A Dataset Documentation: Datasheets for Datasets

Here we answer the questions outlined in the datasheets for datasets paper by Gebru et al. [21].

A.1 Motivation

For what purpose was the dataset created? CLEVRTEX was created to serve as the next challenging benchmark for unsupervised multi-object segmentation methods. It trades simpler visuals for confounding aspects such as texture, irregular shapes, and a variety of materials.

Who created the dataset (e.g., which team, research group) and on behalf of which entity (e.g., company, institution, organisation)? The dataset has been constructed by the research group "Visual Geometry Group" at the Engineering Science Department, University of Oxford.

Who funded the creation of the dataset? The dataset is created for research purposes at VGG. L. K. is funded by EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems EP/S024050/1. I. L. is supported by the EPSRC programme grant Seebibyte EP/M013774/1 and ERC starting grant IDIU-638009. C. R. is supported by Innovate UK (project 71653) on behalf of UK Research and Innovation (UKRI) and by the European Research Council (ERC) IDIU-638009.

A.2 Composition

What do the instances that comprise the dataset represent (e.g., documents, photos, people, countries)? The dataset consists of images featuring simulated scenes and segmentation, depth, normal, albedo, and shadow masks available, and metadata detailing scene composition.

How many instances are there in total (of each type, if appropriate)? There are 50 000 instances in the main CLEVRTEX dataset. 20 000 in each variant, PLAINBG, VARBG, GRASSBG and CAMO. There is also a further 10 000 instances in the testing-only variant OOD.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? The dataset is a sample of the near-infinite set of possible arrangements under our sampling distribution. Please see Section 3.1 for a description of the process to sample the scene.

What data does each instance consist of? Each instance consists of the RGB scene image, depth, normal, albedo, and shadow masks (all PNG), and further metadata (JSON) detailing object positions, shapes, scales, and materials used. We use only the RBG image for training during the benchmarking process and segmentation masks and metadata to evaluate.

Is there a label or target associated with each instance? For the task explored in this paper, unsupervised multi-object segmentation, the target labels are the segmentation masks, which are not used during training.

Is any information missing from individual instances? No.

Are relationships between individual instances made explicit (e.g., users' movie ratings, social network links)? No, there are no relationships between different instances.

Are there recommended data splits (e.g., training, development/validation, testing)? Yes, we adopt 10%/10%/80% test/val/train splits for the datasets by instance index, with the exception of OOD variant, which is used for evaluation only. The rationale behind splits is that the data comes from the same generation process for each variant and can already be considering randomized. Simply using an image index to separate the splits makes both data-loading easy and removes the need to distribute canonical split indexes.

Are there any errors, sources of noise, or redundancies in the dataset? No.

Is the dataset self-contained, or does it link to or otherwise rely on external resources (e.g., websites, tweets, other datasets)? The dataset is self-contained.

Does the dataset contain data that might be considered confidential (e.g., data that is protected by legal privilege or by doctor-patient confidentiality, data that includes the content of individuals' non-public communications)? No.

Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety? No.

Does the dataset relate to people? If not, you may skip the remaining questions in this section. No.

Does the dataset identify any subpopulations (e.g., by age, gender)? NA

Is it possible to identify individuals (i.e., one or more natural persons), either directly or indirectly (i.e., in combination with other data) from the dataset? NA

Does the dataset contain data that might be considered sensitive in any way (e.g., data that reveals racial or ethnic origins, sexual orientations, religious beliefs, political opinions or union memberships, or locations; financial or health data; biometric or genetic data; forms of government identification, such as social security numbers; criminal history)? NA

A.3 Collection process

How was the data associated with each instance acquired? The data was generated.

What mechanisms or procedures were used to collect the data (e.g., hardware apparatus or sensor, manual human curation, software program, software API)? The images were rendered using Blender 2.9.3 software on generic systems.

If the dataset is a sample from a larger set, what was the sampling strategy (e.g., deterministic, probabilistic with specific sampling probabilities)? See the similar question in the Composition section.

Who was involved in the data collection process (e.g., students, crowdworkers, contractors) and how were they compensated (e.g., how much were crowdworkers paid)? The authors were involved in the process of generating this dataset.

Over what timeframe was the data collected? The datasets were rendered over a period of several weeks.

