JDSSV

Journal of Data Science, Statistics, and Visualisation

November 2025, Volume V, Issue IX. *doi: 10.52933/jdssv.v5i9.160*

Bridging the Inference Gap in Multimodal Variational Autoencoders

Agathe Senellart
Université Paris Cité, Inria, Inserm, HeKA, F-75015 Paris

Stéphanie Allassonnière
Université Paris Cité, Inria, Inserm, HeKA, F-75015 Paris

Abstract

From medical diagnosis to autonomous vehicles, many critical applications rely on the integration of multiple heterogeneous data modalities. Multimodal Variational Autoencoders offer versatile and scalable methods for generating unobserved modalities from observed ones. Recent models using mixture-of-experts aggregation suffer from theoretical limitations that reduce generation quality on complex datasets. In this article, we propose a novel latent-variable model which is able to learn both joint and conditional distributions without introducing mixture aggregation. Our model follows a multistage training process; after learning the joint distribution with variational inference, we learn the conditional distributions using normalizing flows and a new, theoretically grounded objective function. Importantly, we also propose extracting the semantic content shared between modalities in a pre-training stage and incorporating these representations into the inference distributions to enhance generative coherence. Our method achieves state-of-the-art results on several benchmark datasets.

Keywords: Contrastive learning, multimodality, normalizing flows, variational autoencoders.

1. Introduction

In many real-world applications, including medicine, autonomous driving, or robotics, information is conveyed through heterogeneous modalities—such as images, signals, or text. In the medical field, for instance, a patient's condition is described through sonograms, MRI, blood tests results, and clinical notes. Leveraging these diverse views jointly leads to richer representations and deeper insights into the underlying phenomena. Two central challenges in multimodal machine learning are: (i) *learning relevant joint representations* and (ii) *generating coherent and diverse samples*, conditionally or jointly, across modalities.

Multimodal variational autoencoders are latent generative models that can address both challenges simultaneously. In recent years, several approaches have been proposed to extend the variational autoencoder (Kingma and Welling 2014) (VAE) to efficiently model multimodal data. Some of them suggest training *coordinated* VAEs with a similarity constraint between latent spaces (Higgins et al. 2018; Yin et al. 2017). In other works, a single latent space is used to jointly represent all modalities (Suzuki et al. 2016; Wu and Goodman 2018; Shi et al. 2019). Among these models, one popular approach is to aggregate modalities using a simple function such as product-of-experts (Wu and Goodman 2018), or mixture-of-experts (Shi et al. 2019). Aggregation approaches are computationally efficient and scalable, but come with critical limitations (Daunhawer et al. 2022; Sutter et al. 2021). For instance, the product-of-experts approach can lead to imbalances between modalities, allowing one to dominate or override information from others (Shi et al. 2019). On the other hand, recent theoretical work (Daunhawer et al. 2022) shows that mixture-based models introduce an irreducible inference gap, which degrades sample quality. More specifically, such models tend to generate samples close to the expectation, failing to capture the diversity of the true distribution.

To address these issues, we propose an original **multi-stage modeling approach** that avoids aggregation entirely. First, we learn the joint distribution. Then, we use normalizing flows (NF) (Rezende and Mohamed 2016) and a principled objective function to learn the conditional distributions. This decoupled strategy bypasses the theoretical limitations of mixture-based models.

Crucially, we further enhance conditional generation by incorporating **shared representations**, extracted in a pre-training step via contrastive learning (Poklukar et al. 2022) or Deep Canonical Correlation Analysis (DCCA)(Andrew et al. 2013). These representations capture **the semantic content shared across modalities** and are used to guide the approximation of the posteriors, ensuring that **cross-modal generations remain semantically coherent**, thereby improving generative quality.

Previous multimodal variational autoencoders (VAEs) have primarily sought to disentangle shared semantic content from modality-specific information through the introduction of additional latent variables and modifications to the VAE loss function (Lee and Pavlovic 2021; Sutter et al. 2021; Palumbo et al. 2023, 2024). In contrast, our approach uses techniques explicitly designed for extracting shared information in a *pre-training* stage.

We demonstrate the effectiveness of our method on several benchmark datasets. Our model outperforms existing approaches, particularly on challenging settings such as the Translated PolyMNIST dataset.

