,

# Benchmarking Multimodal Variational Autoencoders: CdSprites+ Dataset and Toolkit Supplementary Material

**Gabriela Sejnova** **Michal Vavrecka** **Karla Stepanova**
Czech Institute of Informatics, Robotics and Cybernetics
Czech Technical University in Prague, Prague, Czech Republic
gabriela.sejnova@cvut.cz

## 1 Overview

In Section 2, we provide additional details for the CdSprites+ dataset. In Section 3, we describe the technical information and detailed results for the experiments presented in the paper and Section 4 reports the reproduced results from original papers using our toolkit.

## 2 CdSprites+ dataset statistics

The presented version of the benchmark dataset comprises 5 different levels of difficulty, where each level varies in the number of included features (see Table 1 for their overview). The default size of the dataset is depicted in Table 1 for each level, although the user can easily generate a larger set of the data. The scripts for calculation of the cross- and joint- coherency accuracies use separate batches of testing data provided in the toolkit.

In all levels, the source of noise in the images is their random position and rotation - in levels 1-4, the shapes are located around the whole image with a variance of 25 pixels along both $x$ and $y$ axes. In Level 5, the shapes are shifted in random quadrants where their position also varies with a variance of 8 pixels. The positions are configured so that the whole shapes are always fitting the image. In Levels 2-5 where we vary the size, the proportion of the small objects to the big objects is 1:5.

You can also see a PCA visualization of the CdSprites+ Level 5 dataset in Fig. 4.

Table 1: Statistics of the CdSprites+ benchmark dataset. We show the number of train/validation samples and the number of various shapes, colours, object poses (meaning quadrants which are distinguished in captions) and backgrounds used in each difficulty level. The text captions only describe features that vary (e.g. in level 1, the text descriptions only include the shape name). The colours and backgrounds are all textured when they vary.

| Level | Train Samples | Validation Samples | Shapes | Sizes | Colours | Positions | Backgrounds |
|---|---|---|---|---|---|---|---|
| 1 | 67 500 | 7 500 | 3 | 1 | 1 | 1 | 1 |
| 2 | 108 000 | 12 000 | 3 | 2 | 1 | 1 | 1 |
| 3 | 270 000 | 30 000 | 3 | 2 | 5 | 1 | 1 |
| 4 | 540 000 | 60 000 | 3 | 2 | 5 | 4 | 1 |
| 5 | 864 000 | 96 000 | 3 | 2 | 5 | 4 | 2 |

Table 2: Training and inference times for each model trained on our CdSprites+ dataset. The models were trained for 150 epochs on Levels 1-2 and for 250 epochs on Levels 3-5, we thus show these times separately. We show the mean values over all seeds and different latent dimensionalities, the standard deviation is shown as ±.

| Model | Per epoch (s) | Per training (min)<br>(Levels 1-2) | Per training (min)<br>(Levels 3-5) | Inference time (s) |
|---|---|---|---|---|
| MMVAE | 203 ±2 | 525 ±8 | 860 ±10 | 152 ±6 |
| MVAE | 150 ±20 | 397 ±25 | 645 ±32 | 135 ±19 |
| MoPoE | 127 ±12 | 324 ±9 | 542 ±10 | 126 ±11 |
| DMVAE | 200 ±3 | 500 ±6 | 841 ±7 | 148 ±8 |

### 2.1 Using character-wise embeddings

We choose to use character-wise embeddings for CdSprites+ rather than word embeddings. While this choice was made to increase the difficulty of the text modality in our dataset, character embeddings have been recently used also in several other works as this approach brings specific advantages.

Firstly, character-wise embedding does not require a pre-defined vocabulary of possible input words. This can be useful e.g. in incremental learning scenarios where the whole vocabulary is not known prior to training beginning. Secondly, the model can be tested for robustness after training by inputting sentences with misspelt words (e.g., “sqaare” instead of “square”) to see if the model can generate correct images. With word-level embedding, this is not possible as replacing entire words will change the feature or create a nonsensical query (e.g., “left square” instead of “blue square”).

Please note that we expect the users to use the same encoder and decoder networks (i.e. character transformer networks) for the CdSprites+ benchmark to provide a restricted and fair comparison to other models. Should the users want to use CdSprites+ outside our toolkit for their custom evaluation, they can as well use word-level embeddings as we provide raw strings for the CdSprites+ text modality.

## 3 Benchmark study results

Here we provide the specific training configuration and hyperparameters used for the experiments on the CdSprites+ dataset as listed in the paper. We also report the detailed results for hyperparameter grid search in terms of the cross- and joint-generation accuracies.

### 3.1 Training configuration

All our experiments were trained with the GeForce GTX 1080 and NVIDIA Tesla V100 GPU cards, the mean computation times for training and inference are shown in Table 2. We used the Adam optimizer, the learning rate of $1e^{-4}$ and all experiments were repeated for 5 seeds (we report standard deviations for the results in the tables). We trained for 150 epochs for Levels 1 and 2 and for 250 epochs in the case of Levels 3-5. In the hyperparameter grid search, we varied the latent dimensionality (16, 24, 32) for all 5 dataset levels and the MVAE, MMVAE and MoPoE models. In the case of DMVAE, the latent dimensionality was different as there are private (modality-dependent) and shared latents. We thus chose different values for the comparison. We used a fixed value of 10 for both private latents and varied the shared latents with values 10, 16 and 24. In Tables 1-5, we show this as the total number of latent dimensions, i.e. 30 (10 shared and 2 × 10 private), 36 (16 shared and 2 × 10 private) and 46 (24 shared and 2 × 10 private).

