A Benchmark Suite for Systematically Evaluating Reasoning Shortcuts: Supplementary Material

Samuele Bortolotti
U. Trento

Emanuele Marconato
U. Trento

Tommaso Carraro
Fondazione Bruno Kessler

Paolo Morettin
U. Trento

Emile van Krieken
U. Edinburgh

Antonio Vergari
U. Edinburgh

Stefano Teso
U. Trento

Andrea Passerini
U. Trento

1 Code, Data Sets and Generators

In the following, we discuss: 1) code and data licensing Section 1.1, 2) how the data was collected and organised Section 1.4, 3) what kind of information it contains Section 1.5, 4) how it should be used ethically and responsibly Section 1.2, 5) how it will be made available and maintained Section 1.3. All data, generators, metadata, and experimental code for reproducing the results are available at: `https://unitn-sml.github.io/rsbench`.

Detailed statistics for each data set using the default configuration are reported in Table 1.

Table 1: **Detailed statistics about the *default* data sets in** `rsbench`. For generators, the number of concepts k is configurable; in `CLE4EVR`, n and m are the minimum and maximum number of objects.

Task	Info $\mathbf{x}$	Info $\mathbf{c}$	Info $\mathbf{y}$	Train	Val	Test	OOD
`MNMath`	$28k \times 28$	k digits, 10 values each	cat multilabel	custom	custom	custom	custom
`MNAdd-Half`	56×28	2 digits, 10 values each	cat (19 values)	$2,940$	840	420	$1,080$
`MNAdd-EvenOdd`	56×28	2 digits, 10 values each	cat (19 values)	$6,720$	$1,920$	960	$5,040$
`MNLogic`	$28k \times 28$	k digits, 2 values each	binary	custom	custom	custom	custom
`Kand-Logic`	$3 \times 192 \times 64$	3 objects per image 3 shapes 3 colors	binary	$4,000$	$1,000$	$1,000$	–
`CLE4EVR`	320×240	n to m objects per image 10 shapes 10 colors 2 materials 3 sizes	binary	custom	custom	custom	custom
`BDD-OIA`	1280×720	21 binary concepts	bin multilabel, 4 labels	$16,082$	$2,270$	$4,572$	–
`SDD-OIA`	469×387	21 binary concepts	bin multilabel, 4 labels	$6,820$	$1,464$	$1,464$	$1,000$

1.1 Licensing

Code. Most of our code is available under the `BSD 3-Clause` license. The `CLE4EVR` and `SDD-OIA` generators are derived from the `CLEVR` code base, which is available under the `BSD` license. The `Kand-Logic` generator is derived from the `Kandinsky-patterns` code base, which is available under the `GPL-3.0` license, and so is our generator.

Data. `MNMath`, `MNAdd-Half`, `MNAdd-EvenOdd` and `MNLogic` are derived from `MNIST` [1], which is distributed under `CC-BY-SA 3.0`, and so are our data sets and generated data. `BDD-OIA` is derived

Submitted to the 38th Conference on Neural Information Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks. Do not distribute.

from BDD-100k [2], which is distributed under a BSD 3-Clause license, and so is our data set. Data sets and generated data for Kand-Logic and SDD-OIA are available under a CC-BY-SA 4.0 license.

1.2 Ethical Statement

rsbench is a collection of datasets aimed at exploring challenges related to concept quality, particularly focusing on identifying reasoning shortcuts. It also includes a formal verification tool to assess how often these shortcuts occur in specific configurations. Essentially, rsbench aims to help investigating concept quality in neural, neuro-symbolic and foundation models. Although this is not its intended purpose, such a benchmark may inadvertently used to improve models designed for harmful applications. However, to our knowledge, our work does not directly threaten individuals or society. Additionally, since most datasets are synthetically generated, they do not cause harm during creation. BDD-OIA, just like BDD-100k, could in principle be used to train models that aim to cause harm. We expressly disapprove of this usage.

1.3 Hosting and Maintenance Plan

The data is openly available on Zenodo at https://zenodo.org/doi/10.5281/zenodo.11612555. The data set generators are freely available on Github. The repository is linked in our website: https://github.com/unitn-sml/rsbench.

