
Transferring disentangled representations: bridging the gap between synthetic and real images

Jacopo Dapuelto Nicoletta Noceti Francesca Odone
jacopo.dapuelto@edu.unige.it
{nicoletta.noceti, francesca.odone}@unige.it
MaLGa-DIBRIS, Università degli studi di Genova, Genova, Italy

Abstract

Developing meaningful and efficient representations that separate the fundamental structure of the data generation mechanism is crucial in representation learning. However, Disentangled Representation Learning has not fully shown its potential on real images, because of correlated generative factors, their resolution and limited access to ground truth labels. Specifically on the latter, we investigate the possibility of leveraging synthetic data to learn general-purpose disentangled representations applicable to real data, discussing the effect of fine-tuning and what properties of disentanglement are preserved after the transfer. We provide an extensive empirical study to address these issues. In addition, we propose a new *interpretable* intervention-based metric, to measure the quality of factors encoding in the representation. Our results indicate that some level of disentanglement, transferring a representation from synthetic to real data, is possible and effective.

1 Introduction

Developing meaningful, reusable and efficient representations is a critical step in representation learning [1, 58, 57, 67]. Disentangled Representation Learning (DRL) [1, 40, 24, 67] aims to learn models that can identify and disentangle underlying Factors of Variation (FoVs), hidden in the observable data. These models encode them in an interpretable and compact shape [31, 9, 1, 73], independently from the task at hand [22, 39, 66, 67]. Moreover, DRL enhances explainability, robustness, and generalization capacity across various applications [67]. Disentangled representations have been shown useful for various downstream tasks, such as FoVs prediction [41, 40], image generation [72, 48, 43, 42, 59] and translation [21, 19, 38], fair classification [56, 39], abstract reasoning [66, 63], domain adaptation [35], and out-of-distribution (OOD) generalization [11, 20].

While all the abovementioned methods may rely on different definitions of disentanglement (see just as examples [1, 23, 64]), and in this sense a comprehensive comparison is hard, they usually share the observation that some level of supervision on the FoVs is beneficial for disentanglement. However, labelling every single factor to achieve fully supervised disentanglement is costly or even unfeasible [70, 52]. For this reason, DRL has been mostly validated on synthetic or simulated data, usually acquired on purpose [11, 41, 60], and there is a limited understanding of the potential of DRL to address general-purpose representation tasks, as well as the specific challenges of the real world (e.g. the presence of clutter and occlusion, correlation between factors [11], etc.). Such challenges may prevent the model from learning perfectly disentangled representations [65].

In this work, we propose the adoption of Disentangled Representation (DR) transfer to deal with complex realistic/real dataset. DL transferring was explored in [20], where Source models learnt in an unsupervised manner were transferred to a Target dataset, by transferring hyperparameters. The authors observed a limited effectiveness in the direct transfer of representations. Instead, Dittadi et al. [11] found out that disentangled representation can help in OOD generalization from a simulated to a smaller real dataset. In both cases, the study involved very specific types of dataset, built to emulate

the real one in every detail. Recently, Fumero et al. [18] addressed disentanglement in real data without the need for FoVs annotation, leveraging the knowledge extracted from a diversified set of supervised tasks to learn a common disentangled representation to be transferred to real settings. We follow a different direction, setting up a very straightforward and generalizable procedure: we resort to a weakly supervised approach [41, 25, 26] to learn DRs on Source datasets where the FoVs are known and annotated, to then transfer (with no supervision) such representation to a Target dataset where the FoVs are not known or available. Our final aim is to consider real datasets as a Target, while synthetic data (where FoVs annotation is easy to obtain) can be employed as a Source.

The paper presents three main contributions: **(1) a novel metric** to assess the quality of disentanglement, which is *interpretable*, classifier-free and informative on the structure of the latent representation; **(2) a DR transfer methodology** to Target datasets without FoV annotation; **(3) an extensive experimental analysis** that considers different (Source, Target) pairs and quantitatively assesses the expressiveness of the learnt DR on Target of different nature (including the case where the gap between Source and Target is large), taking into consideration the main expected properties of disentangled representation. We discuss the role of fine-tuning and the need to reason on the distance between Source and Target datasets.

The paper is organized as follows. In Section 2, we propose and discuss our new intervention-based metric, OMES. In Section 3.3, we introduce our transfer approach to DRL, and provide a thorough analysis of different types of transfer scenarios (synthetic to synthetic, synthetic to real, real to real). Section 5 is left to the conclusions.

2 Evaluating the quality of disentanglement

2.1 Background

While there is no universally accepted definition of disentanglement, there is common agreement on the properties that a DR should have [12, 54, 66, 1, 57, 2]:

Modularity [55]: A factor influences only a portion of the representation space, and only this factor influences this subspace. This is achievable if the FoV are independent, meaning that a variation in one FoV does not affect others.

Compactness [55]: The subset of the representation space affected by a FoV should be as small as possible (ideally, only one dimension). This property is also called *completeness* in [13].

Explicitness [54]: DR should explicitly describe the factors, thus it should favour FoVs classification.

The taxonomy presented in [12] groups all metrics in three families (see a summary in Table 6 in App.): **Intervention-based** metrics compare codes by intervention, either creating subsets of data in which one or more factors are kept constant (*BetaVAE* [24] and *FactorVAE* [27]), or in which only one factor is varying (*RF-VAE* [28]), and predicting which factors were involved in the intervention; **Predictor-based** metrics use regressors or classifiers to predict factors from DR (*DCI Disentanglement* [13] and *SAP* [32]) or intervened subsets (*BetaVAE*, *FactorVAE* and *RF-VAE*); **Information-based** metrics leverage information theory principles, such as mutual information, to quantify factor-DR relationships (*Mutual Information Gap (MIG)* [8, 12], *MED* [6], *Modularity* [55] and *InfoMEC* [25]).

Intervention-based metrics have the advantage of providing control over the factor and the corresponding representation. However, they are all based on classifiers, thus they depend on method, hyperparameter settings and model capacity. The latter consideration can be extended to all Predictor-based metrics. On the other hand, Information-based methods are mainly ground on the computation of Mutual Information, which is dependent on an estimator and its parameters [51, 7].

Motivated by these limitations, we introduce in the next section a new metric, to the best of our knowledge, the *first* classifier-free intervention-based metric.

2.2 Our metric: OMES

OMES (*Overlap Multiple Encoding Scores*) is an intervention-based metric measuring the quality of factor encoding in the representation while providing information about its structure: we measure *modularity*, analyzing how the FoVs overlap, and *compactness*, detecting and quantifying how a factor is encoded in the dimensions of the representation.

Algorithm 1 Compute association matrix S between dimensions and FoVs

Require: $D_\Phi = [\mathbf{R}^1, \mathbf{R}^2, \mathbf{k}]$, n $\triangleright n$ number of FoVs
Ensure: $\mathbf{R}^1, \mathbf{R}^2 \in \mathbb{R}^{N \times m}$, $\mathbf{k} \in \mathbb{R}^N$ $\triangleright m = |\mathcal{A}|$, N number of pairs in D

- 1: $S \leftarrow \text{ZEROS}(m, n)$
- 2: **for** $j = 1$ to n **do**
- 3: $\mathbf{R}_j^1 = \mathbf{R}^1[\mathbf{k} == j, :]$ and $\mathbf{R}_j^2 = \mathbf{R}^2[\mathbf{k} == j, :]$
- 4: **for** $h = 1$ to m **do**
- 5: $PC = \text{PearsonCorr}(\mathbf{R}_j^1[:, h], \mathbf{R}_j^2[:, h])$
- 6: $S[h, j] = 1 - \text{abs}(PC)$
- 7: **end for**
- 8: **end for**
- 9: **return** S $\triangleright m \times n$ matrix

Algorithm 2 Overlap score of FoV j : OS

Require: matrix S , FoV index j
Ensure: $S \in \mathbb{R}^{m \times n}$

- 1: scores $\leftarrow \text{ZEROS}(m)$
- 2: **for** $h = 1$ to m **do** \triangleright dimension h
- 3: $i_S \leftarrow \text{ZEROS}(n)$; $i_S[j] = 1$
- 4: scores[h] = $1 - \text{MAE}(i_S, S[h, :])$
- 5: **end for**
- 6: **return** POOLING(scores)

Algorithm 3 Encoding score of FoV j : MES

Require: matrix S , FoV index j
Ensure: $S \in \mathbb{R}^{m \times n}$

- 1: scores $\leftarrow \text{ZEROS}(m)$
- 2: **for** $h = 1$ to m **do** \triangleright dimension h
- 3: $i_S \leftarrow \text{ZEROS}(m)$; $i_S[h] = 1$
- 4: scores[h] = $1 - \text{MAE}(i_S, S[:, j])$
- 5: **end for**
- 6: **return** POOLING(scores)

Given an image X , with Φ its mapping into a d -dimensional latent disentangled space, $\Phi(X) = r$, $r \in \mathbb{R}^d$. We discard dimensions whose empirical standard deviation is extremely small (< 0.05), meaning that the dimensions are inactive [68, 10]. This leaves us with a subset of $m \leq d$ active dimensions, to which we will refer in the following.

Let D be a dataset formed by image pairs, $D = \{(X_i^1, X_i^2, k_i)\}_{i=1}^N$, where X_i^1, X_i^2 are two images that differ for only the FoV k_i .

OMES requires computing a weighted *association* matrix S between the dimensions of the representation and the FoVs, with higher association values if the factor is encoded in a certain dimension (see Algorithm 1): we consider the representations of the image pairs in dataset D , obtaining D_Φ . In matrix notation we may write it as $D_\Phi = [\mathbf{R}^1, \mathbf{R}^2, \mathbf{k}]$, where the pair \mathbf{R}^1 and \mathbf{R}^2 are $N \times m$ -dimensional matrices with each row i is the representation of the i -th image pair $\Phi(X_i^1) = r_i^1$ and $\Phi(X_i^2) = r_i^2$ respectively. For each FoV k we extract the rows of D_Φ such that the i -th entry of vector \mathbf{k} is $k_i = k$, we call this set D_Φ^k . Each entry $S[h, j]$ of the association matrix relates a dimension h of the estimated disentangled representation with a FoV j . Its value is in the range $[0, 1]$ with elements close to 1 corresponding to a dimension that effectively captures the variations of a FoV. The association is based on a correlation analysis: since the samples from D_Φ^k are paired to differ only for the FoV k , we expect a good representation *not to correlate* where such FoV is encoded. To ease interpretability, we transform the obtained values (see Algorithm 1, line 6) so that high values denote a strong association between dimension and FoV.

When the model exhibits perfect disentanglement, each row and column of the association matrix S present just one element with a high association, corresponding to the only dimension where the factor is encoded. We thus measure the level of disentanglement through similarity with an ideal array, where the association matrix shows all 0s but in the positions of the correct associations, where there are 1s.

We rely on the above considerations to derive our metric as a linear combination of two main contributions. The *Overlap Score* (OS) penalizes the overlap of different FoVs in the same dimensions (Algorithm 2 — in this case, each row of S , associated with a dimension, is compared with the ideal array) and hence measures Modularity, while the *Multiple Encoding Score* (MES) penalizes the encoding of the same factor into different dimensions (Algorithm 3 — in this case each column

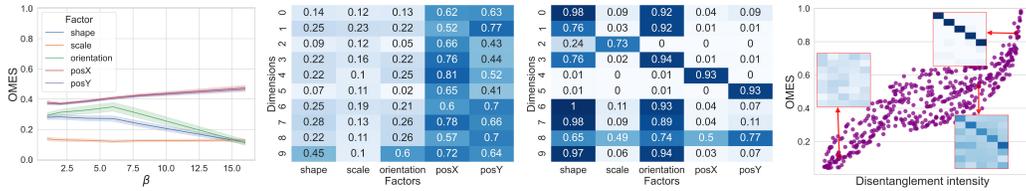


Figure 1: Dataset *Noisy-dSprites*: **Left**: Scores of the proposed metric for each FoV, α is fixed to 0.5. **Center Left**: Association matrix of an unsupervised model ($\beta = 6$). **Center Right**: Association matrix of a weakly-supervised model. **Right**: Scores of synthetic Association matrices simulating underfitting, partial disentanglement and almost perfect disentanglement.

of S corresponding to a FoV is compared with the ideal array) measuring Compactness. In both algorithms, we derive a vector summarizing the contribution of all dimensions for the FoV. The final score in $[0, 1]$ (higher values meaning higher disentanglement) can be obtained with a pooling (either MAX or AVERAGE) on the vector. The OMES metric is computed as

$$OMES(S) = \frac{1}{n} \sum_{j=1}^n \alpha OS(S, j) + (1 - \alpha) MES(S, j). \quad (1)$$

With $\alpha = 0$, OMES only measures the Compactness of the representation (MES component); with $\alpha = 1$, instead, our metric measures the Modularity only (with OS). Values of α in the interval $(0, 1)$ can be used to balance the importance of both contributions.

Relation with existing metrics. To the best of our knowledge, the only metrics capturing more than one property are DCI [13] and the very recent InfoMEC [25]. Differently from DCI, our metric is intervention-based with no influence on the choice of the specific classifier that may inevitably impact the results, as observed in [7]. With respect to InfoMEC, that must be applied to quantized latent codes, our metric is more general and accepts continuous latents.

OMES is based on the intervention of the FoVs, thus we require the FoV to be (at least partially) known: in particular, samples are coupled so that they differ in one FoV only. In this, OMES differs from existing intervention-based metrics [24, 27] in which the intervention is the opposite (samples have only one FoV in common). Our pairing requires less supervision, and it is usually easier to obtain during data acquisition (for instance, from videos [41]). In addition, it has been shown that this type of pairing provides more guarantees on disentanglement properties [60, 41].

Finally, compared to Information-based methods, we exploit Correlation instead of Mutual Information, hence we do not need its estimation that can be sensitive to parameters choice (e.g. granularity of the discretization [7]) and choice of estimator [51, 7].

2.3 OMES assessment

We now analyze OMES, extending previous studies on the unsupervised [40] and weakly supervised [41] setting. As for the *unsupervised case*, we exploit available 5400 trained models from [40] (3 datasets, 6 values for β , 50 random seeds, 6 unsupervised methods: β -VAE [24], FactorVAE [27], β -TCVAE [8], DIP-VAE-I [32], DIP-VAE-II[32], AnnealedVAE [5]); in this section we report an analysis on Noisy-dSprites, the remaining 2 benchmarks can be found in the Appendix B.1 and B.2. Instead, for the *weakly supervised case* trained models are not available, so we reproduce the models as in [41] training them on Shapes3D and on other datasets that can be found in Appendix B.3.

OMES interpretation. The metric, by construction, allows us to compute the overall score and a score for each FoV separately: we can thus interpret the effect of hyperparameters on the single FoV, and evaluate the FoV separately in each dimension of the representation. Moreover, by inspecting the metric at a factor level, we may identify uneven behaviours (e.g. models performing similarly on average but for different contributions from the factors).

