreground objects compared to DINOSAUR [62], StableLSD [32], and SPOT [36].

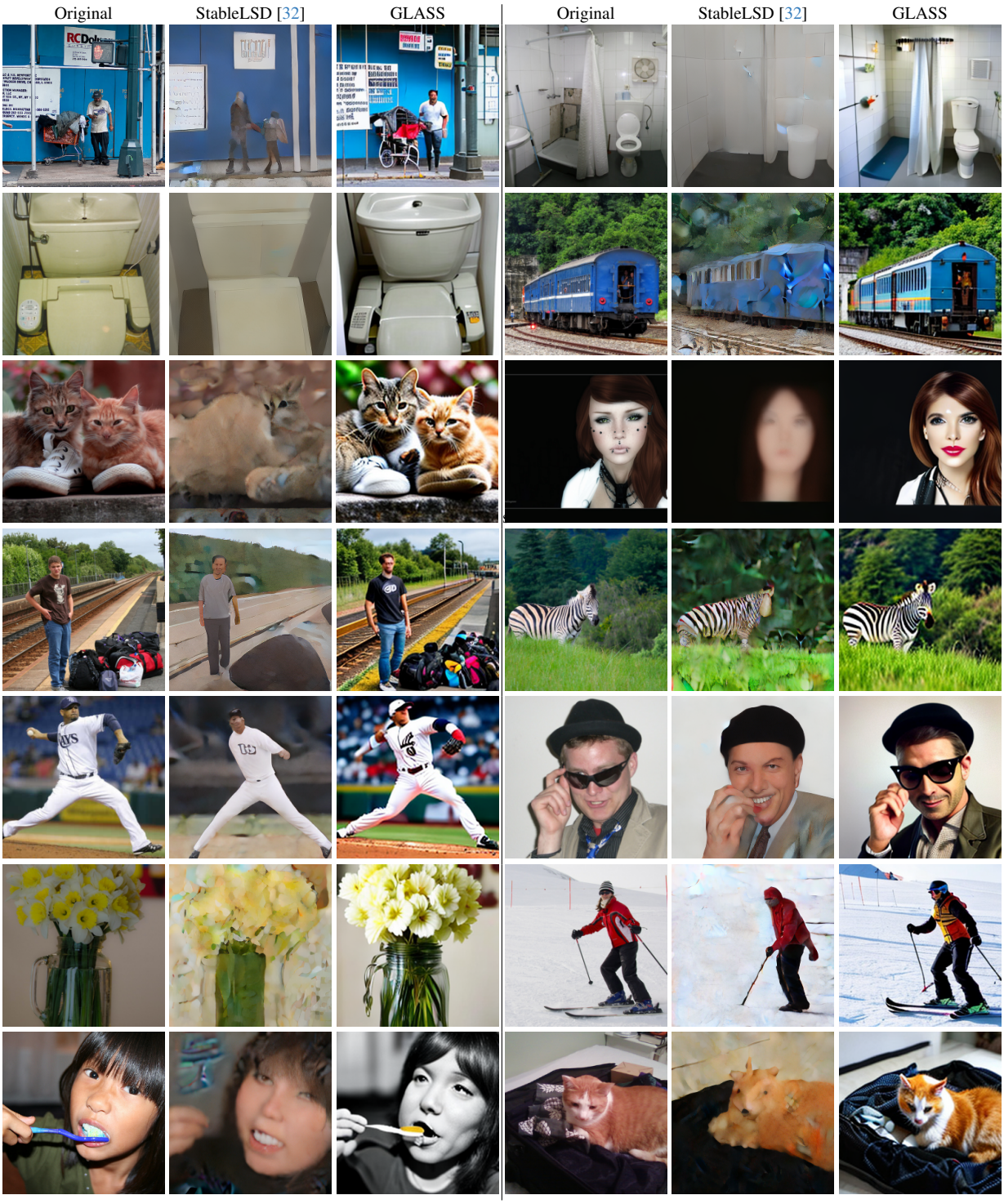

Figure 11. **Qualitative comparison for conditional image generation.** GLASS and GLASS† not only learn to decompose scenes meaningfully, but the learned slot can reconstruct the input scene more faithfully and with higher fidelity than StableLSD [32].

