# Reproducible Machine Learning

SalmonBagel: Bonnie Hu[#1], Niloofar Khoshsiyar[#2], Chianyu Liu[#3]

[#]*COMP 551 Applied Machine Learning, McGill University, Canada*

[1]guanqing.hu@mail.mcgill.ca, 260556970

[2]niloofar.khoshsiyar@mcgill.ca, 260515304

[3]qianyu.liu@mail.mcgill.ca, 260576898

***Abstract*--- Reproducibility refers to the ability of a researcher to duplicate the result of a prior study. While it should be a rudimentary condition for any research, majority of researchers tend to believe that there is a crisis in literature reproducibility [1]. With this is mind, our goal is to duplicate the work of a submission from the International Conference on Learning Representations and evaluate its reproducibility.**

***Keywords*--- reproducibility, generative models, product-of-experts inference network, joint variational autoencoder**

## I. Introduction

One of the challenges facing machine learning research today and scientific research community in general is to ensure that results published in academic journals and conference proceedings are reliable and reproducible. In support of this, the goal of this project is to investigate the reproducibility of empirical results submitted to the International Conference on Learning Representations. Namely, we aim to reproduce the experiments, results, and claims reported in Generative Models of Visually Grounded Imagination [2] [3].

This paper is organized as such: we first provide background information about variational autoencoder and summarize the proposed solution in Section II. Then, we outline the metrics and methods we followed to evaluate said solution's reproducibility in Section III. In Section IV and V, we discuss our results from reproducing the experiments and compare them to the claims made in the paper. Finally, in Section 0 and VII, we conclude this paper with some discussion based on our observations and findings.

## II. Paper Overview

The phrase *visually grounded imagination* means the ability to create images from novel semantic concepts. To understand this better, we use a two-person game as an example. The first person imagines a visual concept $C$ and communicates a description of it, which we call $y$. Then the second person interprets it, as $z$, and to test how well the concept was understood, he *grounds* the description by communicating back a set of real images, $S$, which he *imagined* to describe the concept. Finally, the first person validates how well the set $S$ matches the concept $C$. The steps that are explained above are similar to how the authors approached this problem.

One of the main focuses of this paper is to be able to handle different levels of abstraction in the descriptions. For example, a fully defined description of a face image can be "black haired, not smiling, female", whereas an incomplete description or an abstract concept can be only "black haired". What differentiates this paper with the previous works is that the proposed model can generate images even though the attributes are not fully specified.

For this goal, the authors propose an extension to variational autoencoders (VAE) in multimodal setting, by introducing a joint generative model (JVAE). In this scenario, we have both an image $x$ and an attribute

**BiVCCA**

$$J(x,y,\theta,\phi) = \mu\left(\mathrm{E}_{q_{\phi_x}(z|x)}[\log p_\theta(x,y|z)] - KL\left(q_{\phi_x}(z|x), p_\theta(z)\right)\right) + (1-\mu)\left(\mathrm{E}_{q_{\phi_x}(z|y)}[\log p_\theta(x,y|z)] - KL\left(q_{\phi_y}(z|y), p_\theta(z)\right)\right) \quad (1)$$

**JMVAE**

$$J(x,y,\theta,\phi) = elbo(x,y,\theta,\phi) - \alpha\left[KL\left(q_\phi\left(z|x,y\right), q_{\phi_x}(z|x)\right)\right) + KL\left(q_\phi\left(z|x,y\right), q_{\phi_y}(z|y)\right)\right)\Big] \quad (2)$$

**TELBO**

$$L\left(\theta,\phi,\phi_x,\phi_y\right) = \mathrm{E}_{\tilde{p}(x,y)}[elbo(x,y,\theta,\phi) + elbo(x,\theta,\phi) + elbo(y,\theta,\phi)] \quad (3)$$

vector $y$ as variables, with a novel objective function called triple evidence lower bound, or TELBO. The model is trained with paired data $\{x_n, y_n\}$, however at the test time unpaired data is used, which means either a description or an image. To do so, they use three inference networks $q(z|x,y)$, $q(z|x)$ and $q(z|y)$ which then allow for a translation between an image and a description, $p(y|x)$ and $p(x|y)$.