2. Background

Let's assume that we observe multimodal samples $X = (x_1, x_2, \ldots, x_M)$ with M modalities from an unknown distribution $p(X)$. We aim to approximate this joint distribution as well as the conditional distributions with parametric ones $p_\theta(X)$, $p_\theta(x_j|x_i)$ for any $1 \leq i \neq j \leq M$. $p_\theta(x_j|x_i)$ refers to the distribution of modality x_j *given* x_i.

In the VAE framework, one assumes that there exists a shared latent representation z, from which all modalities can be generated with parametric distributions $(p_\theta(x_j|z))_j$ called *decoders*. For instance, for an image modality x_1, $p_\theta(x_1|z)$ can be a Gaussian distribution $\mathcal{N}(\mu_\theta(z), \Sigma_\theta(z))$ whose mean and variance are given by a neural network. Following previous work (Suzuki et al. 2016), each modality is supposed to be conditionally independent of the others given z, such that the joint model writes:

$$p_\theta(X, z) = p_\theta(X|z)p_\theta(z) = p_\theta(z)\prod_{j=1}^{M} p_\theta(x_j|z), \tag{1}$$

where $p_\theta(z)$ is a *prior* distribution over the latent variable and θ refers to all parameters used to model the prior and the decoders. In that framework, the two goals mentioned above (model the joint and conditional distributions) translate as follows: firstly, we want to learn the best possible θ to model the observations. Secondly, we want to approximate the *inference distributions* $p_\theta(z|(x_j)_{j\in S})$ to infer the latent variable from any given subset of modalities $S \in \mathcal{P}(M)$, where $\mathcal{P}(M) = \{S|S \subset [|1, M|] \text{ and } S \neq \emptyset\}$. If we can infer z from observed modalities, we can then generate unobserved modalities with the decoders $(p_\theta(x_j|z))_{1\leq j\leq M}$. In the rest of the article, we note $x_S \coloneqq (x_j)_{j\in S}$ to simplify notations.

2.1. Estimating the generative model's parameters

Given N multimodal observations $(X^{(i)})_{1\leq i\leq N}$, a natural objective to estimate θ is to optimize the log-likelihood of the data (Kingma and Welling 2014):

$$\theta^* \in \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \log p_\theta(X^{(i)}) = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{N} \left(\log \int_z p_\theta(X^{(i)}, z)dz\right).$$

Since this objective is intractable, one can resort to Variational Inference (Jordan et al. 1998; Kingma and Welling 2014) by introducing an auxiliary parametric distribution $q_\phi(z|X)$ allowing us to derive an unbiased estimate of the likelihood of the data:

$$\widehat{p}_\theta(X, z) = \frac{p_\theta(X, z)}{q_\phi(z|X)} \quad \text{such that} \quad p_\theta(X) = \mathbb{E}_{q_\phi(z|X)}\left[\widehat{p}_\theta\right]. \tag{2}$$

Then, using Jensen's inequality allows us to derive a lower bound on $\log p_\theta(X)$, referred to as the Evidence Lower Bound (ELBO).

$$\begin{aligned}\log p_\theta(X) &= \log \mathbb{E}_{q_\phi(z|X)}\left[\widehat{p}_\theta\right] \\ &\geq \mathbb{E}_{q_\phi(z|X)}\left[\log p_\theta(X|z)\right] - KL(q_\phi(z|X)||p_\theta(z)) = \mathcal{L}(X; \theta, \phi),\end{aligned} \tag{3}$$

where $KL(q_\phi(z|X)||p_\theta(z))$ refers to the Kullback-Leibler (KL) divergence between the joint posterior and the prior distribution.

This bound is tractable and can be optimized through stochastic gradient descent (Kingma and Welling 2014). Noteworthy, the first term can be seen as a reconstruction error and the second as a regularization term encouraging latent embeddings to follow the prior distribution (Ghosh et al. 2020). The distribution $q_\phi(z|X)$ is generally called the *encoder* and one may prove that:

$$\mathcal{L}(X;\theta,\phi) = \log p_\theta(X) - KL(q_\phi(z|X)||p_\theta(z|X)). \tag{4}$$

This implies that maximizing $\mathcal{L}(X;\theta,\phi)$ with respect to ϕ leads to minimizing the Kullback-Leibler divergence between the true posterior $p_\theta(z|X)$ and its variational approximation $q_\phi(z|X)$ (Kingma and Welling 2014). Some models also rely on variations of Equation (3) to learn θ. (Sutter et al. 2021; Palumbo et al. 2023) add a β factor to weigh the KL term in Equation (3). That hyperparameter can be tuned to promote disentanglement in the latent space (Higgins et al. 2017); by increasing the KL term, it increases pressure on the latent variables to be independent, so that a single unit might encode a single generative factor. Other models use a k-sampled importance weighted estimate of the log-likelihood (IWAE bound) (Shi et al. 2019; Palumbo et al. 2023) or replace the KL with a Jensen-Shannon divergence (Sutter et al. 2020).