We used the default training dataset size and validation split as reported in the statistics Table 1. In Tables 3, 4, 5, 6 and 7, we show results for the MVAE, MMVAE, DMVAE and MoPoE models and the compared latent dimensionalities. Standard deviations over 5 seeds are shown in brackets.

Table 3: Level 1 comparison of accuracies for the four evaluated models trained on our CdSprites+ dataset. *Strict* refers to percentage of completely correct samples (sample pairs in joint generation), *Feats* shows the average percentage of correct features (Level 1 has only 1 feature and *Feats* and *Strict* are thus the same) and *Letters* shows the mean percentage of correctly reconstructed letters.(*Dim*) is the latent space dimensionality.

| Model (Dim) | Txt→Img<br>Strict % | Txt→Img<br>Feats % | Img→Txt<br>Strict % | Img→Txt<br>Feats % | Img→Txt<br>Letters % | Joint<br>Strict % | Joint<br>Feats % |
|---|---|---|---|---|---|---|---|
| MMVAE (16-D) | 47 (14) | N/A | **64 (3)** | N/A | **88 (2)** | **17 (10)** | N/A |
| MVAE (16-D) | **52 (3)** | N/A | 63 (8) | N/A | 86 (2) | 5 (9) | N/A |
| DMVAE (30-D) | 33 (4) | N/A | 4 (5) | N/A | 25 (2) | 4 (6) | N/A |
| MoPoE (16-D) | 33 (3) | N/A | 10 (17) | N/A | 26 (7) | 16 (27) | N/A |
| MMVAE (24-D) | 55 (15) | N/A | 42 (3) | N/A | 31 (12) | 0 (0) | N/A |
| MVAE (24-D) | **55 (4)** | N/A | **61 (3)** | N/A | **82 (1)** | 3 (2) | N/A |
| DMVAE (36-D) | 36 (1) | N/A | 3 (3) | N/A | 21 (2) | **9 (13)** | N/A |
| MoPoE (24-D) | 35 (3) | N/A | 4 (2) | N/A | 24 (6) | 1 (1) | N/A |
| MMVAE (32-D) | 48 (3) | N/A | 36 (2) | N/A | 26 (2) | 0 (0) | N/A |
| MVAE (32-D) | **53 (5)** | N/A | **60 (2)** | N/A | **82 (2)** | **1 (1)** | N/A |
| DMVAE (46-D) | 34 (2) | N/A | 3 (2) | N/A | 20 (9) | 0 (0) | N/A |
| MoPoE (32-D) | 36 (5) | N/A | 2 (1) | N/A | 23 (7) | 0 (0) | N/A |

### 3.2 Used architecture

For all evaluated models, we used the standard ELBO loss function with the $\beta$ parameter fixed to 1. For the MVAE Wu & Goodman (2018) model, we used the sub-sampling approach where the model is trained on all subsets of modalities (i.e. images only, text only and images+text). For the image encoder and decoder, we used 4 fully connected layers with ReLU activations. In the case of the text, we used a Transformer network with 8 layers, 2 attention heads, 1024 hidden features and a dropout of 0.1.

### 3.3 Evaluation metrics

After training, we used the script for automated evaluation (provided in our toolkit) to compute the cross- and joint-coherency of the models. For cross-coherency, we generated a 10000-sample test dataset using the dataset generator and used first the images, and then captions as input to the model to reconstruct the missing modality. For joint coherency, we generated 1000 traversal samples over each dimension of the latent space (i.e. 32000 samples for a 32-D latent space) and fed these latent vectors into the models to reconstruct both captions and images.

For both the cross- and joint-coherencies, we report the following metrics: *Strict*, *Feat*, and *Letters* to provide more information on what the models are capable to do. In the first ( *Strict*) metrics, we considered the text sample as accurate only if all letters in the description were 100 % accurate, i.e. we did not tolerate any noise. For the image outputs, we considered the images as correct only if all the attributes for the given difficulty level could be detected using our pre-trained classifiers (i.e. correct classification for the shape, colour, size, position or background). For joint coherency, we considered the generated pair as correct only when both the image and captions fulfilled these criteria and were semantically matching.

For the feature-level metrics, we calculated the percentage of correctly reconstructed/generated features (e.g. whole words or image attributes such as shape) and reported the mean percentage of correct features per sample. For the image-caption cross-generation accuracy, we also calculated the average percentage of correct letters per output sample.

In the following section, we report the mean accuracies for both cross- and joint-coherency - these numbers describe the proportion of the correct outputs to all outputs.