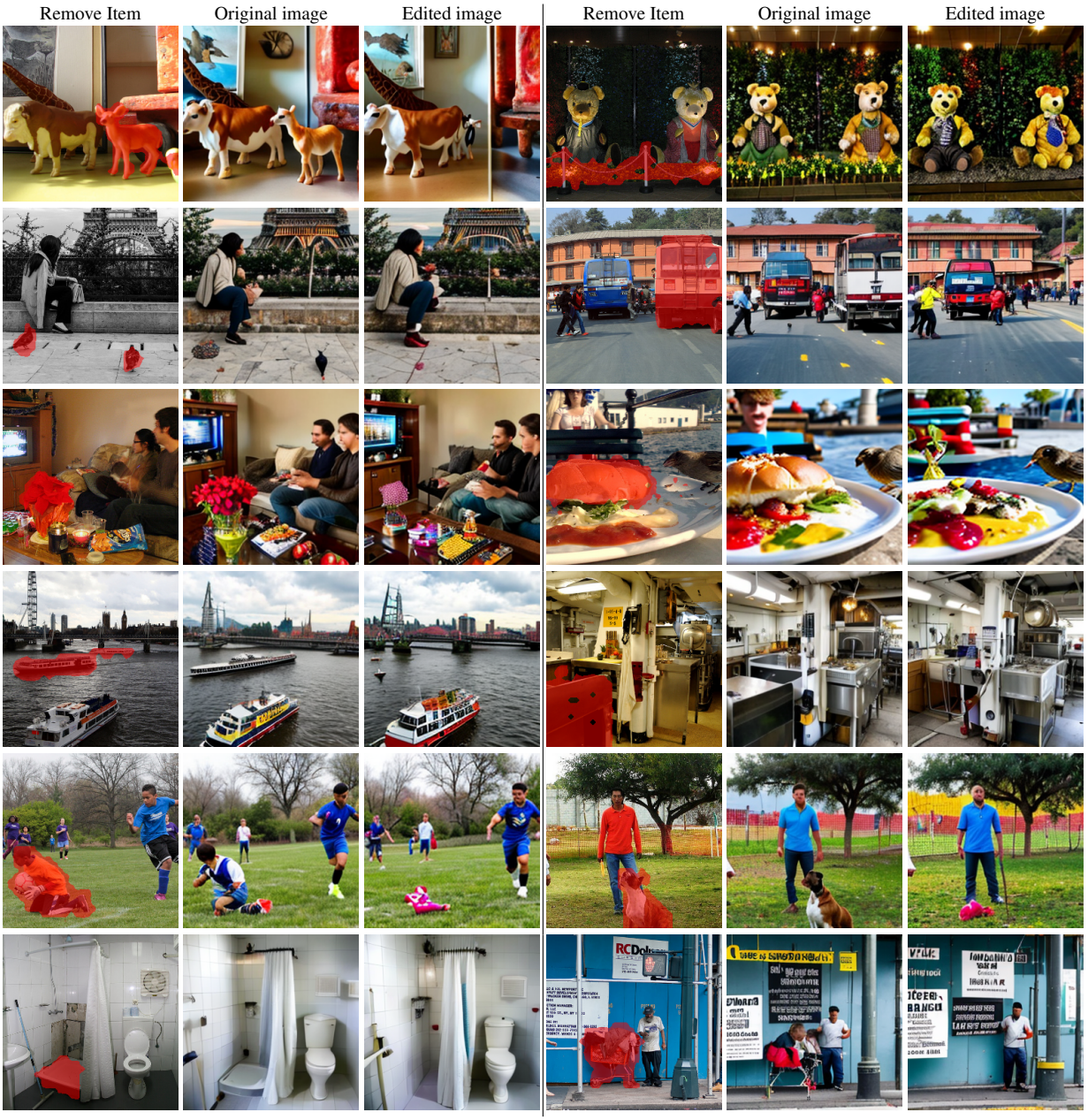

Figure 12. **Compositional generation.** GLASS enables the compositional image generation of real-world complex scenes. Here, the masked object *(in red)* is the slot to be removed from the original image. The original image is the reconstructed image from the slots of the input image.

