The Journal of Open Source Software

MultiVae: A Python package for Multimodal Variational Autoencoders on Partial Datasets

Agathe Senellart[1,2*¶] **and Stéphanie Allassonnière**[2*]

1 Université de Paris-Cité **2** Inria **3** Inserm ¶ Corresponding author * These authors contributed equally.

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

In partnership with

AMERICAN ASTRONOMICAL SOCIETY

This article and software are linked with research article DOI 10.3847/xxxxx <- update this with the DOI from AAS once you know it., published in the Astrophysical Journal <- The name of the AAS journal..

Summary

In recent years, there has been a major boom in the development of multimodal machine learning models. Among open topics, representation (fusion) and genera- tion of multimodal data are very active fields of research. Recently, Multimodal Variational Autoencoders (VAEs) have been attracting growing interest for both tasks, thanks to their versatility, scalability, and interpretability as probabilistic latent variable models. They are also particularly interesting models in the partially observed setting, as most of them can learn even with missing data. This last point makes them particularly interesting for research fields such as the medical field, where missing data are commonplace.

In this article, we present MultiVae, an open-source Python library for bringing together unified implementations of multimodal VAEs. It has been designed for easy, customizable use of these models on fully or partially observed data. This library also facilitates the development and benchmarking of new algorithms by integrating several popular datasets, variety of evaluation metrics and tools for monitoring and sharing models.

Multimodal Variational Autoencoders

In Multimodal Machine Learning, two goals are generally targeted: (1) Learn a shared representation from multiple modalities; (2) Learn to generate one missing modality given the ones that are available.

Multimodal Variational Autoencoders aim at solving both issues at the same time. These models learn a latent representation z of all modalities in a lower dimensional common space and learn to *decode* z to generate any modality (Suzuki & Matsuo, 2022).
Let $X = (x_1, x_2, ...x_M)$ contain M modalities. In the VAE setting, we suppose that the generative process behind the observed data is the following:

$$z \sim p(z) \qquad \forall 1 \leq i \leq M, x_i|z \sim p_\theta(x_i|z) \tag{1}$$

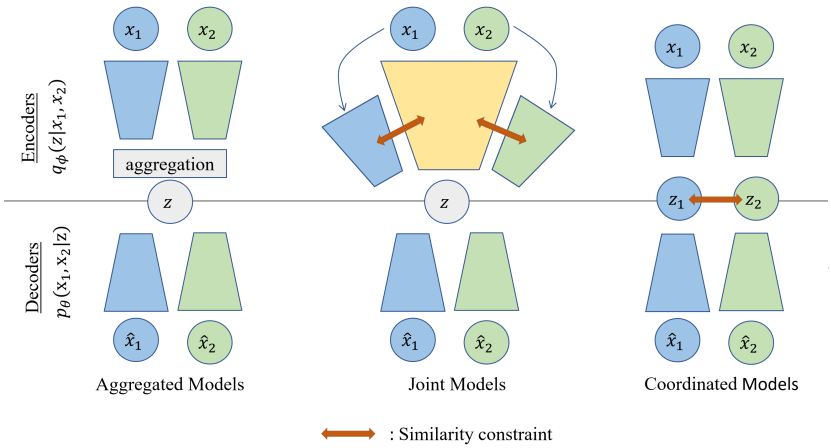
where $p(z)$ is a prior distribution that is often fixed, and $p_\theta(x_i|z)$ are called *decoders* and are parameterized by neural network. Typically, $p_\theta(x_i|z) = \mathcal{N}(x_i; \mu_\theta(z), \sigma_\theta(z))$ where $\mu_\theta, \sigma_\theta$ are neural networks. We aim to learn these *decoders* that translate z into the high dimensional data x_i. At the same time, we aim to learn an *encoder* $q_\phi(z|X)$ that map the multimodal observation to the latent space. $q_\phi(z|X)$ is also parameterized by a neural network. Derived from variational inference (Kingma & Welling, 2014), the VAE objective writes:

$$\mathcal{L}(X) = \mathbb{E}_{q_\phi(z|X)}\left(\sum_i \ln(p_\theta(x_i|z))\right) - KL(q_\phi(z|X)|p(z))$$

A simple interpretation of this objective is to see that the first term is a reconstruction loss and the second term is a regularization term that avoids overfitting. A typical training of a

multimodal VAE consists in encoding the data with the encoder, reconstructing each modality with the decoders and take a gradient step to optimize the loss $\mathcal{L}(X)$.