Were any ethical review processes conducted (e.g., by an institutional review board)? No.

Does the dataset relate to people? If not, you may skip the remainder of the questions in this section. No.

A.4 Preprocessing/cleaning/labeling

Was any preprocessing/cleaning/labeling of the data done (e.g., discretization or bucketing, tokenization, part-of-speech tagging, SIFT feature extraction, removal of instances, processing of missing values)? No, the dataset was generated together with labels.

Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data (e.g., to support unanticipated future uses)? NA

Is the software used to preprocess/clean/label the instances available? NA

A.5 Uses

Has the dataset been used for any tasks already? In the paper we show and benchmark the intended use of this dataset for unsupervised multi-object segmentation setting.

Is there a repository that links to any or all papers or systems that use the dataset? We will be listing these on the website.

What (other) tasks could the dataset be used for? We include additional information maps when generating this dataset, which could be used for exploring value of using extra modalities for supervision or as targets. As mentioned before, we also generated necessary metadata for CLEVR-like QA task.

Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses? No.

Are there tasks for which the dataset should not be used? This dataset is meant for research purposes only.

A.6 Distribution

Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created? No.

How will the dataset will be distributed (e.g., tarball on website, API, GitHub)? The dataset and related evaluation code is available on the website https://www.robots.ox.ac.uk/~vgg/research/clevrtex/ allowing users to download and read-in the data.

When will the dataset be distributed? The dataset is available now.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? CC-BY.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? The original textures used in rendering objects are copyrighted by Poliigon Pty Ltd and cannot be redistributed to a third party. This only applies to texture images used in creating this dataset. The materials used for main dataset are freely available under non-commercial license and we include instructions to retrieve them alongside the generation code. Textures used in evaluation-only OOD variant are not available free of charge (we obtained them under a commercial license), but their catalogue is similarly included with the code. The dataset instances themselves do not have IP-based restrictions.

Do any export controls or other regulatory restrictions apply to the dataset or to individual instances? Not that we are are of. Regular UK laws apply.

A.7 Maintenance

Who is supporting/hosting/maintaining the dataset? The dataset is supported by the authors and by the VGG research group. The main contact person is Laurynas Karazija.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)? The authors of this dataset can be reached at their e-mail addresses: *{laurynas,chrisr,iro}@robots.ox.ac.uk*.

Is there an erratum? If errors are found and erratum will be added to the website.

Will the dataset be updated (e.g., to correct labeling errors, add new instances, delete instances)? Any potential future updates or extension will be communicated via the website. The dataset will be versioned.

If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances (e.g., were individuals in question told that their data would be retained for a fixed period of time and then deleted)? NA

Will older versions of the dataset continue to be supported/hosted/maintained? We plant to continue hosting older versions of the dataset.

If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so? Yes, we make the dataset generation code available.

A.8 Other questions

Is your dataset free of biases? Yes.

Can you guarantee compliance to GDPR? No, we are unable to comment on legal issues.

A.9 Author statement of responsibility

The authors confirm all responsibility in case of violation of rights and confirm the licence associated with the dataset and its images.

B Dataset

The dataset can be accessed at https://www.robots.ox.ac.uk/~vgg/research/clevrtex. In CLEVRTEX and its variants, each instance contains:

1. RBG scene image
2. semantic mask image
3. depth mask image
4. shadow mask image
5. albedo mask image
6. normal mask image
7. Metadata JSON, which further details:
   (a) number of objects
   (b) background material
   (c) shape of each object
   (d) size of each object
   (e) rotation of each object
   (f) scene (3D) coordinates of each object
   (g) image (2D) coordinates of each object
   (h) material of each object
   (i) color (only relevant on VARBG) of each object
   (j) scene directions (CLEVR metadata)
   (k) object relationships (CLEVR metadata)

All images are provided as PNG. We also provide code for reading in the dataset and evaluation utilities for general performance metrics and per-shape/material/size breakdown. The dataset is provided under the CC-BY license.

C Supplementary Material

C.1 Data

All images are center-cropped to a 192×192 patch and further downsampled to 128×128 pixels as a pre-processing step before being fed to the models. This introduces partially visible objects in the datasets, removes uninteresting empty edges of the scenes, and lowers the computational load. Many of the benchmarked models were developed to work with such resolution. We include helper code to load our datasets for convenience. For CLEVR we are using a version that includes segmentation masks for evaluation[5], for which we adopt the standard 70k/15k/15k train/validation/test splits.

