
On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset

Muhammad Waleed Gondal^{1*†} Manuel Wüthrich^{1*} Đorđe Miladinović²
Francesco Locatello¹² Martin Breidt³ Valentin Volchkov¹ Joel Akpo¹
Olivier Bachem⁴ Bernhard Schölkopf¹ Stefan Bauer^{1†}

¹Max Planck Institute for Intelligent Systems

²Department of Computer Science ETH Zurich

³Max Planck Institute for Biological Cybernetics

⁴Google Research, Brain Team

Abstract

Learning meaningful and compact representations with disentangled semantic aspects is considered to be of key importance in representation learning. Since real-world data is notoriously costly to collect, many recent state-of-the-art disentanglement models have heavily relied on synthetic toy data-sets. In this paper, we propose a novel data-set which consists of over one million images of physical 3D objects with seven factors of variation, such as object color, shape, size and position. In order to be able to control all the factors of variation precisely, we built an experimental platform where the objects are being moved by a robotic arm. In addition, we provide two more datasets which consist of simulations of the experimental setup. These datasets provide for the first time the possibility to systematically investigate how well different disentanglement methods perform on real data in comparison to simulation, and how simulated data can be leveraged to build better representations of the real world. We provide a first experimental study of these questions and our results indicate that learned models transfer poorly, but that model and hyperparameter selection is an effective means of transferring information to the real world.

1 Introduction

In representation learning it is commonly assumed that a high-dimensional observation \mathbf{X} is generated from low-dimensional factors of variation \mathbf{G} . The goal is usually to revert this process by searching for a latent embedding \mathbf{Z} which replicates the underlying generative factors \mathbf{G} , e.g. shape, size or color. Learning well-*disentangled* representations of complex sensory data has been identified as one of the key challenges in the quest for artificial intelligence (AI) [2, 45, 31, 3, 48, 29, 54], since they should contain all the information present in the observations in a compact and interpretable structure [2, 26, 8] while being independent from the task at hand [15, 33].

Disentangled representations may be useful for (semi-)supervised learning of downstream tasks, transfer and few-shot learning [2, 49, 39]. Further, such representations allow to filter out nuisance factors [27], to perform interventions and to answer counterfactual questions [44, 50, 45]. First

*These authors contributed equally.

†Correspondence to: waleed.gondal@tue.mpg.de, stefan.bauer@inf.ethz.ch

applications of algorithms for learning disentangled representations have been applied to visual concept learning, sequence modeling, curiosity-based exploration or even domain adaptation in reinforcement learning [51, 30, 42, 20, 22, 34, 54]. The research community is in general agreement on the importance of this paradigm and much progress has been made in the past years, particularly on the algorithmic level [e.g. 18, 24], fundamental understanding [e.g. 17, 52] and experimental evaluation [38]. However, research has thus far focused on synthetic toy datasets.

The main motivation for using synthetic datasets is that they are cheap, easy to generate and the independent generative factors can be easily controlled. However, real-world recordings exhibit *imperfections* such as chromatic aberrations in cameras and complex surface properties of objects (e.g. reflectance, radiance and irradiance), making transfer learning from synthetic to real data a nontrivial task. Despite the growing importance of the field and the potential societal impact in the medical domain or fair decision making [e.g. 6, 10, 37], the performance of state-of-the-art disentanglement learning on real-world data is unknown.

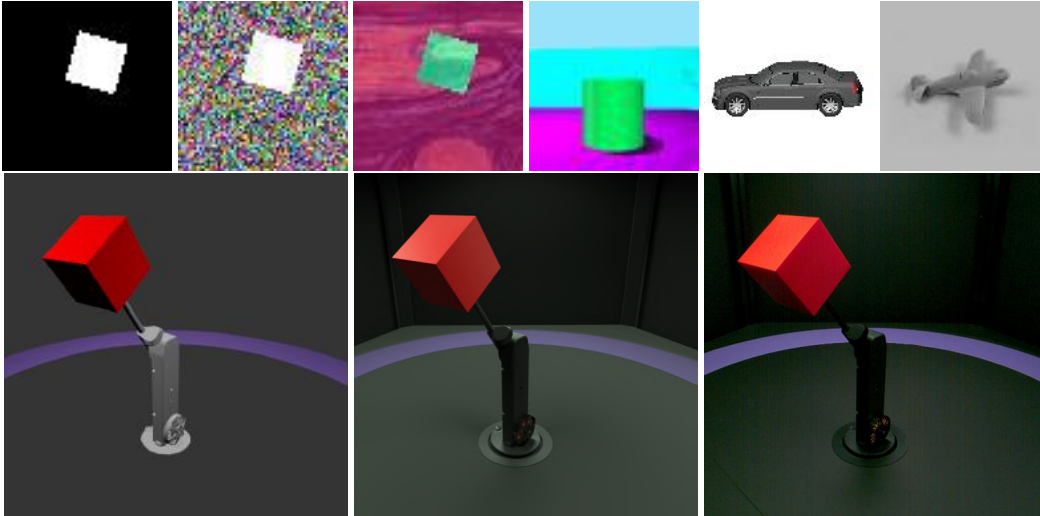


Figure 1: **Datasets:** In the top row samples from previously published datasets are shown from left to right: dSprites, Noisy-dSprites, Scream-dSprites, 3dshapes, Cars3D and SmallNORB. In the second row (again left to right) we provide simple simulated data, highly realistically simulated data and real-world data examples of the newly collected dataset.

To bridge the gap between simulation and the physical world, we built a recording platform which allows to investigate the following research questions: (i) How well do unsupervised state-of-the-art algorithms transfer from rendered images to physical recordings? (ii) How much does this transfer depend on the quality of the simulation? (iii) Can we learn representations on low dimensional recordings and transfer them from the current state-of-the-art of 64×64 images to high quality images? (iv) How much supervision is necessary to encode the necessary inductive biases? (v) Are the confounding and distortions of real-world recordings beneficial for learning disentangled representations? (vi) Can we disentangle causal mechanisms [44, 28, 29, 45] in the data generating process? (vii) Are disentangled representations useful for solving the real-world downstream tasks?

While answering all of the above questions is beyond the scope of this paper, our key contributions can be summarized as follows:

- We introduce the first *real-world 3D data set* recorded in a controlled environment, defined by 7 *factors of variation*: object color, object shape, object size, camera height, background color and two degrees of freedom of motion of a robotic arm. The dataset is made publicly available³.
- We provide synthetic images produced by computer graphics with two levels of realism. Since the robot arm and the objects are printed from a 3D template, we can ensure a close similarity between the realistic renderings and the real-world recordings.

³https://github.com/rr-learning/disentanglement_dataset

- The collected dataset of physical 3D objects consists of over one million images, and each of the two simulated datasets contains the same number of images as well.
- We investigate the role of inductive bias and the transfer of different hyper-parameter settings between the different simulations and the real-world and the requirements on the quality of the simulation for a successful transfer.

2 Background and Related Work

We assume a set of observations of a (potentially high dimensional) random variable \mathbf{X} which is generated by K unobserved causes of variation (generative factors) $\mathbf{G} = [G_1, \dots, G_K]$ (i.e., $\mathbf{G} \rightarrow \mathbf{X}$) that do not cause each other. These latent factors represent *elementary ingredients* to the causal mechanism generating \mathbf{X} [44, 45]. The elementary ingredients $G_i, i = 1, \dots, K$ of the causal process work on their own and are changeable without affecting others, reflecting the independent mechanisms assumption [49]. However, for some of the factors a hierarchical structure may exist for which this may only hold true when seeing the hierarchical structure as a whole as one component. The graphical model corresponding to this framework and adapted from [52] is depicted in figure 2. The hierarchical structure of the factors G_{K-1}^1 and G_{K-1} might represent one compositional process e.g. connected joints of a robot arm. The most commonly accepted understanding of *disentanglement* [2] is that each learned feature in \mathbf{Z} should capture one factor of variation in \mathbf{G} .

Current state-of-the-art disentanglement approaches use the framework of variational auto-encoders (VAEs) [25]. The (high-dimensional) observations \mathbf{x} are modelled as being generated from some latent features \mathbf{z} with chosen prior $p(\mathbf{z})$ according to the probabilistic model $p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$. The generative model $p_\theta(\mathbf{x}|\mathbf{z})$ as well as the proxy posterior $q_\phi(\mathbf{z}|\mathbf{x})$ can be represented by neural networks, which are optimized by maximizing the variational lower bound (ELBO) of $\log p(\mathbf{x}_1, \dots, \mathbf{x}_N)$

$$\mathcal{L}_{VAE} = \sum_{i=1}^N \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}^{(i)})} [\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p(\mathbf{z}))$$

Since the above objective does not enforce any structure on the latent space, except for some similarity to the typically chosen isotropic Gaussian prior $p(\mathbf{z})$, various proposals for more structure imposing regularization have been made. Using some sort of supervision [e.g. 43, 4, 35, 40, 9] or proposing completely unsupervised [e.g. 19, 24, 7, 27, 13] learning approaches. [19] proposed the β -VAE penalizing the Kullback-Leibler divergence (KL) term in the VAE objective more strongly, which encourages similarity to the factorized prior distribution. Others used techniques to encourage statistical independence between the different components in \mathbf{Z} , e.g., FactorVAE [24] or β -TCVAE [7], while DIP-VAE proposed to encourage factorization of the inferred prior $q_\phi(\mathbf{z}) = \int q_\phi(\mathbf{z}|\mathbf{x})p(\mathbf{x})d\mathbf{x}$. For other related work we refer to the detailed descriptions in the recent empirical study [38].