Table 4: Level 2 comparison of accuracies for the 4 models trained on our CdSprites+ dataset. *Strict* refers to the percentage of completely correct samples (sample pairs in joint generation), *Feats* shows the average percentage of correct features (Level 2 has 2 features) and *Letters* shows the mean percentage of correctly reconstructed letters.(*Dim*) is the latent space dimensionality.

| Model (Dim) | Txt→Img<br>Strict % | Txt→Img<br>Feats % | Img→Txt<br>Strict % | Img→Txt<br>Feats % | Img→Txt<br>Letters % | Joint<br>Strict % | Joint<br>Feats % |
|---|---|---|---|---|---|---|---|
| MMVAE (16-D) | **18 (4)** | 0.8 (0.1)/2 | 41 (20) | 1.4 (0.2)/2 | 85 (4) | **3 (3)** | **0.6 (0.1)/2** |
| MVAE (16-D) | 16 (1) | 0.8 (0.0)/2 | **55 (27)** | **1.5 (0.3)/2** | **91 (6)** | 1 (1) | 0.3 (0.3)/2 |
| DMVAE (30-D) | 15 (2) | 0.8 (0.0)/2 | 4 (1) | 0.4 (0.0)/2 | 30 (2) | 0 (0) | 0.2 (0.1)/2 |
| MoPoE (16-D) | 10 (3) | 0.8 (0.0)/2 | 8 (7) | 0.7 (0.1)/2 | 40 (4) | 1 (1) | 0.2 (0.1)/2 |
| MMVAE (24-D) | 17 (5) | 0.4 (0.0)/2 | 16 (0) | 0.4 (0.0)/2 | 40 (2) | 1 (1) | 0.2 (0.0)/2 |
| MVAE (24-D) | 16 (3) | 0.4 (0.0)/2 | **52 (9)** | **0.8 (0.0)/2** | **86 (1)** | **5 (6)** | 0.3 (0.0)/2 |
| DMVAE (36-D) | **18 (2)** | **0.9 (0.0)/2** | 5 (1) | 0.4 (0.0)/2 | 24 (1) | 0 (0) | 0.2 (0.2)/2 |
| MoPoE (24-D) | 8 (3) | 0.8 (0.0)/2 | 13 (3) | 0.8 (0.1)/2 | 35 (3) | 1 (1) | **0.5 (0.1)/2** |
| MMVAE (32-D) | **17 (1)** | 0.4 (0.0)/2 | 16 (0) | 0.5 (0.0)/2 | 43 (2) | 0 (0) | 0.1 (0.0)/2 |
| MVAE (32-D) | 16 (4) | 0.8 (0.1)/2 | **40 (13)** | **1.8 (0.1)/2** | **87 (1)** | **11 (9)** | **0.8 (0.0)/2** |
| DMVAE (46-D) | **17 (1)** | **0.8 (0.0)/2** | 3 (1) | 0.4 (0.1)/2 | 24 (2) | 0 (0) | 0.1 (0.1)/2 |
| MoPoE (32-D) | 7 (2) | **0.8 (0.0)/2** | 10 (8) | 0.8 (0.1)/2 | 33 (1) | 0 (0) | 0.3 (0.2)/2 |

Table 5: Level 3 comparison of accuracies for the 4 models trained on our CdSprites+ dataset. *Strict* refers to the percentage of completely correct samples (sample pairs in joint generation), *Feats* shows the average percentage of correct features (Level 3 has 3 features) and *Letters* shows the mean percentage of correctly reconstructed letters.(*Dim*) is the latent space dimensionality.

| Model (Dim) | Txt→Img<br>Strict % | Txt→Img<br>Feats % | Img→Txt<br>Strict % | Img→Txt<br>Feats % | Img→Txt<br>Letters % | Joint<br>Strict % | Joint<br>Feats % |
|---|---|---|---|---|---|---|---|
| MMVAE (16-D) | 6 (2) | 1.2 (0.2)/3 | 2 (3) | 0.6 (0.2)/3 | 31 (5) | 0 (0) | 0.4 (0.1)/3 |
| MVAE (16-D) | 6 (0) | 1.3 (0.0)/3 | **22 (12)** | **2.1 (0.1)/3** | **85 (3)** | 0 (0) | **0.5 (0.1)/3** |
| DMVAE (30-D) | 4 (0) | 1.2 (0.0)/3 | 0 (0) | 0.4 (0.1)/3 | 22 (2) | **1 (1)** | **0.5 (0.1)/3** |
| MoPoE (16-D) | **6 (1)** | **1.6 (0.0)/3** | 0 (0) | 0.1 (0.1)/3 | 21 (5) | 0 (0) | 0.0 (0.0)/3 |
| MMVAE (24-D) | 4 (3) | 1.2 (0.3)/3 | 1 (1) | 0.8 (0.2)/3 | 31 (6) | 0 (0) | 0.3 (0.0)/3 |
| MVAE (24-D) | **7 (1)** | **1.3 (0.0)/3** | **45 (3)** | **2.4 (0.0)/3** | **91 (1)** | 0 (0) | **0.6 (0.0)/3** |
| DMVAE (36-D) | 3 (2) | 1.1 (0.1)/3 | 0 (0) | 0.2 (0.0)/3 | 18 (1) | 0 (0) | 0.1 (0.0)/3 |
| MoPoE (24-D) | 7 (4) | 1.3 (0.1)/3 | 0 (0) | 0.7 (0.1)/3 | 32 (0) | 0 (0) | 1.1 (0.1)/3 |
| MMVAE (32-D) | 5 (3) | 1.1 (0.1)/3 | 1 (1) | 0.6 (0.1)/3 | 28 (2) | 0 (0) | 0.0 (0.0)/3 |
| MVAE (32-D) | **8 (2)** | 1.3 (0.0)/3 | **59 (4)** | **2.5 (0.1)/3** | **93 (1)** | 0 (0) | **0.5 (0.1)/3** |
| DMVAE (46-D) | 5 (1) | 1.1 (0.0)/3 | 0 (0) | 0.1 (0.1)/3 | 15 (1) | 0 (0) | 0.1 (0.0)/3 |
| MoPoE (32-D) | **8 (2)** | **1.5 (0.1)/3** | 0 (1) | 0.6 (0.1)/3 | 28 (1) | 0 (0) | 0.5 (0.2)/3 |