2.2. Choice of the approximate inference distribution

A simple choice is to model the approximate posterior $q_\phi(z|X)$ as a Gaussian distribution $\mathcal{N}(\mu_\phi(X), \Sigma_\phi(X))$, where a dedicated joint encoder network takes all modalities as input and outputs the parameters $\mu_\phi(X), \Sigma_\phi(X)$. By maximizing $\mathcal{L}$, we obtain an estimation of θ and an approximation of the joint posterior $p_\theta(z|X)$ with $q_\phi(z|X)$. However, we do not have access to the remaining subset posteriors $(p_\theta(z|x_S))_{S\in\mathcal{P}(M)}$ which are *intractable*. To estimate these posterior distributions, two approaches have been proposed, which we detail in the following paragraphs.

2.3. Surrogate distributions and learning objectives

Firstly, a few models such as JMVAE (Suzuki et al. 2016), or TELBO (Vedantam et al. 2018) introduce surrogate parametric distributions $(q_{\phi_S}(z|x_S))_{S\in\mathcal{P}(M)}$ and train them with an additional loss function to approximate the desired posterior distributions. However, those models use a large number of parameters since the joint posterior $q_\phi(z|X)$ and each approximate posterior $(q_{\phi_S}(z|x_S))_{S\in\mathcal{P}(M)}$ use a dedicated network encoder. The number of parameters then scales with the number of subsets $|\mathcal{P}(M)| = 2^M$. For this reason, this type of model was less explored, especially since they produced less coherent results than *aggregated models* proposed afterwards.

2.4. Aggregated models

Aggregated models compute the joint posterior $q_\phi(z|X)$ as an aggregation of unimodal encoders $q_{\phi_j}(z|x_j)$ for $1 \leq j \leq M$. Wu and Goodman (2018) uses a product-of-experts (PoE) operation $q_\phi(z|X) \propto p_\theta(z)\prod_{j=1}^{M} q_{\phi_j}(z|x_j)$ while Shi et al. (2019) uses a mixture-of-experts (MoE). Many variants were then introduced such as mixture-of-product-of-experts (Sutter et al. 2021) or generalized product-of-experts (Lawry Aguila et al. 2023). During inference, each subset posterior $p_\theta(z|x_S)$ is approximated with the

subset aggregation of the unimodal encoders $(q_{\phi_j}(z|x_j))_{j\in S}$. For instance, with a PoE approach: $p_\theta(z|x_S)$ is approximated by $q_\phi(z|x_S) \propto \prod_{j\in S} q_{\phi_j}(z|x_j)$. This approach has the advantage of scaling linearly with the number of modalities. However, although the ELBO objective implies that $q_\phi(z|X)$ will approximate the true joint posterior $p_\theta(z|X)$ in the KL sense (as seen in 2.1), **there is no theoretical garanty that the *subset* aggregations $q_\phi(z|x_S)$ will approximate the subset posteriors $p_\theta(z|x_S)$.**

Furthermore, Daunhawer et al. (2022) show that all *mixture-based* models suffer from a fundamental limitation that caps their generative quality. More precisely, for these models, there is a generative discrepancy $\Delta(X)$ between the log-likelihood of the data and the ELBO:

$$\mathbb{E}_{\hat{p}(X)}(\log(p_\theta(X))) \geq \mathbb{E}_{\hat{p}(X)}(\mathcal{L}(X;\theta,\phi)) + \Delta(X), \tag{5}$$

where $\hat{p}(X)$ is the observed empirical distribution. $\Delta(X)$ is strictly positive and only depends on the law of X and the mixture components (Daunhawer et al. 2022). Using $\log(p_\theta(X)) - \mathcal{L}(X;\theta,\phi) = KL\left(q_\phi(z|X)||p_\theta(z|X)\right)$, one can rewrite (5) as:

$$\mathbb{E}_{\hat{p}(X)}\left(KL(q_\phi(z|X)||p_\theta(z|X))\right) \geq \Delta(X). \tag{6}$$

This lower bound implies that the approximate joint posterior $q_\phi(z|X)$ can only approach the true joint posterior $p_\theta(z|X)$ up to $\Delta(X) > 0$. Daunhawer et al. (2022) detail in extensive experiments how these generative discrepancy results in a diminished quality of generated samples.