Fig. 1 (Left) shows the metric scores for different values of β keeping the different FoV separated: the FoVs less affected by reconstruction (e.g. PosX and PosY) exhibit an increasing disentanglement score as β grows. On the other hand, Shape and Orientation present a maximum value around $\beta = 6$ and then decrease because they are more susceptible to the reconstruction quality, which degrades for larger values of β . In Fig. 1 (Center Left) an association matrix S is generated from one

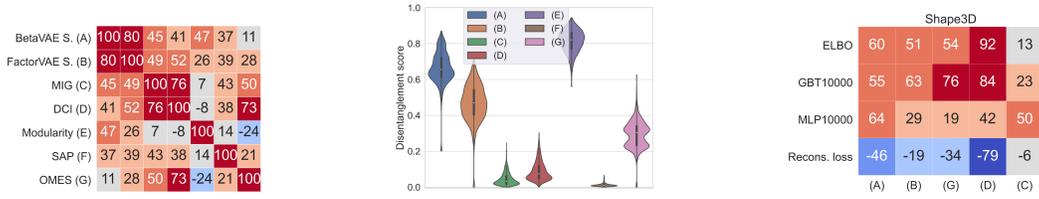


Figure 2: **Left:** Rank-correlation between metrics of models trained on *Noisy-dSprites*. **Center:** Scores distribution of the metrics on Noisy-dSprites. **Right:** Rank correlations (Spearman) of ELBO, reconstruction loss, and the test accuracy of a GBT and a MLP classifier trained on 10,000 labelled data points with disentanglement metrics. In all plots OMES is computed with $\alpha = 0.5$.

of the unsupervised models (β -VAE) trained with $\beta = 6$: *Shape* and *Orientation* are encoded in the same dimension (overlapping) and produce lower values because of the reconstruction, while *PosX* and *PosY* are encoded in multiple dimensions and mostly overlapping. *Scale* does not seem to be well represented. In Fig. 1 (Center Right) we report the association matrix obtained by a weakly-supervised model (Ada-GVAE): the factors *PosX*, *PosY* and *Scale* are disentangled while *Shape* and *Orientation* are encoded in the representation with high intensity in different dimensions, with overlaps. Fig.1 (Right) shows OMES values ($\alpha = 0.5$) computed over synthetic association matrices S , obtained by perturbing the ideal (diagonal) one. The perturbation aims to simulate 3 scenarios: noisy matrices where the disentanglement can be more or less strongly derived; models exhibiting partial disentanglement; models with no disentanglement. As it can be appreciated, the metric score nicely reflects the disentanglement intensity.

Agreement of OMES with other disentanglement metrics. We extend the analysis in [40]. The rank-correlation (e.g. Fig. 2 (Left)) between the previously proposed metrics and our OMES (for $\alpha = 0.5$) shows that the latter has a high level of correlation with MIG and DCI, but mild correlation with BetaVAE, FactorVAE (OMES is based on the opposite intervention type), and Modularity. This is consistent on all benchmarks. We show in Fig. 2 (Center) the score distribution of the metrics, computed on the whole set of models. We observe OMES produces a wider range of values with respect to MIG and DCI: our metric looks more descriptive, similarly to BetaVAE and FactorVAE.

Agreement with performance metrics. Similarly to [41], we consider the more informative weakly-supervised setting and discuss the rank correlation between our metric and performance evaluations (ELBO, reconstruction loss, and test error of FoVs classifier). Our analysis, reported in Fig. 2 (right) shows that OMES performs similarly to DCI, negatively correlated with the Reconstruction loss, and positively with the ELBO. It also correlates with the performances of GBT10000 (the classifier we will use in the experiments) while it mildly does with MLP10000. This empirical evidence is in line with what was observed in [11]. *The correlation with the classification score is a sign that OMES is able to capture the property of explicitness of the representation, although it is not directly measured by our metric.* It is worth mentioning the correlation of OMES with the performance metrics is more stable than what is obtained by other metrics across different datasets (see the Appendix B.3).

3 Transferring disentangled representations

Fully unsupervised disentangled representation learning has been shown unsatisfactory in many scenarios [40]. However, annotating the FoVs can be a very critical and uncertain process. In this section, we propose a general-purpose methodology for transferring disentangled representations learned from supervised synthetic or simulated data to an unsupervised dataset (in terms of the FoVs). This approach allows us to evaluate the effectiveness of disentangled representations transfer, and its potential in real-world applications.

3.1 Our methodology and research questions

Most of the focus in learning disentangled representations has been on synthetic datasets whose ground truth factors exhibit perfect independence by design [44, 53, 15, 34, 4]. Instead, real-world

Table 1: Summary of the datasets and their properties. * in the $\#FoV$ refers to the possible presence of hidden factors.

Dataset	Real	3D	Occlusions	#FoV	Independence	Complete annotation	Resolution	#Images
dSprites	✗	✗	✗	5	✓	✓	64 × 64	737K
Noisy-dSprites	✗	✗	✗	5	✓	✓	64 × 64	737K
Color-dSprites	✗	✗	✗	6	✓	✓	64 × 64	4,4M
Noisy-Color-dSprites	✗	✗	✗	6	✓	✓	64 × 64	4,4M
Shapes3D	✗	✓	✓	6	✓	✓	64 × 64	480K
Isaac3D	✗	✓	✓	9	✓	✓	128 × 128	737K
Coil100-Augmented	✓	✓	✓	4	✓	✓	128 × 128	1,1M
RGB-D Objects	✓	✓	✓	3*	✗	✗	256 × 256	35K

scenarios present several challenges that we want to investigate in our analysis.

We consider β -VAE models with weakly supervised learning specifically we adopted Ada-GVAE [41], for its simplicity and its sampling strategy similar to our metric, on a *Source* Dataset, using pairs of images that differ in ℓ factors of variation. We set $\ell = 1$ as it was shown to lead to higher disentanglement [41]. Following [11], we vary the parameter β in $\{1, 2\}$, sufficient to achieve high disentanglement with weak supervision [11, 41].

We evaluate the quality of the disentanglement in a transfer learning scenario, assessing the transferability of the disentangled representation on a *Target* Dataset, with the final aim of targeting real scenarios. The evaluation we report considers our metric OMES, as well as DCI and MIG, the most widely used metrics in the literature [50, 18, 17, 14, 69, 25, 45]. Moreover, in accordance to [40, 41, 11], we evaluate the quality of the disentanglement also in terms of accuracy w.r.t. a downstream classification task, with a classifier per FoV. We evaluate the latter in two modalities: (1) Considering the entire representation and (2) Selecting with OMES the dimension of the representation that best encodes the FoVs. Our analysis addresses three main research questions:

Q1 - *How well does disentanglement transfer, and how much does it depend on the distance between Source and Target Dataset?*

We will consider different transfer learning scenarios (syn2syn, syn2real, real2real) and pairs (*Source*, *Target*) datasets with different distances.

Q2 - *Which properties of a DR are preserved on the Target Dataset?*

We will discuss *Explicitness* of the representation (through FoV classification), *Compactness* (analysing the component MES of OMES, the MIG metric, as well as the performances of the one-dimensional representation), *Modularity* (relying on the OS component of OMES and on DCI).

Q3 - *How effective is fine-tuning on the disentanglement?*

We will consider the performances of the FoVs classification, the compactness and the modularity on the *Target dataset* before and after fine-tuning.

3.2 Datasets

In our analysis, we consider both synthetic and real datasets offering different challenges, a summary of their properties is in Tab. 1. Some of the datasets are *DRL-compliant*, meaning that there is full independence between the FoVs (this is reported in column *Independence*), and FoVs appear in all their possible combinations. This is easy to achieve if the dataset is specifically tailored for DRL, but it can not be easily obtained in general.

dSprites[44] is a dataset of 2D shapes generated from 5 ground truth FoVs: Shape, Scale, Rotation, x and y Positions. Variants of the dataset have been proposed: in **Noisy-dSprites** the background is filled with uniform noise; **Color-dSprites** includes Color as an additional FoV; **Noisy-Color-dSprites** adds uniform noise to the latter. We refer to them as: **N-dSprites**, **C-dSprites** and **N-C-dSprites**.

Shapes3D [4] is a dataset of 3D shapes, generated from 6 ground truth FoVs: Floor colour, Wall colour, Object colour, Scale, Shape and Orientation. It is characterized by the presence of *Occlusions*.

Isaac3D [47] is a synthetic dataset of a 3D scene of a kitchen where a robot arm is holding objects in a variety of configurations. It is characterized by 9 *real-world complex* FoVs, including robot movements, camera height, environmental conditions (e.g. lighting).

There are few real datasets available specifically meant for DRL. [20] is a collection of datasets covering the transitions from simulated to real data, which is, however, not fully available at the moment. [11] is not appropriate for our analysis since the real data section is very small compared to the complexity of the task. We consider instead real benchmarks proposed for classification tasks, chosen to reflect some of the real-world challenges but possessing some "semantic connection" with the synthetic dataset we refer to, e.g. in terms of the expected FoVs. This allows us to reason on the potential of transferability. Example images are in Appendix C.1.

Coil is derived from Coil100 [46]. The original dataset contains 7200 real color images of 100 objects. The objects were placed on a motorized turntable against a black background. The turntable was rotated to vary object pose w.r.t. a fixed camera, producing self-occlusions and 2D silhouette changes. We augment the original dataset with two additional FoV, a planar rotation (9 angles) and a scaling (18 values). Therefore, we identify 4 FoV (**Objects**, **Pose**, **Rotation** and **Scale**) that, by construction, are independent. To consider in our analysis a real dataset visually related to dSprite, we derived a binary version of Coil, called **Coil(bin)**, by applying Otsu's thresholding [49].

RGBD-Objects [33] is a dataset of 300 common household objects acquired by a RGB-D camera. The objects are organized into 51 **Categories** and a varying number of *instances* for each category. For each object, 3 video sequences have been acquired with different camera heights (**Elevation**) so that the object is viewed from different angles while rotating (**Pose**). Then, images have been cropped so that the object is always in a central position. For our experiments, we used a subset with one object instance per category to make it semantically similar to Coil100 but with the additional complexity of variability in the background, presence of occlusions and clutter. Hence, we control 3 FoVs (**Category**, **Elevation**, **Pose**), but other factors are *hidden or not annotated* (e.g. **Background**, **Illumination**, etc.) due to a realistic acquisition protocol. We refer to RGBD-Objects as **RGBD**. We also use a variant of the dataset, including depth maps only, referred to as **RGBD(depth)**.

3.3 Experimental analysis

Implementation details. We trained 20 different models (10 random seeds \times 2 values of β) for each *Source* dataset. We adopted the same training strategy as in [11] (see Appendix C.2). As for FoVs classification, following [11, 41], we consider Gradient Boosted Trees (GBT) [16] and a Multilayer Perceptron (MLP) [37] with 2 hidden layers of size 256. Since the specific choice of a classifier is not crucial for our analysis, here we report GBT, MLP can be found in Appendix C.7. Fine-tuning to the *Target* dataset of the VAE models is unsupervised and it is carried out for 50k steps.

Tables description. The tables group different experiments based on the *Target* dataset. For each FoV, we report under the name the number of values the factor can assume (i.e. its *granularity*). The tables report the average classification performance over the 20 models, before and after fine-tuning. The latter is reported in parenthesis in terms of gain or loss w.r.t. the performance before the fine-tuning. *All* is the average performance of all FoVs.

The column *Pruned* highlights the two different representation modalities: if the classifier is trained on the whole representation (**X**), or using only one dimension, i.e. the one showing the strongest encoding of a certain FoV according to the OMES metric (**✓**). As already mentioned, a good performance of the former is an indication of explicitness, while the latter is a positive sign of compactness. Tables also report metrics assessing Modularity (our MES and DCI) and Compactness (our OS and MIG).

Note that we exploit the interpretability of OMES in the transfer learning process to select the most representative dimension of the representation for the classification (the "Pruned" columns).

(1) Synthetic to synthetic. As a baseline, we consider the case in which both *Source* and *Target* datasets are synthetic and we have access to the annotation of the FoVs, they are DRL-compliant. If *Source* and *Target* have the same FoVs (S=dSprites with T=Noisy-dSprites or S=Color-dSprites with T=Noisy-Color-dSprites, see Table 2) we observe that pruning the representation to just one dimension maintains, on average, stable performances. This shows that the *compactness* of the representation is preserved for the *Target* dataset, both before and after fine-tuning.

Fine-tuning allows for improved performance in terms of *explicitness* preserving the remaining properties of the representation, also in the case of the pruned representation. The **Orientation** FoV is difficult in these datasets as it suffers from reconstruction errors. We increase complexity by adding a new FoV to the *Target* dataset (S=dSprite with T=Color-dSprite, see Table 2).

Table 2: Quantitative evaluation of transferred disentangled models using the dSprites family of datasets. We transfer from a Source (ST) to a Target Dataset (TD). We report the average classification accuracy obtained with GBT on the full and the pruned representations (see text). The last columns on the right report a comparison between disentanglement metrics, including MES and OS.

SD	TD	Pruned	Mean accuracy on FoVs(%)							Modularity(%)		Compactness(%)	
			Color (7)	Shape (3)	Scale (6)	Orientation (40)	PosX (32)	PosY (32)	All	Our (OS)	DCI	Our (MES)	MIG
dSprites	N-dSprites	✗		61.1	47.1	6.7	17.8	17.1	30.0				
		✓		(+11.2)	(+12.0)	(+7.5)	(+27.9)	(+28.1)	(+17.3)	31.8	22.0	32.1	13.9
	C-dSprites	✗	30.8	94.2	86.9	44.8	76.3	75.7	68.1				
		✓	(+35.4)	(-1.9)	(+0.5)	(-4.2)	(-5.1)	(-5.3)	(+3.2)	61.0	42.9	72.3	34.0
C-dSprites	N-C-dSprites	✗	26.2	76.7	77.0	17.1	74.9	74.6	57.8				
		✓	(+6.8)	(+2.6)	(+2.4)	(+1.0)	(-4.1)	(-4.3)	(+0.7)	(+1.6)	(+10.4)	(+3.5)	(+0.3)
	N-C-dSprites	✗	33.0	41.6	28.6	2.9	9.0	9.40	20.7				
		✓	(+65.2)	(+7.3)	(+16.8)	(+0.9)	(+19.7)	(+20.2)	(+21.7)	28.9	5.4	29.0	1.8
			29.1	39.3	25.3	2.6	6.0	5.3	17.9	(+0.3)	(+2.2)	(-1.9)	(+1.0)
			(+52.3)	(+4.5)	(+11.2)	(+0.4)	(+7.1)	(+7.4)	(+13.8)				

Table 3: Target dataset: Shapes3D (see Table 2).

SD	TD	Pruned	Mean accuracy on FoVs(%)							Modularity(%)		Compactness(%)	
			Floor Hue (10)	Wall Hue (10)	Object Hue (10)	Scale (8)	Shape (4)	Orientation (15)	All	Our (OS)	DCI	Our (MES)	MIG
dSprites		✗	78.0	80.3	43.6	25.4	55.5	35.3	53.0				
		✓	(+14.1)	(+13.1)	(+25.6)	(+9.6)	(+18.6)	(+0.8)	(+13.6)	26.9	9.6	23.9	5.3
	Shapes3D	✗	63.5	59.5	28.5	21.3	43.6	23.4	40.0				
		✓	(-2.0)	(+4.2)	(+13.7)	(+4.4)	(+11.4)	(-0.8)	(+5.1)	(+0.6)	(+9.0)	(+1.1)	(-0.4)
C-dSprites	Shapes3D	✗	82.1	79.4	46.8	30.1	53.7	39.8	55.3				
		✓	(+8.0)	(+12.3)	(+28.8)	(+10.4)	(+32.6)	(+5.1)	(+16.2)	30.6	13.7	28.2	8.4
	Isaac3D	✗	60.6	52.34	30.2	25.3	44.0	29.4	40.3				
		✓	(-5.8)	(+3.2)	(+12.1)	(+4.0)	(+18.5)	(-0.1)	(+5.3)	(+0.8)	(+7.4)	(+2.1)	(-1.4)
Isaac3D		✗	62.8	62.6	39.8	30.5	51.9	21.8	44.9				
		✓	(+26.2)	(+28.2)	(+37.9)	(+10.3)	(+35.2)	(+28.1)	(+27.6)	24.3	3.6	21.2	1.5
		✗	44.8	42.5	26.1	22.3	41.6	13.8	31.8				
		✓	(+8.3)	(+14.4)	(+16.5)	(+6.6)	(+20.3)	(+14.5)	(+13.4)	(+6.1)	(+16.7)	(+8.1)	(+5.2)

All FoVs in common between Source and Target are effectively classified, again except Orientation. As for the new FoV (Color), we report lower performances, but we can appreciate a significant improvement with fine-tuning if we exploit a global representation. Instead, we observe a lower improvement with the pruned representation, suggesting that the new factor is not encoded in one single dimension.