C.2 Metrics

As previously mentioned, prior work [15, 24, 40] evaluated using the adjusted Rand index (ARI) metric calculated only on pixels that correspond to the foreground objects, filtered using ground-truth data. We share the concern of some authors [15, 44] that such evaluation protocol does not account for whether objects are considered a part of the background and how well models segment object boundaries. Instead, we opt for the mIoU metric, familiar from the supervised segmentation setting. The predicted objects are matched with ground truth segments using the Hungarian matching algorithm, which assigns only a single predicted component to each true mask, maximizing overall overlap. A mean is taken over all objects, including the background. We provide side-by-side comparison of these metrics on all benchmarked models in Tables 5 and 6. We chose mIoU in favor of ARI metric, as it weights all objects equally irrespective of their size. ARI is based on counting pairs, thus it gives larger regions such backgrounds more weight.

[5] Available at https://github.com/deepmind/multi_object_datasets.

Table 5: Benchmark results on CLEVR, CLEVRTEX, CAMO, and OOD comparing ARI-FG and mIoU metrics. Results are shown ($\pm\sigma$) calculated over 3 runs.

Model	CLEVR		CLEVRTEX		OOD		CAMO	
	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)
▣ SPAIR* [12]	77.13 ± 1.92	65.95 ± 4.02	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
▣ SPACE [38]	22.75 ± 14.04	26.31 ± 12.93	17.53 ± 4.13	9.14 ± 3.46	12.71 ± 3.44	6.87 ± 3.32	10.55 ± 2.09	8.67 ± 3.50
▣ GNM [30]	65.05 ± 4.19	59.92 ± 3.72	53.37 ± 0.67	42.25 ± 0.18	48.43 ± 0.86	40.84 ± 0.30	15.73 ± 0.89	17.56 ± 0.74
▣ MN [51]	72.12 ± 0.64	56.81 ± 0.40	38.31 ± 0.70	10.46 ± 0.10	37.29 ± 1.04	12.13 ± 0.19	31.52 ± 0.87	8.79 ± 0.15
▣ DTI [44]	89.54 ± 1.44	48.74 ± 2.17	79.90 ± 1.37	33.79 ± 1.30	73.67 ± 0.98	32.55 ± 1.08	72.90 ± 1.89	27.54 ± 1.55
⊞ GenV2 [16]	57.90 ± 20.38	9.48 ± 0.55	31.19 ± 12.41	7.93 ± 1.53	29.04 ± 11.23	8.74 ± 1.64	29.60 ± 12.84	7.49 ± 1.67
⊞ eMORL [14]	93.25 ± 3.24	50.19 ± 22.56	45.00 ± 7.77	12.58 ± 2.39	43.13 ± 9.28	13.17 ± 2.58	42.34 ± 7.19	11.56 ± 2.09
⊞ MONet [6]	54.47 ± 11.41	30.66 ± 14.87	36.66 ± 0.87	19.78 ± 1.02	32.97 ± 1.00	19.30 ± 0.37	12.44 ± 0.73	10.52 ± 0.38
⊞ SA [40]	95.89 ± 2.37	36.61 ± 24.83	62.40 ± 2.23	22.58 ± 2.07	58.45 ± 1.87	20.98 ± 1.59	57.54 ± 1.01	19.83 ± 1.41
⊞ IODINE [24]	93.81 ± 0.76	45.14 ± 17.85	59.52 ± 2.20	29.17 ± 0.75	53.20 ± 2.55	26.28 ± 0.85	36.31 ± 2.57	17.52 ± 0.75

Table 6: Results on PLAINBG,VARBG, and GRASSBG variants, comparing ARI-FG and mIoU metrics.