Table 6: Level 4 comparison of accuracies for the 4 models trained on our CdSprites+ dataset. *Strict* refers to the percentage of completely correct samples (sample pairs in joint generation), *Feats* shows the average percentage of correct features (Level 4 has only 4 features) and *Letters* shows the mean percentage of correctly reconstructed letters.(*Dim*) is the latent space dimensionality.

| Model (Dim) | Txt→Img<br>Strict % | Txt→Img<br>Feats % | Img→Txt<br>Strict % | Img→Txt<br>Feats % | Img→Txt<br>Letters % | Joint<br>Strict % | Joint<br>Feats % |
|---|---|---|---|---|---|---|---|
| MMVAE (16-D) | 2 (0) | **1.6 (0.2)/4** | 0 (0) | 0.4 (0.4)/4 | 15 (3) | 0 (0) | 0.1 (0.1)/4 |
| MVAE (16-D) | 0 (0) | 1.3 (0.0)/4 | 0 (0) | 0.2 (0.3)/4 | 16 (5) | 0 (0) | 0.3 (0.6)/4 |
| DMVAE (30-D) | 1 (1) | 1.4 (0.0)/4 | 0 (0) | **0.5 (0.1)/4** | **18 (1)** | 0 (0) | **0.5 (0.1)/4** |
| MoPoE (16-D) | **3 (1)** | **1.6 (0.2)/4** | 0 (0) | **0.5 (0.1)/4** | 16 (3) | 0 (0) | 0.1 (0.1)/4 |
| MMVAE (24-D) | 3 (3) | **1.7 (0.4)/4** | **1 (2)** | 0.7 (0.4)/4 | **27 (9)** | 0 (0) | **0.5 (0.2)/4** |
| MVAE (24-D) | **4 (1)** | 1.2 (0.1)/4 | 0 (1) | **2.4 (0.0)/4** | 14 (1) | 0 (0) | 0.2 (0.1)/4 |
| DMVAE (36-D) | 0 (1) | 1.3 (0.0)/4 | 0 (0) | 0.2 (0.0)/4 | 14 (1) | 0 (0) | 0.2 (0.0)/4 |
| MoPoE (24-D) | 2 (1) | 1.4 (0.0)/4 | 0 (0) | 0.7 (0.1)/4 | 21 (3) | 0 (0) | 0.1 (0.2)/4 |
| MMVAE (32-D) | 1 (1) | 1.6 (0.0)/4 | 0 (0) | 0.9 (0.0)/4 | 21 (0) | 0 (0) | 0.2 (0.0)/4 |
| MVAE (32-D) | 2 (1) | 1.1 (0.1)/4 | 0 (1) | **1.1 (0.0)/4** | 12 (3) | 0 (0) | **0.4 (0.2)/4** |
| DMVAE (46-D) | 1 (1) | 1.2 (0.0)/4 | 0 (0) | 0.1 (0.0)/4 | 14 (1) | 0 (0) | 0.1 (0.0)/4 |
| MoPoE (32-D) | **4 (0)** | **1.7 (0.1)/4** | 0 (0) | 0.5 (0.3)/4 | **20 (3)** | 0 (0) | 0.2 (0.2)/4 |

Table 7: Level 5 comparison of accuracies for the 4 models trained on our CdSprites+ dataset. *Strict* refers to the percentage of completely correct samples (sample pairs in joint generation), *Feats* shows the average percentage of correct features (Level 5 has 5 features) and *Letters* shows the mean percentage of correctly reconstructed letters. (*Dim*) is the latent space dimensionality.

| Model (Dim) | Txt→Img<br>Strict % | Txt→Img<br>Feats % | Img→Txt<br>Strict % | Img→Txt<br>Feats % | Img→Txt<br>Letters % | Joint<br>Strict % | Joint<br>Feats % |
|---|---|---|---|---|---|---|---|
| MMVAE (16-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | 0.4 (0.2)/5 | 16 (0) | 0 (0) | 0.7 (0.4)/5 |
| MVAE (16-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | **0.6 (0.0)/5** | **27 (1)** | 0 (0) | 0.2 (0.2)/5 |
| DMVAE (30-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | 0.6 (0.1)/5 | 18 (2) | 0 (0) | **0.7 (0.1)/5** |
| MoPoE (16-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | 0.3 (0.2)/5 | 15 (1) | 0 (0) | 0.5 (0.7)/5 |
| MMVAE (24-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | 0.6 (0.1)/5 | 17 (2) | 0 (0) | 0.5 (0.1)/5 |
| MVAE (24-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | 0.6 (0.0)/5 | **25 (3)** | 0 (0) | 0.3 (0.0)/5 |
| DMVAE (36-D) | **1 (0)** | 1.8 (0.0)/5 | 0 (0) | 0.6 (0.1)/5 | 14 (0) | 0 (0) | 0.5 (0.1)/5 |
| MoPoE (24-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | **0.7 (0.0)/5** | 17 (1) | 0 (0) | **1.0 (0.0)/5** |
| MMVAE (32-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | 0.4 (0.1)/5 | 15 (0) | 0 (0) | 0.5 (0.4)/5 |
| MVAE (46-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | 0.6 (0.1)/5 | **24 (2)** | 0 (0) | 0.6 (0.1)/5 |
| DMVAE (32-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | 0.4 (0.1)/5 | 14 (1) | 0 (0) | 0.4 (0.1)/5 |
| MoPoE (32-D) | 0 (0) | 1.8 (0.0)/5 | 0 (0) | **0.7 (0.3)/5** | 17 (2) | 0 (0) | **1.1 (0.1)/5** |