1.4 Data Collection

rsbench makes uses of two pre-existing data collections, namely MNIST and BDD-OIA. In this section, we briefly describe this data and how it is collected.

MNIST: The MNIST [1] dataset is a well known collection of handwritten digits, consisting of $60,000$ training images and $10,000$ test images. Each image is a 28×28 grayscale image of a numerical digit ranging from 0 to 9. The dataset was created by Yann LeCun, Corinna Cortes and Christopher J.C. Burges. MNAdd-EvenOdd and MNAdd-Half build on the MNIST dataset [3, 4]. MNLogic and MNMath, two datasets that can be generated from rsbench, make use of MNIST images.

BDD-OIA: BDD-OIA [5] is a dataset based on BDD-100K [2] dataset. BDD-100K is a large collection consisting of driving video data, developed by researchers at the University of California, Berkeley. The dataset is suitable for multitask learning, ranging from object detection to semantic segmentation and object tracking. It contains $100,000$ videos and images, collected under diverse driving conditions, times of day, and geographic locations. The data is annotated with labels including bounding boxes, lane marking, and drivable area segmentation. For further information, please refer to the original paper [2].

1.5 Data Generators

Each rsbench data generator comprises two Python components: the *generator* proper samples new data, and the associated *parser* reads the configuration from a YAML file. The latter also validates the configuration, *i.e.*, check for required fields and ensure the logical formulas work as intended. Users can also configure the generators through the command line. Generated images are stored in PNG format, and ground-truth annotations as JOBLIB metadata.

Shared configuration options. All generators support a set of basic command line settings: config: path to the YAML configuration file; output_dir: path to the output directory; n_samples: number of samples to be generated; log_level: verbosity level; seed: RNG seed, for reproducibility;

They all comply with the following YAML settings: symbols: names of the logic symbols (concepts) that appear in the knowledge; the order is managed internally by rsbench; logic: formal specification of the knowledge as a sympy formula, used for computing the ground truth labels; prop_in_distribution: proportion of examples to put in the in-distribution sets (train, validation, and test), up to 100%; combinations_in_distribution: what combinations of concept values should be included in the in-distribution sets. `val_prop`: proportion of examples to put in the validation set; `test_prop`: proportion of examples to put in the test set;

Non-Blender generators: `MNMath`, `MNLogic`, and `Kand-Logic`. The generator first parses the `YAML` configuration file, then proceeds to randomly sample the required number of examples. It generates a series of label and concept assignments that comply with the combinations combinations specified by the config file, if any. The ground-truth label is computed using the knowledge K. For `MNMath`, which is multi-class and multi-label, this involves splitting the configurations between classes or random sampling. Before the generation of the dataset, `rsbench` automatically checks whether the sampled configurations produce labels that are either all false or all true, and returns an error to the user if such a condition is found.

If the `prop_in_distribution` flag is set, the specified ratio is assigned to the in-distribution datasets (training, validation, and test), while the remaining settings are allocated to the out-of-distribution datasets. An equal number of examples are then assigned to both positive and negative configurations chosen for training, testing, and validation. This is achieved by sampling configurations alternately from positive and negative sides, with replacement. Depending on the dataset, examples are generated, and information such as labels and concepts are stored as `JOBLIB` metadata.

Finally, `rsbench` provides the option to specify a compression type (*e.g.*, zip) for storing the dataset, ensuring efficient storage and easy distribution.

Blender-based generators. Generating 3D images involves running scripts from within Blender, which requires a different setup. These scripts read all configuration from the command line and specified configuration files. Options include the positions of shapes (`shape_dir`) and materials (`material_dir`), the output directories (`output_image_dir` for the examples and `output_scene_dir` for metadata), the image resolution (`width`, `height`), and details bout the rendering step (like `render_tile_size`, `render_num_samples`, `camera_jitter`, `light_jitter`). The rendering engine used for `CLE4EVR` is `CYCLES`, while `SDD-OIA` uses the `EEVEE` rendering engine to speed up rendering, although this can be easily changed by the user.