To further increase the distance between Source and Target, we consider pairs for which the semantics of the FoVs are the same, but they are different in appearance, granularity, and composition (S=Color-dSprite with T=Shapes3D, see Table 3): we can observe that even without fine-tuning, the latent representation allows the classification of the dominant FoVs of the dataset, i.e. Floor Hue and Wall Hue, also when focusing on a single dimension. Fine-tuning positively affects the average classification accuracy, especially when using the whole representation.

We finally reason on the gap between Source and Target datasets in terms of complexity. When the Source is simpler than the Target but still they have some FoVs in common, possibly with different appearances, (e.g. S=Shapes3D, T=Isaac3D, Table 9 and Table 10) we can appreciate the effectiveness of transfer and fine-tuning for all metrics. Conversely, when the Source is much more complex than the Target (e.g. when S=Isaac3D, T=Shapes3D) one could expect the richness in the Source to be directly transferrable to the simpler Target. However, we observe that the finetuning is still beneficial for all the disentanglement metrics. This can be explained by the “domain” dependence of VAE models.

Discussion. Disentanglement transfers well between synthetic datasets with the same FoVs, w.r.t. all the properties. If the Target includes new FoVs, fine-tuning is necessary for the new FoV, but also for the entire representation, as compactness and modularity are partially degraded by the new FoV. When the Source and Target become significantly different, fine-tuning is also beneficial. We can conclude that when both Source and Target are synthetic and DRL-compliant, the properties of disentangled representation are preserved before and after fine-tuning, especially when the datasets have FoVs in common even though they have different appearance.

(2) Synthetic to Real. We now analyse the potential of transferring a disentangled representation from an appropriately generated Synthetic Source (DRL-compliant) to a Real Target. We first consider Real

Table 4: Target dataset: Coil100 and variants (see Table 2).

SD	TD	Pruned	Mean accuracy on FoVs(%)					Modularity(%)		Compactness(%)	
			Object (100)	Pose (72)	Orientation (18)	Scale (9)	All	Our (OS)	DCI	Our (MES)	MIG
dSprites	Coil (bin)	✗	13.2 (+3.9)	1.7 (-0.1)	48.1 (+1.7)	38.6 (+7.4)	25.4 (+3.2)	37.8 (+2.6)	11.4 (+2.7)	25.7 (+2.1)	5.1 (+1.8)
		✓	4.8 (-0.2)	1.5 (+0.1)	18.0 (-5.2)	33.3 (+2.7)	14.4 (-0.7)				
C-dSprites		✗	23.3 (+20.9)	1.6 (+0.1)	34.6 (+13.2)	38.6 (+7.6)	24.5 (+10.4)	33.9 (+3.0)	10.4 (+1.8)	27.2 (+0.0)	4.1 (+2.2)
		✓	13.4 (+0.1)	1.5 (-0.1)	11.5 (+1.9)	33.2 (-3.2)	14.9 (-0.3)				
Shapes3D	Coil	✗	16.6 (+24.9)	1.5 (+0.1)	17.7 (+26.9)	32.5 (+12.4)	17.1 (+16.1)	31.7 (-0.2)	3.2 (+3.5)	25.0 (-1.0)	1.7 (+0.8)
		✓	6.2 (+4.6)	1.39 (+0.0)	9.72 (+4.5)	23.5 (+3.6)	10.21 (+3.2)				
Coil(bin)		✗	20.9 (+12.6)	1.6 (+0.0)	36.2 (+10.0)	30.4 (+12.8)	22.3 (+8.8)	36.3 (-5.2)	10.4 (-5.3)	26.1 (-2.4)	5.6 (-3.5)
		✓	7.3 (+2.6)	1.5 (-0.1)	12.8 (+1.8)	27.2 (-0.4)	12.2 (+1.0)				

Table 5: Target dataset: RGBD-Objects and variants (see Table 2).

SD	TD	Pruned	Mean accuracy on FoVs(%)				Modularity(%)		Compactness(%)	
			Category (51)	Elevation (4)	Pose (263)	All	Our (OS)	DCI	Our (MES)	MIG
dSprites		✗	64.3 (+4.3)	83.0 (+2.8)	0.3 (+0.0)	49.2 (+2.4)	34.3 (+0.6)	11.0 (+0.5)	22.1 (-0.2)	3.4 (+0.5)
		✓	34.1 (-4.3)	67.3 (-1.5)	0.3 (+0.0)	33.9 (-1.9)				
Shapes3D	RGBD (depth)	✗	56.2 (+15.4)	79.5 (+7.8)	0.3 (+0.0)	45.3 (+7.7)	35.3 (-0.8)	12.0 (-0.9)	22.4 (-0.5)	2.0 (+0.8)
		✓	25.9 (+11.7)	65.2 (+4.8)	0.3 (+0.0)	30.5 (+5.5)				
Coil(bin)		✗	17.3 (+5.5)	58.7 (+6.7)	0.3 (+0.0)	25.4 (+4.1)	35.6 (-1.5)	11.8 (+0.3)	23.5 (-2.0)	4.3 (-2.4)
		✓	9.3 (+3.2)	56.6 (-1.4)	0.3 (+0.0)	22.1 (+0.6)				
C-dSprites		✗	86.4 (+2.5)	63.7 (+1.3)	0.3 (+0.0)	50.1 (+1.3)	35.7 (-0.7)	7.8 (-2.3)	24.7 (-1.5)	1.8 (-0.4)
		✓	36.7 (+1.6)	48.6 (-0.8)	0.2 (+0.1)	28.5 (+0.3)				
Coil	RGBD	✗	83.5 (+6.9)	61.1 (+3.5)	0.3 (+0.1)	48.3 (+3.5)	35.0 (+0.0)	4.2 (+0.6)	23.5 (-0.3)	2.0 (-1.1)
		✓	35.7 (+0.1)	47.6 (-0.7)	0.3 (+0.0)	27.8 (-0.2)				
Coil(bin)		✗	79.3 (+9.3)	59.2 (+5.6)	0.3 (+0.0)	46.3 (+5.0)	35.3 (-0.5)	5.1 (-0.2)	24.2 (-1.2)	1.3 (-0.2)
		✓	36.3 (+0.4)	48.6 (-0.0)	0.2 (+0.0)	28.4 (+0.2)				

Targets with FoVs independence. In Table 4, we first analyse (S=Color-dSprite with T=Coil): the Target dataset shares some FoVs with the Source (such as Scale, inplane Orientation, and Object, the latter related to shape), but their variability and granularity may be very different (e.g. the shape/object can assume 3 values on Source and 100 on Target). Other FoV are new, such as Pose (encoding 3D rotations). The accuracy we achieve is very uneven on the different FoVs, in particular Pose, since the Source does not incorporate any 3D information. We notice an improvement with fine-tuning but also a degradation in performances with the pruned representation (a sign the representation does not produce a good disentanglement on the Target). An exception to this last comment is the FoV Scale, whose accuracy does not degrade significantly with one dimension only (an indication this FoV is well represented in one dimension).

With the same Target, we assess a representation which is incorporating some level of 3D information (S=Shapes3D, with T=Coil, Table 4). This choice does not bring any benefit since the synthetic Shapes3D includes a very simplified form of pose variation. Orientation accuracy degrades, as it is not captured by the Source (indeed, with fine tuning this performance improves). Considering the Target dataset, it clearly presents several new challenges w.r.t. Source ones, we produce a last experiment with a simplified binary version of the Target, meant to be more similar to the binary images in dSprite. In this case, some FoVs are very well represented (Orientation and Scale), while Object presents low performances due to the decrease in the image descriptive power caused by binarization.

We now consider another real Target, RGBD-Object (see Table 5). As for the former, we consider Color-dSprite as a synthetic source: here, the same considerations about the Pose we discussed before are valid, and on the other FoVs, we observe the global representation is effective (and marginally improved by fine-tuning), while the pruned representation leads to lower performances, as a sign the representation is not perfectly disentangled on the Target. Here again, we investigate the possibility of transferring to simplified versions of the dataset (in this case, we consider the Depth channel only). The FoV Elevation improves significantly as it is not directly affected by color information. Concerning Modularity (Table 4 and Table 5), fine-tuning seems to have a small influence. In particular, we observe a small degradation for RGBD-Objects (Table 5), coherent with the challenges of a real dataset with unknown/hidden factors perturbing the encoding of the FoV.

Discussion. If the Source is synthetic (and DRL-compliant), and the Target is Real, the quality of transferring seems to depend on the distance between datasets and is not even across different FoVs: FoVs that are more similar between the datasets are more easily represented, and they exhibit better compactness properties. Fine-tuning is bringing a significant benefit in Explicitness and some

(limited) benefit in Modularity and Compactness. If the Target incorporates unknown hidden factors, as we may expect to happen in the real world, Modularity and Compactness transfer worse, and the benefit of fine-tuning is limited.

(3) Real to Real. We conclude by discussing the possibility of transferring from a DRL-compliant real dataset to another real one. As a first task, we consider as a Source a simplified version of the Target (specifically, S=Coil-binary, with T=Coil): the source should encode the factors not related to RGB, while the finetuning should improve the disentanglement and the explicitness of the representation. However, this is not the case with Coil100, whose representations degrade the Modularity, and the finetuning only affects the entire representation.

We then consider a larger variation between Real Source and Real Target (specifically, S=Coil with T=RGBD-Object, see Table 5): we obtain similar results to those of Color-dSprites as Source Dataset (comparable Explicitness), with a reduction on the performances obtained by the pruned representation. Notice that adopting the binary Coil as a Source causes only a limited reduction in Explicitness, and this was somewhat unexpected as we have a large gap in complexity between Source and Target. Our experiments did not consider RGBD-Objects acting as a Source dataset, not being DRL-compliant.

Discussion. Using a real DRL-compliant dataset as a Source, we do not appreciate any benefit. Fine-tuning is not particularly effective. At the same time, we notice that some level of disentanglement transfer can be observed.

4 Limitations

A limitation of our current work is the adoption of a specific family of approaches (VAE-based). The generalization of our finding to more recent vector-based approaches (e.g. [71, 62, 48, 36]) needs further investigation. However, each family of approaches for disentanglement learning *follows specific paradigms that may require tailored designs for transfer learning*. In other words, while the general transfer methodology is still applicable, it might need proper tuning to perform optimally depending on the particular learning approach.

5 Conclusions

In this paper, we discussed the potential of transferring a Disentangled Representation as a strategy to address disentanglement in real data. We learned the representation from a Source Dataset in a weakly supervised manner and transferred it to a Target Dataset, where supervision on the FoVs was difficult or impossible to obtain. We identified three main scientific questions, summarised in Section 3.2, which we recall to draw conclusions on our study. Starting from question **Q2**, on the properties of disentangled representations that are preserved after transferring, we may conclude Explicitness is usually well maintained, while Modularity and Compactness are reduced as we move from synthetic to real. More precisely, we appreciate a degradation in the global metrics (such as OS and ME), while on the compactness through the analysis of the 1-dimensional pruned representations, we notice that some FoV may transfer very well.

As for **Q3**, we may observe that fine-tuning is almost always beneficial, and it never causes any harm. **Q1**, a much wider question discussing under what circumstances transfer is effective, leads us to conclude that some structural similarity between Source and Target datasets is necessary, including similar ranges/granularity of variations of related factors. A quantification of the similarity among datasets is still under investigation; *the results of our study suggest one could design synthetic data to capture/disentangle specific factors of interest*.

Future directions. Currently, we are exploring quantitative methods to assess the distance between Source and Target datasets. In the near future will target more specific applications, such as biomedical image classification or action recognition from videos, to discuss and relate the general results we are reporting in this paper to more specific and challenging domains.

Acknowledgement

We acknowledge the financial support from PNRR MUR Project PE0000013 "Future Artificial Intelligence Research (FAIR)", funded by the European Union – NextGenerationEU, CUP J33C24000430007

References

- [1] Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence* **35**(8), 1798–1828 (2013)
- [2] Bouchacourt, D., Tomioka, R., Nowozin, S.: Multi-level variational autoencoder: Learning disentangled representations from grouped observations. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 32 (2018)
- [3] Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349* (2015)
- [4] Burgess, C., Kim, H.: 3d shapes dataset. <https://github.com/deepmind/3dshapes-dataset/> (2018)
- [5] Burgess, C.P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., Lerchner, A.: Understanding disentangling in β -vae. *arXiv preprint arXiv:1804.03599* (2018)
- [6] Cao, J., Nai, R., Yang, Q., Huang, J., Gao, Y.: An empirical study on disentanglement of negative-free contrastive learning. *Advances in Neural Information Processing Systems* **35**, 1210–1222 (2022)
- [7] Carbonneau, M.A., Zaidi, J., Boilard, J., Gagnon, G.: Measuring disentanglement: A review of metrics. *IEEE transactions on neural networks and learning systems* (2022)
- [8] Chen, R.T., Li, X., Grosse, R.B., Duvenaud, D.K.: Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems* **31** (2018)
- [9] Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *Advances in neural information processing systems* **29** (2016)
- [10] Dang, H., Huu, T.T., Nguyen, T.M., Ho, N.: Beyond vanilla variational autoencoders: Detecting posterior collapse in conditional and hierarchical variational autoencoders. In: *The Twelfth International Conference on Learning Representations* (2024), <https://openreview.net/forum?id=4zZFG1iCl9>
- [11] Dittadi, A., Träuble, F., Locatello, F., Wüthrich, M., Agrawal, V., Winther, O., Bauer, S., Schölkopf, B.: On the transfer of disentangled representations in realistic settings. *arXiv preprint arXiv:2010.14407* (2020)
- [12] Do, K., Tran, T.: Theory and evaluation metrics for learning disentangled representations. In: *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net (2020), <https://openreview.net/forum?id=HJgK0h4Ywr>
- [13] Eastwood, C., Williams, C.K.: A framework for the quantitative evaluation of disentangled representations. In: *International conference on learning representations* (2018)
- [14] Eddahmani, I., Pham, C.H., Napoléon, T., Badoc, I., Fouefack, J.R., El-Bouz, M.: Unsupervised learning of disentangled representation via auto-encoding: A survey. *Sensors* **23**(4), 2362 (2023)
- [15] Fidler, S., Dickinson, S., Urtasun, R.: 3d object detection and viewpoint estimation with a deformable 3d cuboid model. *Advances in neural information processing systems* **25** (2012)
- [16] Friedman, J.H.: Greedy function approximation: a gradient boosting machine. *Annals of statistics* pp. 1189–1232 (2001)