In order to handle concepts with various levels of abstraction, the paper offers a product of experts (POE) inference network such that if no attributes are specified the posterior is the same as the prior. As the concept becomes less abstract, meaning more attributes are specified, the posterior becomes narrower and includes less diverse set of images.

The last step is to evaluate the generated set of images, for which the authors propose an evaluation criteria called *the 3 C's of visual imagination*. The 3 Cs refers to correctness, coverage, and compositionality. First, correctness is a metric to assess whether each image in the set of outputs $S$ is consistent with its given description $y$. Second, coverage checks the diversity of the missing attributes across different images. Third, compositionality, which is the essence of imagination, is a measure of how well the model works for novel combinations of attributes, which were not seen in training.

The paper trains the JVAE model on two datasets, MNIST-with-attributes (MNIST-A) and CelebA. The proposed objective TELBO is compared to two other objectives JMVAE [4] and BiVCCA [5]. The three models shares the same architecture when running on MNIST-A dataset, while differs in the objective functions. Mathematically, these objective functions are given as Equation (1), (2), and (3). More details on these objectives and their significance can be found in the original paper. The hyperparameters used in paper are chosen so that they maximize correctness on a validation set.

## III. Problem Representation

### A. Reproducibility scope

On both MNIST-A and CelebA datasets, the author ran the experiments on an independent and identically distributed (i.i.d.) split dataset and a compositional split dataset, where the latter is used to evaluation compositionality. Given the augmented dataset has $10 \times 2 \times 3 \times 4 = 240$ possible label combinations, the compositional split refers to a dataset with non-overlapping label combinations. The goal is that when the model is fed a compositionally novel concept, it is able to generate more diverse set of images.

In comparison, we decided to narrow to scope of this project by focusing on reproducing the results of the paper [2] for i.i.d MNIST-A dataset, using the source code provided by the author.

### B. Reproducibility metrics

To examine the reproducibility of a scientific literature, we followed a series of defined metrics [6]. The metrics largely focus on the availability of source materials and feasibility of the reimplementation task. Table 1 is a list of indexed metrics we followed to assess the reproducibility of this paper.

TABLE 1 SUMMARY OF REPRODUCIBILITY METRICS

| | **Reproducibility Metrics** |
|---|---|
| 1 | Availability of datasets |
| 2 | Availability of code (including version and dependencies) |
| 3 | Availability all hyperparameters |
| 4 | Alignment between the paper and the code |
| 5 | Clarity of code & paper |
| 6 | Details of computing infrastructure used |
| 7 | Computation requirements (time, memory, number/type of machines) |
| 8 | Reimplementation effort (time, and expertise) |
| 9 | Interaction with the authors |

### C. Observations

As most of evaluation criteria in the metrics involves the presence of the source code, we began this project by reaching out to the principal author of the paper via email, who agreed to share with us the source code from the experiments, as well as two augmented MNIST-A and CelebA datasets.

We would like to point out that the MNIST-A dataset provided by the author is a non-proprietary version that is similar but not identical to the dataset used during their experiments. The author assured us that we should expect the reproduced results to be close to the results reported in the paper.

The source code includes experiments for all three objectives: TELBO, JMVAE, and BiVCCA. To distinguish these experiments, we will refer to them by their respective objective functions for the rest of the paper. From the paper and the source code, we made the following observations.

First, the paper does not reference the hardware nor the infrastructure used to run the experiments. However, from our email exchange with the author, we learned that the authors used a Nvidia Titan X

graphics card, which leads to a training time of around 27 hours for the TELBO experiment.

Second, the author tested multiple values for their hyperparameters, namely, they tested different values of $\alpha_y$ in all three experiments and $private\_py\_scaling$ for TELBO. Apart from those, other hyperparameters are defined a single value in the source code. The best practice hyperparameters are not specified in the paper, but can be found in the source code repository.

Lastly, we observed that the authors may have run the experiments on a cluster of computers. The authors used Slurm, a cluster management and job scheduling system for Linux. Running the experiments on multiple computers can greatly ease the training process, allowing users to try out different hyperparameters or datasets.

## IV. Algorithm Selection and Implementation

### A. Hardware setup

We ran our experiments on Google Compute Engine, using one Nvidia Tesla P100 accelerator. We initially used one Nvidia Tesla K80 accelerator, and abandoned it because of time constraint. As suggested by the author, the K80 accelerators are high precision GPUs for scientific computing and tend to be sub-optimal for most deep learning purposes. With the K80 setup, the training time is approximately 3 times that of our P100 setup.