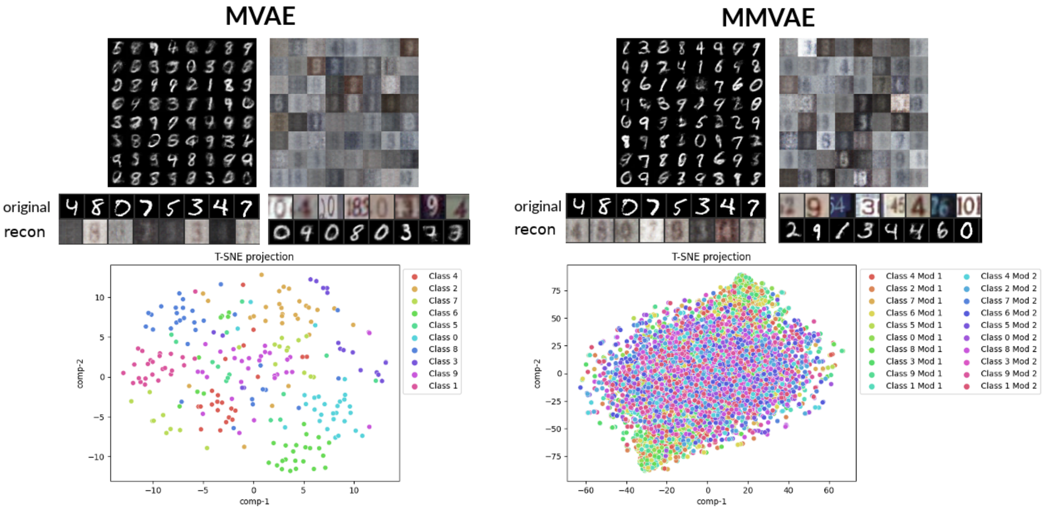

Figure 1: Results for the MVAE and MMVAE models trained on the MNIST-SVHN dataset using our toolkit. For MMVAE, we used the DREG objective as proposed by the authors, MVAE was trained with ELBO. We used the encoder and decoder networks from the original implementations. The top figures are traversals for each modality, below we show cross-generated samples. The bottom figures are T-SNE visualizations of the latent space - please note that for MVAE we show samples from the single joint posterior, while for MMVAE we show samples for both modality-specific distributions.

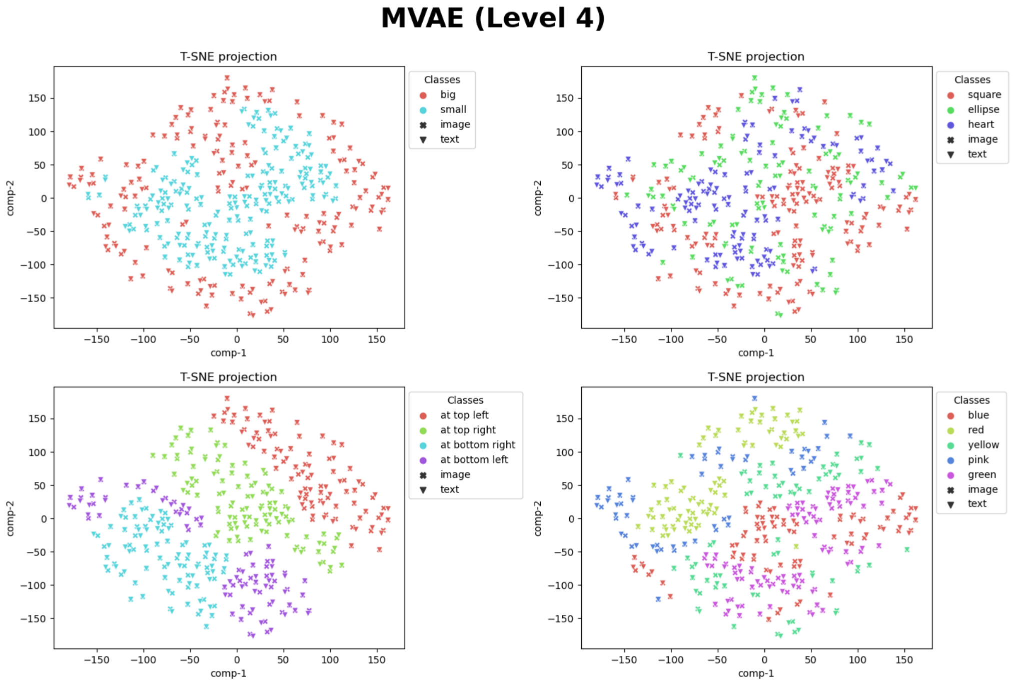

Figure 2: T-SNE visualizations for the MVAE model's (16-D) joint latent space trained on CdSprites+ Level 4. We show the latent space for each of the 4 features (size, shape, position and colour) individually.