2.1 Established Datasets for the Unsupervised Learning of Disentangled Representations

Real-world data is costly to generate and groundtruth is often not available since significant confounding may exist. To bypass this limitation, many recent state-of-the-art disentanglement models [55, 24, 7, 18, 8] have heavily relied on synthetic toy datasets, trying to solve a simplified version of the problem in the hope that the conclusions drawn might likewise be valid for real-world settings. A quantitative summary of the most widely used datasets for learning disentangled representations is provided in table 1.

Dataset Descriptions: For quantitative analysis, *dSprites*⁴ is the most commonly used dataset. This synthetic dataset [18] contains binary 2D images of hearts, ellipses and squares in low resolution.

⁴<https://github.com/deepmind/dsprites-dataset>

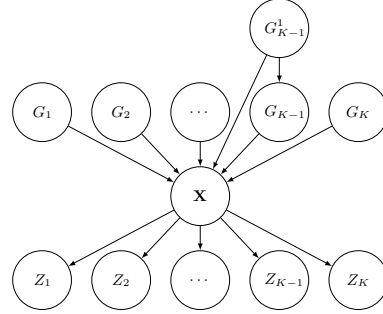


Figure 2: Graphical Model, where $\mathbf{G} = (G_1, G_2, \dots, G_K)$ are the generative factors (color, shape, size, ...) and \mathbf{X} the recorded images. The aim of disentangled representation learning is to learn variables Z_i that capture the independent mechanisms G_i .

Dataset	Factors of Variation	Resolution	# of Images	3D	Real-World
dSprites	5	64×64	737,280	✗	✗
Noisy dSprites	5	64×64	737,280	✗	✗
Scream dSprites	5	64×64	737,280	✗	✗
SmallNORB	5	128×128	48,600	✓	✗
Cars3D	3	64×64	17,568	✓	✗
3dshapes	6	64×64	480,000	✓	✗
<i>MPI3D-toy</i>	7	64×64	1,036,800	✓	✗
<i>MPI3D-realistic</i>	7	256×256	1,036,800	✓	✗
<i>MPI3D-real</i>	7	512×512	1,036,800	✓	✓

Table 1: Summary of the properties of different datasets. The newly contributed datasets are emphasized.

In *Color-dSprites* the shapes are colored with a random color, *Noisy-dSprites* considers white-colored shapes on a noisy background and in *Scream-dSprites* the background is replaced with a random patch in a random color shade extracted from the famous The Scream painting [41]. The dSprites shape is embedded into the image by inverting the color of its pixel. The *SmallNORB*⁵ dataset contains images of 50 toys belonging to 5 generic categories: four-legged animals, human figures, airplanes, trucks, and cars. The objects were imaged by two cameras under 6 lighting conditions, 9 elevations (30 to 70 degrees every 5 degrees), and 18 azimuths (0 to 340 every 20 degrees) [32]. For *Cars3D*⁶, 199 CAD models from [14] were used to generate 64×64 color renderings from 24 rotation angles each offset by 15 degrees [46]. Recently, *3dshapes* was made publicly available⁷, a dataset of 3D shapes procedurally generated from 6 ground truth independent latent factors. These factors are floor colour, wall colour, object colour, scale, shape and orientation [24].

3 Bridging the Gap Between Simulation and the Real World: A Novel Dataset

While other real-world recordings, e.g. *CelebA* [36], exist, they offer only qualitative evaluations. However, a more controlled dataset is needed to quantitatively investigate the effects of inductive biases, sample complexity and the interplay of simulations and the real-world.

3.1 Controlled Recording Setup

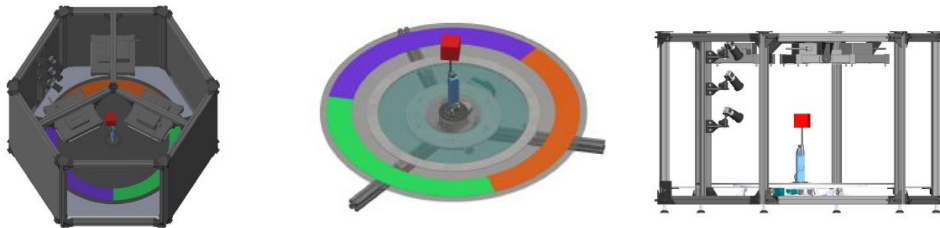


Figure 3: Renderings of the developed robotic platform: On the left a view from a 30° angle from the top (note that one panel in front and the top panels have been removed such that the interior of the platform is visible. During recordings, the platform is entirely closed. Middle: the robotic arm carrying a red cube (the entire cage is hidden). Right: frontal view without the black shielding (note the three cameras at different heights).

⁵<https://cs.nyu.edu/~ylclab/data/norb-v1.0-small/>

⁶<http://www.scottreed.info/files/nips2015-analogy-data.tar.gz>

⁷<https://github.com/deepmind/3dshapes-dataset/>

In order to record a controlled dataset of physical 3D objects, we built the mechanical platform illustrated in figure 3. It consists of three cameras mounted at different heights, a robotic manipulator carrying a 3D printed object (which can be swapped) in the center of the platform and a rotating table at the bottom. The platform is shielded with black sheets from all sides to avoid any intrusion of external factors (e.g. light) and the whole environment is relatively uniformly illuminated by three light sources installed within the platform.

3.1.1 Factors of Variation

The generative factors of variation G mentioned in section 2 are listed in the following for our recording setup.

Object Color: All objects have one of six different colors: red (255, 0, 0), green (0, 255, 0), blue (0, 0, 255), white (255, 255, 255), olive (210,210,80) and brown (153,76,0) (see figure 4).

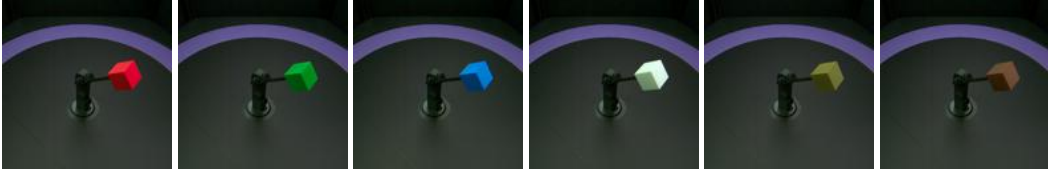


Figure 4: We show all the object colors while maintaining the other factors constant.

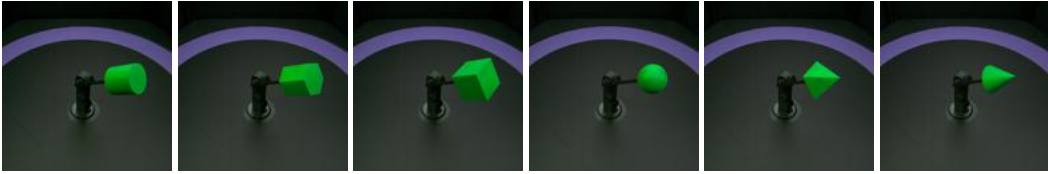


Figure 5: We show all object shapes while maintaining all other factors constant.

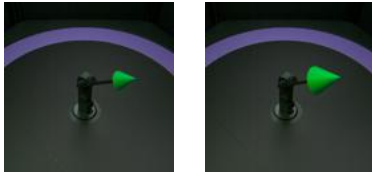


Figure 6: We show the two object sizes while maintaining all other factors constant.

Object Shape: There are objects of four different shapes in the dataset: a cylinder, a hexagonal prism, a cube, a sphere, a pyramid with square base and a cone. All objects exhibit rotational symmetries about some axes, however the kinematics of the robot are such that these axes never align with the degrees of freedom of the robot. This is important because it ensures that the robot degrees of freedom are observable given the images.

Object Size: There are objects of two different sizes in the dataset, categorized as large (roughly 65mm in diameter) and small (roughly 45 mm in diameter).

Camera Height: The dataset is recorded with three cameras mounted at three different heights (see figure 7 on the right), which represents another factor of variation in the images.