- [17] Fumero, M., Cosmo, L., Melzi, S., Rodola, E.: Learning disentangled representations via product manifold projection. In: Meila, M., Zhang, T. (eds.) Proceedings of the 38th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 139, pp. 3530–3540. PMLR (18–24 Jul 2021), <https://proceedings.mlr.press/v139/fumero21a.html>
- [18] Fumero, M., Wenzel, F., Zancato, L., Achille, A., Rodolà, E., Soatto, S., Schölkopf, B., Locatello, F.: Leveraging sparse and shared feature activations for disentangled representation learning. *Advances in Neural Information Processing Systems* **36** (2024)
- [19] Gabbay, A., Hoshen, Y.: Scaling-up disentanglement for image translation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6783–6792 (2021)
- [20] Gondal, M.W., Wuthrich, M., Miladinovic, D., Locatello, F., Breidt, M., Volchkov, V., Akpo, J., Bachem, O., Schölkopf, B., Bauer, S.: On the transfer of inductive bias from simulation to the real world: a new disentanglement dataset. *Advances in Neural Information Processing Systems* **32** (2019)
- [21] Gonzalez-Garcia, A., Van De Weijer, J., Bengio, Y.: Image-to-image translation for cross-domain disentanglement. *Advances in neural information processing systems* **31** (2018)
- [22] Goodfellow, I., Lee, H., Le, Q., Saxe, A., Ng, A.: Measuring invariances in deep networks. *Advances in neural information processing systems* **22** (2009)
- [23] Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., Lerchner, A.: Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230* (2018)
- [24] Higgins, I., Matthey, L., Pal, A., Burgess, C.P., Glorot, X., Botvinick, M.M., Mohamed, S., Lerchner, A.: beta-vae: Learning basic visual concepts with a constrained variational framework. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, 2017, Conference Track Proceedings. OpenReview.net (2017), <https://openreview.net/forum?id=Sy2fzU9g1>
- [25] Hsu, K., Dorrell, W., Whittington, J., Wu, J., Finn, C.: Disentanglement via latent quantization. *Advances in Neural Information Processing Systems* **36** (2024)
- [26] Kahana, J., Hoshen, Y.: A contrastive objective for learning disentangled representations. In: European Conference on Computer Vision. pp. 579–595. Springer (2022)
- [27] Kim, H., Mnih, A.: Disentangling by factorising. In: International Conference on Machine Learning. pp. 2649–2658. PMLR (2018)
- [28] Kim, M., Wang, Y., Sahu, P., Pavlovic, V.: Relevance factor vae: Learning and identifying disentangled factors. *arXiv preprint arXiv:1902.01568* (2019)
- [29] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014)
- [30] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013)
- [31] Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.: Deep convolutional inverse graphics network. *Advances in neural information processing systems* **28** (2015)
- [32] Kumar, A., Sattigeri, P., Balakrishnan, A.: Variational inference of disentangled latent concepts from unlabeled observations. *arXiv preprint arXiv:1711.00848* (2017)
- [33] Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: 2011 IEEE international conference on robotics and automation. pp. 1817–1824. IEEE (2011)
- [34] LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 2, pp. II–104. IEEE (2004)

- [35] Lee, S., Cho, S., Im, S.: Dranet: Disentangling representation and adaptation networks for unsupervised cross-domain adaptation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 15252–15261 (2021)
- [36] Lin, Z., Thekumparampil, K., Fanti, G., Oh, S.: Infogan-cr and modelcentrality: Self-supervised model training and selection for disentangling gans. In: international conference on machine learning. pp. 6127–6139. PMLR (2020)
- [37] Lippmann, R.: Book review: Neural networks, a comprehensive foundation, by simon haykin. *Int. J. Neural Syst.* **5**(4), 363–364 (1994). <https://doi.org/10.1142/S0129065794000372>, <https://doi.org/10.1142/S0129065794000372>
- [38] Liu, Y., Sangineto, E., Chen, Y., Bao, L., Zhang, H., Sebe, N., Lepri, B., Wang, W., De Nadai, M.: Smoothing the disentangled latent style space for unsupervised image-to-image translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10785–10794 (2021)
- [39] Locatello, F., Abbati, G., Rainforth, T., Bauer, S., Schölkopf, B., Bachem, O.: On the fairness of disentangled representations. *Advances in neural information processing systems* **32** (2019)
- [40] Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., Bachem, O.: Challenging common assumptions in the unsupervised learning of disentangled representations. In: international conference on machine learning. pp. 4114–4124. PMLR (2019)
- [41] Locatello, F., Poole, B., Rätsch, G., Schölkopf, B., Bachem, O., Tschannen, M.: Weakly-supervised disentanglement without compromises. In: International Conference on Machine Learning. pp. 6348–6359. PMLR (2020)
- [42] Lu, Z., Wu, C., Chen, X., Wang, Y., Bai, L., Qiao, Y., Liu, X.: Hierarchical diffusion autoencoders and disentangled image manipulation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5374–5383 (2024)
- [43] Ma, L., Sun, Q., Georgoulis, S., Van Gool, L., Schiele, B., Fritz, M.: Disentangled person image generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 99–108 (2018)
- [44] Matthey, L., Higgins, I., Hassabis, D., Lerchner, A.: dsprites: Disentanglement testing sprites dataset. <https://github.com/deepmind/dsprites-dataset/> (2017)
- [45] Mo, S., Sun, Z., Li, C.: Representation disentanglement in generative models with contrastive learning. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1531–1540 (2023)
- [46] Nayar, Murase, H.: Columbia object image library: Coil-100. Tech. Rep. CUCS-006-96, Department of Computer Science, Columbia University (February 1996)
- [47] Nie, W.: High resolution disentanglement datasets. <https://github.com/NVlabs/High-res-disentanglement-datasets> (2019)
- [48] Nie, W., Karras, T., Garg, A., Debnath, S., Patney, A., Patel, A.B., Anandkumar, A.: Semi-supervised stylegan for disentanglement learning. In: Proceedings of the 37th International Conference on Machine Learning. pp. 7360–7369 (2020)
- [49] Otsu, N.: A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics* **9**(1), 62–66 (1979)
- [50] Pandey, A., Fanuel, M., Schreurs, J., Suykens, J.A.: Disentangled representation learning and generation with manifold optimization. *Neural Computation* **34**(10), 2009–2036 (2022)
- [51] Paninski, L.: Estimation of entropy and mutual information. *Neural computation* **15**(6), 1191–1253 (2003)
- [52] Pham, C.H., Ladjal, S., Newson, A.: Pca-ae: Principal component analysis autoencoder for organising the latent space of generative networks. *Journal of Mathematical Imaging and Vision* **64**(5), 569–585 (2022)

- [53] Reed, S., Sohn, K., Zhang, Y., Lee, H.: Learning to disentangle factors of variation with manifold interaction. In: International conference on machine learning. pp. 1431–1439. PMLR (2014)
- [54] Ridgeway, K.: A survey of inductive biases for factorial representation-learning. CoRR **abs/1612.05299** (2016), <http://arxiv.org/abs/1612.05299>
- [55] Ridgeway, K., Mozer, M.C.: Learning deep disentangled embeddings with the f-statistic loss. *Advances in neural information processing systems* **31** (2018)
- [56] Sarhan, M.H., Navab, N., Eslami, A., Albarqouni, S.: Fairness by learning orthogonal disentangled representations. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIX 16. pp. 746–761. Springer (2020)
- [57] Schmidhuber, J.: Learning factorial codes by predictability minimization. *Neural computation* **4**(6), 863–879 (1992)
- [58] Schölkopf, B., Locatello, F., Bauer, S., Ke, N.R., Kalchbrenner, N., Goyal, A., Bengio, Y.: Toward causal representation learning. *Proceedings of the IEEE* **109**(5), 612–634 (2021)
- [59] Shi, Y., Yang, X., Wan, Y., Shen, X.: Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11254–11264 (2022)
- [60] Shu, R., Chen, Y., Kumar, A., Ermon, S., Poole, B.: Weakly supervised disentanglement with guarantees. arXiv preprint arXiv:1910.09772 (2019)
- [61] Sønderby, C.K., Raiko, T., Maaløe, L., Sønderby, S.K., Winther, O.: Ladder variational autoencoders. *Advances in neural information processing systems* **29** (2016)
- [62] Song, Y., Keller, A., Sebe, N., Welling, M.: Flow factorized representation learning. *Advances in Neural Information Processing Systems* **36** (2024)
- [63] Steenbrugge, X., Leroux, S., Verbelen, T., Dhoedt, B.: Improving generalization for abstract reasoning tasks using disentangled feature representations. arXiv preprint arXiv:1811.04784 (2018)
- [64] Suter, R., Miladinovic, D., Schölkopf, B., Bauer, S.: Robustly disentangled causal mechanisms: Validating deep representations for interventional robustness. In: International Conference on Machine Learning. pp. 6056–6065. PMLR (2019)
- [65] Träuble, F., Creager, E., Kilbertus, N., Locatello, F., Dittadi, A., Goyal, A., Schölkopf, B., Bauer, S.: On disentangled representations learned from correlated data. In: International Conference on Machine Learning. pp. 10401–10412. PMLR (2021)
- [66] Van Steenkiste, S., Locatello, F., Schmidhuber, J., Bachem, O.: Are disentangled representations helpful for abstract visual reasoning? *Advances in neural information processing systems* **32** (2019)
- [67] Wang, X., Chen, H., Tang, S., Wu, Z., Zhu, W.: Disentangled representation learning (2023)
- [68] Wang, Y., Blei, D., Cunningham, J.P.: Posterior collapse and latent variable non-identifiability. *Advances in Neural Information Processing Systems* **34**, 5443–5455 (2021)
- [69] Watters, N., Matthey, L., Burgess, C.P., Lerchner, A.: Spatial broadcast decoder: A simple architecture for learning disentangled representations in vaes. arXiv preprint arXiv:1901.07017 (2019)
- [70] Xiang, S., Gu, Y., Xiang, P., Chai, M., Li, H., Zhao, Y., He, M.: Disunknown: Distilling unknown factors for disentanglement learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14810–14819 (October 2021)
- [71] Yang, T., Wang, Y., Lu, Y., Zheng, N.: Disdiff: Unsupervised disentanglement of diffusion probabilistic models. *Advances in Neural Information Processing Systems* **36** (2024)

- [72] Zhu, J.Y., Zhang, Z., Zhang, C., Wu, J., Torralba, A., Tenenbaum, J., Freeman, B.: Visual object networks: Image generation with disentangled 3d representations. *Advances in neural information processing systems* **31** (2018)
- [73] Zhu, X., Xu, C., Tao, D.: Where and what? examining interpretable disentangled representations. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5861–5870 (2021)

Table 6: Summary of different metrics for disentanglement learning. L and K are the numbers of latent variables and ground truth factors, respectively.

Metric	Intervention based	Information based	Predictor based	Classifier	#Classifiers	Measured Property
BetaVAE	✓	✗	✓	Linear/majority-vote	1	Modularity
FactorVAE	✓	✗	✓	Linear/majority-vote	1	Modularity
SAP	✗	✗	✓	Threshold value	$L \times K$	Compactness
MIG	✗	✓	✗	None	0	Compactness
DCI	✗	✗	✓	LASSO	K	Modularity
				/		Compactness
Modularity	✗	✓	✗	Random forest	0	Explicitness
				None		Modularity
OMES(Our)	✓	✓	✗	None	0	Modularity Compactness

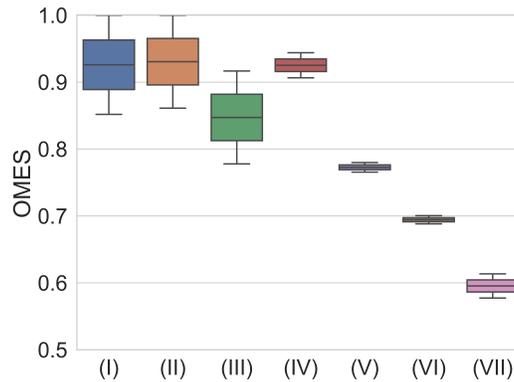


Figure 3: Boxplot of the distribution of OMES comparing 7 association matrices S , for different α values: (I), (II) and (III) are the results of simulated scenarios where only Overlap (I) and Multiple Encoding (II) or both (III) are represented, (IV), (V), (VI) are obtained with weak-supervision (respectively, on datasets Shapes3D, Color-dSprites, Noisy-dSprites; (VII) Noisy-dSprite with unsupervised model.

A Evaluating the quality of disentanglement

Table 6 reports the main characteristics of the well-established and most used disentanglement metrics. Note that OMES is the only one both Interventional-based and Information-based, measuring Modularity and Compactness.

B OMES assessment

In this section, we report the evaluation of the 1800 models trained on *Noisy-dSprites*, 1800 models trained on *SmallNORB* and 1800 models trained on *Cars3D*, all from [40]. Here we report the extensions of the results in Section 2.3

B.1 OMES interpretation

Fig. 4 shows our metric OMES scores for different values of β keeping the different FoV separated, for Noisy-dSprites (**Left**), SmallNORB (**Center**) and Cars3D (**Right**). α is fixed to 0.5.

In addition, Figure 3 shows the range of values with different α for a selection of S . We include 3 synthetic cases ((I), (III), (IV)) producing high scores, and (V) generated from Shapes3D, is very similar. The scores of the Noisy-dSprites models ((VI), (VII)) are lower, as it is a more challenging dataset. Color-dSprites (V) is easier to disentangle and output values in between Shapes3D and Noisy-dSprites. Note how the boxplots generated from real models produce a smaller range of values w.r.t the simulated cases; the choice of α does not appear critical in the real case.

B.2 Agreement of OMES with other disentanglement metrics

Fig. 4 shows the metric scores for different values of β keeping the different FoV with $\alpha = 0.5$, the 3 benchmark datasets (Noisy-dsprites (Left), SmallNORB (Center), Cars3D (Right)). In general the greater the β the higher the disentanglement but the factors strictly related to reconstruction quality fail to be encoded in the representation.

Fig. 5 shows the distribution of the disentanglement metrics, extending the plot from [40] with our metric OMES computed with different values of $\alpha \in \{0.0, 0.3, 0.5, 0.8, 1.0\}$. We observe the higher the α the less variable the distributions of our metrics, meaning that the models are more similar in terms of *Multiple Encoding* than they are in terms of *Overlap*.

Fig. 6 shows the rank correlations of the disentanglement metrics of the models trained on *Noisy-dSprites*, extending the plot from [40] with our metric OMES computed with different values of $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$. We observe the higher the α (*Multiple Encoding*) the higher the correlations with BetaVAE Score and FactorVAE Score and negative correlations with Modularity.

Analogously, Fig. 7 shows the rank correlations of the disentanglement metrics of the models trained on *SmallNORB*, extending the plot from [40] with our metric OMES computed with different values of $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

Finally, Fig. 8 shows the rank correlations of the disentanglement metrics of the models trained on *Cars3D*, extending the plot from [40] with our metric OMES computed with different values of $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

B.3 Agreement with performance metrics

Fig. 9 shows Rank correlation with ELBO, reconstruction loss, and test error of FoVs classifier for the models trained on the weak-supervised setting that was shown to be more interesting for this analysis. We consider all the different Source datasets used in the transfer experiments.

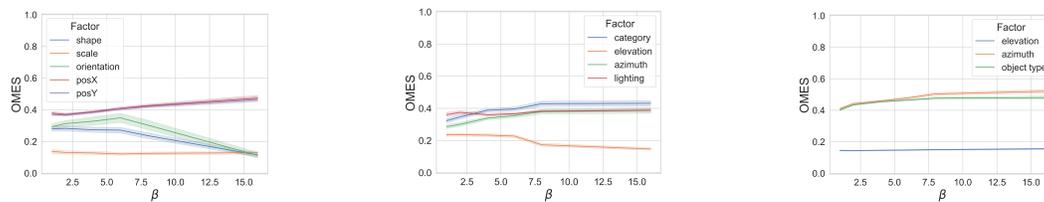


Figure 4: Scores of the proposed metric for each FoV (α is fixed to 0.5) of the 5400 models in [40]: 1800 models trained on Noisy-dsprites (Left); 1800 models trained on SmallNORB (Center); 1800 models trained on Cars3D (Right).