Furthermore, we setup 3 identical instances to run each experiment independently.

### B. Running the code

Following the instructions from the source code package, we first set up a virtual environment and installed the dependencies required to run the code. We also modified the training script in order to run only the best practice hyperparameters without Slurm.

We ran each experiment once, with the following hyperparameters (consistent with variable names in the source code):

Table 2 Summary of hyperparameters used in experiments

| Hyperparameters | Value |
|---|---|
| num_training_steps | 250000 |
| alpha_x | 1 |
| product_of_experts | 1 |
| num_latent | 10 |
| alpha_y | 50 |
| l1_reg | 5e-6 |
| TELBO specific | |
| stop_elboy_gradient | 1 |
| stop_elbox_gradient | 1 |
| private_py_scaling | 50 |
| JMVAE specific | |
| jmvae_alpha | 1 |
| BiVCCA specific | |
| bivcca_mu | 0.7 |

For each experiment, the training time and number of steps are summarized below:

Table 3 Summary of training time

| Experiment | Training Speed (s/step) | # Steps | Training Time (approx. to the nearest hour) |
|---|---|---|---|
| TELBO | 0.36 | 250000 | 25 |
| JMVAE | 0.27 | 250000 | 19 |
| BiVCCA | 0.29 | 50000* | 4 |

* Number of steps reduced due to time limitation.

## V. Testing and Validation

### A. Reproducing results

The results below are obtained by running the code provided by the authors. The datasets was split into train, val, and test set of 85%, 5%, and 10% respectively as mentioned in the paper.

#### 1) Validation results

To measure correctness and coverage, the author trained an observation classifier on the i.i.d dataset. The observation classifier plays the role of a human observer that checks not only that the images look realistic, but also that they have the desired attributes. However, it is important to note that while the observation classifier ease the validation and test process, it is not always correct. Table 4 is a summary of the accuracies of the classifier on different attributes.

Every model is validated using the observation classifier to evaluate the images it generated in terms of the accuracies of class label, scale, orientation, location, and overall accuracy.

Our experimental results, organized in Table 5, convey similar performance relations of the three methods as what is shown in the paper. JMVAE method gets the highest validation accuracies and the overall accuracy of TELBO is 5.54% lower than JMVAE. BiVCCA gives the lowest results with an average accuracy at 67.41%.

Table 4 Accuracy of the observation classifier given in the paper

| Label | Accuracy (%) |
|---|---|
| class label | 91.18 |
| scale | 90.56 |
| orientation | 92.23 |
| location | 100 |

TABLE 5 VALIDATION RESULTS SUMMARY

| Label | Accuracy (%) | | |
|---|---|---|---|
| | **TELBO** | **JMVAE** | **BiVCCA** |
| Class label | 78.23 | 90.43 | 21.03 |
| Scale | 85.66 | 89.78 | 82.49 |
| Orientation | 86.55 | 92.38 | 67.82 |
| Location | 100 | 99.99 | 98.30 |
| **Overall** | **87.61** | **93.15** | **67.41** |

The validation output also includes several similar sheets of images. The difference between them is not specified in the paper. One example of image output generated by TELBO can be seen in Figure 1 where the images with one or more wrong specifications are labeled in red. The meaning of each number in the specification is not explained in the paper either, but it is easily derived from the figure. A summary of specification meanings can be seen in Table 6.

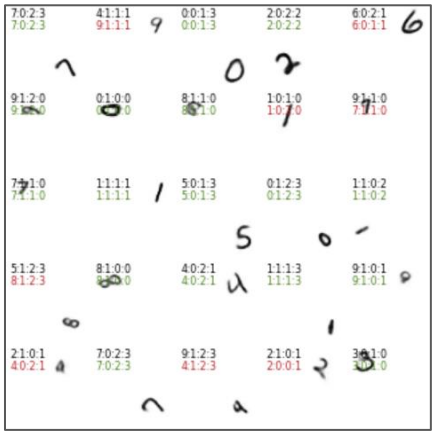

Figure 1 Image result of our experiment for TELBO

TABLE 6 MEANING OF IMAGE LABEL IN FIGURE 1

| Index | Meaning | Values |
|---|---|---|
| 1 | Digit | 0-9 |
| 2 | Size | 0: big |
| | | 1: small |
| 3 | Rotation | 0: rotate right |
| | | 1: no rotation |
| | | 2: rotate left |
| 4 | Position | 0: top left |
| | | 1: top right |
| | | 2: bottom left |
| | | 3: bottom right |

*2) Test results: image evaluation using 3 C's*

The image evaluation results using 3 C's on the test set can be seen in Table 7. The results from the paper can be seen in Figure 2. Same table format has been used for comparison convenience.