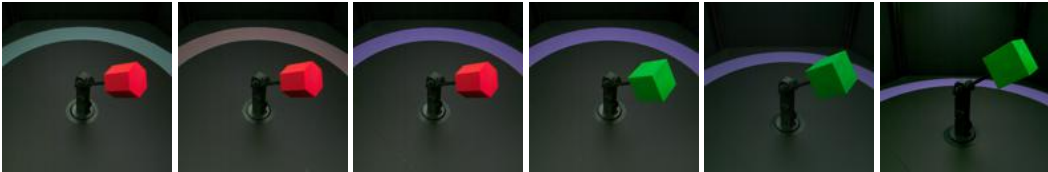


Figure 7: Three images on the left: we vary the background color. Three images on the right: we vary the camera height.

Background Color: The rotation table (see figure 7) allows us to change background color. Note that for all images in the dataset we orient the table in such a way that only one background color is visible at a time. The colors are: sea green, salmon and purple.

Degrees of Freedom of the Robotic Arm: Each object is mounted on the tip of the manipulator shown in figure 3. This manipulator has two degrees of freedom, a rotation about a vertical axis at the base and a second rotation about a horizontal axis. We move each joint in a range of 180° in 40 equal steps (see figure 8 and figure 9). Note that these two factors of variation are independent, just like all other factors (i.e. we record all possible combinations between the two).

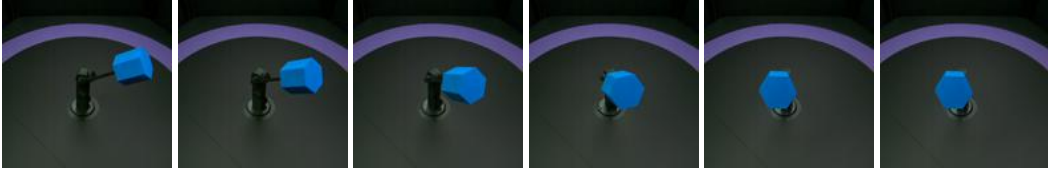


Figure 8: Motion along first DOF while maintaining the other factors constant. Note that in total we record 40 steps, here we only show 6 due to space constraints.

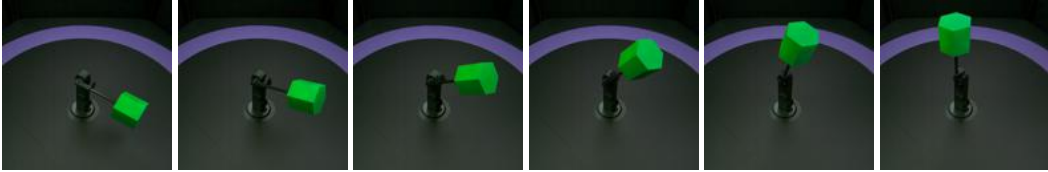


Figure 9: Motion along second DOF while maintaining the other factors constant. Note that in total we record 40 steps, here we only show 6 due to space constraints.

3.2 Simulated Data

In addition to the real-world dataset we recorded two simulated datasets of the same setup, hence all factors of variation are identical across the three datasets. One of the simulated datasets is designed to be as realistic as possible and the synthetic images are visually practically indistinguishable from real images (see figure 1 middle). For the second simulated dataset we used a deliberately simplified model (see figure 1 left), which allows to investigate transfer from simplified models to real data.

The synthetic data was generated using Autodesk 3ds Max(2018). Most parts of the scene were imported from SolidWorks CAD files that were originally used to construct the experimental stage including the manipulator and 3D printing of the test objects. The surface shaders are based on Autodesk Physical material with hand-tuned shading parameters, based on full resolution reference images. The camera poses were initialized from the CAD data and then manually fine-tuned using reference images. The realistic synthetic images were obtained using the Autodesk Raytracer (ART) with three rectangular light sources, mimicking the LED panels. The simplified images were rendered with the Quicksilver hardware renderer.

4 First Experimental Evaluations of (unsupervised) Disentanglement Methods on Real-World Data

Some fields have been able to narrow the gap between simulation and reality [56, 5, 23], which has led to remarkable achievements (e.g. for in-hand manipulation [1]). In contrast, for disentanglement methods this gap has not been bridged yet, state-of-the-art algorithms seem to have difficulties to transfer learned representations even between toy datasets [38]. The proposed dataset will enable the community to systematically investigate how such transfer of information between simulations with different degrees of realism and real data can be achieved. In the following we present a first experimental study in this directions.

4.1 Experimental Protocol

We apply all the disentanglement methods (β -VAE, FactorVAE, β -TCVAE, DIP-VAE-I, DIP-VAE-II, AnnealedVAE) which were used in a recent large-scale study [38] to our three datasets. Due to space constraints, the models are abbreviated with numbers one to five in the plots in the same order. We use (disentanglement_lib) and we evaluate on the same scores as [38]. In all the experiments, we used images with resolution 64x64. This resolution is used in the recent large-scale evaluations and by state-of-the-art disentanglement learning algorithms [38]. Each of the six methods is trained on each of the three datasets with five different hyperparameter settings (see table 2 in the appendix for details) and with three different random seeds, leading to a total of 270 models. Each model is trained for 300,000 iterations on Tesla V100 gpus. Details about the evaluation metrics can be found in appendix C.

4.2 Experimental Results

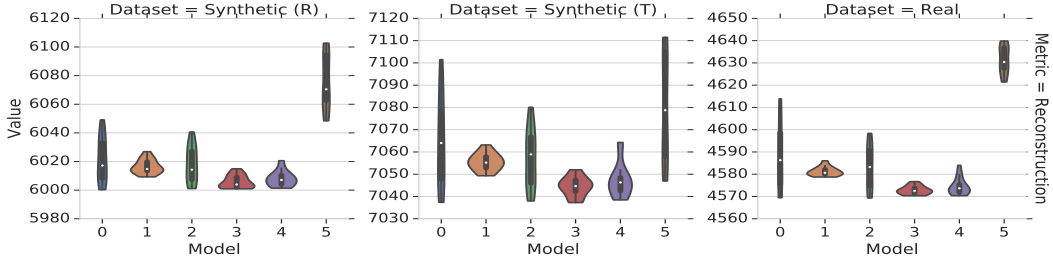


Figure 10: Reconstruction scores of different methods (0= β -VAE, 1=FactorVAE, 2= β -TCVAE, 3=DIP-VAE-I, 4=DIP-VAE-II, 5=AnnealedVAE) on the realistic synthetic dataset, the toy synthetic dataset and the real dataset.

Reconstruction Across Datasets: Figure 10 shows that there is a difference in reconstruction score across datasets: The score is the lowest on real data, followed by the realistic simulated dataset (R) and the simple toy (T) images. This indicates that there is a significant difference in the distribution of the real data compared to the simulated data, and that it is harder to learn a representation of the real data than of the simulated data. However, the relative behaviour of different methods seems to be similar across all three datasets, which indicates that despite the differences, the simulated data may be useful for model selection.

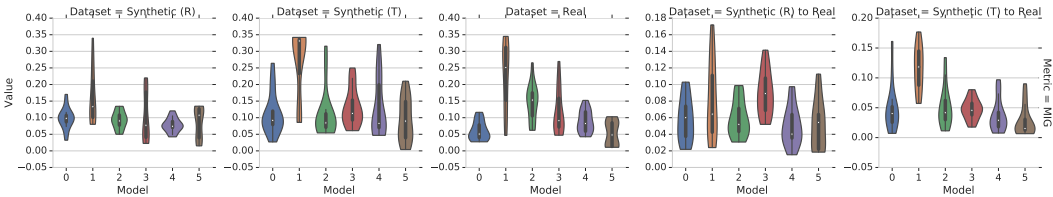


Figure 11: The Mutual Information Gap (MIG) scores attained by different methods for the following evaluations (from left to right): (a) trained and evaluated on synthetic realistic, (b) trained and evaluated on synthetic toy, (c) trained and evaluated on real, (d) trained on synthetic realistic and evaluated on real, (e) trained on synthetic toy and evaluated on real. The variance is due to different hyper-parameters and seeds.

Direct Transfer of Representations: In figure 11 we show the Mutual Information Gap (MIG) scores attained by different methods for different evaluations. The same plots for different metrics look qualitatively similar (see figure 22 in the appendix). Given the high variance, it is difficult to make conclusive statements. However, it seems quite clear that all methods perform significantly better when they are trained and evaluated on the same dataset (three plots on the left). Direct transfer of learned representations from simulated to real data (two plots on the right) seems to work rather poorly.

	Metric = DCI Disentanglement				
Synthetic (R)	100	80	67	85	76
Synthetic (T)	80	100	73	74	80
Real	67	73	100	73	68
Synthetic (R) to Real	85	74	73	100	56
Synthetic (T) to Real	76	80	68	56	100
	Synthetic (R)	Synthetic (T)	Real	Synthetic (R) to Real	Synthetic (T) to Real

Figure 12: Rank-correlation of the DCI disentanglement scores of different models (including hyperparameters) across different data sets.