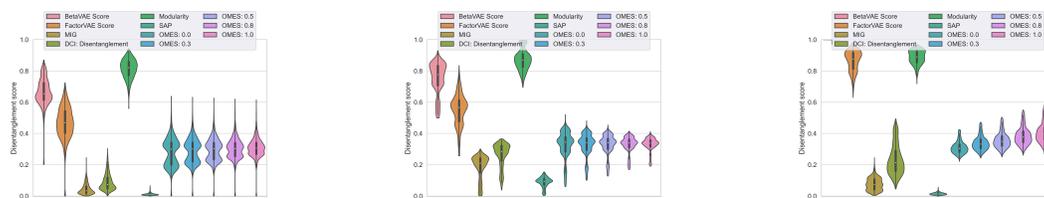


Figure 5: Distribution of different metrics of the 5400 models in [40]: 1800 models trained on Noisy-dsprites (Left); 1800 models trained on SmallNORB (Center); 1800 models trained on Cars3D (Right). This is an extension of the plots in [40], we added our metrics with different values of $\alpha \in \{0.0, 0.3, 0.5, 0.8, 1.0\}$.

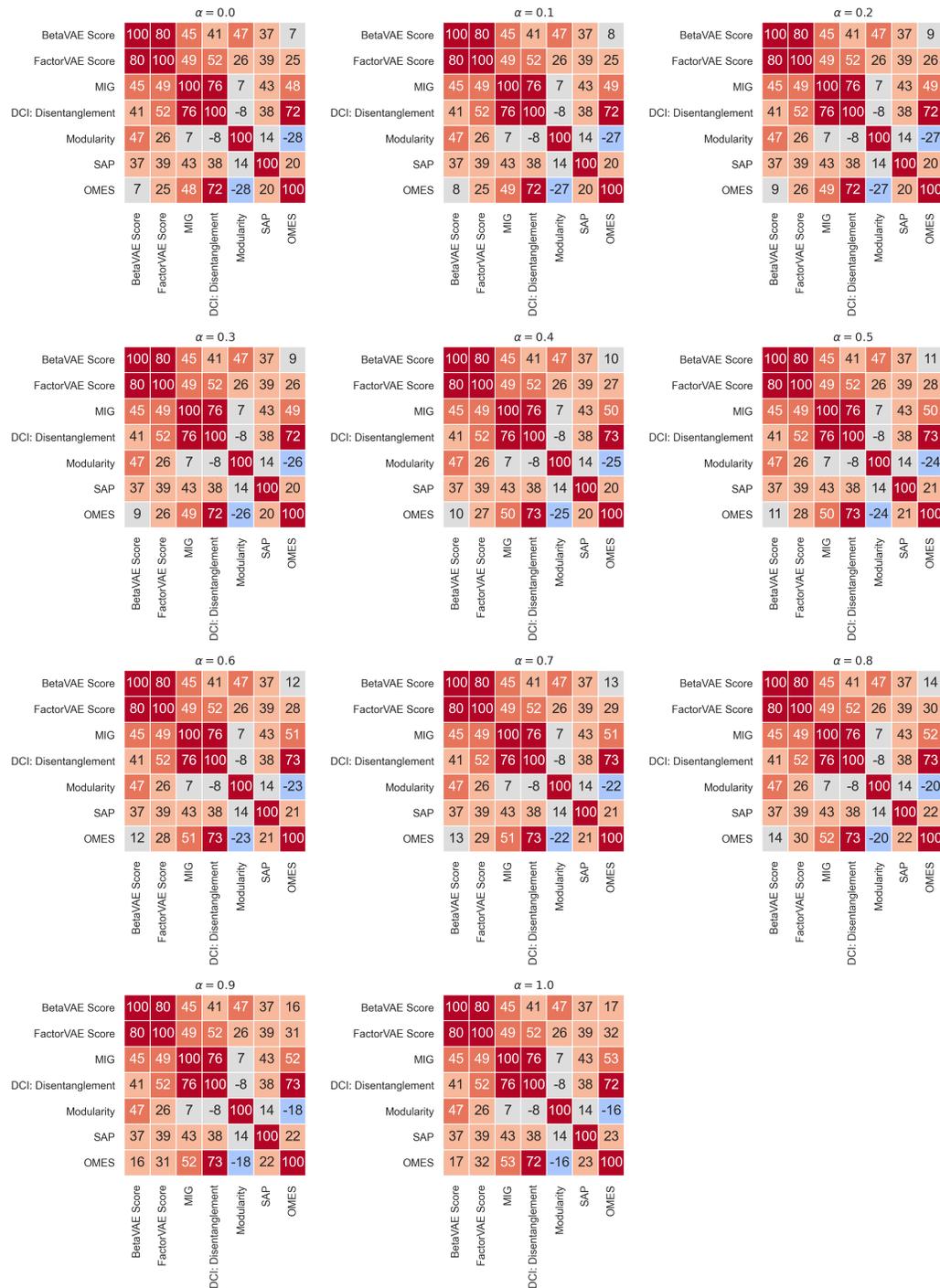


Figure 6: Rank correlation of different metrics on the same dataset (*Noisy-dSprites*) computed on the 1800 models in [40]. This is an extension of the plots in [40], we added our metrics with different values of $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

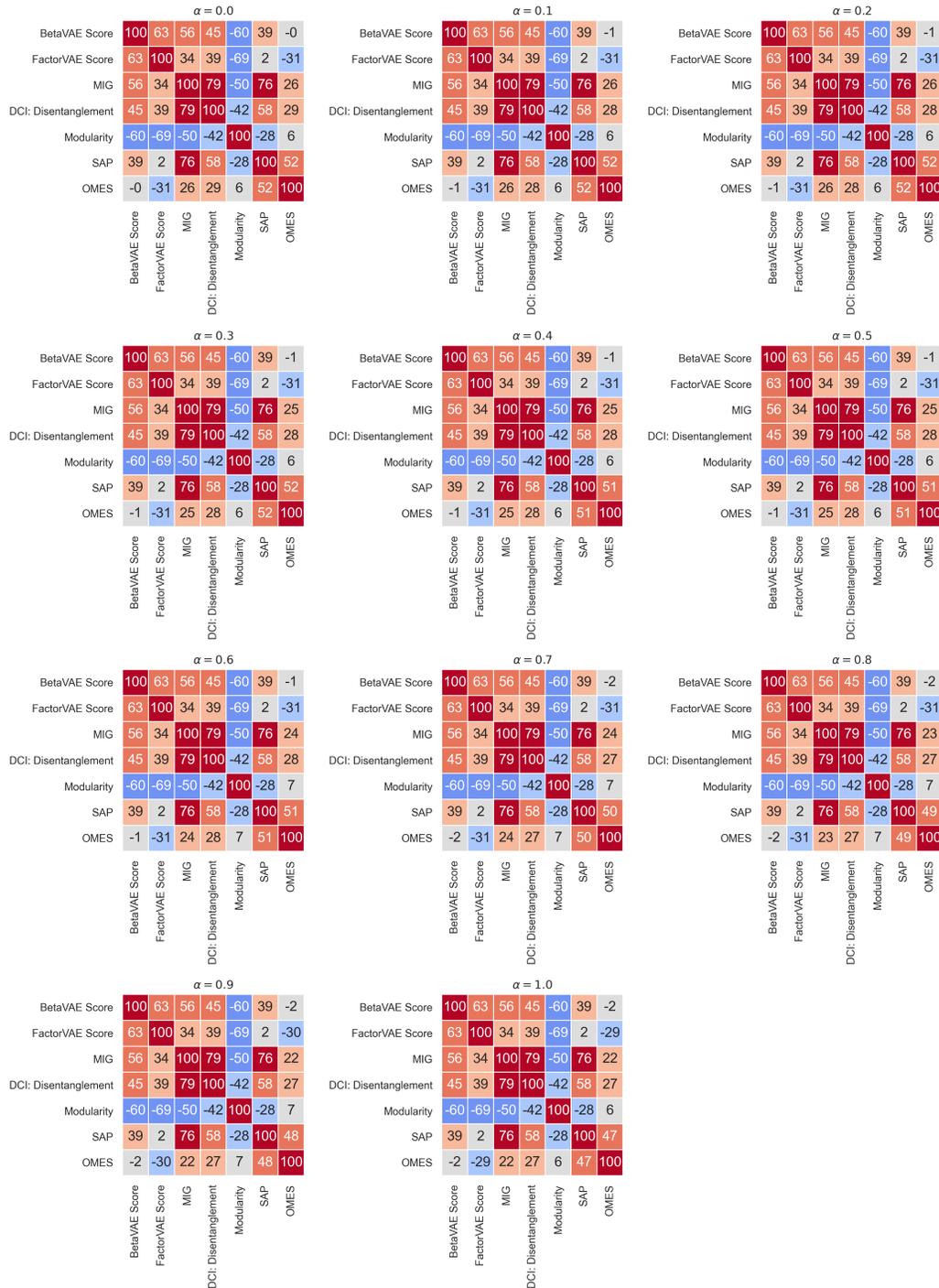


Figure 7: Rank correlation of different metrics on the same dataset (*SmallNORB*) computed on the 1800 models in [40]. This is an extension of the plots in [40], we added our metrics with different values of $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

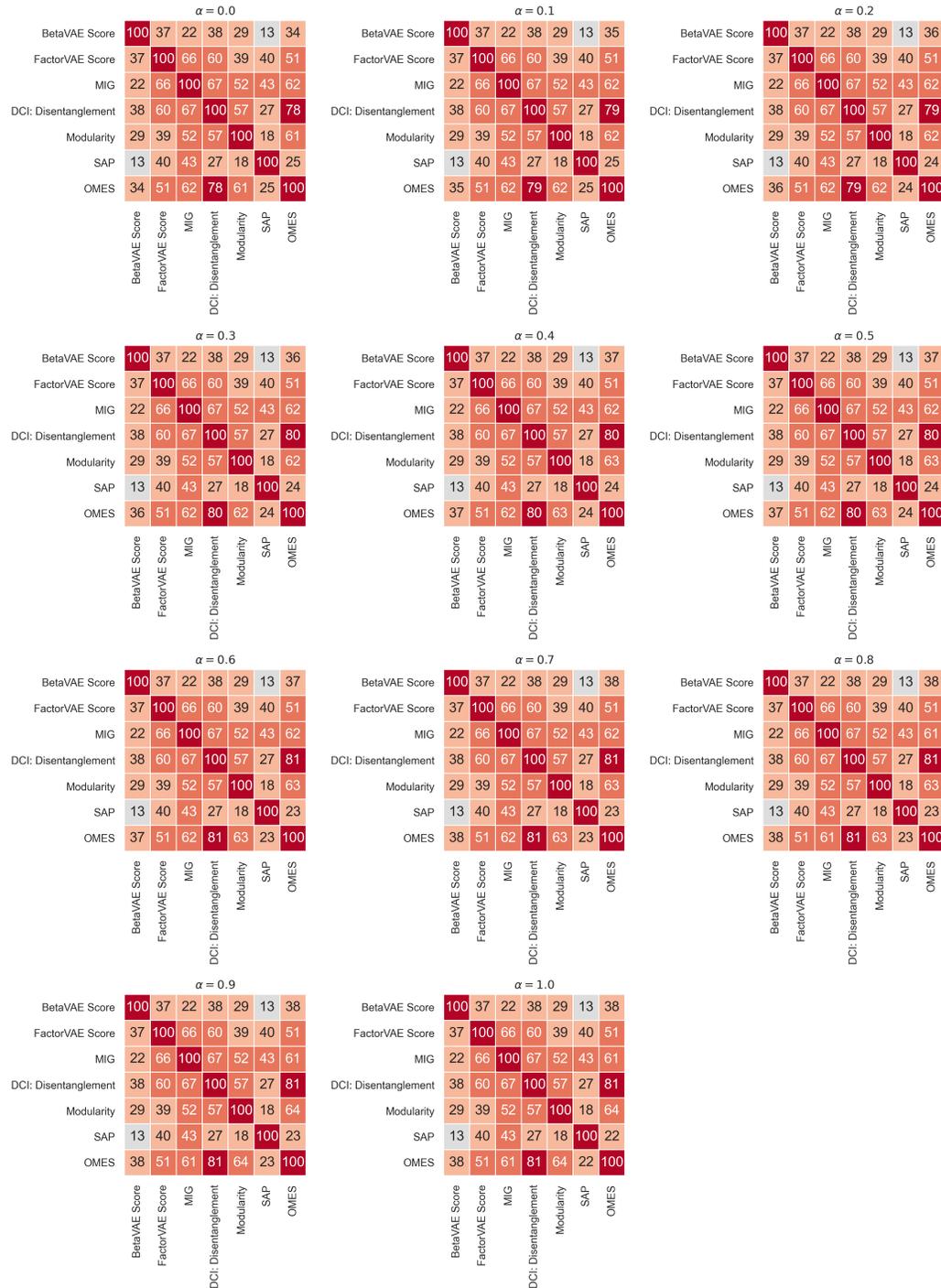


Figure 8: Rank correlation of different metrics on the same dataset (*Cars3D*) computed on the 1800 models in [40]. This is an extension of the plots in [40], we added our metrics with different values of $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

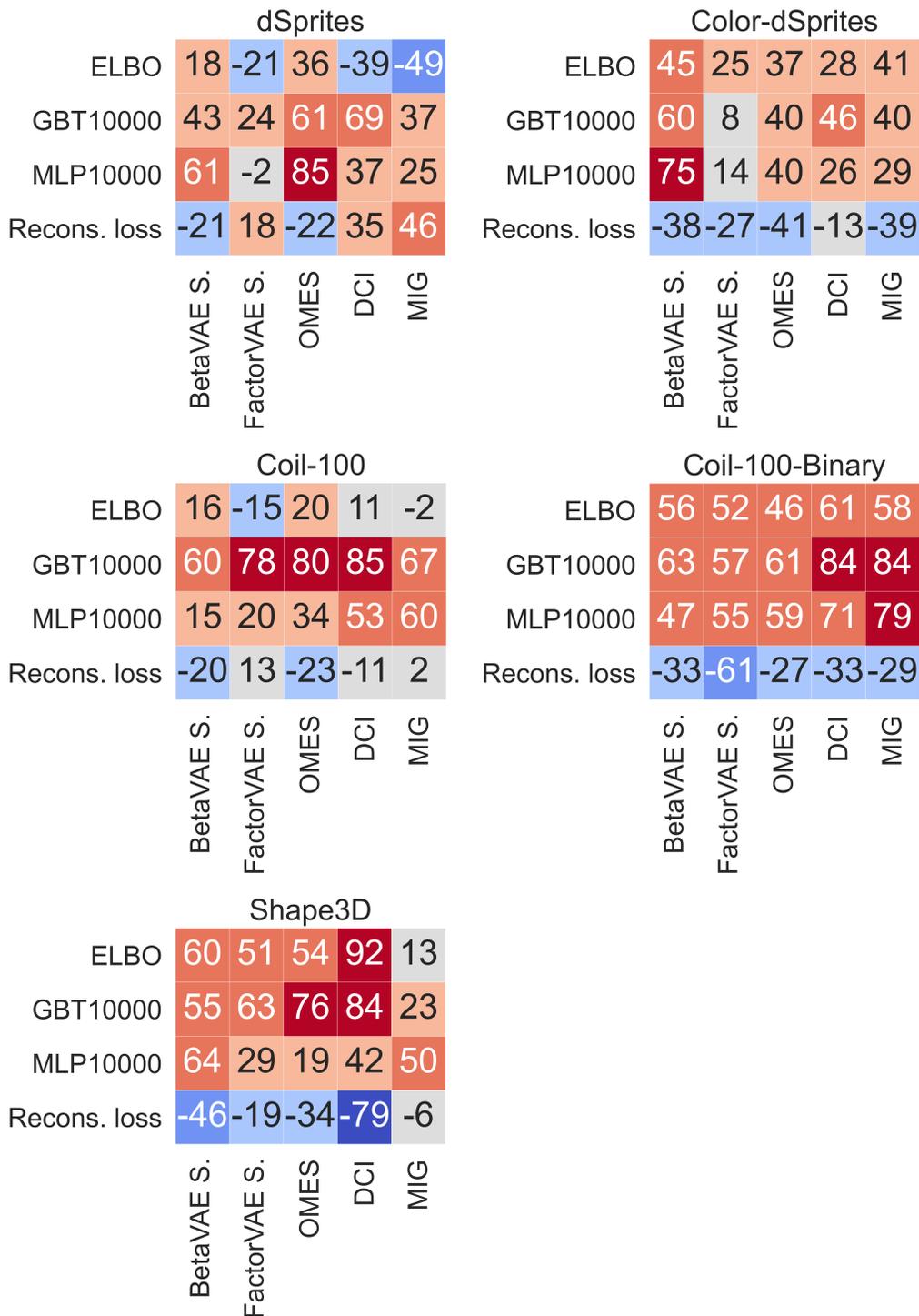
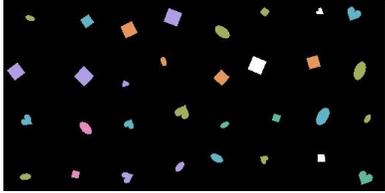
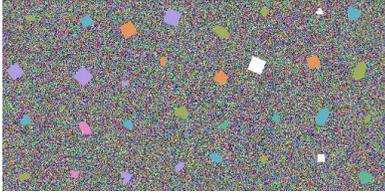
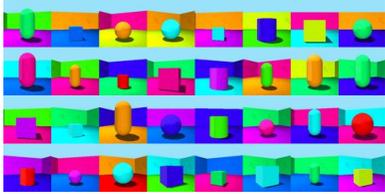
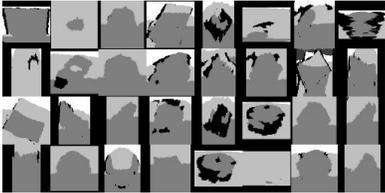


Figure 9: Rank correlations (Spearman) of ELBO, reconstruction loss, and the test accuracy of a GBT and a MLP classifier trained on 10,000 labelled data points with disentanglement metrics. In all plots OMES is computed with $\alpha = 0.5$.