Model	PLAINBG		VARBG		GRASSBG	
	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)	↑ARI-FG (%)	↑mIoU (%)
▣ SPAIR* [12]	51.75	39.32	0.05	0.00	0.00	0.00
▣ SPACE [38]	34.25	31.96	29.36	16.10	32.52	33.85
▣ GNM [30]	40.73	26.49	66.79	49.78	67.31	53.15
▣ MN [51]	38.34	10.16	43.64	11.51	59.79	34.80
▣ DTI [44]	77.74	36.03	81.56	38.82	82.37	37.65
⊞ GenV2 [16]	85.33	24.39	66.04	14.40	21.12	2.88
⊞ eMORL [14]	52.00	29.39	50.18	22.92	69.64	19.38
⊞ MONet [6]	57.10	38.72	51.87	23.73	37.97	21.29
⊞ SA [40]	51.75	39.32	89.78	62.57	43.55	12.88
⊞ IODINE [24]	54.32	23.83	75.33	39.86	66.91	25.76

C.3 Hyper-parameters

Where available in PyTorch, we use the official implementation for the benchmarked methods. Otherwise, we use a re-implementation, checked against the original method, and further verify that it produces similar results to those reported in the corresponding papers. Where the original methods have been applied to CLEVR (or its variant), we employ the same hyper-parameter configuration for CLEVR. For other datasets or methods that have not been trained on CLEVR, we follow a best-effort approach to tuning hyper-parameters.

For MONet [6], we reduced the batch size from 64 to 63 (3×21). IODINE [24] and MONet were trained for 300k iterations instead of 1M as we noticed that no changes to learned configurations, running loss, or performance improvements were taking place after 250k iterations. For MONet, IODINE we found the original configuration worked well enough. For SPACE [38], we concentrated on finding a suitable setting for output standard deviation for foreground and background networks. Despite higher values being crucial for both Genesis and GNM models, we could not identify a configuration that produced better results than the original 0.15 in our exploration. The following describes any adjustments made to the original configurations for other models.

Slot Attention [40] We use 11 slots on all tests. We varied the number of attention iterations. We have found the model to perform the best when trained using 3. We maintained the original learning rate, batch size, and optimizer settings and trained for the suggested 500k iterations.

Efficient MORL [14] We increase the number of components to 11 and change the input resolution to 128×128 to be inline with other methods studied. GECO reconstruction target is further adjusted to account for change in resolution. We use the value of $-108\,000$ for CLEVR and PLAINBG. We use higher values of $-61\,000$ for VARBG and GRASSBG and $-73\,000$ for CLEVRTEX, OOD, and CAMO, due to more complex backgrounds. We considered a set of $\{-8\,000, -48\,000, -61\,000, -69\,000, -73\,000, -108\,000, -112\,000\}$, selecting the best performing ones. eMORL†: following the release of CLEVRTEX, the codebase of eMORL has been updated including configuration settings for CLEVRTEX. The authors provided us with trained models that show better performance (Table 3) in our evaluation.

GNM [30] We use a 4×4 slot grid with total of 16 slots and a latent dimension of 64 for objects and 10 for background. We found the model extremely sensitive to the output standard deviation.

We found values 0.2 on CLEVR and 0.5 on CLEVRTEX worked well. It is worth noting that in our testing, with values of 0.4 and 0.6, GNM could not learn to segment the scene. We trained for 300k iteration.

GenesisV2 [16] We focused our hyper-parameter selection on the output standard deviation and GECO [47] objective. On CLEVR we used GECO goal of 0.5655 and output standard deviation of 0.7, which was crucial for model to learn as lower values did not produce good segmentations. On CLEVRTEX we lowered the GECO goal to 0.5, which outperformed CLEVR setting.

SPAIR* [12] As mentioned before, we incorporated a background VAE network into SPAIR by using a convolutional encoder and a spatial broadcast decoder [55]. We also replaced MLP-based glimpse decoder with a similar spatial broadcast decoder. Additionally, we added an extra convolution in the backbone network to handle inputs of 128×128 size. In this configuration, SPAIR had 16 slots. We set the latent dimension of objects to 64, and background to 1 on CLEVR and 4 on CLEVRTEX. We trained for 250k iterations using a batch size of 128, Adam optimizer, learning rate of 1e-4, with gradient clipping when norm exceeded 1.0. We used β value of 2.7. On CLEVR we used the output standard deviation of 0.15. On CLEVRTEX, we annealed the value from 0.5 to 0.15 over 50k iterations. On CLEVR, the object presence prior hyper-parameter s was annealed from 0.0001 to 0.99 over 10k, on CLEVRTEX, over 50k iterations.

Table 7: Architecture of component networks changed in SPAIR*.