Transfer of Hyperparameters: We have seen that transferring representations directly from simulated to real data seems to work poorly. However, it may be possible to instead transfer information at a higher-level, such as the choice of the method and its hyperparameters as an inductive bias.

In order to quantitatively evaluate whether such a transfer is possible, we pick the model (including hyperparameters) which performs best in simulation (according to a metric chosen at random), and we compute the probability of outperforming (according to a metric and seed chosen randomly) a model which was chosen at random. If no transfer is possible, we would expect this probability to be 50%.

However, we find that model selection from realistic simulated renderings (R) outperforms random model selection 72% of the time while transferring the model from the simpler synthetic images (T) to real-world data even beats random selection 78% of the time.

This finding is confirmed by figure 12, where we show the rank-correlation of the performance of models (including hyperparameters) trained on one dataset with the performance of these models trained on another dataset. The performance of a model trained on

some dataset seems to be highly correlated with the performance of that model trained on any other dataset. In figure 12 we use the DCI disentanglement metric as a score, however, qualitatively similar results can be observed using most of the disentanglement metrics (see figure 25 in the appendix).

Summary These results indicate that the simulated and the real data distribution have some similarities, and that these similarities can be exploited through model and hyperparameter selection. Surprisingly, it seems that the transfer of models from the synthetic toy dataset may work even better than the transfer from the realistic synthetic dataset.

5 Conclusions

Despite the intended applications of disentangled representation learning algorithms to real data in fields such as robotics, healthcare and fair decision making [e.g. 6, 10, 20], state-of-the-art approaches have only been systematically evaluated on synthetic toy datasets. Our work effectively complements related efforts [e.g. 38] to address current challenges of representation learning, offering the possibility of investigating the role of inductive biases, sample complexity, transfer learning and the use of labels using real-world images.

A key aspect of our datasets is that we provide rendered images of increasing complexity for the *same* setup used to capture the real-world recordings. The different recordings offer the possibility of investigating the question if disentangled representations can be transferred from simulation to the real world and how the transferability depends on the degree of *realism* of the simulation. Beyond the evaluation of representation learning algorithms, the proposed dataset can likewise be used for other tasks such as 3D reconstruction and scene rendering [12] or learning compositional visual concepts [21]. Furthermore, we are planning to use the novel experimental setup for recording objects with more complicated shapes and textures under more difficult conditions, such as dependence among different factors.

Acknowledgments

This research was partially supported by the Max Planck ETH Center for Learning Systems and Google Cloud. We thank Alexander Neitz and Arash Mehrjou for useful discussions. We would also like to thank Felix Grimmering, Ludovic Righetti, Stefan Schaal, Julian Viereck and Felix Widmaier whose work served as a starting point for the development of the robotic platform in the present paper.

References

- [1] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *arXiv preprint arXiv:1808.00177*, 2018.
- [2] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [3] Yoshua Bengio, Yann LeCun, et al. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5):1–41, 2007.
- [4] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In *AAAI*, 2018.
- [5] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Advances in Neural Information Processing Systems*, pages 343–351, 2016.
- [6] Agisilaos Chartsias, Thomas Joyce, Giorgos Papanastasiou, Scott Semple, Michelle Williams, David Newby, Rohan Dharmakumar, and Sotirios A Tsaftaris. Factorised spatial representation learning: application in semi-supervised myocardial segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 490–498. Springer, 2018.
- [7] Tian Qi Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *Advances in Neural Information Processing Systems*, pages 2610–2620, 2018.
- [8] Xi Chen, Yan Duan, Rein Houthoofd, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pages 2172–2180, 2016.
- [9] Brian Cheung, Jesse A Livezey, Arjun K Bansal, and Bruno A Olshausen. Discovering hidden factors of variation in deep networks. In *Workshop at International Conference on Learning Representations*, 2015.
- [10] E. Creager, D. Madras, J-H. Jacobson, M. Weis, K. Swersky, T. Pitassi, and R. Zemel. Flexibly fair representation learning by disentanglement. In *ICML*, page to appear, 2019.
- [11] Cian Eastwood and Christopher K. I. Williams. A framework for the quantitative evaluation of disentangled representations. In *International Conference on Learning Representations*, 2018.
- [12] SM Ali Eslami, Danilo Jimenez Rezende, Frederic Besse, Fabio Viola, Ari S Morcos, Marta Garnelo, Avraham Ruderman, Andrei A Rusu, Ivo Danihelka, Karol Gregor, et al. Neural scene representation and rendering. *Science*, 360(6394):1204–1210, 2018.
- [13] Babak Esmaeili, Hao Wu, Sarthak Jain, Siddharth Narayanaswamy, Brooks Paige, and Jan-Willem Van de Meent. Hierarchical disentangled representations. *arXiv preprint arXiv:1804.02086*, 2018.
- [14] Sanja Fidler, Sven Dickinson, and Raquel Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *Advances in neural information processing systems*, pages 611–619, 2012.
- [15] Ian Goodfellow, Honglak Lee, Quoc V Le, Andrew Saxe, and Andrew Y Ng. Measuring invariances in deep networks. In *Advances in neural information processing systems*, pages 646–654, 2009.
- [16] David Ha and Jürgen Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.
- [17] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

- [18] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.
- [19] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.
- [20] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. Darla: Improving zero-shot transfer in reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1480–1490. JMLR. org, 2017.
- [21] Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bosnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. Scan: Learning hierarchical compositional visual concepts. *arXiv preprint arXiv:1707.03389*, 2017.
- [22] Irina Higgins, Nicolas Sonnerat, Loic Matthey, Arka Pal, Christopher P Burgess, Matko Bošnjak, Murray Shanahan, Matthew Botvinick, Demis Hassabis, and Alexander Lerchner. Scan: Learning hierarchical compositional visual concepts. In *International Conference on Learning Representations*, 2018.
- [23] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. *arXiv preprint arXiv:1812.07252*, 2018.
- [24] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.
- [25] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.
- [26] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015.
- [27] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. In *International Conference on Learning Representations*, 2017.
- [28] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.
- [29] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. Building machines that learn and think like people. *Behavioral and Brain Sciences*, 40, 2017.
- [30] Adrien Laversanne-Finot, Alexandre Pere, and Pierre-Yves Oudeyer. Curiosity driven exploration of learned disentangled goal spaces. In *Conference on Robot Learning*, pages 487–504, 2018.
- [31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.
- [32] Yann LeCun, Fu Jie Huang, Leon Bottou, et al. Learning methods for generic object recognition with invariance to pose and lighting. In *CVPR (2)*, pages 97–104. Citeseer, 2004.
- [33] Karel Lenc and Andrea Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *IEEE conference on computer vision and pattern recognition*, pages 991–999, 2015.

- [34] Timothée Lesort, Natalia Díaz-Rodríguez, Jean-François Goudou, and David Filliat. State representation learning for control: An overview. *Neural Networks*, 2018.
- [35] Yen-Cheng Liu, Yu-Ying Yeh, Tzu-Chien Fu, Wei-Chen Chiu, Sheng-De Wang, and Yu-Chiang Frank Wang. Detach and adapt: Learning cross-domain disentangled deep representation. *arXiv preprint arXiv:1705.01314*, 2017.
- [36] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.
- [37] Francesco Locatello, Gabriele Abbati, Tom Rainforth, Stefan Bauer, Bernhard Schölkopf, and Olivier Bachem. On the fairness of disentangled representations. *arXiv preprint arXiv:1905.13662*, 2019.
- [38] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.
- [39] Francesco Locatello, Michael Tschannen, Stefan Bauer, Gunnar Rätsch, Bernhard Schölkopf, and Olivier Bachem. Disentangling factors of variation using few labels. *arXiv preprint arXiv:1905.01258*, 2019.
- [40] Michael F Mathieu, Junbo Jake Zhao, Junbo Zhao, Aditya Ramesh, Pablo Sprechmann, and Yann LeCun. Disentangling factors of variation in deep representation using adversarial training. In *Advances in Neural Information Processing Systems*, pages 5040–5048, 2016.
- [41] Edvard Munch. The scream, 1893.
- [42] Ashvin V Nair, Vitchyr Pong, Murtaza Dalal, Shikhar Bahl, Steven Lin, and Sergey Levine. Visual reinforcement learning with imagined goals. In *Advances in Neural Information Processing Systems*, pages 9209–9220, 2018.
- [43] Siddharth Narayanaswamy, T Brooks Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. In *Advances in Neural Information Processing Systems*, pages 5925–5935, 2017.
- [44] Judea Pearl. *Causality*. Cambridge university press, 2009.
- [45] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. MIT press, 2017.
- [46] Scott E Reed, Yi Zhang, Yuting Zhang, and Honglak Lee. Deep visual analogy-making. In *Advances in neural information processing systems*, pages 1252–1260, 2015.
- [47] Karl Ridgeway and Michael C Mozer. Learning deep disentangled embeddings with the f-statistic loss. In *Advances in Neural Information Processing Systems*, pages 185–194, 2018.
- [48] Jürgen Schmidhuber. Learning factorial codes by predictability minimization. *Neural Computation*, 4(6):863–879, 1992.
- [49] Bernhard Schölkopf, Dominik Janzing, Jonas Peters, Eleni Sgouritsa, Kun Zhang, and Joris Mooij. On causal and anticausal learning. In *International Conference on Machine Learning*, pages 1255–1262, 2012.
- [50] P. Spirtes, C. Glymour, and R. Scheines. *Causation, prediction, and search*. Springer-Verlag. (2nd edition MIT Press 2000), 1993.
- [51] Xander Steenbrugge, Sam Leroux, Tim Verbelen, and Bart Dhoedt. Improving generalization for abstract reasoning tasks using disentangled feature representations. In *Workshop on Relational Representation Learning at Neural Information Processing Systems*, 2018.