Most multimodal VAEs differ in how they construct the encoder $q_\phi(z|X)$. In Figure **??**, we summarize several approaches: *Aggregated models* (Shi et al., 2019; Sutter et al., 2021; Wu & Goodman, 2018) use a mean or a product operation to aggregate the information coming from all modalities, where *Joint models* (Senellart et al., 2023; Suzuki et al., 2016; Vedantam et al., 2018) uses a neural network taking all modalities as input. Finally *coordinated models* (Tian & Engel, 2019; Wang et al., 2017) uses different latent spaces but add a constraint term in the loss to force them to be similar.



Recent extensions of multimodal VAEs include additional terms to the loss, or use multiple (Palumbo et al., 2023) or hierarchical (Dorent et al., 2023; Vasco et al., 2022) latent spaces to more comprehensively describe the multimodal data. Aggregated models have a natural way of learning on incomplete datasets: for an incomplete sample X, we use only the available modalities to encode the data and compute the loss $\mathcal{L}(X)$. However, except in MultiVae, there doesn't exist an implementation of these models that can be used on incomplete datasets in a straightforward manner.

Data Augmentation

Another application of these models is Data Augmentation (DA): from sampling latent codes z and decoding them, *fully synthetic multimodal* samples can be generated to augment a dataset. Data augmentation has been proven useful in many data-intensive deep learning applications (Chadebec et al., 2023). In a dedicated module `multivae.samplers`, we propose different ways of sampling latent codes z to further explore the generative abilities of these models.

Statement of need

Although multimodal VAEs have interesting applications in different fields, the lack of easy-to-use and verified implementations might hinder applicative research. With MultiVae, we offer unified implementations, designed to be easy to use by non-specialists and even on incomplete data. To this end, we offer online documentation and tutorials. In order to propose reliable implementations of each method, we tried to reproduce, whenever possible, a key result from the original paper. Some works similar to ours have grouped together model implementations: the Multimodal VAE Comparison Toolkit includes 4 models and the Pixyz library groups 2 multimodal models (https://github.com/masa-su/pixyz/blob/main/examples/jmvae.ipynb). The work closest to ours and released while we were developping our library is `multi-view-ae` (**?**), which contains a dozen of models. We compare in a summarizing table below, the different features of each work. Our library complements what already exists: our API is quite different compared to previous work, the models implemented are not all the same, and for those

we have in common, our implementation offers additional parameterization options. Indeed, for each model, we've made sure to offer great flexibility on parameters and to include all implementation details present in the original codes that boost results. What's more, our library offers many additional features: compatibility with incomplete data, which we consider essential for real-life applications, and a range of tools dedicated to the research and development of new algorithms: benchmark datasets, metrics modules and samplers, for testing and analyzing models. Our library also supports distributed training and straightforward model sharing via HuggingFace Hub. Therefore our work complements existing options and addresses different needs.

List of models and features

In the Table below, we list available models and features, and compare to previous work. This symbol (✓*) indicates that our implementation include additional options.

Models/ Features	MultiVae	multi-view-ae
JMVAE	✓*	✓
MVAE	✓*	✓
MMVAE	✓*	✓
MoPoE	✓*	✓
DMVAE	✓	✓
MVTCAE	✓	✓
MMVAE+	✓*	✓
CMVAE	✓	
Nexus	✓	
CVAE	✓	
MHVAE	✓	
TELBO	✓	
JNF	✓	
MCVAE		✓
mAAE		✓
DVCCA		✓
mWAE		✓
mmJSD		✓
gPoE		✓
Support of Incomplete datasets	✓	
GMM Sampler	✓	
MAF Sampler, IAF Sampler	✓	
Metric: Likelihood, Coherences, FIDs, Reconstruction, Clustering	✓	
Inline Datasets	✓	
Model sharing via Hugging Face	✓	

Description of the software

Our implementation is based on PyTorch and is inspired by the architecture of (Chadebec et al., 2022). The implementations of the models are collected in the module `multivae.models`.

Each model class is accompanied by a configuration dataclass gathering the collection of any relevant hyperparameter which enables them to be saved and loaded straightforwardly. The models are implemented in a unified way, so that they can be easily integrated within the `multivae.trainers`. Trainers are also accompanied by a training configuration dataclass used to specify any training-related hyperparameters (number of epochs, optimizers, schedulers, etc..). Models that have a multistage training [50, 40] benefit from their dedicated trainer that makes them as straightforward to use as other models. Partially observed datasets can be conveniently handled using the `IncompleteDataset` class that contains masks informing on missing or corrupted modalities in each sample. For Data Augmentation purposes the module `multivae.samplers` regroups different ways generating fully synthetic data. Finally, the MultiVae library also integrates an evaluation pipeline for all models with common metrics such as likelihoods, coherences, FID scores [18] and visualizations.