The generators build on the implementation of [6]. The images are stored as `PNG`s, while the metadata, in `JSON` format, contains information about concepts, ground truth labels, object bounding boxes, object positions, and relationships between objects (*e.g.*, that one object is behind another). Unlike the synthetic data generation case, these scripts currently do not offer an option to compress the dataset, though this is a future contribution under consideration.

1.6 `MNMath` Data Generator

Additional `YAML` config for `MNMath` are the number of digits per image (`num_digits`) and the subset of candidate digits (`digit_values`). The code expects `num_digits` names for `symbols`: the first one is assigned to the first digit, the second symbol to the second digit, and so on. With `logic`, the user can provide the system of equations. With `combinations_in_distribution`, the user can have fine-grained control over the in-distribution data (*e.g.*, specifying "0234" means that the in-distribution data contains 0 2 3 4).

1.7 `MNLogic` Data Generator

The `YAML` file allows to specify the number of Boolean variables in the formula, as well as the formula itself. The knowledge defaults to the k-bit `XOR`. `rsbench` includes a script for generating random ℓ-CNF formulas, which can be readily used with `MNLogic` by setting `xor_rule` to false and `logic` to the target formula. If `use_mnist` is set, the input images are of size $(k \cdot 28) \times 28$ and obtained by concatenating k `MNIST` digits, one per bit. Otherwise, the code defaults to the setup of [3], where the inputs are encoded as $k \times 1$ black-and-white images, one pixel per bit.

You can filter what types of data appear in-distribution with `combinations_in_distribution` (*e.g.*, specifying 0101 means the in-distribution data contains 0 1 0 1).

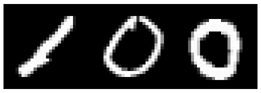
Table 2: Example of MNMath data

YAML config	JOBLIB metadata	PNG data

num_digits: 2
symbols:
  - a
  - b
logic:
  - 2*a + b
  - a + b
	{
	    ’label’: [6, 7],
	    ’meta’: {
	        ’concepts’: [
	            [2, 2],
	            [3, 4]
	        ]
	    }
	}	

Table 3: Example of MNLogic data

YAML config	JOBLIB metadata	PNG data

n_digits: 3
xor_rule: False
symbols:
  - a
  - b
  - c
logic:
    Or(And(a, b), Not(c))
use_mnist: True
	{
	    ’label’: True,
	    ’meta’: {
	        ’concepts’: [
	            True,
	            False,
	            False
	        ]
	    }
	}
		

1.8 Kand-Logic Data Generator

The YAML file allows specifying: n_shapes, the number of primitives per figure; n_figures: the number of figures per input image; colors, a subset of {red, yellow, blue}; shapes: a subset of {square, circle, triangle}. The first two symbols are associated to the first primitive in the first image, and refer to its shape and color, respectively; the next two to the second primitive, and so on for all primitives and figures in the input. logic applies to each individual figure. The ground-truth label of an image (consisting of multiple figures) is specified by aggregator_symbols and aggregator_logic. These give names to the variables holding the truth value for each figure, and how these values are aggregated to yield the ground-truth label, respectively.

The user can specify which data combination to generate in-distribution by setting combinations_in_distribution (*e.g.*, specifying • "red, square" • "blue, square" • "blue, square" means the in-distribution data contains an image made of a red square and two blue squares).

1.9 CLE4EVR Data Generator

The data generation process for CLE4EVR closely resembles that of previous datasets. To generate the datasets, the program samples various configurations, specifically the number of objects, shapes, colors, and sizes. These configurations are then divided into positive and negative sets based on the whether they satisfy the knowledge logic. The sets are used to generate images while maintaining a balanced ratio of positive and negative ground-truth samples.