- [52] Raphael Suter, Djordje Miladinovic, Bernhard Schölkopf, and Stefan Bauer. Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 2019.
- [53] Jakub M Tomczak and Max Welling. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.
- [54] Sjoerd van Steenkiste, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem. Are disentangled representations helpful for abstract visual reasoning? *arXiv preprint arXiv:1905.12506*, 2019.
- [55] Nicholas Watters, Loic Matthey, Christopher P Burgess, and Alexander Lerchner. Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. *arXiv preprint arXiv:1901.07017*, 2019.
- [56] Jingwei Zhang, Lei Tai, Peng Yun, Yufeng Xiong, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Vr-goggles for robots: Real-to-sim domain adaptation for visual control. *IEEE Robotics and Automation Letters*, 4(2):1148–1155, 2019.

A Platform

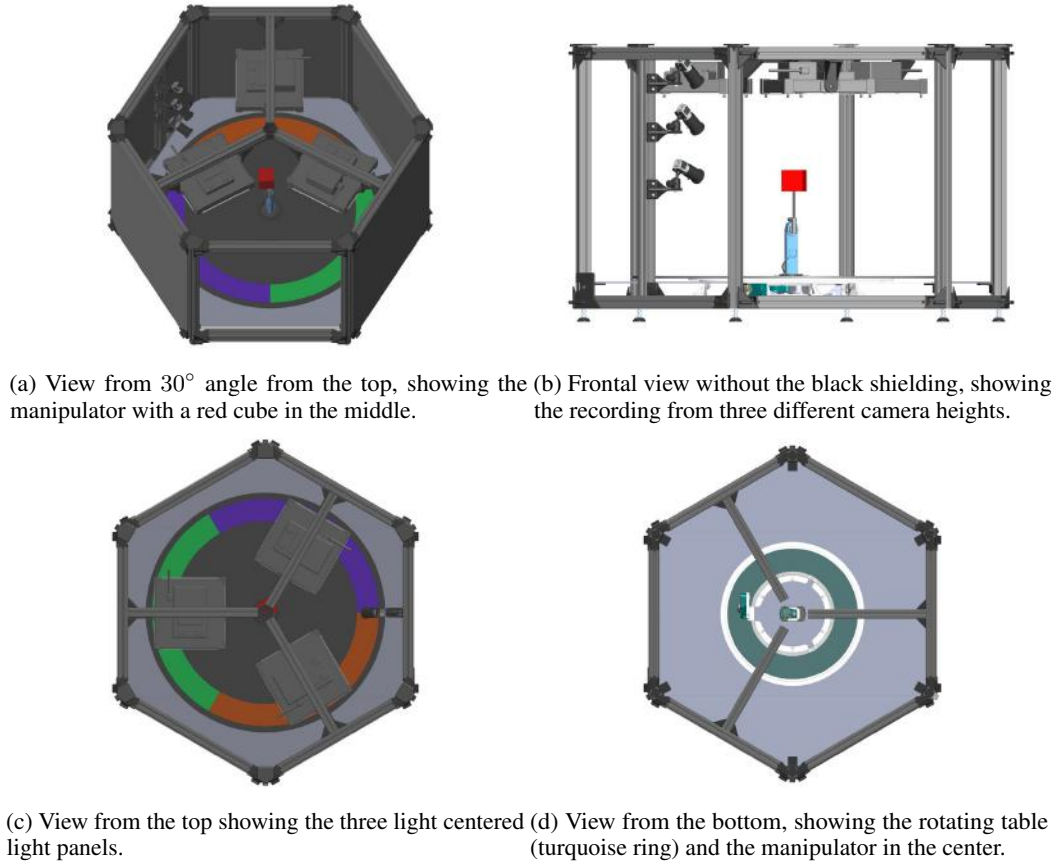


Figure 13: Different rendered images of the recording setup.

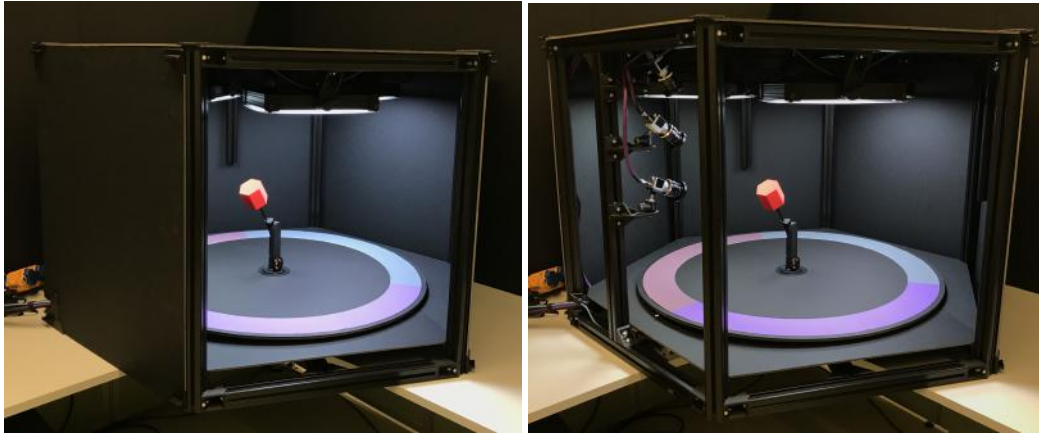


Figure 14: The mechanical platform for recording the real-world dataset.

A.1 Difference between Realistic Simulations and Real-World Images

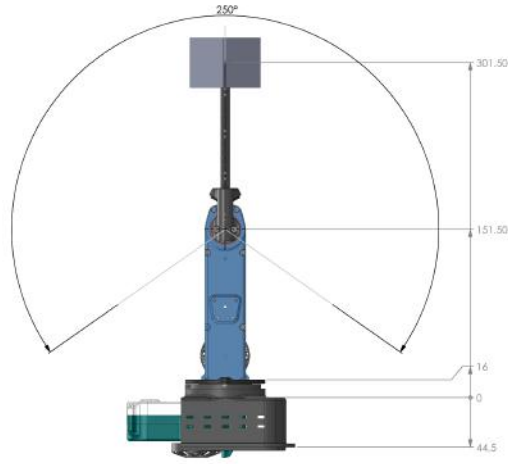
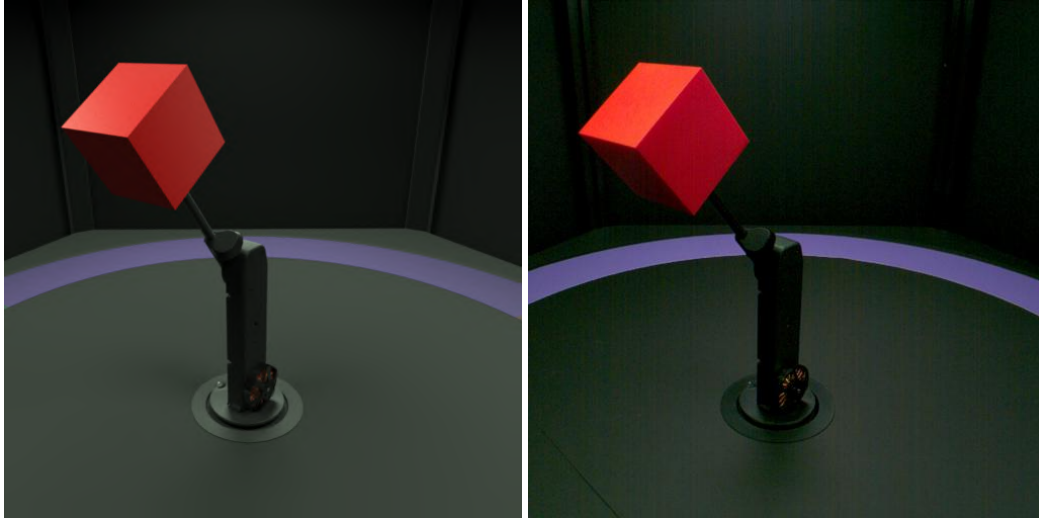


Figure 15: A technical drawing of the manipulator, dimensions are given in mm.