Figure 13. **Compositional generation.** GLASS enables the compositional image generation of real-world complex scenes. Here, the masked object *(in red)* is the slot to be added (Added item) to the original image resulting in the edited image.

Num. of slots	$mIoU_i$	mBO_i
7	**38.9**	**40.6**
14	27.0	28.0
21	24.2	25.3

Table 12. **Effect of number of slots on GLASS.** As with all slot-attention methods, GLASS is also sensitive to the number of slots used in the slot-attention module.

Caption type	$mIoU_i$	mBO_i
BLIP-2	38.9	40.6
ShareGPT-4V	38.3	40.7
Template-based	**40.0**	**41.2**

Table 13. **Effect of caption generation method on GLASS.** We find that our method is robust to the choice of language module to generate the captions. Interestingly, if we can access a ground-truth object occurrence statistics dataset, a template-based caption scheme outperforms learnt language-based methods.

composes objects/concepts via text-based prompts. For example, "class label 1 — class label 2" generates an image containing class labels 1 and 2. On the other hand, GLASS is a compositional method that first extracts objects/concepts from an image and then generates the image. **Note**: Both these models try to address the compositional generation problem. However, they are not directly comparable as GLASS relies on input images for extracting concepts while Composable Diffusion uses the text. Fig. 9 shows that GLASS can extract concepts/slots from one image and transfer them to another to create a high-fidelity image, which faithfully contains both concepts from the source and destination image. In contrast, [85] sometimes is unable to compose certain concepts given in the text prompt, *e.g.*, in Fig. 9 (*top-row*) no person is generated for the prompt "PERSON AND PIZZA".

G. Additional Ablations

Effect of number of slots. We test the dependence of GLASS on the number of slots in the slot-attention module. GLASS is normally trained with 7 slots, consistent with previous OCL models. Tab. 12 additionally shows results for instance-aware object discovery task when GLASS is trained with 14 and 21 slots. As seen, increasing the number of slots leads to a decrease in $mIoU_i$ and mBO_i, indicating that the model cannot segment the objects correctly. This is because if the number of objects is larger than the number of objects in the scene, the scene is over-segmentated. The larger the number of slots, the more slots bind to object parts.

Effect of caption-generation module. Table 13 shows an analysis of the effect of the caption generation module on the instance-aware object discovery task. GLASS uses a BLIP-2 [43] model. We also test the performance against a more powerful captioning model, namely ShareGPT-4V [81], and a simple template-based pipeline.

In the template-based pipeline, we first determine the empirical probability of each class appearing in the image using ground-truth class labels from the COCO dataset and the empirical probability of a certain number of objects appearing in a COCO image. Following this, we first sample the number of objects and then sample class labels from the learnt object occurrence distribution. After this, we populate the standard template "A high-quality image of <obj(i)>, <obj(i+1)> ... <obj(k)>; <obj(i)> <obj(i+1)> ... <object (k)>" using the class labels of the sampled objects. **Note:** For this template-based pipeline, no input image is needed for caption generation. However, the dataset statistics in terms of the probability of occurrence of the objects and the probability of a certain number of objects in an image are required.

We find that our approach is not very sensitive to the particular choice of captioning model. Interestingly, the template-based approach slightly outperforms both captioning models, showing that we largely need to ensure that the generated images possess the appropriate object occurrence statistics.

References

[81] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Conghui He, Jiaqi Wang, Feng Zhao, and Dahua Lin. ShareGPT4V: Improving large multi-modal models with better captions. In *ECCV*, pages 370–387, 2024.

[82] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, pages 991–998, 2011.

[83] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, pages 193–218, 1985.

[84] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[85] Nan Liu, Shuang Li, Yilun Du, Antonio Torralba, and Joshua B. Tenenbaum. Compositional visual generation with composable diffusion models. In *ECCV*, 2022.

[86] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in Adam. In *ICLR*, 2017.

[87] Quang Ho Nguyen, Truong Tuan Vu, Anh Tuan Tran, and Khoi Nguyen. Dataset Diffusion: Diffusion-based synthetic data generation for pixel-level semantic segmentation. In *NeurIPS*, pages 76872–76892, 2023.

[88] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, pages 846–850, 1971.