Table 4: Example of `Kand-Logic` data

YAML config	JOBLIB metadata	PNG data

colors:
  - red
  - yellow
  - blue
shapes:
  - circle
  - square
  - triangle
symbols:
  - shape_1
  - color_1
  ...
  - shape_3
  - color_3
logic:
    (Eq(color_1, color_2) &
    Eq(shape_1, shape_2) &
    Ne(shape_1, shape_3)) |
    ...)
    # two equal one diff
aggregator_symbols:
  - pattern_1
  - pattern_2
  - pattern_3
aggregator_logic:
    pattern_1 &
    pattern_2 &
    pattern_3
	{
	    'label': True,
	    'meta': {
	        'concepts': [
	            [6, 2,
	            5, 1,
	            6, 2],
	
	            [6, 1,
	            5, 2,
	            6, 1],
	
	            [5, 2,
	            5, 2,
	            4, 1]
	        ]
	    }
	}	

`rsbench` allows users to customize various aspects of data generation, including the number of objects, whether occlusion is permitted, and the dimensions of the image. The occlusion check, which uses Blender rendering, can be slow for many objects due to rejection sampling.

`rsbench` by default includes two materials (rubber and metal), nine shapes, and eight predefined colors, with options to create custom blend files and specify RGB values. Default object sizes are large, medium, and small, but users can fully customize these settings in a configuration file.

The `symbols` for each object, are be defined in the following the order: color, shape, material, and size.

1.10 `SDD-OIA` Data Generator

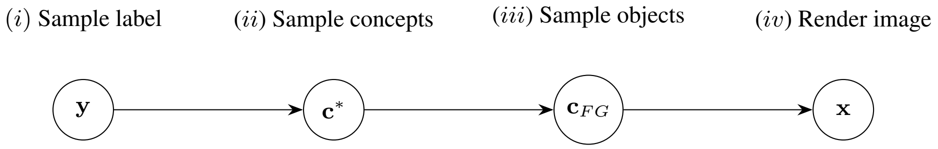
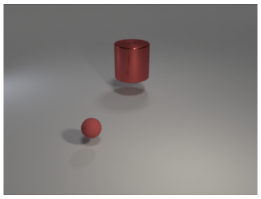


Figure 1: Illustration of the sampling process of `SDD-OIA`

Regarding `SDD-OIA`, `rsbench` allows users to specify parameters such as the number of samples, number of configurations to be generated, and image size.

Table 5: Example of CLE4EVR data

YAML config	JSON metadata	PNG data

symbols:
  - color_1
  - shape_1
  - mat_1
  - size_1
  - color_2
  - shape_2
  - mat_2
  - size_2
logic: |
    And(
      Eq(color_1, color_2),
      Eq(shape_1, shape_2),
      Eq(mat_1, mat_2),
      Eq(size_1, size_2)
)
	{
	    "label": 0,
	    "concepts": [
	    [
	      [
	        0,
	        1,
	        0,
	        0,
	        0,
	        0,
	        0,
	        0
	      ],
	}
		

For SDD-OIA, the data generation approach differs from other datasets in rsbench and follows a Bayesian network [7]. The process involves first (i) sampling the actions $\mathbf{y}$ from $p(\mathbf{y})$, ensuring that the overall dataset is balanced in the labels, *i.e.*, $p(\mathbf{y})$ is the uniform distribution. (ii) Second, we sample the ground-truth concepts $\mathbf{c}^*$ from the conditional $p(\mathbf{c}^* \mid \mathbf{y})$. Then, (iii) the concepts $\mathbf{c}^*$ specify a fine-grained distribution of objects in the scene, denoted as $\mathbf{c}_{FG}$, which are sampled through $p(\mathbf{c}_{FG}|\mathbf{c}^*)$. Next, the fine-grained objects are used to generate the scene. This step is deterministic and yields the final image $\mathbf{x}$. The crossroads scene is essentially a grid where objects' positions are specified by the fine-grained variables $\mathbf{c}_{FC}$. This ensures the concepts $\mathbf{c}^*$ are visible from the car's camera. The scene is then rendered with blender. The process is shown in Fig. 1. All steps in the sampling procedure ensure that all concepts can be retrieved from the image (respecting assumption **A1** in Appendix A.3) and that labels can be predicted uniquely from concepts $\mathbf{c}^*$ (respecting assumption **A2** in Appendix A.3).

A key aspect of SDD-OIA is its customizable data generation process, which involves sampling the concepts and constructing the scene. This necessitates a hard-coded compositional framework to correctly position the camera and objects, ensuring visibility from the car's perspective. This approach enables the creation of a high-quality synthetic neuro-symbolic dataset, where objects, sample quantities, and distribution ratios are fully customizable. Like other datasets, SDD-OIA maintains a balanced distribution across all actions. Users can configure model selection, object dimensions, and the probabilities for sampling different objects by adjusting the categorical distribution weights or the hard-coded matrix configuration.