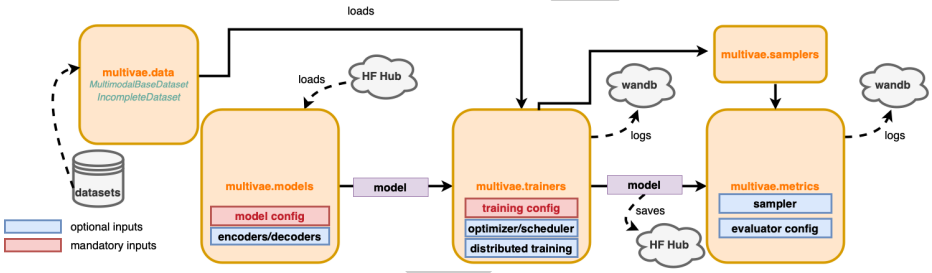


Figure 1: Code structure

Documentation

The main features are illustrated through tutorials made available either as notebooks or scripts allowing users to get started easily. An online documentation is also made available at https://multivae.readthedocs.io/en/latest.

Acknowledgements

We are grateful to the authors of all the original implementations of the models included in MultiVae.

References

Chadebec, C., Thibeau-Sutre, E., Burgos, N., & Allassonnière, S. (2023). Data augmentation in high dimensional low sample size setting using a geometry-based variational autoencoder. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(3), 2879–2896. https://doi.org/10.1109/TPAMI.2022.3185773

Chadebec, C., Vincent, L., & Allassonniere, S. (2022). Pythae: Unifying Generative Autoencoders in Python - A Benchmarking Use Case. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems* (Vol. 35, pp. 21575–21589). Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2022/file/872f0e04ef95be7970d9a9d74b198fdf-Paper-Datasets_and_Benchmarks.pdf

Dorent, R., Haouchine, N., Kogl, F., Joutard, S., Juvekar, P., Torio, E., Golby, A. J., Ourselin, S., Frisken, S., Vercauteren, T., Kapur, T., & Wells, W. M. (2023). Unified brain MR-ultrasound synthesis using multi-modal hierarchical representations. In *Medical image*

computing and computer assisted intervention – MICCAI 2023 (pp. 448–458). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-43999-5_43

Kingma, D. P., & Welling, M. (2014). *Auto-Encoding Variational Bayes*. arXiv. http://arxiv.org/abs/1312.6114

Palumbo, E., Daunhawer, I., & Vogt, J. E. (2023). *MMVAE+: ENHANCING THE GENERATIVE QUALITY OF MULTIMODAL VAES WITHOUT COMPROMISES*.

Senellart, A., Chadebec, C., & Allassonnière, S. (2023). Improving multimodal joint variational autoencoders through normalizing flows and correlation analysis. *arXiv Preprint arXiv:2305.11832*.

Shi, Y., Siddharth, N., Paige, B., & Torr, P. H. S. (2019). Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models. *arXiv:1911.03393 [Cs, Stat]*. http://arxiv.org/abs/1911.03393

Sutter, T. M., Daunhawer, I., & Vogt, J. E. (2021). Generalized Multimodal ELBO. *ICLR*.

Suzuki, M., & Matsuo, Y. (2022). A survey of multimodal deep generative models. *Advanced Robotics*, *36*(5-6), 261–278. https://doi.org/10.1080/01691864.2022.2035253

Suzuki, M., Nakayama, K., & Matsuo, Y. (2016). Joint Multimodal Learning with Deep Generative Models. *arXiv:1611.01891 [Cs, Stat]*. http://arxiv.org/abs/1611.01891

Tian, Y., & Engel, J. (2019). Latent Translation: Crossing Modalities by Bridging Generative Models. *ArXiv*.

Vasco, M., Yin, H., Melo, F. S., & Paiva, A. (2022). Leveraging hierarchy in multimodal generative models for effective cross-modality inference. *Neural Networks*, *146*, 238–255.

Vedantam, R., Fischer, I., Huang, J., & Murphy, K. (2018). Generative Models of Visually Grounded Imagination. *arXiv:1705.10762 [Cs, Stat]*. http://arxiv.org/abs/1705.10762

Wang, W., Yan, X., Lee, H., & Livescu, K. (2017). *Deep Variational Canonical Correlation Analysis*. arXiv. https://doi.org/10.48550/arXiv.1610.03454

Wu, M., & Goodman, N. (2018). Multimodal Generative Models for Scalable Weakly-Supervised Learning. *Advances in Neural Information Processing Systems*, *31*. https://proceedings.neurips.cc/paper/2018/hash/1102a326d5f7c9e04fc3c89d0ede88c9-Abstract.html