When the label is fully specified (# Attributes equals to 4), the correctness of TELBO is 0.28% better than the result in the paper, while the correctness of JMVAE and BiVCCA are about 3.5% lower than the paper results. When the test concepts are abstract with one or more attributes missing. The results of the paper in Figure 2 show that the correctness score of JMVAE starts to drop, while TELBO and BiVCCA remain stable. In our test results in Table 7, however, the correctness score for JMVAE remains stable, while the correctness scores of TELBO and BiVCCA increase.

In terms of coverage (diversity), the paper results in Figure 4 shows that TELBO is higher than the other methods. The results of our experiments give the same trend as well.

TABLE 7 IMAGE EVALUATION OF OUR EXPERIMENTS

| Method | # Attributes | Coverage (%) | Correctness (%) |
|---|---|---|---|
| i.i.d. split * | | | |
| TELBO | 4 | - | $82.36 \pm 0.28$ |
| JMVAE | | - | $81.58 \pm 0.17$ |
| BiVCCA | | - | $63.46 \pm 0.45$ |
| TELBO | 3 | $89.04 \pm 0.57$ | $83.36 \pm 0.40$ |
| JMVAE | | $88.11 \pm 0.45$ | $80.97 \pm 0.29$ |
| BiVCCA | | $78.49 \pm 0.60$ | $69.31 \pm 0.51$ |
| TELBO | 2 | $89.95 \pm 0.17$ | $83.19 \pm 0.64$ |
| JMVAE | | $89.45 \pm 0.21$ | $79.46 \pm 0.91$ |
| BiVCCA | | $83.33 \pm 0.41$ | $72.11 \pm 1.04$ |
| TELBO | 1 | $91.20 \pm 0.15$ | $84.50 \pm 1.33$ |
| JMVAE | | $90.03 \pm 0.13$ | $81.28 \pm 1.04$ |
| BiVCCA | | $85.33 \pm 0.22$ | $71.87 \pm 1.49$ |

* Only i.i.d. split was reproduced.

| Method | #Attributes | Coverage (%) | Correctness (%) |
|---|---|---|---|
| iid split | | | |
| TELBO | 4 | - | $82.08 \pm 0.56$ |
| JMVAE | | - | $85.15 \pm 0.26$ |
| BiVCCA | | - | $67.38 \pm 0.69$ |
| TELBO | 3 | $91.14 \pm 0.53$ | $81.63 \pm 0.38$ |
| JMVAE | | $88.52 \pm 0.37$ | $82.00 \pm 0.37$ |
| BiVCCA | | $85.28 \pm 0.68$ | $70.68 \pm 0.87$ |
| TELBO | 2 | $90.32 \pm 0.57$ | $82.03 \pm 1.37$ |
| JMVAE | | $87.89 \pm 0.69$ | $81.02 \pm 1.05$ |
| BiVCCA | | $85.09 \pm 0.76$ | $72.33 \pm 2.31$ |
| TELBO | 1 | $90.94 \pm 0.19$ | $83.67 \pm 1.70$ |
| JMVAE | | $88.70 \pm 0.35$ | $81.58 \pm 1.78$ |
| BiVCCA | | $85.53 \pm 0.27$ | $68.36 \pm 2.21$ |
| Compositional split | | | |
| TELBO | 4 | - | $75.61 \pm 1.43$ |
| JMVAE | | - | $76.86 \pm 1.30$ |
| BiVCCA | | - | $68.58 \pm 1.02$ |

(a) Evaluation of different approaches on the test set. Higher numbers are better. We report standard deviation across 5 splits of the test set.