Aggregated models that are only based on a product-of-experts such as MVAE (Wu and Goodman 2018) or MVTCAE (Hwang et al. 2021), avoid this issue but generally show lower generative coherence than mixture-based models (Sutter et al. 2021; Palumbo et al. 2023), especially when sampling from a single modality.

2.5. Other approaches and recent developments

Coordinated VAEs. Using the terminology of Suzuki and Matsuo (2022), coordinated VAEs—such as those proposed by Higgins et al. (2018) and Yin et al. (2017)—are multimodal models that impose a similarity constraint between the latent representations of each modality. They are not aggregation-based; instead, they aim to align the individual posteriors $(q_{\phi_j}(z_j|x_j))_{1\leq j\leq M}$. Since they do not approximate the joint posterior $p_\theta(z|X)$ or any subset posteriors $p_\theta(z|x_S)$ for S containing more than one modality, inference conditioned on multiple modalities is not supported by this approach.

Additional latent spaces. To improve the diversity of the generations, methods have been proposed with more complex generative models including multiple, hierarchical (Vasco et al. 2022) or independent (Sutter et al. 2021; Lee and Pavlovic 2021; Daunhawer et al. 2021) latent spaces. An additional goal of these models is to separate into different latent spaces the information shared across modalities from modality-specific factors. Palumbo et al. (2023) show that these models are sensitive to the *shortcut* issue, referring to shared information leaking into the modality specific latent spaces. They propose the MMVAE+ model with an amended ELBO loss and modalities' specific priors to limit that phenomenon. However, while (Lee and Pavlovic 2021; Daunhawer et al. 2021) were based on PoE, the MMVAE+ suffers from the limitations that come with mixture aggregation.

Enhanced generation with diffusion models. Palumbo et al. (2024) propose a hierarchical multimodal VAE with diffusion decoders to improve the generation quality. Interestingly enough, the proposed diffusion decoder improvement can be applied to many VAE based methods including the one presented in this article. However, their CMVAE model is still based on a mixture aggregation which limits its expressivity. In contrast, Bounoua et al. (2024) propose to learn separate representations for each modality and use a multimodal diffusion model to capture their interactions. Their method bypasses the limitations of mixture-based models but does not learn a joint fused representation across modalities.

3. Proposed Method

As mentioned in 2.3, the JMVAE and TELBO models use a joint encoder for modelling $q_\phi(z|X)$ and therefore avoid the inference gap of mixture-based models. We propose a new method in the same line of work but solving the scalability issue and improving generative quality. Our method **decouples the training of the joint generative model and the approximation of the posteriors** in two separate steps:

- Step 1: Train a variational autoencoder to learn the generative model $p_\theta(X)$ as well as an approximation of the joint posterior $q_\phi(z|X)$.
- Step 2: For conditional generation, approximate the unimodal posteriors $p_\theta(z|x_j)$ with normalizing flows (Rezende and Mohamed 2015) based distributions $q_{\phi_j}(z|x_j)$ for $1 \leq j \leq M$ using a novel learning objective.

For the subset posteriors, we show that, for any $S \in \mathcal{P}(M)$, we can approximate $p_\theta(z|x_S)$ with a product-of-experts $\prod_{j\in S} q_{\phi_j}(z|x_j)/p_\theta(z)^{|S|-1}$. This way, no additional network needs to be trained and our framework scales linearly with the number of modalities. Note that this PoE is only used during *inference* which means that it doesn't suffer from the same limitations as PoE aggregated models. In the following subsections, we detail each step of our method, and then we introduce an **improvement for the second step that leverages information shared across modalities**.