Table 7: Datasets info and examples. Specifically, the *Variant* column shows the corresponding variants of the original dataset

Dataset FoV	Original	Variant																
<table border="1"> <thead> <tr> <th colspan="2">Color-dSprites</th> </tr> <tr> <th>FoV</th> <th># values</th> </tr> </thead> <tbody> <tr> <td>Color</td> <td>7</td> </tr> <tr> <td>Shape</td> <td>3</td> </tr> <tr> <td>Scale</td> <td>6</td> </tr> <tr> <td>Orientation</td> <td>40</td> </tr> <tr> <td>PosX</td> <td>32</td> </tr> <tr> <td>PosY</td> <td>32</td> </tr> </tbody> </table>	Color-dSprites		FoV	# values	Color	7	Shape	3	Scale	6	Orientation	40	PosX	32	PosY	32		
Color-dSprites																		
FoV	# values																	
Color	7																	
Shape	3																	
Scale	6																	
Orientation	40																	
PosX	32																	
PosY	32																	
<table border="1"> <thead> <tr> <th colspan="2">Shapes3D</th> </tr> <tr> <th>FoV</th> <th># values</th> </tr> </thead> <tbody> <tr> <td>Floor Hue</td> <td>10</td> </tr> <tr> <td>Wall Hue</td> <td>10</td> </tr> <tr> <td>Object Hue</td> <td>10</td> </tr> <tr> <td>Scale</td> <td>8</td> </tr> <tr> <td>Shape</td> <td>4</td> </tr> <tr> <td>Orientation</td> <td>15</td> </tr> </tbody> </table>	Shapes3D		FoV	# values	Floor Hue	10	Wall Hue	10	Object Hue	10	Scale	8	Shape	4	Orientation	15		
Shapes3D																		
FoV	# values																	
Floor Hue	10																	
Wall Hue	10																	
Object Hue	10																	
Scale	8																	
Shape	4																	
Orientation	15																	
<table border="1"> <thead> <tr> <th colspan="2">Coil100-Augmented</th> </tr> <tr> <th>FoV</th> <th># values</th> </tr> </thead> <tbody> <tr> <td>Object</td> <td>100</td> </tr> <tr> <td>Pose</td> <td>72</td> </tr> <tr> <td>Orientation</td> <td>18</td> </tr> <tr> <td>Scale</td> <td>9</td> </tr> </tbody> </table>	Coil100-Augmented		FoV	# values	Object	100	Pose	72	Orientation	18	Scale	9						
Coil100-Augmented																		
FoV	# values																	
Object	100																	
Pose	72																	
Orientation	18																	
Scale	9																	
<table border="1"> <thead> <tr> <th colspan="2">RGBD Objects</th> </tr> <tr> <th>FoV</th> <th># values</th> </tr> </thead> <tbody> <tr> <td>Category</td> <td>51</td> </tr> <tr> <td>Elevation</td> <td>4</td> </tr> <tr> <td>Pose</td> <td>263</td> </tr> </tbody> </table>	RGBD Objects		FoV	# values	Category	51	Elevation	4	Pose	263								
RGBD Objects																		
FoV	# values																	
Category	51																	
Elevation	4																	
Pose	263																	

C Transfer experiment

Here we provide additional information about the architecture of the models for the transfer experiments. Moreover, we include the tables reporting the average performances of the GBT and MLP classifiers.

C.1 Datasets

Table 7 reports the main information about the used dataset, e.g. FoV and number of classes, together with some examples of the original dataset, plus some samples of one variant of the given dataset, such as Color-dSprites and Noisy-Color-dSprites. The only exception is Shapes3D which does not have any variant, so different samples drawn from the same dataset are shown.

C.2 Architecture

Table 8 shows the architecture of the model used for all the transfer experiments. We trained multiple models with 10 different random seeds for each value of β . Hence, we obtain 20 models for each *Source* dataset. We adopted the Adam optimizer [29] with default parameters, batch size=64 and 400k steps. We used linear deterministic warm-up [11, 61, 3] over the first 50k training steps. We maintained the latent dimension fixed to 10 for all the experiments.

Table 8: Encoder and Decoder architecture for the transfer experiments.

Encoder	Decoder
Input: $64 \times 64 \times \#channels$ 4×4 conv, 32 LeakyRelu(0.02), stride 2 4×4 conv, 64 LeakyRelu(0.02), stride 2 4×4 conv, 128 LeakyRelu(0.02), stride 2 Flatten $2 \times$ FC 10	Input: \mathbb{R}^{10} FC 8192 LeakyRelu(0.02) FC $8 \times 8 \times 128$ 4×4 upconv, 64 LeakyRelu(0.02), stride 2 4×4 upconv, 32 LeakyRelu(0.02), stride 2 4×4 upconv, #channels Sigmoid, stride 2

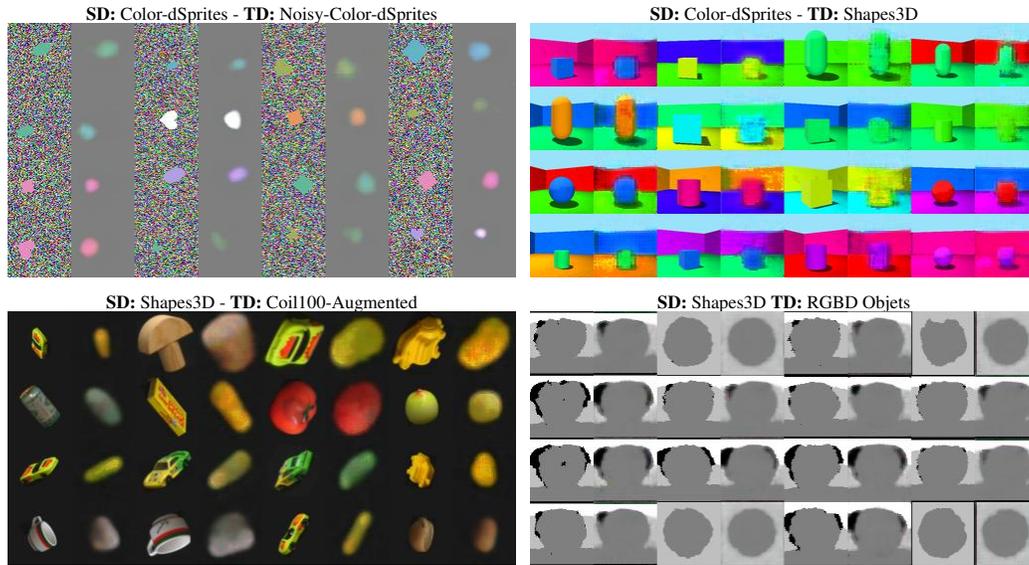


Figure 10: Some reconstructions generated from the fine-tuned models of different Source (SD) Target (TD) couples.

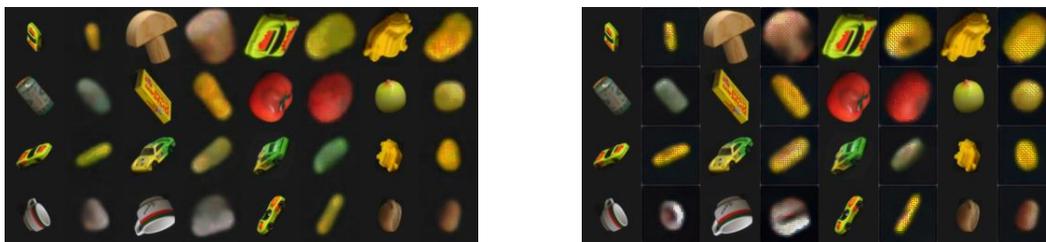


Figure 11: **(Left)** Reconstruction of samples of Coil100 of a fine-tuned model trained originally on Color-dSprites, same as in Fig. 10. **(Right)** Reconstruction of samples of Coil100 of a model trained from scratch on it.

C.3 Transfer reconstruction

Some reconstructions, generated by the VAE models **after** they are fine-tuned, are depicted in Fig. 10. The quality of the reconstruction is good even if the encoding is obtained by training the model first on a completely different source dataset and then fine-tuning the model for a few iterations.

Fig. 11 (Right) shows the reconstruction of the same samples of Coil100-Augmented from the model trained from scratch on it. Comparing the latter with the reconstructions of Fig. 11(Left) it can be observed that the quality is comparable (with some exceptions), and so with the fine-tuned models, we are not losing much information from the data.

Table 9: Transfer from Shapes3D (Source) to Isaac3D (Target). Average classification *accuracy* over the 20 models of the GBT classifier, before and after fine-tuning (see Table 2).

Mean accuracy on FoVs(%)										
Pruned	Object shape (3)	Object scale (4)	Camera height (4)	X-movement (8)	Y-movement (5)	Light intensity (4)	Light y-direction (6)	Object color (4)	Wall color (4)	All
✗	34.9 (+5.1)	54.0 (+34.6)	39.2 (+17.4)	33.9 (+29.6)	23.0 (+4.3)	83.6 (+14.0)	85.4 (+12.7)	29.8 (+13.9)	78.1 (+18.9)	51.3 (+16.7)
✓	33.8 (+3.6)	40.1 (+24.1)	33.5 (+11.1)	24.7 (+15.3)	21.6 (+2.7)	69.4 (+17.3)	67.5 (+14.4)	27.0 (+7.3)	61.5 (+16.6)	42.1 (+12.5)

Table 10: Transfer from Shapes3D (Source) to Isaac3D (Target). The *Compactness* and *Modularity* scores of the same models of Table 9.

Pruned	Modularity(%)		Compactness(%)	
	Our (OS)	DCI	Our (MES)	MIG
✗	25.1	6.3	21.2	2.2
✓	(+9.7)	(+16.1)	(+10.2)	(+5.9)

C.4 Transfer protocol

Each GBT and MLP classifiers are trained on the latent representation r extracted from the Encoder of the Ada-GVAE models so that the train split comprises 10000 samples and the test split is of 5000 samples.

With *unsupervised* fine-tuning on the Target dataset, it means that the model is trained as a simple VAE [30].

C.5 GBT & MLP performance distribution

In this section, we report the performance distribution of the GBT classifiers on the target FoV (see Fig. 12), the figures on the left depict the performances on the representation before fine-tuning and on the right are depicted the results after the fine-tuning of the representation. Fig. 13 shows the performance distribution of the MLP classifiers on the target FoV, the figures on the left depict the performances on the representation before fine-tuning and on the right depict the results after the fine-tuning of the representation.

C.6 Transfer from Shapes3D to Isaac3D

In this section, we report the results of the transfer from Shapes3D to Isaac3D, see Table 9 for the Explicitness and Table 10 for Modularity and Compactness.

We observe that the transfer is overall effective according to all the disentanglement scores even though the Target dataset is much more complex than the Source. If we compare the results of the transfer from Shapes3D to other real datasets such as Coil100 (Table 4) and RGBD-Objects (Table 5), we notice that in the case of Isaac3D the boost of finetuning is more noticeable.

This suggests that, if source and target have common FoVs with similar appearance we can obtain reasonable performances on a real Target dataset even if the source synthetic dataset is much simpler.

C.7 MLPs performances

In this section, we report the performances of the MLP classifiers on the FoVs, for the sake of comparison we report the scores of the disentanglement metrics on the representation input.

We associate the tables regarding the same Target dataset: Table 11 corresponds to the same representations of Table 2 in the main document; Table 12 to Table 3; Table 13 to Table 4 and Table 14 to Table 5.

Overall we can observe higher performances obtained with MLP, especially on the classifiers trained on the entire representation. This happens because MLP can easily disentangle an entangled representation by observing different dimensions at a time while GBTs partition the input space into regions aligned with the axes, making it harder to observe multiple dimensions at a time and so the FoVs.

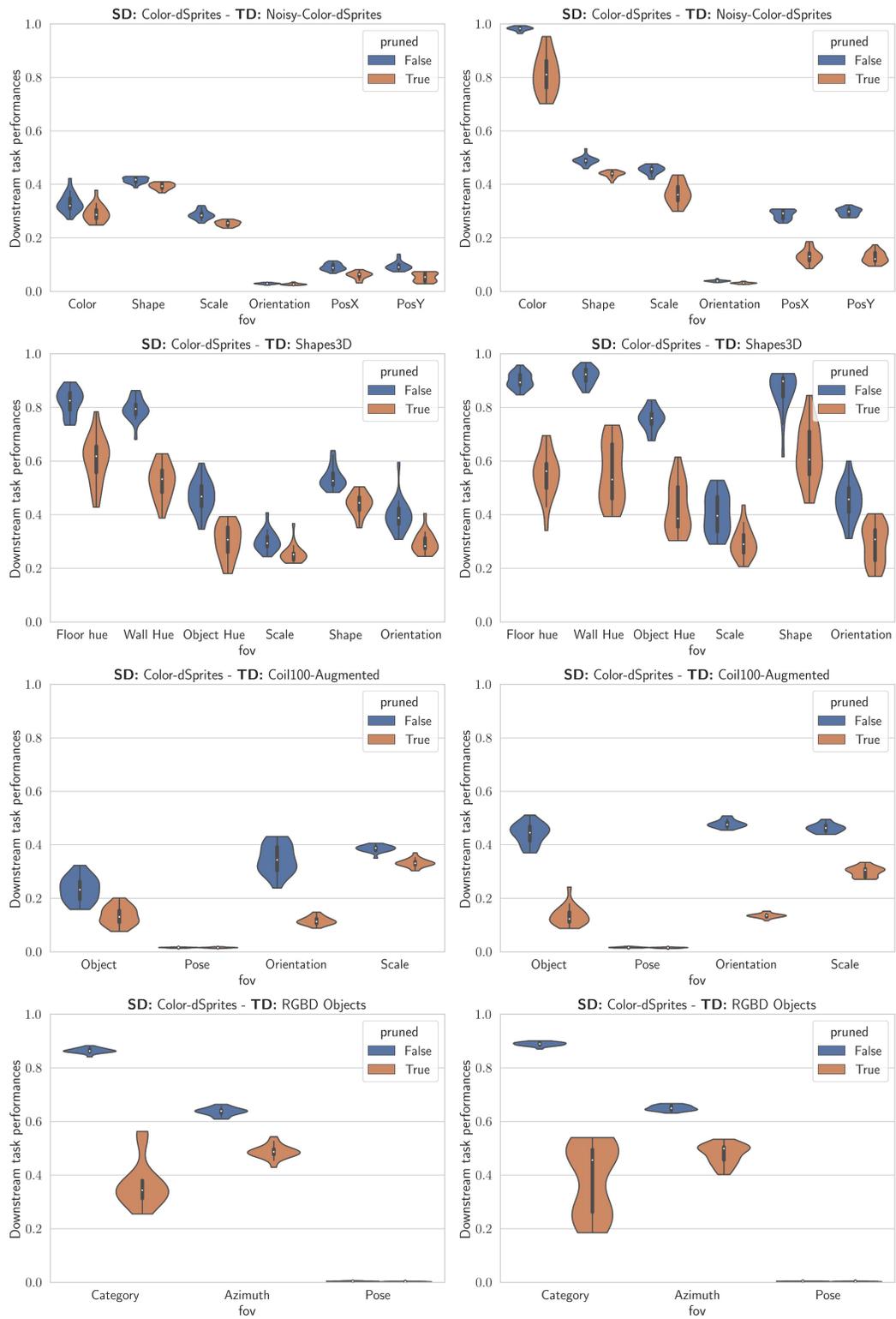


Figure 12: Some examples of the performance distribution of the GBT classifiers before (Left) and after (Right) fine-tuning of different Source (SD) Target (TD) couples.