1.10.1 Assets used in SDD-OIA

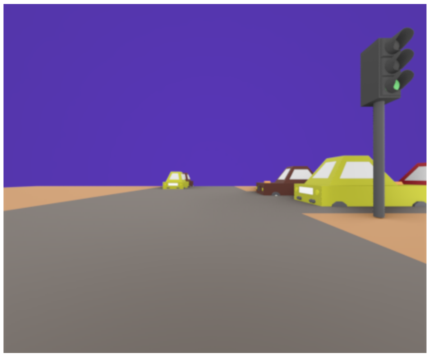
All assets are made available under permissive licenses that allow reuse for non-commercial purposes.

- Author: stunts. Speed Limit Signs [3D model]. Retrieved from https://free3d.com/3d-model/speed-limit-signs-172903.html;
- Author: corrobrocz. Concrete street barrier [3D model]. Retrieved from https://free3d.com/3d-model/concrete-street-barrier-917223.html;
- Author: paulsendesign. Cartoon low poly trees [3D model]. Retrieved from https://free3d.com/3d-model/cartoon-low-poly-trees-895299.html;

Table 6: Example of SDD-OIA data

JSON metadata | **PNG data**

```
{
    "label": [
        0,
        1,
        0,
        1
      ],
      "concepts": {
        "red_light": false,
        "green_light": true,
        "car": false,
        "person": false,
        "rider": false,
        "other_obstacle": false,
        "follow": false,
        "stop_sign": false,
        "left_lane": false,
        "left_green_light": true,
        "left_follow": false,
        "no_left_lane": true,
        "left_obstacle": false,
        "left_solid_line": false,
        "right_lane": true,
        "right_green_light": true,
        "right_follow": true,
        "no_right_lane": false,
        "right_obstacle": false,
        "right_solid_line": false,
        "clear": true
    }
}
```



- Author: roxas. Low Poly Car [3D model]. Retrieved from https://free3d.com/3d-model/low-poly-car-14842.html;
- Author: RokoTheAwesome. Traffic Light [3D model]. Retrieved from https://www.turbosquid.com/3d-models/traffic-light-547022

All the models from free3d are under the Personal Use License, meaning the models are available for free but only for personal or non-commercial use. In contrast, the models from TurboSquid are under the Standard 3D Model License, which permits the use of TurboSquid models in various commercial projects, such as games and movies. This license allows the creation and distribution of your end-products without reproduction limitations to any target market or audience indefinitely. However, the license prohibits making the models themselves directly available to end-users, so rsbench redirects to the asset URL.

1.11 BDD-OIA Data

Data for BDD-OIA are those previously published in [5]. BDD-OIA images are selected from BDD-100k only including franes with complicated scenes where multiple actions {forward, stop, left, right} are possible. This includes situations with multiple objects present. Following [5], all images are manually annotated for ground-truth actions and 21 associated binary concepts. The dataset contains 16k frames for training, (with annotated labels and concepts); 2k frames for validation, and 4.5k frames for testing. The table on the right from the previous paper [5] reports the overall proportion of labels and concepts.

Concept classes in `BDD-OIA`

Action Category	**Concepts**	**Count**
`move_forward`	`green_light`	7805
	`follow`	3489
	`road_clear`	4838
`stop`	`red_light`	5381
	`traffic_sign`	1539
	`car`	233
	`person`	163
	`rider`	5255
	`other_obstacle`	455
`turn_left`	`left_lane`	154
	`left_green_light`	885
	`left_follow`	365
	`no_left_lane`	150
	`left_obstacle`	666
	`letf_solid_line`	316
`turn_right`	`right_lane`	6081
	`right_green_light`	4022
	`right_follow`	2161
	`no_right_lane`	4503
	`right_obstacle`	4514
	`right_solid_line`	3660

2 Additional Results

Here, we report additional tables for TCAV evaluation complementing the results reported in the main text. All results indicate that TCAV at different layers always attain low $F1$-scores. We also report the $\mathsf{Cls}(C)$ and mAcc_C.