(a) Realistically simulated sample.

(b) Real-world sample.

Figure 16: Comparison of the realistically simulated and the real-world dataset for one example. The detailed procedure and the availability of CAD files used e.g. for the 3D printed robotic arm ensures a close overlap between both examples. Only small details and a high resolution show the differences e.g. the crack in the floor in the left corner which is only visible in the right image or shadings from the light on the object. Both pictures have a resolution of 512×512 .

B A Discussion on Disentangled Representations and their Transfer

Empirically, we observed that some of the best trained models were able to disentangle, though imperfectly, the factors of camera height, background colors, object sizes and the motions along the first and second degrees of freedom. Whereas, they performed poorly in disentangling the factors of some object shapes (for e.g. pyramid and cone) and some colors (for e.g. olive and brown). This may well be because of less pixel variation in the respective factors. The features of camera height and background color cause the most difference (maximum L2 distance) in image space. Similarly the object positions at different joint configurations (not the consecutive frames) also have big L2 distance which may explain why the models focus more on learning these factors.

The image reconstructions become more blurry as we move from the simple simulated dataset to the more complex real-world dataset, which can be seen by the reconstruction scores in figure 10. It has been previously noted that the use of overly simplistic priors like standard normal Gaussians in VAE models [25] can lead to overregularization [53]. To achieve disentanglement in VAE models, [18] put higher weights ($\beta > 1$) on the KL divergence which decreases the reconstruction quality. With the increased complexity in the dataset, the decrease in reconstruction quality becomes even more pronounced, as illustrated in figure 17, 18 and 19.

In figures 20 and 21, we show the reconstruction results of transfers from simple simulated and realistic simulated datasets to the real-world dataset. The models completely fail in transferring the representations from the simple simulated to the real-world data. On the other hand, in the realistic simulated to real-world transfer (figure 21) the models almost always reconstruct the correct background strip color, manipulator pose and the camera height factor. However, the object properties seem to differ a lot. This shows that in the case of complex environments, the models put more focus on learning the environment to get the better reconstruction accuracy than to learn the important but relatively small changing factors. This result for VAE models has also been confirmed by [16].

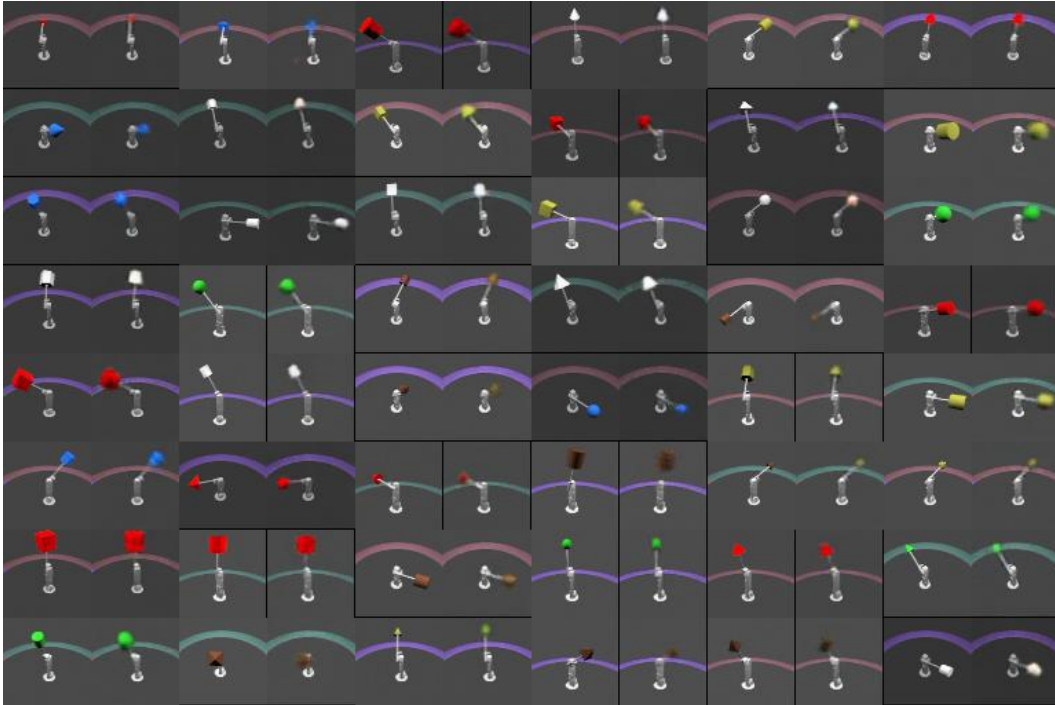


Figure 17: Image reconstruction of Factor VAE model on the low quality simulated dataset.

C Details of the Experimental Protocol

Metrics: Various methods to validate a learned representation for disentanglement based on known ground truth generative factors G have been proposed [e.g. 11, 47, 7, 24]. This has for example

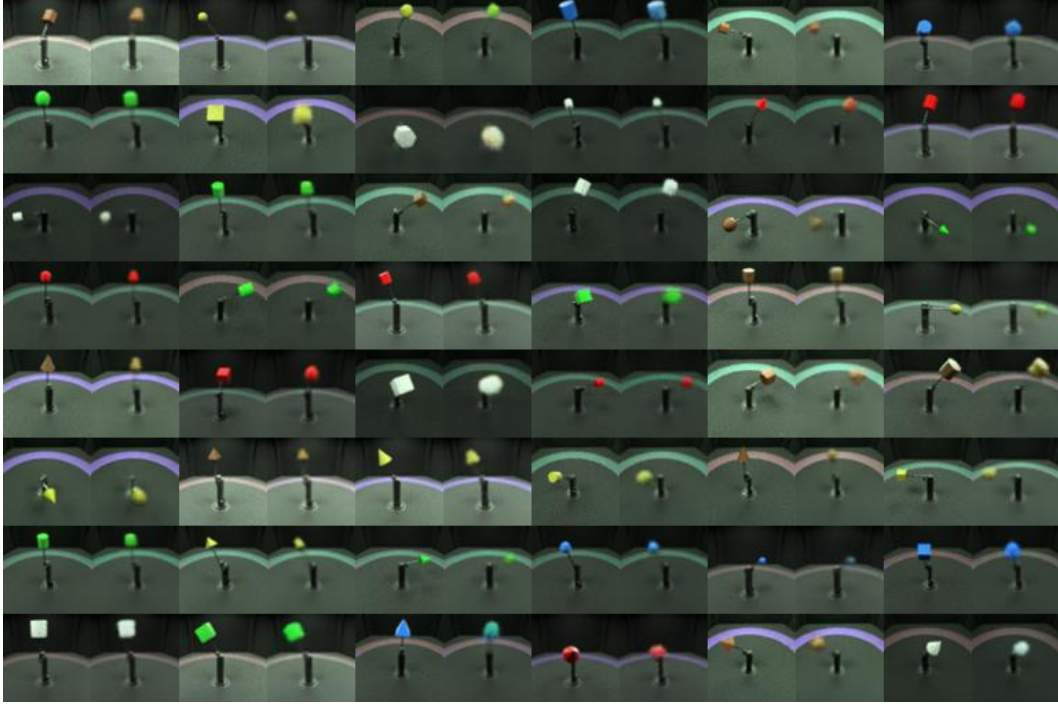


Figure 18: Image reconstruction of Factor VAE model on the realistic simulated dataset.

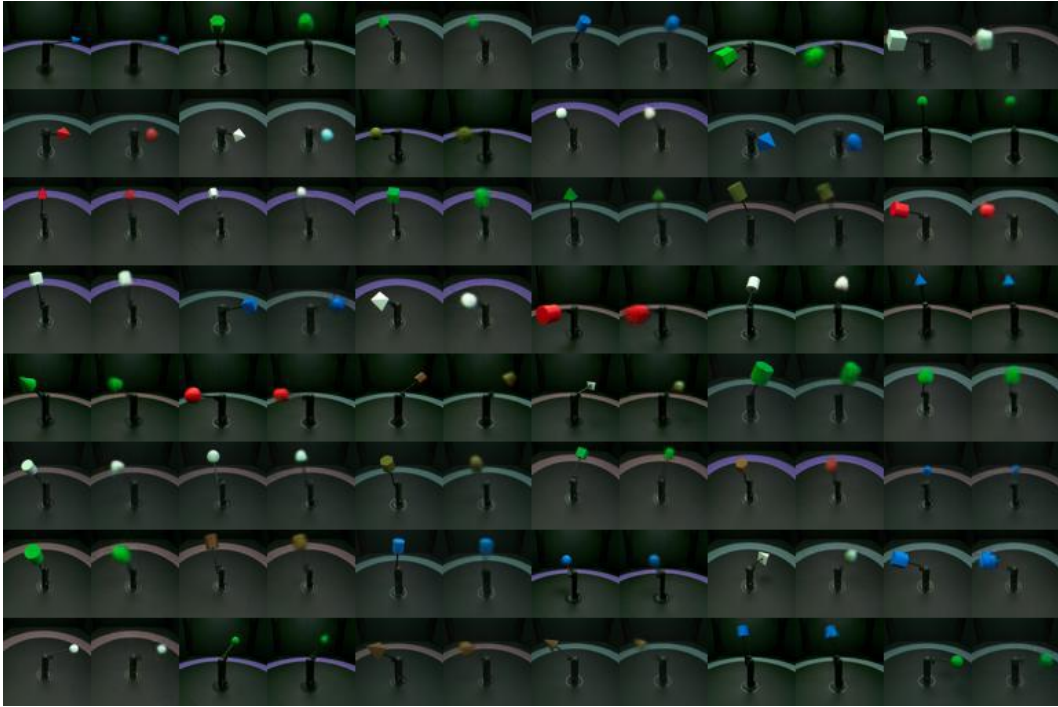


Figure 19: Image reconstruction of Factor VAE model on the real dataset.