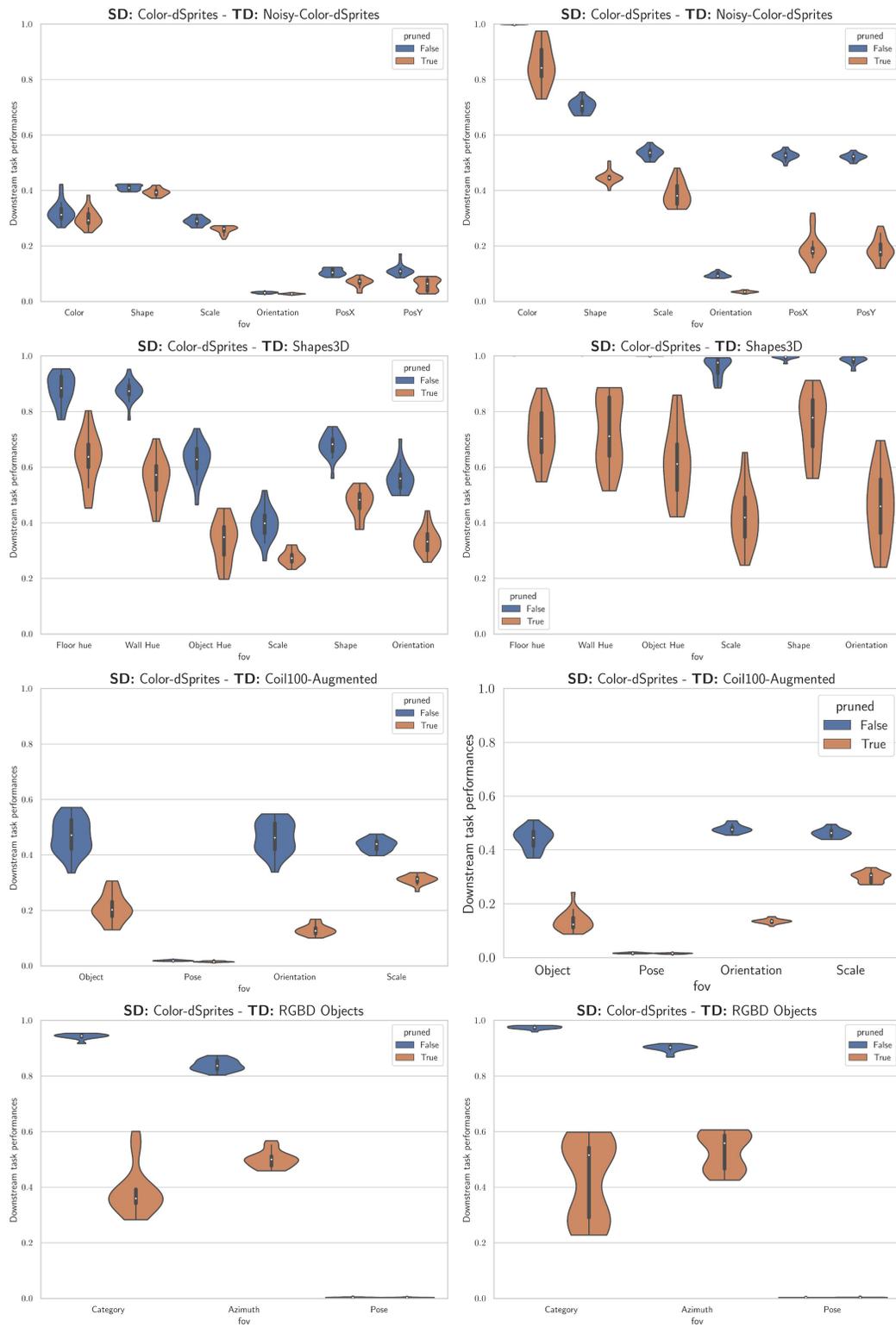


Figure 13: Some examples of the performance distribution of the MLP classifiers before (Left) and after (Right) fine-tuning of different Source (SD) Target (TD) couples.

Table 11: Target dataset: dSprites and variant. The averaged performances of the MLP classifiers and disentanglement metrics, together with the average improvement with the finetuning (in brackets).

SD	TD	Pruned	Mean Accuracy on FoVs(%)							Modularity(%)		Compactness(%)	
			Color (7)	Shape (3)	Scale (6)	Orientation (40)	PosX (32)	PosY (32)	All	Our (OS)	DCI	Our (MES)	MIG
dSprites	N-dSprites	✗		64.1	44.3	8.7	20.1	19.4	31.3				
		✓		(+33.2)	(+45.9)	(+33.7)	(+54.0)	(+54.0)	(+44.2)	31.8	22.0	32.1	13.9
	C-dSprites	✗		50.9	42.9	4.1	15.4	15.9	25.8				
		✓		(+14.3)	(+22.3)	(+8.8)	(+27.7)	(+26.7)	(+20.0)	(+6.3)	(+4.0)	(+3.9)	(+6.9)
C-dSprites	N-C-dSprites	✗	39.3	99.4	96.3	67.6	85.2	83.6	78.6				
		✓	(+45.0)	(+0.2)	(+2.1)	(+5.9)	(-1.9)	(-2.5)	(+8.1)	61.0	42.9	72.3	34.0
	N-C-dSprites	✗	19.3	74.3	78.2	21.1	73.8	73.4	56.7				
		✓	(+13.1)	(+5.6)	(+6.4)	(+3.1)	(-1.8)	(-1.6)	(+4.1)	(+1.6)	(+10.4)	(+3.5)	(+0.3)
C-dSprites	N-C-dSprites	✗	32.2	41.0	28.9	3.1	10.5	11.1	21.1				
		✓	(+67.7)	(+29.4)	(+24.6)	(+6.4)	(+42.1)	(+41.0)	(+35.2)	28.9	5.4	29.0	1.8
	N-C-dSprites	✗	29.9	39.4	25.8	2.7	6.8	6.0	18.4				
		✓	(+55.4)	(+5.3)	(+12.9)	(+0.7)	(+12.4)	(+12.6)	(+16.6)	(+0.3)	(+2.2)	(-1.9)	(+1.0)

Table 12: Target dataset: Shapes3D. The averaged performances of the MLP classifiers and disentanglement metrics, together with the average improvement with the finetuning (in brackets).

SD	TD	Pruned	Mean Accuracy on FoVs(%)							Modularity(%)		Compactness(%)	
			Floor Hue (10)	Wall Hue (10)	Object Hue (10)	Scale (8)	Shape (4)	Orientation (15)	All	Our (OS)	DCI	Our (MES)	MIG
dSprites	Shapes3D	✗	84.7	86.8	57.3	33.4	67.5	51.1	63.4				
		✓	(+15.3)	(+13.2)	(+40.3)	(+55.8)	(+27.9)	(+44.9)	(+32.9)	26.9	9.6	23.9	5.3
	Shapes3D	✗	64.1	61.3	31.9	23.1	45.2	27.0	42.1				
		✓	(+14.6)	(+18.4)	(+33.1)	(+16.7)	(+23.4)	(+13.4)	(+19.9)	(+0.6)	(+9.0)	(+1.1)	(-0.4)
C-dSprites	Shapes3D	✗	88.1	87.4	62.5	39.5	67.7	56.0	66.9				
		✓	(+11.9)	(+12.6)	(+37.5)	(+56.6)	(+31.6)	(+42.3)	(+32.1)	30.6	13.7	28.2	8.4
	Shapes3D	✗	63.2	55.9	33.7	27.5	47.2	33.6	43.5				
		✓	(+8.3)	(+16.6)	(+27.8)	(+14.6)	(+27.7)	(+12.0)	(+17.8)	(+0.8)	(+7.4)	(+2.1)	(-1.4)

This aligns with what is observed in [41, 11]. Given this, it is preferable to observe the performances of the GBT because they can give a clearer about the disentanglement properties of the representation.

C.8 GBT & MLP performances standard deviation

In this section, we report the standard deviation of performances of the GBT and MLP classifiers on the FoVs, for the sake of comparison we also report the standard deviation of scores of the disentanglement metrics on the representation input.

MLP classifiers: Table 19 corresponds to the same representations of Table 11 in the Appendix; Table 20 to Table 12; Table 21 to Table 13 and Table 22 to Table 14.

GBT classifiers: Table 15 corresponds to the same representations of Table 2 in the Appendix; Table 16 to Table 3; Table 17 to Table 4 and Table 18 to Table 5.

Note that the metrics scores in the MLPs and GBTs tables are the same because we refer to the same representations. In general, we can observe the models without fine-tuning have a small standard deviation, while fine-tuned models perform with higher variation.

Table 13: Target dataset: Coil100-augmented and variants. The averaged performances of the MLP classifiers and disentanglement metrics, together with the average improvement with the finetuning (in brackets).

SD	TD	Pruned	Mean Accuracy on FoVs(%)					Modularity(%)		Compactness(%)	
			Object (100)	Pose (72)	Orientation (18)	Scale (9)	All	Our (OS)	DCI	Our (MES)	MIG
dSprites	Coil (bin)	✗	37.2	2.6	55.6	44.6	35.0				
		✓	(+25.7)	(+0.5)	(+13.3)	(+21.6)	(+15.3)	37.8	11.4	25.7	5.1
	Coil (bin)	✗	9.2	1.7	19.5	34.5	16.2				
		✓	(-0.6)	(-0.1)	(-5.0)	(+1.3)	(-1.1)	(+2.6)	(+2.7)	(+2.1)	(+1.8)
C-dSprites	Coil (bin)	✗	46.9	1.9	46.3	43.6	34.7				
		✓	(+41.1)	(+0.8)	(+22.9)	(+23.4)	(+22.1)	33.9	10.4	27.2	4.1
	Coil (bin)	✗	20.8	1.5	12.9	31.2	16.6				
		✓	(-0.6)	(+0.2)	(+2.3)	(-0.2)	(+0.4)	(+3.0)	(+1.8)	(+0.0)	(+2.2)
Shapes3D	Coil (bin)	✗	32.0	1.7	27.3	36.0	24.2				
		✓	(+58.3)	(+1.0)	(+41.6)	(+30.3)	(+32.8)	31.7	3.2	25.0	1.7
	Coil (bin)	✗	10.1	1.5	11.1	24.1	11.7				
		✓	(+8.6)	(+0.1)	(+6.2)	(+3.7)	(+4.7)	(-0.2)	(+3.5)	(-1.0)	(+0.8)
Coil(bin)	Coil (bin)	✗	51.3	2.5	51.1	35.1	35.0				
		✓	(+34.5)	(+0.2)	(+18.8)	(+31.6)	(+21.3)	36.3	10.4	26.1	5.6
	Coil (bin)	✗	12.5	1.7	14.7	28.1	14.3				
		✓	(+5.6)	(-0.1)	(+3.0)	(-0.2)	(+2.1)	(-5.2)	(-5.3)	(-2.4)	(-3.5)

Table 14: Target dataset: RGBD-Objects and variants. The averaged performances of the MLP classifiers and disentanglement metrics, together with the average improvement with the finetuning (in brackets).

SD	TD	Pruned	Mean Accuracy on FoVs(%)				Modularity(%)		Compactness(%)	
			Category (51)	Elevation (4)	Pose (263)	All	Our (OS)	DCI	Our (MES)	MIG
dSprites		✗	73.9 (+9.3)	90.3 (+4.5)	0.3 (-0.1)	54.8 (+4.6)	34.3 (+0.6)	11.0 (+0.5)	22.1 (-0.2)	3.4 (+0.5)
		✓	36.4 (-2.9)	68.6 (-0.5)	0.3 (+0.0)	35.1 (-1.1)				
Shapes3D	RGBD (depth)	✗	70.5 (+15.0)	88.4 (+7.5)	0.3 (-0.1)	53.0 (+7.5)	35.3 (-0.8)	12.0 (-0.9)	22.4 (-0.5)	2.0 (+0.8)
		✓	28.4 (+13.5)	66.7 (+6.4)	0.3 (-0.0)	31.8 (+6.6)				
Coil(bin)		✗	20.8 (+6.2)	62.8 (+6.7)	0.3 (+0.0)	28.0 (+4.3)	35.6 (-1.5)	11.8 (+0.3)	23.5 (-2.0)	4.3 (-2.4)
		✓	10.2 (+3.8)	57.7 (-2.3)	0.3 (+0.0)	22.7 (+0.5)				
C-dSprites		✗	94.3 (+3.2)	83.9 (+6.2)	0.3 (-0.1)	59.5 (+3.1)	35.7 (-0.7)	7.8 (-2.3)	24.7 (-1.5)	1.8 (-0.4)
		✓	38.6 (+4.1)	50.0 (+2.9)	0.3 (+0.0)	29.6 (+2.3)				
Coil	RGBD	✗	95.6 (+2.7)	89.7 (+2.1)	0.2 (+0.0)	61.8 (+1.6)	35.0 (+0.0)	4.2 (+0.6)	23.5 (-0.3)	2.0 (-1.1)
		✓	41.2 (-1.0)	52.7 (-1.6)	0.3 (+0.0)	31.4 (-0.8)				
Coil(bin)		✗	93.7 (+4.1)	86.6 (+3.8)	0.2 (+0.0)	60.2 (+2.6)	35.3 (-0.5)	5.1 (-0.2)	24.2 (-1.2)	1.3 (-0.2)
		✓	40.6 (+1.2)	52.5 (+1.1)	0.3 (+0.1)	31.1 (+0.8)				

Table 15: Target dataset: dSprites and variant. The standard deviation of the performances of the GBT classifiers and disentanglement metrics, together with the standard deviation of the performances with the finetuning (in brackets).