Conv Encoder

Layer	Size/Ch.	Act.	Comment
Conv 3×3	32	ReLU	stride 2
Conv 3×3	32	ReLU	stride 2
Conv 3×3	64	ReLU	stride 2
Conv 3×3	64	ReLU	stride 2
Avg P 1×1			
MLP	128	ReLU	
MLP	$\|\|\mu\|\| + \|\|\sigma\|\|$	Softplus for σ only	

Broadcast Decoder

Layer	Size/Ch.	Act.	Comment
Broadcast			add coord.
Conv 3×3	32	ReLU	no pad
Conv 3×3	32	ReLU	no pad
Conv 3×3	32	ReLU	no pad
Conv 3×3	32	ReLU	no pad
Conv 1×1	4	Sigmoid for masks only	

DTISprites [44] On CLEVR, we used the setting used for CLEVR6 in the original work except for increasing the possible number of objects. We found that using ten slots leads to better segmentation results than setting to 11 as with other models (one more than the max number of objects). On CLEVRTEX, we used color and protective transforms for both sprites and backgrounds.

MarioNette [51] We adjusted the model to learn to select and use from a dictionary of backgrounds, same as sprites. Additionally, we lowered the layer size to 4, using two layers, which gives 32 possible slots of size 64×64. On CLEVRTEX, we increased the sizes of both background and sprite dictionaries to as large as would fit in GPU memory. We trained with 60 sprites and single background on CLEVR, PLAINBG, and GRASSBG increasing the number of backgrounds to 60 on VARBG and CLEVRTEX.

C.4 Extra Figures

Here, we include extra figures listing additional output for all benchmarked models on CLEVR, CLEVRTEX, test sets and variants (Fig. 5). Fig. 6 contains example output of sprite- and glimpse-based models when they fail to learn correct foreground and background elements and learn to tile the image instead.

C.5 Dataset Construction

The main method of how the dataset is constructed is described in Section 3.1. Here, we include additional figures to showcase some steps in the dataset creation and provide catalog of materials used.

Lighting Fig. 8 shows the possible range of randomizing light positions in the scene, from warm closeup light positions with lots of shadows falling onto other objects to distant lights casting small soft shadows onto background even in crowded scenes. Fig. 8 also shows 4 possible shapes at 3 possible scales used in the CLEVRTEX.