Table 7: Concept metrics for each NN layer using TCAV on MNAdd-EvenOdd

	Layer Num	Acc_C	$\mathrm{F}_1(C)$	$\mathsf{Cls}(C)$
$conv_1$	1	0.11 ± 0.03	0.10 ± 0.03	0.00 ± 0.00
$conv_2$	2	0.12 ± 0.03	0.10 ± 0.04	0.01 ± 0.02
fc_1	3	0.12 ± 0.04	0.09 ± 0.05	0.24 ± 0.30
fc_2	4	0.11 ± 0.02	0.07 ± 0.03	0.29 ± 0.34

Table 8: Concept metrics for each NN layer using TCAV on Kand-Logic

	Layer Num	Acc_C	$\mathrm{F}_1(C)$	$\mathsf{Cls}(C)$
$conv1$	1	0.35 ± 0.01	0.34 ± 0.01	0.00 ± 0.01
$conv2$	2	0.35 ± 0.01	0.34 ± 0.01	0.00 ± 0.01
$conv3$	3	0.34 ± 0.01	0.34 ± 0.01	0.00 ± 0.01
$conv4$	4	0.35 ± 0.01	0.34 ± 0.01	0.00 ± 0.01
$conv5$	5	0.35 ± 0.01	0.34 ± 0.01	0.00 ± 0.01
$fc1$	6	0.33 ± 0.01	0.32 ± 0.01	0.00 ± 0.01
$fc2$	7	0.33 ± 0.01	0.31 ± 0.01	0.00 ± 0.01

Table 9: Concept metrics for each NN layer using TCAV on SDD-OIA

	Layer Num	mAcc_C	$\mathrm{mF}_1(C)$	$\mathsf{Cls}(C)$
$conv1$	1	0.48 ± 0.02	0.44 ± 0.01	0.19 ± 0.05
$conv2$	2	0.49 ± 0.02	0.45 ± 0.02	0.20 ± 0.06
$conv3$	3	0.49 ± 0.03	0.45 ± 0.03	0.21 ± 0.09
$conv4$	4	0.48 ± 0.02	0.44 ± 0.01	0.23 ± 0.15
$conv5$	5	0.48 ± 0.02	0.44 ± 0.02	0.30 ± 0.26
$conv6$	6	0.46 ± 0.02	0.43 ± 0.02	0.34 ± 0.33
$fc1$	7	0.50 ± 0.02	0.45 ± 0.03	0.38 ± 0.31
$fc2$	8	0.49 ± 0.02	0.44 ± 0.02	0.43 ± 0.28

Table 10: Concept metrics for each NN layer using TCAV on SDD-OIA with synthetic images.

	Layer Num	mAcc_C	$\mathrm{mF}_1(C)$	$\mathsf{Cls}(C)$
$conv1$	1	0.47 ± 0.02	0.43 ± 0.02	0.18 ± 0.03
$conv2$	2	0.48 ± 0.02	0.44 ± 0.02	0.18 ± 0.03
$conv3$	3	0.49 ± 0.01	0.45 ± 0.01	0.23 ± 0.12
$conv4$	4	0.48 ± 0.03	0.44 ± 0.03	0.23 ± 0.14
$conv5$	5	0.48 ± 0.02	0.44 ± 0.02	0.29 ± 0.25
$conv6$	6	0.48 ± 0.04	0.45 ± 0.04	0.34 ± 0.32
$fc1$	7	0.51 ± 0.03	0.45 ± 0.03	0.38 ± 0.31
$fc2$	8	0.74 ± 0.01	0.42 ± 0.01	0.99 ± 0.01

3 Dataset Documentation: Datasheets for Datasets

Here, we answer the questions posed in the datasheets for datasets paper by Gebru et al [8].

3.1 Motivation

For what purpose was the dataset created? `rsbench` was created to study the phenomenon of reasoning shortcuts (RSs) and concept quality in neuro-symbolic and neural architectures. `rsbench` offers several datasets where RSs occur, as well as a formal verification tool that enables users to verify how many RSs appear in the desired settings.