3.1. Step 1: Training the joint generative model

For learning the generative parameter θ, we optimize the ELBO (see Equation (3)) with a β factor weighting the regularization term. We model the joint encoder $q_\phi(z|X)$ as a Gaussian distribution $\mathcal{N}(\mu_\phi(X), \Sigma_\phi(X))$, with $\mu_\phi(X)$ and $\Sigma_\phi(X)$ given by a neural network taking all modalities as inputs. This step is exactly similar to training a unimodal VAE, and every improvement that was proposed for the unimodal case could be seamlessly adapted here.

3.2. Step 2: Learning the posterior distributions

Once the generative model is learned, we freeze the generative model $p_\theta(X|z)$ and the joint encoder $q_\phi(z|X)$. For $1 \leq j \leq M$ we introduce a surrogate distribution $q_{\phi_j}(z|x_j)$ to approximate the unimodal posterior $p_\theta(z|x_j)$ that is intractable. We propose to maximize the following objective to fit this parametric distribution:

$$\mathcal{L}_j(X;\phi_j) = \mathbb{E}_{q_\phi(z|X)}\Big(\log q_{\phi_j}(z|x_j)\Big). \tag{7}$$

The expectation inside the sum can be estimated by sampling $z \sim q_\phi(z|X)$. Equation (7) shows that during training, the unimodal encoders are *informed* by the joint encoder: a latent variable z is sampled from $q_\phi(z|X)$ and then $\log q_{\phi_j}(z|x_j)$ is maximized. Intuitively, maximizing Equation (7) encourages $q_{\phi_j}(z|x_j)$ to cover all the relevant modes or support of the trained posterior $q_\phi(z|X)$. To further see the relevance of the proposed objective function, we provide the following result:

Proposition 1 *Let $1 \leq j \leq M$ and X_{C_j} the set of all modalities but x_j. $\hat{p}$ refers to the empirical distribution of the data. If $q_{\phi_j}(z|x_j)$ is bounded by a constant C, maximizing $\mathbb{E}_{\hat{p}(X)}(\mathcal{L}_j(X,\phi_j))$ with regard to ϕ_j is equivalent to minimizing:*

$$\mathbb{E}_{\hat{p}(x_j)}\Big(KL(q_\phi^{(avg)}(z|x_j)||q_{\phi_j}(z|x_j))\Big),$$

where $q_\phi^{(avg)}(z|x_j) = \int_{X_{C_j}} q_\phi(z|X)\hat{p}(X_{C_j}|x_j)dX_{C_j}$.

To understand why $q_\phi^{(avg)}(z|x_j)$ serves as a meaningful target distribution, note that if $q_\phi(z|X) = p_\theta(z|X)$ and $p_\theta(X) = \hat{p}(X)$ then $q_\phi^{(avg)}(z|x_j)$ is exactly equal to the true distribution $p_\theta(z|x_j)$. Consequently, if the joint VAE from step 1 is well trained, the unimodal encoders will approximate the true posteriors closely. The proof of this proposition is provided in Appendix C and mostly relies on the Fubini theorem and the fact that $q_\phi(z|X)$ is already trained and frozen in our method. By decoupling the approximation of the generative model and the posteriors, we were able to propose a principled objective for the unimodal encoders.

We use **normalizing flows** to define the parametric distributions $(q_{\phi_j}(z|x_j))_j$ as they allow flexible modeling of complex distributions (Rezende and Mohamed 2015). A flow is an invertible smooth transformation f that can be applied to an initial distribution to create a new one, such that if Z is a random vector with density $q(z)$, then $Z' = f(Z)$ has a density given by:

$$q'(z') = q(z)\left|\det \frac{\partial f^{-1}}{\partial z'}\right| = q(z)\left|\det \frac{\partial f}{\partial z}\right|^{-1}. \tag{8}$$

Combining K transformations $z_K = f_K \circ f_{K-1} \circ \cdots \circ f_1(z_0)$ allows us to gain in complexity of the final distribution. In our case, for each modality $1 \leq j \leq M$, we model the approximate posterior $q_{\phi_j}(z|x_j)$ with the following log-density:

$$\log q_{\phi_j}(z|x_j) = \log q_{\phi_j}^{(0)}(z_0|x_j) - \sum_{k=1}^{K} \log\left|\det \frac{\partial f_k^{(j)}}{\partial z_{k-1}}\right|, \tag{9}$$

where $q_{\phi_j}^{(0)}(z_0|x_j)$ is a simple parametrized Gaussian distribution, the parameters of which are given by neural networks, and $(f_k^{(j)})_{1\leq k\leq K}$ are masked autoregressive flows (Papamakarios et al. 2017). In Section (4.1), we illustrate that this expression allow us to approximate much more precisely the true unimodal posteriors than using a simple multivariate Gaussian. Because of the joint training of normalizing flows during this step, we refer to this model as JNF.