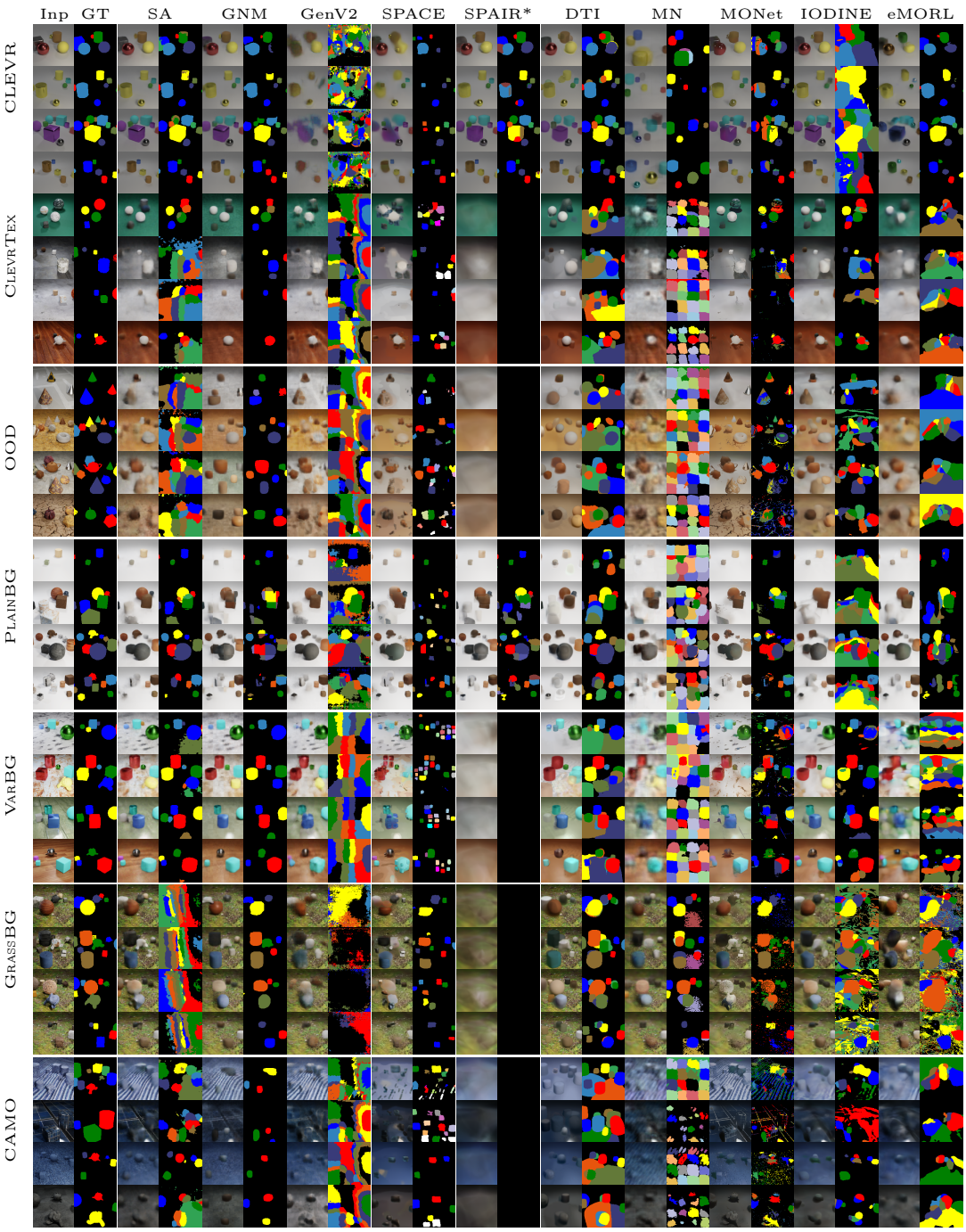


Figure 5: Comparison of various model reconstruction and segmentation outputs on CLEVR, CLEVRTEX and variants. Best viewed digitally.

Shape Adjustments CLEVRTEX features only 4 simple objects. This is mitigated by a range of material-specific geometry adjustments, bumping and transparency mapping applied to the seed shapes. Fig. 9 shows the effect of the shape perturbations in a scene where no other material properties have been applied to the objects.

Camera The camera position is jitterred along with lights. We use a perspective camera with a focal length of 0.035m and 0 shift.