Who created the dataset (*e.g.*, which team, research group) and on behalf of which entity (*e.g.*, company, institution, organisation)? The datasets have been created by the "Structured Machine Learning" research group at the department of Information Engineering and Computer Science of the University of Trento in collaboration with the april Lab at School of Informatics, University of Edinburgh.

Who funded the creation of the dataset? The datasets have been created for research purposes. Funded by the European Union. The views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union, the European Health and Digital Executive Agency (HaDEA) or the European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them. Grant Agreement no. 101120763 - TANGO. PM is supported by the MSCA project GA n°101110960 "Probabilistic Formal Verification for Provably Trustworthy AI - PFV-4-PTAI". AV is supported by the "UNREAL: Unified Reasoning Layer for Trustworthy ML" project (EP/Y023838/1) selected by the ERC and funded by UKRI EPSRC. Emile van Krieken was funded by ELIAI (The Edinburgh Laboratory for Integrated Artificial Intelligence), EPSRC (grant no. EP/W002876/1).

3.2 Composition

What do the instances that comprise the dataset represent (*e.g.*, documents, photos, people, countries)? All datasets contain annotations regarding concepts and labels. `SDD-OIA` comprises synthetically generated images depicting autonomous driving scenarios, such that if they were captured from a car's dashcam, and includes additional information about the scene structure, such as bounding boxes, 2D and 3D coordinates, and spatial relationships among objects. `MNMath`, `MNAdd-Half`, `MNAdd-EvenOdd` and `MNLogic` contain synthetic images of handwritten digits, derived from the `MNIST` dataset. `Kand-Logic` consists of synthetic data showcasing patterns of geometric shapes with various colors. `CLE4EVR` features synthetically generated images representing 3D objects of different shapes, colors, materials, and dimensions; similar to `SDD-OIA`, they include additional scene information. `BDD-OIA` is a real-world, high-stakes dataset comprising images captured from a car's dashcam. For a comprehensive description, please refer to [5].

How many instances are there in total (of each type, if appropriate)? Please refer to Table 1.

Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set? The datasets represent samples from configurations that can be randomly generated according to a grammar. Using the generators, one can filter through various combinations and determine the level of exhaustiveness for generating examples. For a comprehensive overview of each dataset generation process, please consult Section 1.5 and subsequent sections.

What data does each instance consist of? Alongside the images, each dataset sample is annotated with concepts and labels. However, for `SDD-OIA` and `CLE4EVR`, detailed scene information is included, encompassing individual 2D and 3D coordinates, bounding boxes, and spatial relationships between objects. For an complete overview refer to Table 1.

Is there a label or target associated with each instance? Yes, the concept annotations are derived from the data generation process, while the labels are symbolically derived from the knowledge provided to the dataset.

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 connections between different instances.

Are there recommended data splits (*e.g.*, training, development/validation, testing)? Information about the data splits we employed is reported in Appendix B. The user has the freedom to choose the data splits they prefer during the data generation process.

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)? Some of our data sets build on top of established and stable data, namely `MNIST` and (the last frames provided by) `BDD-100k`, for which we provide download links. `SDD-OIA` makes use of external assets, listed in Section 1.10.1. The ready-made `SDD-OIA` data set does not require these assets, but in order to use the generator these have to be obtained separately.

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. `BDD-OIA` contains images depicting pedestrians and bicycle riders. Identifiable information in these images, including anonymization, rights, and risks, is managed by the original `BDD-100k` authors.

Does the dataset identify any subpopulations (*e.g.*, by age, gender)? Please refer to 3.2.

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? Please refer to 3.2.

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)? Please refer to 3.2.

3.3 Collection Process

How was the data associated with each instance acquired? `MNIST` and `BDD-100k` have been obtained from their official repositories, `http://yann.lecun.com/exdb/mnist/` and `https://dl.cv.ethz.ch/bdd100k/data/`, respectively. All other data is synthetically generated.

What mechanisms or procedures were used to collect the data (*e.g.*, hardware apparatus or sensor, manual human curation, software program, software API)? Details about data generations and software programs are discussed in Appendix B.