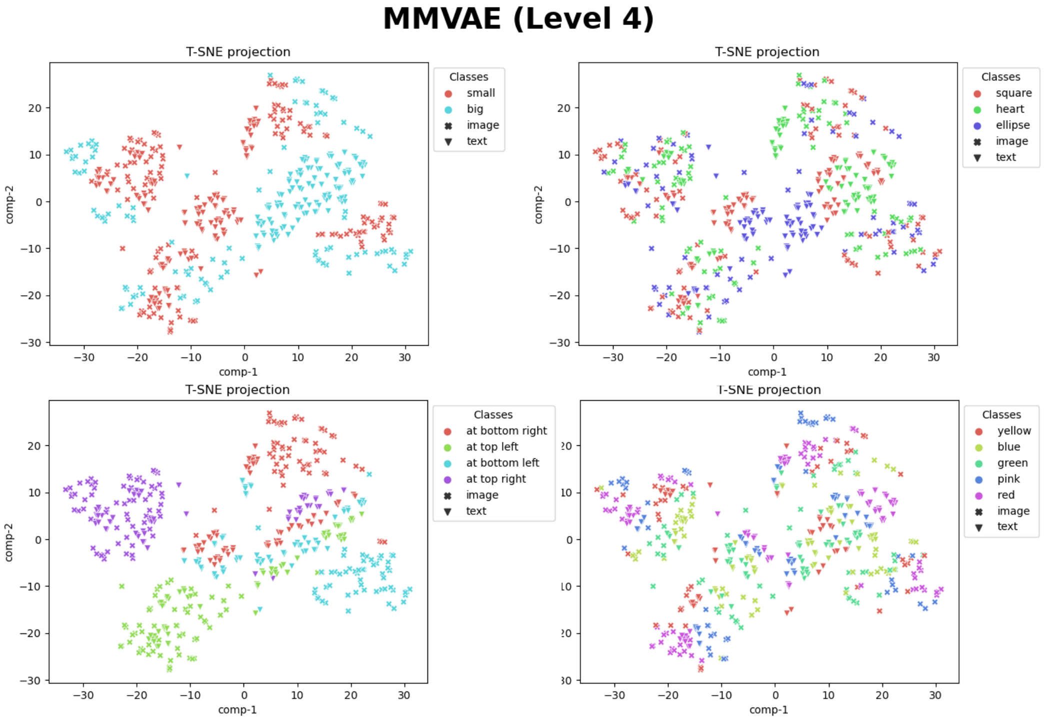

Figure 3: T-SNE visualizations for the MMVAE model's (24-D) unimodal latent spaces trained on CdSprites+ level 4. We show the latent space for each of the 4 features (size, shape, position and colour) individually.

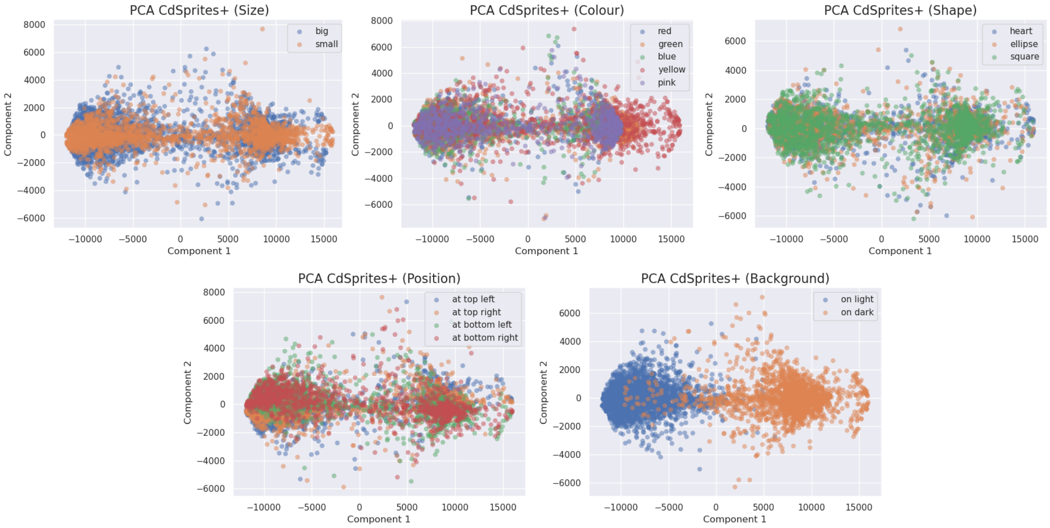

Figure 4: PCA calculated on the images in our CdSprites+ dataset, Level 5. We show a separate figure for each of the 5 features (size, shape, position and colour).