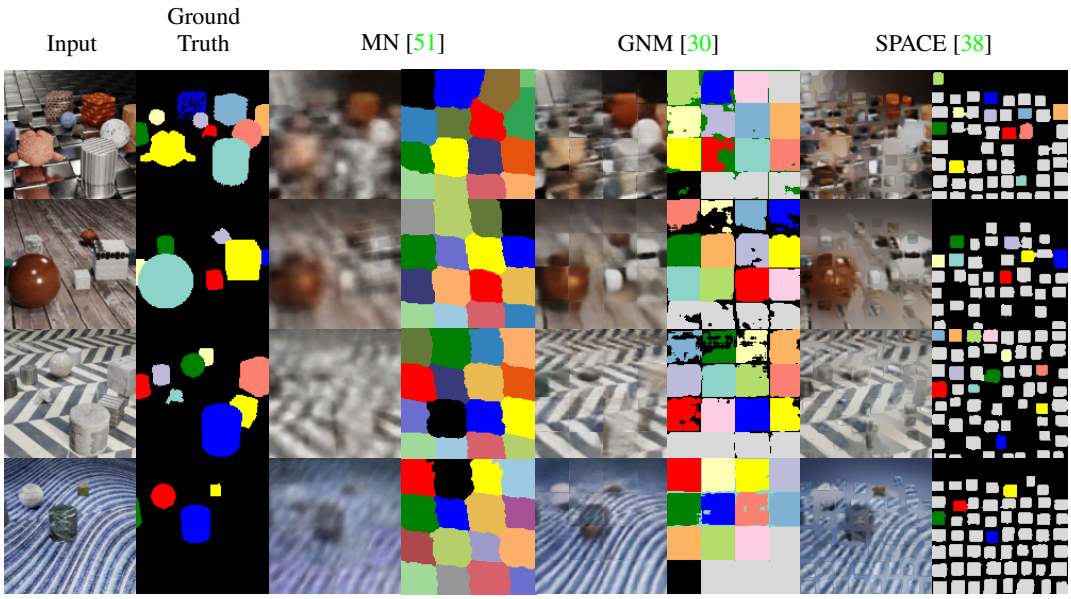
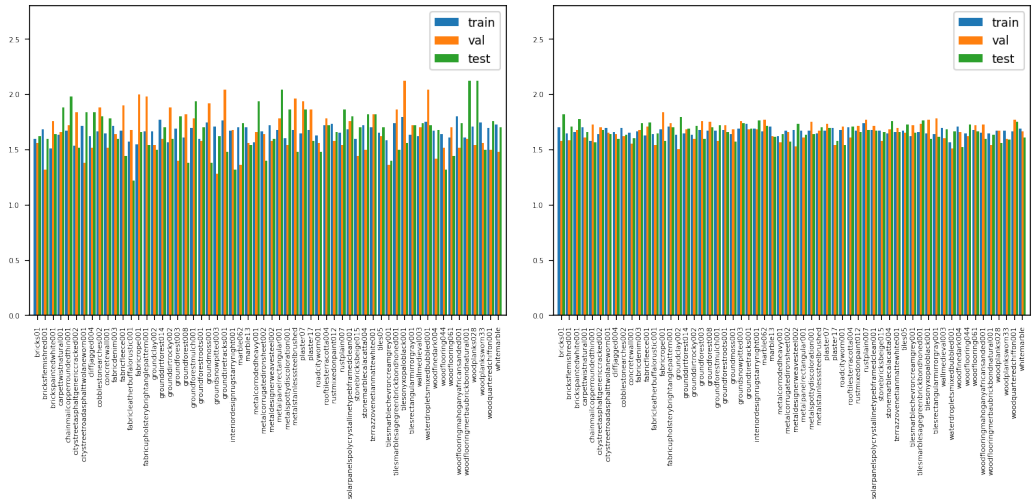


Figure 6: Tiling behaviour common to glimpse- (⊡) and sprite-based (⛹) models. Such tiling occurred whenever the model could not reproduce the foreground and background elements with respective component networks to sufficient accuracy. The models are trained on CLEVRTEX. GNM is shown here trained with output $\sigma = 0.3$.



(a) Background materials

(b) Object materials

Figure 7: Distribution of 60 materials in CLEVRTEX dataset between train/val/test splits, shown as a percentage. (a) shows distribution for the background. (b) shows distribution for objects.

Dataset Splits CLEVRTEX and variants are split into test/val/train datasets using 10%/10%/80% proportions after generation. The splits are made based on the index of the example, that is first 10% form test split. This simple scheme is motivated by the uniform sampling of the scene composition. Fig. 7 shows that this results in roughly proportional distribution of materials for both backgrounds and objects across dataset splits. OOD variant is test-only.

Materials Fig. 10 contains the list of 60 materials used in generating CLEVRTEX and its PLAINBG, VARBG, GRASSBG, and CAMO variants. Please see our generation code for further information. Fig. 11 contains 25 materials used in OOD variant.

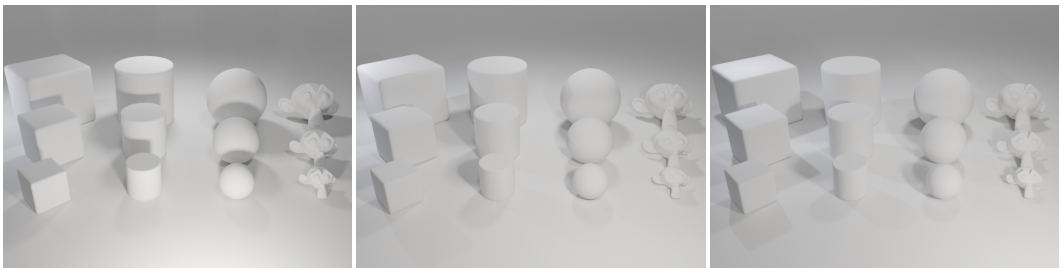


Figure 8: Effects of jittering light positions in the scenes. The images show two extremes with the mean position in the middle. The images also contain a showcase of 4 shapes present in the main CLEVRTEX dataset at 3 possible scales. The scenes are rendered without any materials.