3.3. Sampling from the subset posteriors

Recall that one of our goals is to be able to infer the latent variable z from *any subset of modalities* $S \in \mathcal{P}(M)$. Until now, we have estimated the joint posterior with $q_\phi(z|X)$ and the unimodal posteriors with $q_{\phi_j}(z|x_j)$ for any $j \in [|1, M|]$. Using the same derivation as (Wu and Goodman 2018), we prove that we can approximate any subset posterior using the trained unimodal encoders.

Let $S \in \mathcal{P}(M)$ and $x_S = (x_j)_{j \in S}$:

$$p_\theta(z|x_S) = \frac{p_\theta(x_S|z)p_\theta(z)}{p_\theta(x_S)} = \frac{p_\theta(z)\prod_{j\in S} p_\theta(x_j|z)}{p_\theta(x_S)} = \frac{p_\theta(z)\prod_{j\in S}\{p_\theta(x_j,z)/p_\theta(z)\}}{p_\theta(x_S)} \tag{10}$$

$$= \frac{\prod_{j\in S} p_\theta(z|x_j)p_\theta(x_j)}{p_\theta(z)^{|S|-1}p_\theta(x_S)} = \frac{1}{Z}\frac{\prod_{j\in S} p_\theta(z|x_j)}{p_\theta(z)^{|S|-1}} \approx \frac{1}{Z}\frac{\prod_{j\in S} q_{\phi_j}(z|x_j)}{p_\theta(z)^{|S|-1}} \tag{11}$$

where $1/Z = \{\prod_{j\in S} p_\theta(x_j)\}/p_\theta(x_S)$ is a normalizing constant. We use Equation (1) in the second equality. To sample from this distribution at inference time, we use Hamiltonian Monte Carlo (HMC) sampling (Betancourt 2018) that enables sampling from any distribution with a differentiable density function known up to a multiplicative constant. We recall the algorithm for HMC in Appendix F.

3.4. An improvement of our method leveraging shared information

So far, we have not made any assumption regarding the interactions between modalities $(x_1, x_2, \ldots, x_M)$. However, in many multimodal datasets, there is typically some amount of *shared* semantic information between modalities. For instance, in the MNIST-SVHN dataset (Lecun et al. 1998; Netzer et al. 2011), both images share the same digit as their core semantic content, even though other aspects—like background and style—are **modality-specific**, meaning they do not influence the other modality. **In such settings, to generate an unobserved modality, only the shared semantic content is necessary; the modality-specific details of the observed modalities are not required**. If we can effectively disentangle shared semantics from modality-specific information, we could reduce some of the noise and data variance when learning conditional distributions such as $p_\theta(x_1|x_2)$ by only considering the relevant information in x_2 to predict x_1. Extracting the shared information can simplify the task of modeling $p_\theta(x_1|x_2)$ in order to increase generative coherence.

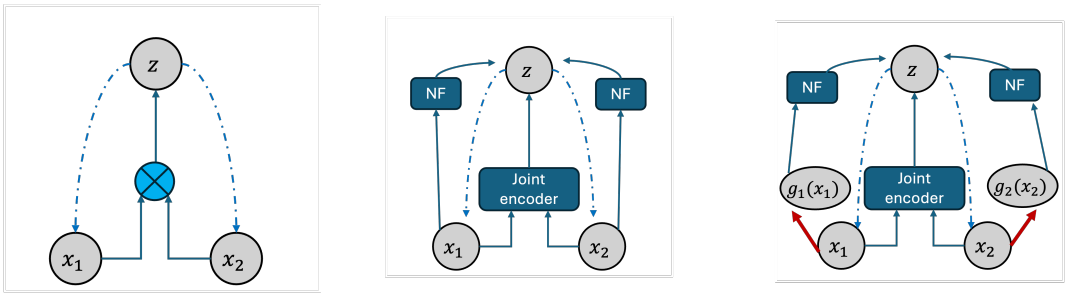
Formally, let us assume that for any $1 \leq j \leq M$ we have a projector g_j such that:

$$\forall i \in [|1, M|], p_\theta(x_i|x_j) = p_\theta(x_i|g_j(x_j)). \tag{12}$$

Morally speaking, g_j extracts the information shared across modalities while tuning out the modality specific information. Then, we can write:

$$p_\theta(x_i|g_j(x_j)) = \int_z p_\theta(x_i|z)p_\theta(z|g_j(x_j))dz. \tag{13}$$