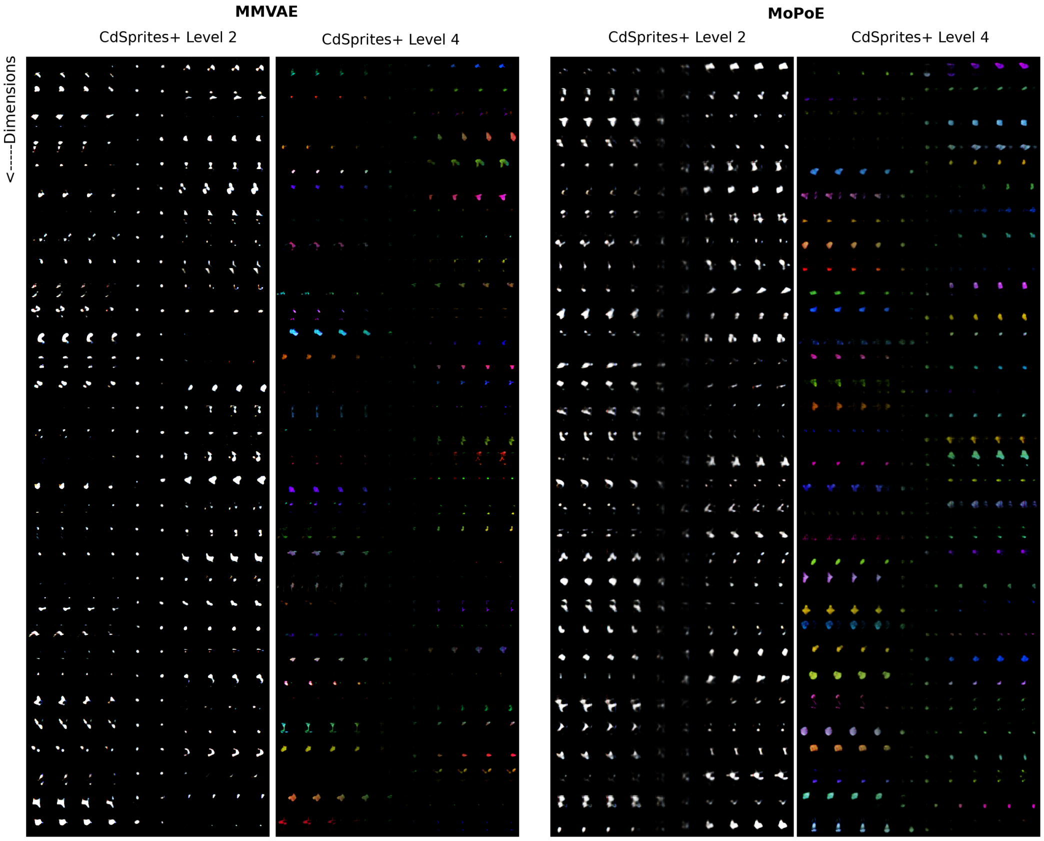

Figure 5: Image traversals for the MMVAE and MoPoE models for the CdSprites+ Levels 2 and 4. Each row is one out of 32 dimensions of the latent space, each column is the single sampled vector from the traversal range (-6,6).

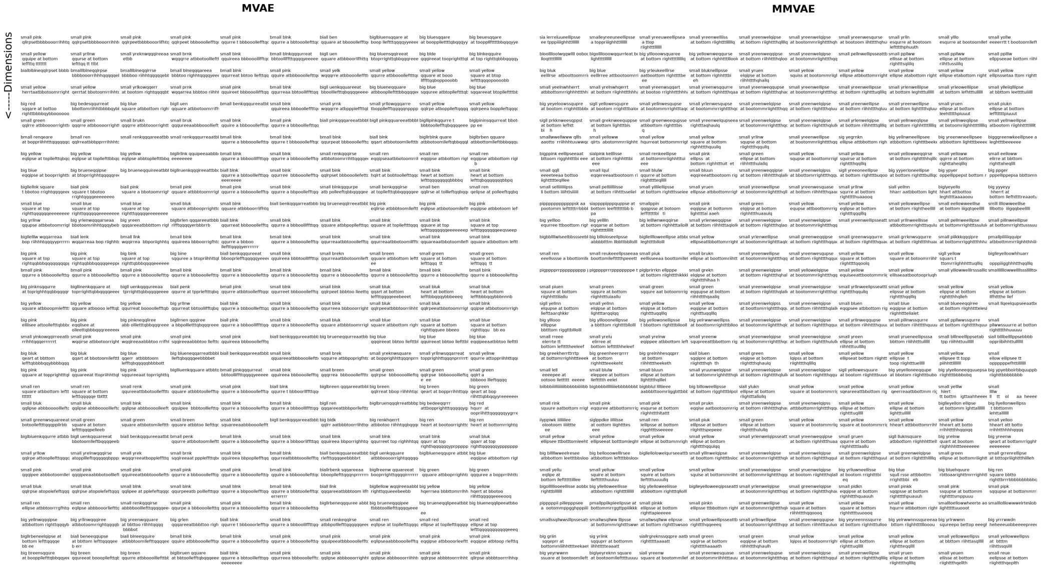

Figure 6: Text traversals for the MVAE and MMVAE models for the CdSprites+ Level 4. Each row is one out of 32 dimensions of the latent space, each column is the single sampled vector from the traversal range (-6,6). Note that we did not set the desired length of the text output, the model thus always generated the maximum number of characters.

### 3.4 Detailed Results

In Tables 3, 4, 5, 6 and 7, we show the comparison of the MVAE, MMVAE, DMVAE and MoPoE models on the 5 difficulty levels of the CdSprites+ dataset. Here we varied the latent dimensionality (16-D to 32-D) with the fixed batch size of 32. The values are the mean cross-generation and joint-generation accuracies over 5 seeds with the standard deviations listed in brackets. According to the *Strict* metrics (with zero noise tolerance, see Sec. 3.3), all models failed in both tasks at Levels 4 and 5. The *Feature* and *Letter* accuracies significantly decrease across levels as the complexity increases. You can see the T-SNE visualizations for the MVAE and MMVAE models trained on Level 4 in Figs. 2 and 3 .