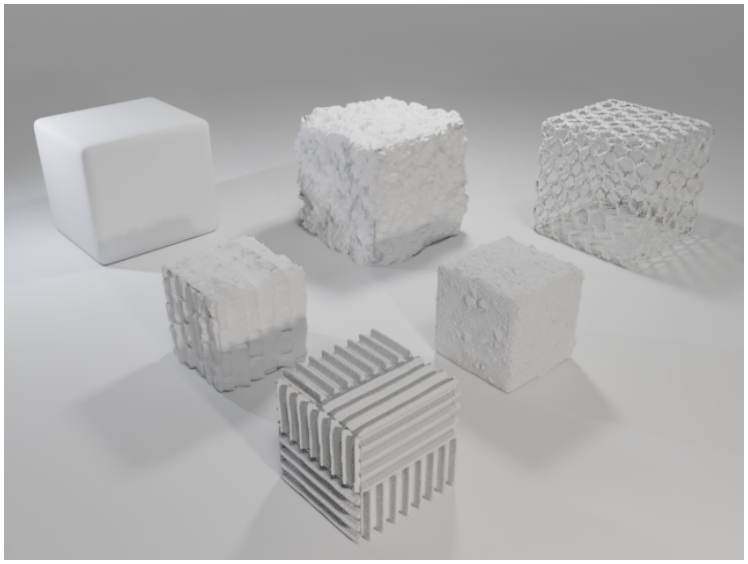


Figure 9: Showcase of a diverse set of shape perturbations applied the basic cube (top left) through a combination of displacement mapping, bumping and transparency mapping. Other material properties are not applied to the objects to show only the displacement details.

Figure 10: Materials used in CLEVRTEX dataset.

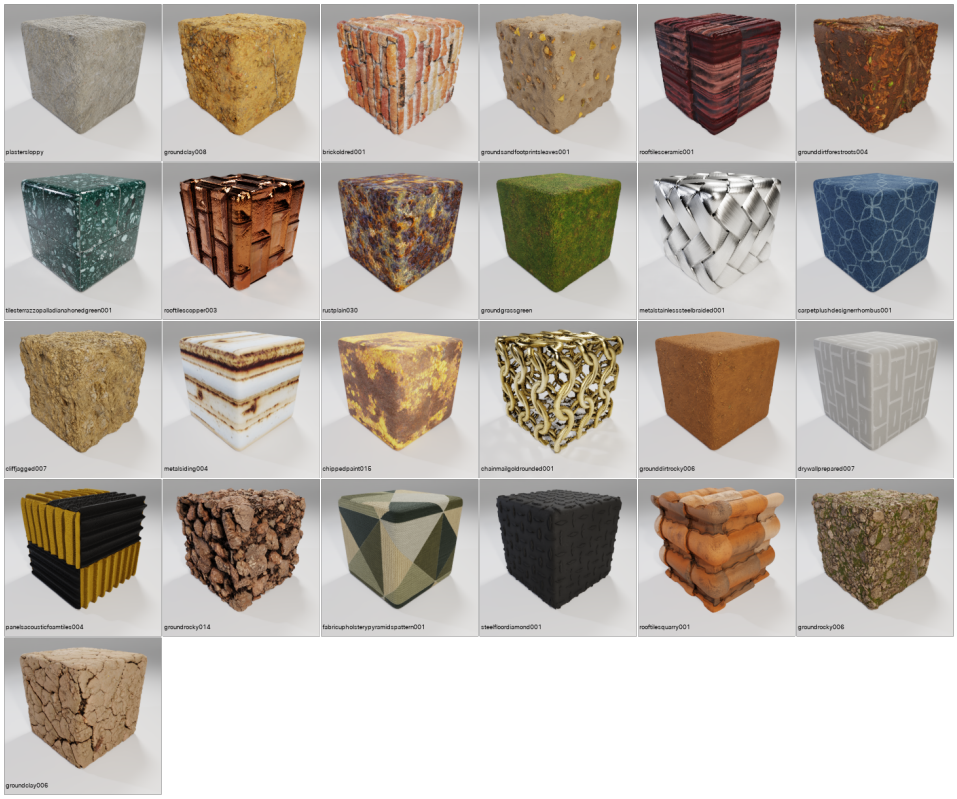


Figure 11: Materials used in OOD dataset variant.