Figure 2 Results on MNIST-A dataset from the paper

The test output also includes 400 mean images. Each image contains two specifications and 10 sub-

images. An example of the image can be seen in Figure 3 with attribute sets '_:_:1:_' (no rotation) and '5:0:1:2' (big 5, no rotation, left bottom). Further explanation from the author is required to understand the meaning of the image.

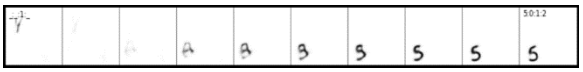

Figure 3 One example of mean image file

### B. Additional JMVAE results

Before we receive the code and datasets from the author, we started by running baseline JMVAE code from its author's Github repository [7]. We ran 500 epochs on MNIST dataset and the running time was 2.5 hours. Table 8 shows our preliminary results of this run. Our results were slightly better than the ones discussed in the JMVAE paper [4].

TABLE 8 JMVAE PRELIMINARY LIKELIHOOD RESULTS

| | In JMVAE paper | Our results |
|---|---|---|
| **Marginal log-likelihood** | -89.20 | -88.58 |
| **Conditional log-likelihood** | -86.97 | -86.31 |

* Higher numbers are better.

The authors of the JMVAE paper tested their model with different alpha settings, and concluded that lower alpha, e.g. $\alpha = 0.01$, produces higher correctness. In this paper, we noticed that the JMVAE alpha hyperparameter is set to 1. Given more time, we would like to explore our JMVAE experiment with different hyperparatmeter set.

## VI. DISCUSSION

Comparing our results in Table 7 and results in the paper in Figure 2, we see a significant resemblance between the two in the performance of TELBO, JMVAE, and BiVCCA models when testing on i.i.d. split dataset. In fact, the average percentage difference is at 1.18%. Given that we ran the source code provided by the author with little change to the script, this finding is expected.

A key difference between our experiments and those of the authors is the number of steps that we used. We ran TELBO and JMVAE for 250,000 steps, and BiVCCA for 50,000 steps, while the authors trained their models for 500,000 steps. The similarity in coverage and correctness implies that 250,000 steps is enough to train a well-performed model. The additional steps in author's experiments did not lead to overfitting, but it did not make a significant improvement either. Moreover, the author noted in their paper that the models do not tend to overfit in their experiment. Indeed, our reproduced results verify this claim.

As the main purpose of this paper is to determine the reproducibility of the results, below we discuss each reproducibility metric from Table 1 in detail:

*Availability of datasets:* An equivalent MNIST-A and CelebA datasets were provided to us upon our request to the authors.

*Availability of code:* The source code is available to us with slight adjustment to fit our hardware and time constraints.

*Availability of hyperparameters:* All hyperparameters are defined in the code. But the set of hyperparameters that yields the highest correctness and coverage is not highlighted.

*Alignment between the paper and the code:* The code implements the architecture and the objective functions described in the paper. Also, our quantitative results match nicely with those in the paper. However, we noticed a misalignment in the qualitative results, namely between the reproduced image and the images in the paper. Reproduced images such as Figure 1 and Figure 3 are not found in the paper, creating some confusion. Moreover, without explanations on the reproduced images, our interpretation of these images may not be correct. Overall, we were not able to obtain a full understanding the process that transforms the raw images we have to the findings presented in the paper. We believe that with more clarification, we can do more qualitative analysis on the output images.

*Clarity of code & paper:* The paper explains the details of the methodology clearly and the code contains straightforward comments and variable names. In addition, the author proactively informed us when an updated version of code is available.

*Computing infrastructure used:* The paper does not provide details on the infrastructure used to run the experiments.

*Computation requirements:* The paper does not provide details on the resource requirements to run the experiments. However, we can estimate training time based on author's running time of TELBO experiment with Nvidia Titan X.

*Reimplementation effort:* Given availability of dataset and source code, the reimplementation effort is

reasonably straightforward for our expertise (or lack thereof) in the subject of generative learning.

*Interaction with the authors:* Throughout this project, we could easily contact the principal author regarding queries in the experiments, and the author always replied to us within a day.

Overall, from our experience, we observe that the main limitation in reproducing this paper is the lack of infrastructural and computational resources required to reproduce the all the experiments performed by the authors. To run the proposed model and two baseline models with one set of hyperparameters for 500,000 steps requires a total of 130 hours using a P100 accelerator. Considering these resource requirements, reproducing this paper may not be suitable for all demographics.