SD	TD	Pruned	Standard deviation Accuracy on FoVs(%)						Modularity(%)		Compactness(%)	
			Color (7)	Shape (3)	Scale (6)	Orientation (40)	PosX (32)	PosY (32)	Our (OS)	DCI	Our (MES)	MIG
dSprites	N-dSprites	✗		0.02 (8.29)	0.02 (6.00)	0.01 (5.55)	0.01 (6.65)	0.02 (7.73)	0.01 (3.43)	0.01 (7.23)	0.01 (2.99)	0.01 (6.29)
		✓		0.06 (8.65)	0.02 (6.98)	0.01 (4.67)	0.01 (12.33)	0.01 (13.49)				
	C-dSprites	✗	0.05 (9.44)	0.02 (1.31)	0.04 (1.89)	0.06 (3.02)	0.01 (1.52)	0.01 (1.68)	0.01 (0.67)	0.01 (1.93)	0.02 (1.02)	0.01 (2.99)
		✓	0.03 (19.28)	0.05 (3.54)	0.05 (2.85)	0.04 (5.67)	0.01 (0.93)	0.01 (1.40)				
C-dSprites	N-C-dSprites	✗	0.03 (0.78)	0.01 (1.53)	0.02 (1.53)	0.00 (0.33)	0.01 (1.72)	0.02 (1.41)	0.01 (2.09)	0.01 (2.99)	0.01 (2.44)	0.01 (2.00)
		✓	0.03 (7.33)	0.01 (1.18)	0.01 (3.99)	0.00 (0.31)	0.01 (2.60)	0.02 (2.15)				

D Experiments Compute Resources

All the experiments have been executed with an NVIDIA Quadro RTX 6000. On average, the training and evaluation of a single Source model take 5 hours. Each fine-tuning and final evaluation takes 1.5 hours. Overall, the whole bunch of transfer experiments and our metric assessment take approximately 1100 hours.

Table 16: Target dataset: Shapes3D. The standard deviation of the performances of the GBT classifiers and disentanglement metrics, together with the standard deviation of the performances with the finetuning (in brackets).

SD	TD	Pruned	Standard deviation Accuracy on FoVs(%)						Modularity(%)		Compactness(%)	
			Floor Hue (10)	Wall Hue (10)	Object Hue (10)	Scale (8)	Shape (4)	Orientation (15)	Our (OS)	DCI	Our (MES)	MIG
dSprites		✗	0.05 (2.77)	0.05 (3.08)	0.05 (9.73)	0.02 (6.58)	0.05 (13.89)	0.06 (6.59)	0.02 (5.60)	0.01 (4.93)	0.02 (4.52)	0.01 (2.13)
		✓	0.06 (10.43)	0.06 (11.44)	0.04 (6.40)	0.02 (5.57)	0.04 (11.17)	0.04 (6.71)				
C-dSprites	Shapes3D	✗	0.09 (8.43)	0.07 (11.10)	0.06 (9.17)	0.03 (5.53)	0.04 (10.63)	0.04 (7.35)	0.02 (3.82)	0.02 (4.57)	0.02 (3.82)	0.01 (2.72)
		✓	0.09 (8.43)	0.07 (11.10)	0.06 (9.17)	0.03 (5.53)	0.04 (10.63)	0.04 (7.35)				

Table 17: Target dataset: Coil100-augmented and variants. The standard deviation of the performances of the GBT classifiers and disentanglement metrics, together with the standard deviation of the performances with the finetuning (in brackets).

SD	TD	Pruned	Standard deviation Accuracy on FoVs(%)				Modularity(%)		Compactness(%)	
			Object (100)	Pose (72)	Orientation (18)	Scale (9)	Our (OS)	DCI	Our (MES)	MIG
dSprites	Coil (bin)	✗	0.01 (0.72)	0.00 (0.20)	0.01 (0.69)	0.01 (0.58)	0.00 (0.67)	0.00 (0.51)	0.00 (0.48)	0.00 (0.51)
		✓	0.00 (0.51)	0.00 (0.13)	0.02 (0.54)	0.01 (1.15)				
C-dSprites		✗	0.05 (3.85)	0.00 (0.17)	0.05 (1.44)	0.01 (1.56)	0.00 (0.48)	0.00 (0.51)	0.00 (0.48)	0.00 (0.51)
		✓	0.03 (3.56)	0.00 (0.15)	0.01 (0.81)	0.01 (1.87)				
Shapes3D	Coil	✗	0.02 (3.12)	0.00 (0.18)	0.01 (1.80)	0.01 (1.30)	0.01 (1.35)	0.01 (1.41)	0.01 (1.07)	0.01 (1.19)
		✓	0.01 (3.94)	0.00 (0.12)	0.01 (2.02)	0.01 (2.98)				
Coil(bin)		✗	0.03 (2.24)	0.00 (0.14)	0.02 (1.48)	0.02 (1.52)	0.01 (1.62)	0.02 (1.52)	0.01 (1.17)	0.01 (1.03)
		✓	0.01 (3.94)	0.00 (0.18)	0.02 (2.57)	0.02 (3.51)				

Table 18: Target dataset: RGBD-Objects and variants. The standard deviation of the performances of the GBT classifiers and disentanglement metrics, together with the standard deviation of the performances with the finetuning (in brackets).

SD	TD	Pruned	Standard deviation Accuracy on FoVs(%)			Modularity(%)		Compactness(%)	
			Category (51)	Elevation (4)	Pose (263)	Our (OS)	DCI	Our (MES)	MIG
dSprites		✗	0.01 (1.79)	0.02 (0.66)	0.00 (0.11)	0.01 (0.47)	0.02 (4.03)	0.01 (0.54)	0.01 (2.47)
		✓	0.05 (5.58)	0.05 (4.65)	0.00 (0.10)				
Shapes3D	RGBD (depth)	✗	0.02 (0.96)	0.02 (0.59)	0.00 (0.11)	0.01 (0.54)	0.02 (4.03)	0.01 (0.54)	0.01 (2.47)
		✓	0.05 (6.07)	0.04 (5.60)	0.00 (0.07)				
Coil(bin)		✗	0.27 (32.60)	0.13 (15.39)	0.00 (0.10)	0.01 (0.53)	0.05 (5.73)	0.01 (0.57)	0.02 (1.31)
		✓	0.14 (18.09)	0.11 (11.46)	0.00 (0.08)				
C-dSprites		✗	0.01 (0.75)	0.01 (0.99)	0.00 (0.11)	0.00 (0.39)	0.02 (2.39)	0.01 (0.77)	0.01 (0.85)
		✓	0.09 (12.79)	0.02 (3.64)	0.00 (0.10)				
Coil	RGBD	✗	0.01 (0.77)	0.01 (1.02)	0.00 (0.11)	0.00 (0.36)	0.01 (2.92)	0.01 (0.67)	0.01 (0.62)
		✓	0.04 (11.12)	0.02 (2.88)	0.00 (0.11)				
Coil(bin)		✗	0.02 (0.70)	0.01 (0.83)	0.00 (0.11)	0.00 (0.42)	0.03 (2.43)	0.01 (0.90)	0.00 (0.64)
		✓	0.10 (12.38)	0.03 (3.76)	0.00 (0.08)				

E Future directions

We briefly summarise the future direction of our work.

On the data set side, we only considered the synthetic datasets applicable to a wider number of tasks, other synthetic datasets exist but either they are limited in the number of FoV or very specific to a task. We limited our analysis to two real datasets, plus their simplified variants, to tackle a broad but limited number of challenges. We will extend our analysis to more complex real datasets with an increasing number of known and unknown factors, but this requires we also design more complex synthetic datasets able to tackle these complications. For the methods, we will explore the effect of the dimensions of the latent space and different kinds of supervision for training the Source model, as well as including (partial) supervision on the fine-tuning. On the applications side, we will analyse

Table 19: Target dataset: dSprites and variant. The standard deviation of the performances of the MLP classifiers and disentanglement metrics, together with the standard deviation of the performances with the finetuning (in brackets).

SD	TD	Pruned	STD Accuracy on FoVs(%)					Modularity(%)		Compactness(%)		
			Color (7)	Shape (3)	Scale (6)	Orientation (40)	PosX (32)	PosY (32)	Our (OS)	DCI	Our (MES)	MIG
dSprites	N-dSprites	✗		0.02 (1.36)	0.03 (3.41)	0.01 (7.36)	0.01 (4.13)	0.02 (5.05)	0.01 (3.43)	0.01 (7.23)	0.01 (2.99)	0.01 (6.29)
		✓		0.06 (10.38)	0.03 (12.67)	0.01 (7.66)	0.02 (13.12)	0.01 (14.77)				
	C-dSprites	✗	0.08 (5.14)	0.00 (0.27)	0.01 (0.60)	0.06 (3.49)	0.05 (5.27)	0.03 (4.01)	0.01 (0.67)	0.01 (1.93)	0.02 (1.02)	0.01 (2.99)
		✓	0.02 (21.72)	0.06 (3.62)	0.06 (3.71)	0.04 (7.10)	0.02 (1.54)	0.02 (1.54)				
C-dSprites	N-C-dSprites	✗	0.04 (0.08)	0.01 (2.34)	0.01 (1.84)	0.00 (0.82)	0.01 (1.53)	0.02 (1.18)	0.01 (2.09)	0.01 (2.99)	0.01 (2.44)	0.01 (2.00)
		✓	0.03 (7.10)	0.01 (2.13)	0.01 (4.27)	0.00 (0.40)	0.02 (5.12)	0.02 (4.01)				

Table 20: Target dataset: Shapes3D. The standard deviation of the performances of the MLP classifiers and disentanglement metrics, together with the standard deviation of the performances with the finetuning (in brackets).

SD	TD	Pruned	STD Accuracy on FoVs(%)						Modularity(%)		Compactness(%)	
			Floor Hue (10)	Wall Hue (10)	Object Hue (10)	Scale (8)	Shape (4)	Orientation (15)	Our (OS)	DCI	Our (MES)	MIG
dSprites	Shapes3D	✗	0.06 (0.03)	0.04 (0.02)	0.07 (5.80)	0.02 (13.47)	0.04 (6.77)	0.06 (4.87)	0.02 (5.60)	0.01 (4.93)	0.02 (4.52)	0.01 (2.13)
		✓	0.05 (7.33)	0.07 (9.73)	0.05 (12.26)	0.03 (12.59)	0.04 (13.96)	0.05 (14.91)				
C-dSprites	Shapes3D	✗	0.05 (0.00)	0.04 (0.00)	0.06 (0.02)	0.06 (3.56)	0.04 (0.78)	0.05 (1.46)	0.02 (3.82)	0.02 (4.57)	0.02 (3.82)	0.01 (2.72)
		✓	0.09 (9.44)	0.08 (12.63)	0.08 (12.60)	0.02 (10.19)	0.05 (11.03)	0.04 (12.73)				

Table 21: Target dataset: Coil100-augmented and variants. The standard deviation of the performances of the MLP classifiers and disentanglement metrics, together with the standard deviation of the performances with the finetuning (in brackets).

SD	TD	Pruned	STD Accuracy on FoVs(%)				Modularity(%)		Compactness(%)	
			Object (100)	Pose (72)	Orientation (18)	Scale (9)	Our (OS)	DCI	Our (MES)	MIG
dSprites	Coil (bin)	✗	0.02 (0.90)	0.00 (0.25)	0.01 (0.67)	0.02 (1.21)	0.00 (0.67)	0.00 (0.51)	0.00 (0.48)	0.00 (0.51)
		✓	0.01 (0.69)	0.00 (0.17)	0.01 (0.70)	0.01 (0.94)				
C-dSprites	Coil	✗	0.02 (0.90)	0.00 (0.25)	0.01 (0.67)	0.02 (1.21)	0.00 (0.48)	0.00 (0.51)	0.00 (0.48)	0.00 (0.51)
		✓	0.01 (0.69)	0.00 (0.17)	0.01 (0.70)	0.01 (0.94)				
Shapes3D	Coil	✗	0.02 (0.91)	0.00 (0.26)	0.01 (0.77)	0.02 (1.16)	0.01 (1.35)	0.01 (1.41)	0.01 (1.07)	0.01 (1.19)
		✓	0.01 (6.04)	0.00 (0.18)	0.01 (3.44)	0.01 (3.44)				
Coil(bin)	Coil	✗	0.04 (1.65)	0.00 (0.19)	0.03 (0.85)	0.02 (2.02)	0.01 (1.62)	0.02 (1.52)	26.1 (-2.4)	0.01 (1.03)
		✓	0.03 (6.86)	0.00 (0.21)	0.02 (3.56)	0.02 (4.13)				

the effect of transferring from different priors (e.g. video sequences, expert knowledge in biological data) and investigate if transferring a disentangled representation will help to increase the level of interpretability of the target representation.

Table 22: Target dataset: RGBD-Objects and variants. The standard deviation of the performances of the MLP classifiers and disentanglement metrics, together with the standard deviation of the performances with the finetuning (in brackets).

SD	TD	Pruned	STD Accuracy on FoVs(%)			Modularity(%)		Compactness(%)	
			Category (51)	Elevation (4)	Pose (263)	Our (OS)	DCI	Our (MES)	MIG
dSprites	RGBD	✗	0.01 (1.07)	0.01 (0.89)	0.00 (0.07)	0.01 (0.47)	0.02 (4.03)	0.01 (0.54)	0.01 (2.47)
		✓	0.07 (7.45)	0.05 (5.32)	0.00 (0.10)				
Shapes3D	RGBD (depth)	✗	0.01 (1.07)	0.01 (0.89)	0.00 (0.07)	0.01 (0.54)	0.02 (4.03)	0.01 (0.54)	0.01 (2.47)
		✓	0.07 (7.45)	0.05 (5.32)	0.00 (0.10)				
Coil(bin)	RGBD	✗	0.33 (38.97)	0.17 (17.92)	0.00 (0.07)	0.01 (0.53)	0.05 (5.73)	0.01 (0.57)	0.02 (1.31)
		✓	0.16 (20.37)	0.12 (13.59)	0.00 (0.09)				
C-dSprites	RGBD	✗	0.01 (0.58)	0.02 (1.22)	0.00 (0.06)	0.00 (0.39)	0.02 (2.39)	0.01 (0.77)	0.01 (0.85)
		✓	0.08 (13.95)	0.03 (6.62)	0.00 (0.07)				
Coil	RGBD	✗	0.01 (0.30)	0.01 (0.87)	0.00 (0.07)	0.00 (0.36)	0.01 (2.92)	0.01 (0.67)	0.01 (0.62)
		✓	0.04 (12.19)	0.02 (5.64)	0.00 (0.08)				
Coil(bin)	RGBD	✗	0.01 (0.32)	0.02 (0.75)	0.00 (0.09)	0.00 (0.42)	0.03 (2.43)	0.01 (0.90)	0.00 (0.64)
		✓	0.11 (13.39)	0.05 (6.53)	0.00 (0.09)				

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The claimed contributions is a metric described in Sec. 2 and a disentangled representation transferring methodology defined and experimentally assessed in Sec. 3.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: In Sec. 3.3 and in Sec. 4 (Limitations), we discuss the potential and the limitations of the methodology.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory Assumptions and Proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not provide theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental Result Reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: In Section 3.3 and in Appendix C we provide all the information to reproduce the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide the code as supplementary material, while the datasets we used are all publicly available.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental Setting/Details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In Sections 3.3 and Section 2 and in Appendix C we provide all the details to understand the experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment Statistical Significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: While we report in the main document the Tables of the averaged measurements, we report in Tables of the same structure the standard deviation of our transfer experiments in Appendix C.8.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments Compute Resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: In Appendix D we are reporting an approximation of the time execution and GPU hardware.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code Of Ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines?>

Answer: [Yes]

Justification: We read the NeurIPS Code of Ethics and the paper conforms with it (we do not involve human subjects or participants, we used already existing datasets respecting their terms of usage, etc.).

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader Impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [NA]

Justification: There is no social impact.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: There is no risk of misuse because our work does not include generating tasks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: The exploitation of already existing models have been properly cited with the original authors.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide the code with instructions about the usage, reproducibility of experiments and license.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The work does not include experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The work does not include experiments with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.