been expressed as the mutual information of a single latent dimension Z_i with generative factors G_1, \dots, G_K [47], where in the ideal case each Z_i has some mutual information with one generative factor G_k but none with all the others. Similarly, [11] trained predictors (e.g., Lasso or random forests) for a generative factor G_k based on the representation \mathbf{Z} . In a disentangled model, each

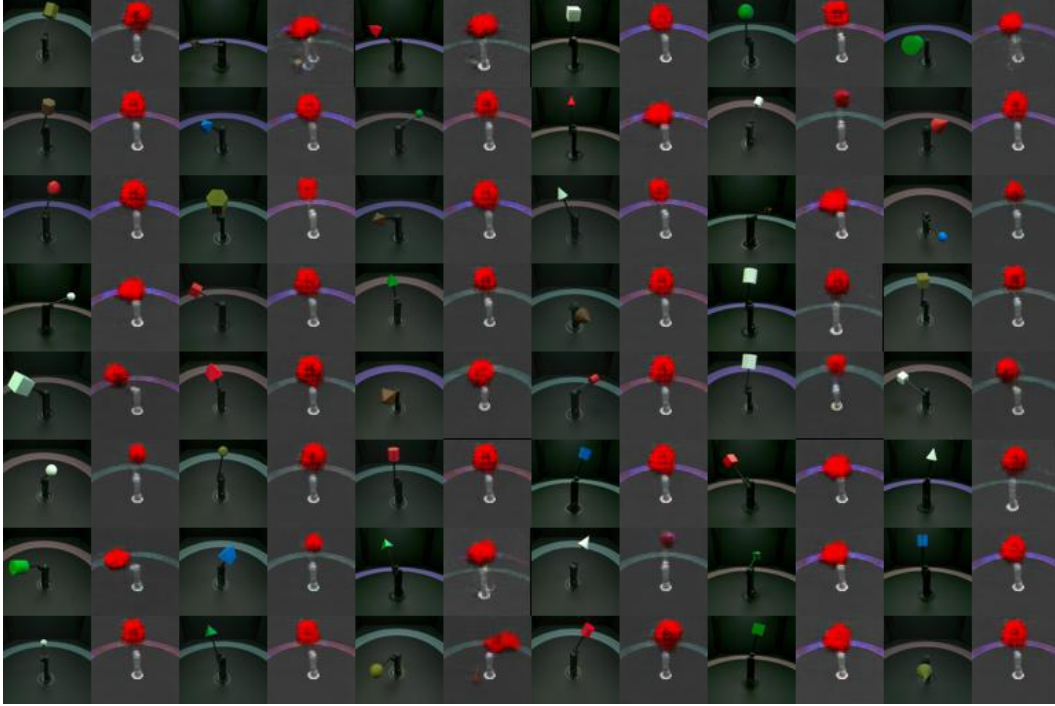


Figure 20: Image reconstructions of the Factor VAE model trained on toy dataset and tested on the real-world dataset. All images in the uneven columns are real, and to the right of each real image is its reconstruction.

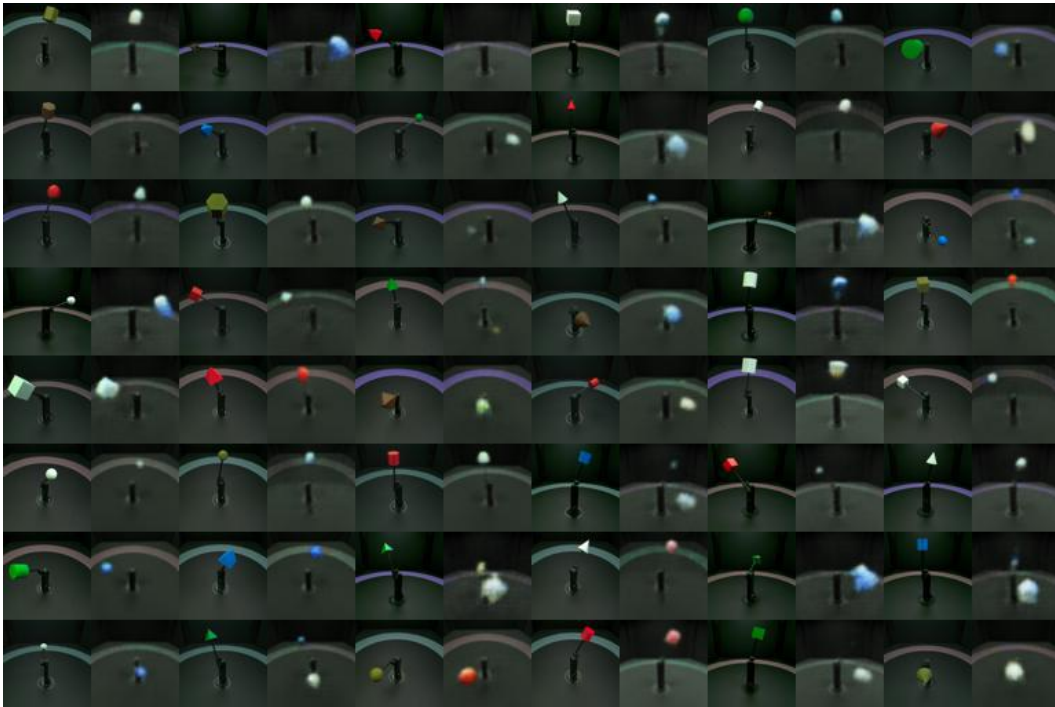


Figure 21: Image reconstructions of the Factor VAE model trained on realistic simulated dataset and tested on the real-world dataset. All images in the uneven columns are real, and to the right of each real image is its reconstruction.

dimension Z_i is only useful (i.e., has high feature importance) to predict one of those factors. [52] proposed an interventional robustness score. The graphical model of [52] adapted to our setup is illustrated in figure 2. Another form of validation, especially without known generative factors is the visual inspection of "latent traversals" [see e.g. 7].

Architecture All the models used the same convolutional encoder and decoder architecture with the fixed latent size of 10.

Model	Parameter	Values
β -VAE	β	[1, 2, 4, 6, 8]
AnnealedVAE	c_{max}	[5, 10, 25, 50, 75]
	iteration threshold	100000
	γ	1000
FactorVAE	γ	[10, 20, 30, 40, 50]
DIP-VAE-I	λ_{od}	[1, 2, 5, 10, 20]
	λ_d	$10\lambda_{od}$
DIP-VAE-II	λ_{od}	[1, 2, 5, 10, 20]
	λ_d	λ_{od}
β -TCVAE	β	[1, 2, 4, 6, 8]

Table 2: Hyperparameters of different methods.

Table 3: Encoder and Decoder architecture for the main experiment.

Encoder	Decoder
Input: $64 \times 64 \times$ number of channels	Input: 10
4×4 conv, 32 ReLU, stride 2	FC, 256 ReLU
4×4 conv, 32 ReLU, stride 2	FC, $4 \times 4 \times 64$ ReLU
4×4 conv, 64 ReLU, stride 2	4×4 upconv, 64 ReLU, stride 2
4×4 conv, 64 ReLU, stride 2	4×4 upconv, 32 ReLU, stride 2
FC 256, F2 2×10	4×4 upconv, 32 ReLU, stride 2
	4×4 upconv, number of channels, stride 2

Training Hyperparameters The training hyperparameters were kept fixed for each of the considered methods.

Table 4: Common hyperparameters for training.