## VII. Conclusion

In conclusion, we learned that in order to facilitate reproducibility of the scientific research work, it is important to be as transparent as possible about the methodology, if possible, to make the code open source and accessible to the peer researchers.

## VIII. Statement of Contribution

**Bonnie Hu**: Reproduced TELBO results, contributed to the report

**Niloofar Khoshsiyar**: Reproduced BiVCCA results, contributed to the report

**Chianyu Liu**: Reproduced JMVAE results, contributed to the report

This project is the product of a joint effort of Bonnie, Niloofar, and Chianyu. ***We hereby state that all work presented in this report is that of these authors***.

## IX. Acknowledgment

We would like to acknowledge the principal author for the kind guidance and assistance.

## X. References

[1] M. Baker, "1500 Scientist lift the lid on reproducibility," *Nature*, 25 May 2016.

[2] R. Vedantam, I. Fischer, J. Huang and K. Murphy, "Generative Models of Visually Grounded Imagination," in *International Conference on Learning Representations*, 2017.

[3] "google/joint_vae," GitHub, 2017. [Online]. Available: https://github.com/google/joint_vae. [Accessed 15 Dec 2017].

[4] M. Suzuki, K. Nakayama and Y. Matsuo, "Joint Multimodal Learning with Deep Generative Models," in *International Conference on Learning Representations*, 2017.

[5] W. Wang, X. Yan, H. Lee and K. Livescu, "Deep Variational Canonical Correlation Analysis," in *International Conference on Learning Representations*, 2016.

[6] J. Pineau, "Reproducibility Metrics," Montreal, 2017.

[7] "masa-su/Tars," GitHub, 2017. [Online]. Available: https://github.com/masa-su/Tars. [Accessed 15 Dec 2017].

## XI. Appendix

The challenge that this paper is addressing is to generate images from novel abstract concepts and compositionally novel concrete concepts, input as a set of attribute descriptions. The authors propose a product-of-experts inference network using a joint variational autoencoder model, with a new objective called triple evidence lower bound, or TELBO. It is claimed in the paper that this objective performs on par with same architecture with JMVAE objective and significantly outperforms one with BiVCCA objective in correctness. Also, it is shown in the paper that TELBO outperforms both baselines when the given attributes in an image description are not fully specified (coverage and compositionality).

In an effort to reproduce the results of this paper and confirm the claims, our team worked on a project aiming to replicate the methodology of the authors. In this path, we find the paper well-written and clear. The source code is publicly available on Github, which facilitates the reproduction of the models. While the MNIST-A dataset used in paper is proprietary to Google, the provided datasets are very similar but not identical to the ones that the authors used. Furthermore, the details of the infrastructure and computational setup were not mentioned in the paper.

In our project, we decided to narrow to scope of this project by focusing on reproducing the results of the paper for i.i.d MNIST-A dataset, using the source code provided by the author. We ran our experiments on a Google Compute Engine, using one Nvidia Tesla P100 accelerator. The hyperparameter set that gives the best performance is missing. Our testing results has a significant resemblance with the results in the paper. The average percentage difference between our and their results is at 1.18%. This is a small difference, therefore confirming the methodology and results presented in this paper.

The code implements the architecture and the objective functions described in the paper. Also, our quantitative results match nicely with those in the paper. However, we noticed a misalignment in the qualitative results, namely between the reproduced image and the images in the paper. Our reproduced images are not found in the paper, creating some confusion. Therefore, we were not able to obtain a full understanding the process that transforms the raw images we have to the findings presented in the paper. We believe that with more clarification, we can do more qualitative analysis on the output images.

A key difference between our experiments and those of the authors is the number of steps that we used. We ran TELBO and JMVAE for 250,000 steps, and BiVCCA for 50,000 steps, while the authors trained their models for 500,000 steps. The similarity in coverage and correctness implies that 250,000 steps is enough to train a well-performed model. The additional steps in author's experiments did not lead to overfitting, but it did not make a significant improvement either. Moreover, the author noted in their paper that the models do not tend to overfit in their experiment. Indeed, our reproduced results verify this claim.

Overall, the process of reproducing this paper's result is straightforward. Feasibility of the experiments large depends on the time and hardware setup. Given more time and resources, we believe that we could reproduce additional experiments such as examining different hyperparameter settings and running the models on CelebA dataset.