## 4 Verifying correctness of model implementation

To verify the correctness of our implementation for each model, we have reproduced selected experiments from the original papers using our toolkit. We provide both the original and our results below.

### 4.1 MMVAE

To verify that our implementation of the MMVAE (Shi et al., 2019) model is correct, we reproduced the experiments using the MNIST-SVHN dataset. We used the same model configuration and parameters as in the original report, i.e. the Mixture-of-Experts mixing with the DREG training objective, latent size 20 and 30 samples drawn from the joint posterior and the likelihood scaling for each modality was adjusted according to the varying dimensionalities. We used the same encoder and decoder architectures as in the original paper. After training, we calculated the joint- and cross-coherencies using the adapted original evaluation script (please see the original paper for the evaluation details). The results are shown in Table 8, the config files for reproducing the experiment are also provided on our GitHub. Please note that the results in Table 8 are different from those

Table 8: Reproduced MNIST-SVHN results for the MMVAE model with our multimodal VAE toolkit. We show the digit classification accuracies (%) of latent variables (*MNIST*) and (*SVHN*), and the probability of digit matching (%) for cross- and joint-generation. For our results, we also show in brackets the variance of the results calculated over 3 seeds.

| Version | MNIST | SVHN | MNIST $\rightarrow SVHN$ | SVHN $\rightarrow MNIST$ | Joint |
|---|---|---|---|---|---|
| Original | 91.3 | 68.0 | 86.4 | 69.1 | 42.1 |
| Reproduced (Ours) | 87.6 (5.2) | 70.4 (4.6) | 82.7 (5.2) | 72.5 (4.9) | 45.3 (3.1) |

Table 9: Reproduced FashionMNIST results for the MVAE model with our multimodal VAE toolkit. We show the estimated marginal log-likelihoods (lower is better). For our results, we also show in brackets the variance of the results calculated over 3 seeds.

| Version | $logp(x_1)$ | $logp(x_1, x_2)$ | $logp(x_1\|x_2)$ |
|---|---|---|---|
| Original | -232.535 | -233.007 | -230.695 |
| Reproduced (Ours) | -234.15 (1.52) | -233.89 (2.61) | -232.56 (3.12) |

Table 10: Reproduced PolyMNIST results for the MoPoE model with our multimodal VAE toolkit. We show the Coherence Accuracy (%) of conditionally generated samples (excluding the input modality) (*1 Mod*, *2 Mods*, *3 Mods* and *4 Mods* stand for the number of input modalities) and the joint coherence (*Joint*). For our results, we also show in brackets the variance of the results calculated over 3 seeds.

| Version | 1 Mod | 2 Mods | 3 Mods | 4 Mods | Joint |
|---|---|---|---|---|---|
| Original | 67 | 78 | 80 | 83 | 12 |
| Reproduced (Ours) | 66 (4) | 73 (5) | 81 (3) | 82 (5) | 11 (3) |

Table 11: Reproduced MNIST-SVHN results for the DMVAE model with our multimodal VAE toolkit. We show the probability of digit matching (%) for cross- and joint-generation. For our results, we also show in brackets the variance of the results calculated over 3 seeds.

| Version | MNIST $\rightarrow SVHN$ | SVHN $\rightarrow MNIST$ | Joint |
|---|---|---|---|
| Original | 88.1 | 83.7 | 44.7 |
| Reproduced (Ours) | 84.5 (4.7) | 82.2 (3.1) | 44.9 (3.6) |

depicted in the main paper, Table 3. This is because here we unified the training hyperparameters with the original paper setup. However, we found that setting the likelihood scaling to 1 for both modalities produces more balanced results (in terms of MNIST/SVHN accuracies) and used thus this setup for the comparative study.

### 4.2 MVAE

In the original MVAE paper (Wu & Goodman, 2018), the results are reported in terms of marginal log-likelihoods. We reproduced the FashionMNIST experiment with a 64-D latent space, batch size 100, and likelihood scaling of 10 for the labels and 1 for the images, as reported in the public code. The results can be seen in Table 9.

### 4.3 MoPoE

For verification that the MoPoE model (Sutter et al., 2021) is correct, the reproduction was performed on the PolyMNIST dataset. Based on the original implementation, we used the 512-D latent space, Laplace prior distributions, and $\beta = 2.5$. After training, we calculated the cross-coherencies conditioned on 1, 2, 3 or 4 modalities as reported in the paper. The results are shown in Table 10.

### 4.4 DMVAE

We reproduced the MNIST-SVHN experiment for the DMVAE model (Lee & Pavlovic, 2021). The reproduced model configuration included shared latent dimensionality $Dim_{shared} = 10$, the private latent dimensionalities were $Dim_{MNIST} = 1$ and for $Dim_{SVHN} = 4$. The used $\beta$ parameter was 1, and batch size 100. We used the same encoder and decoder networks and an adapted script for calculating the cross- and joint-coherencies. The results are in Table 11.

## References

Mihee Lee and Vladimir Pavlovic. Private-shared disentangled multimodal vae for learning of latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1692–1700, 2021.

Yuge Shi, Brooks Paige, Philip Torr, et al. Variational mixture-of-experts autoencoders for multimodal deep generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

Thomas M. Sutter, Imant Daunhawer, and Julia E Vogt. Generalized multimodal ELBO. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=5Y21V0RDBV`.

Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. *Advances in Neural Information Processing Systems*, 31, 2018.