Parameter	Values
Batch size	64
Latent space dimension	10
Optimizer	Adam
Adam: beta1	0.9
Adam: beta2	0.999
Adam: epsilon	1e-8
Adam: learning rate	0.0001
Decoder type	Bernoulli
Training steps	300000

D Detailed Experimental Results

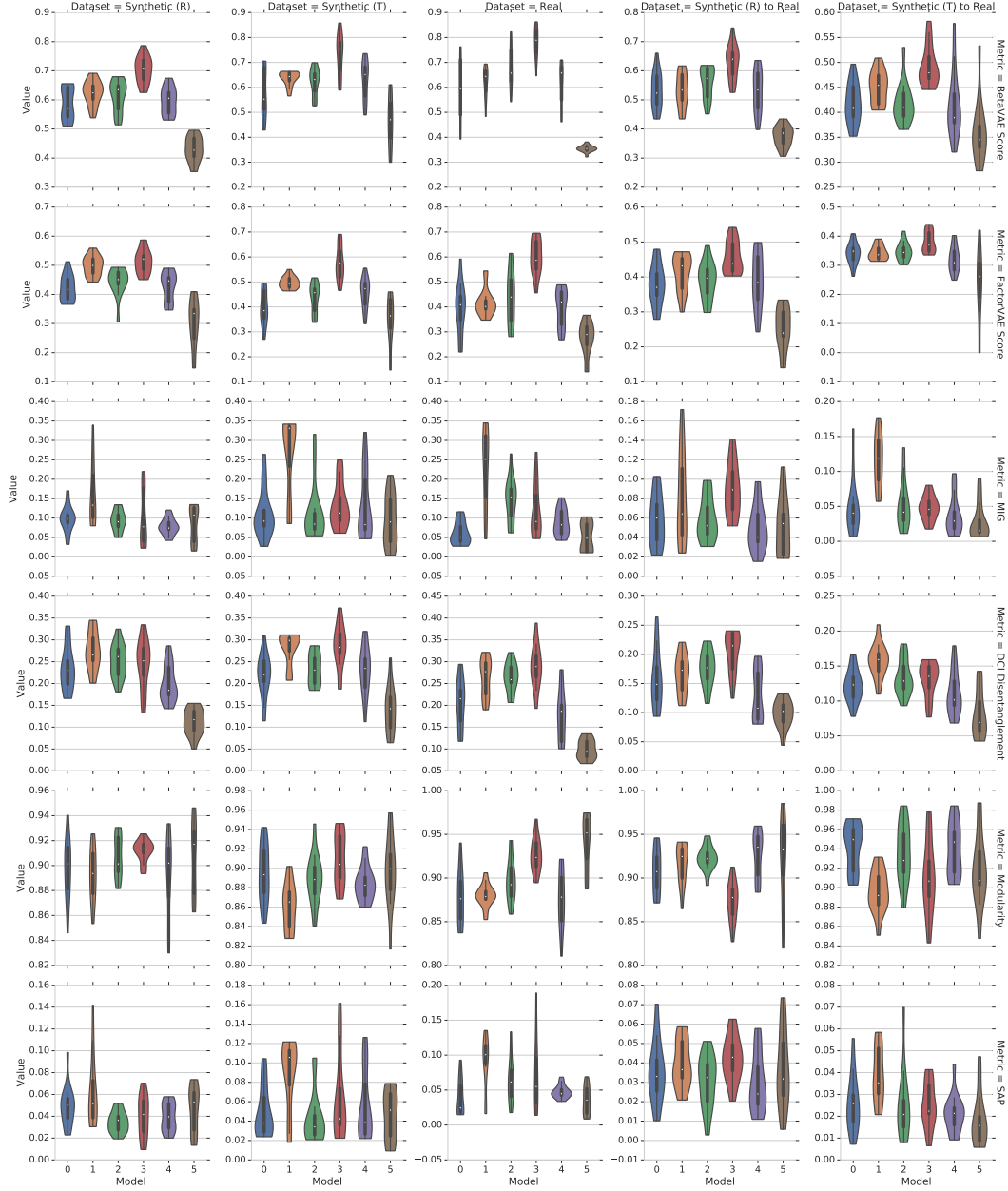


Figure 22: Disentanglement scores attained by different methods for different metrics (row) and evaluations (column, from left to right): (a) trained and evaluated on synthetic realistic, (b) trained and evaluated on synthetic toy, (c) trained and evaluated on real, (d) trained on synthetic realistic and evaluated on real, (e) trained on synthetic toy and evaluated on real. Methods are abbreviated (0= β -VAE, 1=FactorVAE, 2= β -TCVAE, 3=DIP-VAE-I, 4=DIP-VAE-II, 5=AnnealedVAE).



Figure 23: Rank correlation of different metrics on different data sets. Overall, we observe that all metrics except for Modularity seem to be at least mildly correlated on all data sets.

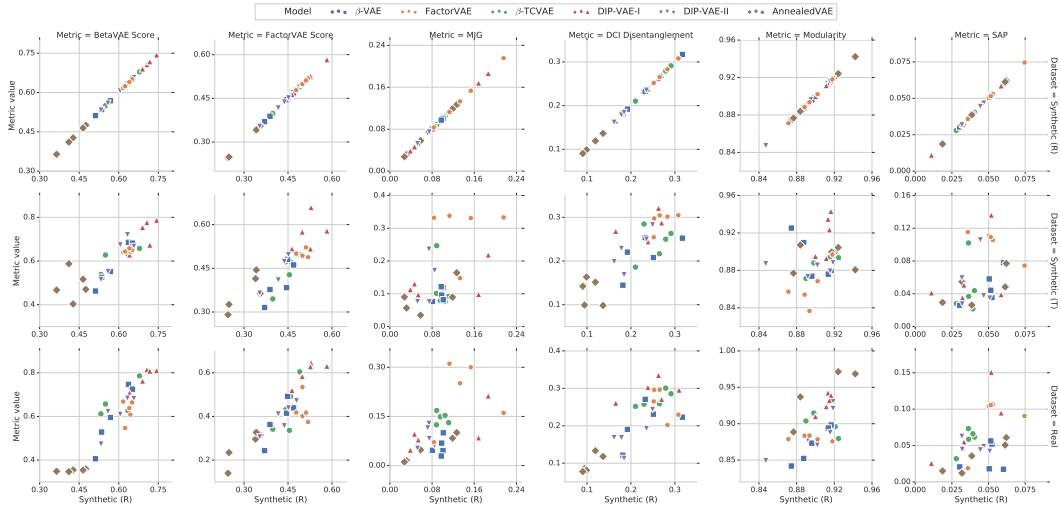


Figure 24: Disentanglement scores on real data vs simulated datasets. There seems to be a positive correlation for at least three metrics.

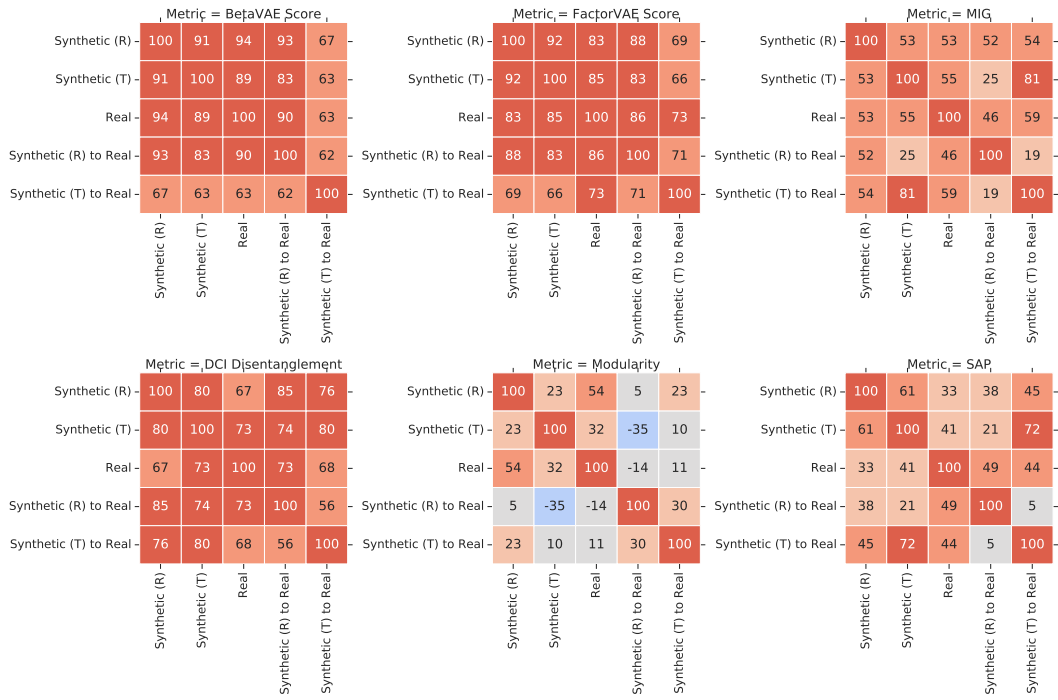


Figure 25: Rank-correlation of different disentanglement metrics across different data sets. Good hyperparameters seem to transfer well between datasets according to at least three metrics.