If the dataset is a sample from a larger set, what was the sampling strategy (*e.g.*, deterministic, probabilistic with specific sampling probabilities)? Please refer to the similar question in Section 3.2.

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 these datasets.

Over what timeframe was the data collected? The datasets were generated over a span of several days.

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. `BDD-OIA` is the only dataset relating to people, please refer to Section 3.2.

3.4 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 datasets were generated along with labels and concept annotations.

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

3.5 Uses

Has the dataset been used for any tasks already? In the paper, we demonstrate and benchmark the intended use of these datasets for evaluating concept quality and exploring RSs. `MNAdd-EvenOdd`, `MNAdd-Half`, and `CLE4EVR` have been utilized in previous studies [4, 3, 9] to investigate RSs and concept quality.

Is there a repository that links to any or all papers or systems that use the dataset? Yes, https://unitn-sml.github.io/rsbench/.

What (other) tasks could the dataset be used for? `SDD-OIA` and `CLE4EVR` offer additional information regarding the scene, including the 3D and 2D coordinates of objects, their bounding boxes, and the relationships between objects within the scene. This spatial data enables various applications such as object discovery, object detection, and reasoning over the scene's structure.

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? These datasets are meant for research purposes only.

3.6 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 datasets, data generators, and related evaluation code are available on the website, enabling users to generate, download, and test their model on the data. Each dataset is provided in `zip` format and can be downloaded from the Zenodo link on the website.

When will the dataset be distributed? The datasets employed in the paper are available now on the website.

Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)? Please refer to Section 1.1.

Have any third parties imposed IP-based or other restrictions on the data associated with the instances? `SDD-OIA` makes use of assets taken from `https://free3d.com` and `https://www.turbosquid.com`. See Section 1.10.1 for the full list and associated licenses. Other instances of datasets 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.

3.7 Maintenance

Who is supporting/hosting/maintaining the dataset? The datasets are supported by the authors and will be actively maintained by the "Structured Machine Learning" research group in the future. For the hosting and maintenance plan, please refer to Section 1.3.

How can the owner/curator/manager of the dataset be contacted (e.g., email address)? The authors of `rsbench` can be contacted via their email addresses: samuele.bortolotti@unitn.it, emanuele.marconato@unitn.it.

Is there an erratum? If errors are found, an 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 extensions will be communicated via the website. The datasets 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)? The only dataset involving people is `BDD-OIA`, plase refer to Section 3.2.

Will older versions of the dataset continue to be supported/hosted/maintained? We plan 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, the dataset generation code is available on our website.

3.8 Other Questions

Is your dataset free of biases? Our data sets are designed to induce a particular type of bias, namely reasoning shortcuts, in models, for the purpose of studying them. The data itself however is not biased towards human factors such as gender, ethnicity, age, etc.

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

3.9 Author Statement of Responsibility

The authors assume full responsibility for any rights violations and confirm the license associated with the datasets and their images.

References

[1] Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

[2] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2636–2645, 2020.

[3] Emanuele Marconato, Stefano Teso, Antonio Vergari, and Andrea Passerini. Not all neuro-symbolic concepts are created equal: Analysis and mitigation of reasoning shortcuts. In *NeurIPS*, 2023.

[4] Emanuele Marconato, Samuele Bortolotti, Emile van Krieken, Antonio Vergari, Andrea Passerini, and Stefano Teso. BEARS Make Neuro-Symbolic Models Aware of their Reasoning Shortcuts. *arXiv preprint arXiv:2402.12240*, 2024.

[5] Yiran Xu, Xiaoyin Yang, Lihang Gong, Hsuan-Chu Lin, Tz-Ying Wu, Yunsheng Li, and Nuno Vasconcelos. Explainable object-induced action decision for autonomous vehicles. In *CVPR*, pages 9523–9532, 2020.

[6] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[7] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[8] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III au2, and Kate Crawford. Datasheets for datasets, 2021.

[9] Emanuele Marconato, Gianpaolo Bontempo, Elisa Ficarra, Simone Calderara, Andrea Passerini, and Stefano Teso. Neuro symbolic continual learning: Knowledge, reasoning shortcuts and concept rehearsal. In *ICML*, 2023.