That is, to generate modality x_i from modality x_j, we can learn to approximate $p_\theta(z|g_j(x_j))$ which might be simpler than approximating $p_\theta(z|x_j)$ if we use relevant functions $(g_j)_{1\leq j\leq M}$.



(a) Aggregated Models (b) JNF model (c) JNF-Shared model

Figure 1: Graphical models in the case $M = 2$. Dashed lines represent decoders, solid lines represent encoders, and red arrows represent the projectors that were jointly pretrained to extract shared information. "NF" refers to normalizing flows.

We propose an improvement of our method, in which we use functions $(g_j)_{1\leq j\leq M}$ *pretrained* to extract shared information across modalities and model the distributions $q_{\phi_j}(z|g_j(x_j))$ instead of $q_{\phi_j}(z|x_j)$. In that case, we model $q_{\phi_j}(z|g_j(x_j))$ with normalizing flows and use the adapted loss function for step 2 (Section (3.2)):

$$\mathcal{L}_j^{(shared)}(X;\phi_j) = \mathbb{E}_{q_\phi(z|X)}\Big(\log q_{\phi_j}(z|g_j(x_j))\Big). \tag{14}$$

The proof of Proposition 1 can be adapted to show that maximizing $\mathcal{L}_j^{(shared)}$ over the training set, leads to minimizing the KL-divergence between $q_{\phi_j}(z|g_j(x_j))$ and $q_\phi^{(avg)}(z|g_j(x_j)) = \int_X q_\phi(z|X)\hat{p}(X|g_j(x_j))dX$. This is detailed in Appendix C.

Extracting information shared across modalities. How can we learn relevant functions $(g_j)_{1\leq j\leq M}$ that would verify Equation (12)? Many methods have been proposed to extract information shared across modalities, and since the best method may vary depending on the dataset, we treat this as a flexible and data dependent component of our approach. In our experiments, we tried two general methods: Deep Canonical Correlation Analysis (DCCA) (Andrew et al. 2013) and contrastive learning (CL) (Poklukar et al. 2022). In both cases, the projectors $(g_j)_{1\leq j\leq M}$ are trained *jointly* to learn *similar* representations across modalities. For the projections $(g_j(x_j))_{1\leq j\leq M}$ to be *similar* across modalities, the projectors have to extract shared information while discarding unrelated information. The notion of similarity is defined differently in both methods: DCCA maximizes the correlation between projections $Corr(g_i(X_i), g_j(X_j))$ while CL optimizes cosine similarity $\{g_i(X_i)^T g_j(X_j)\}/(||g_i(X_i)||||g_j(X_j)||)$ for each pair i, j of modalities. We detail each method in Appendix B. We *conjecture* that using these methods, we can extract meaningful statistics $(g_j(x_j))_{1\leq j\leq M}$ verifying Equation (12) and check this assumption in our experiments. Results presented in Section (4) and Appendix D.7 support this hypothesis.

We refer to this improvement of our method as *JNF-Shared*. To the best of our knowledge, this is the only work proposing to model posterior distributions conditioned on pretrained semantic representations rather than the entire data. In Figure 1, we summarize our models in the case $M = 2$.

4. Experiments

In this section, we first illustrate our method on a toy dataset, and then compare results against state-of-the-art methods on 4 benchmark datasets.

4.1. Toy dataset

We construct a toy dataset with two black-and-white image modalities: x_1 (a square) and x_2 (a circle), whose sizes vary independently. Each shape is either full or empty, and this binary class label is shared across modalities—if the circle is full, so is the square. Figure 2.a presents samples of this toy dataset. We perform the first step of our method on this dataset (see 3.1), which is training a simple joint VAE with a two-dimensional latent space that we can visualize. In Figure 2, we can see how this joint latent space is structured, with the full shapes on one side (blue dots) and the empty shapes on the other side (red dots). In Figure 2 ,the intensities of the colors indicate the size distribution with larger squares encoded away from the center.

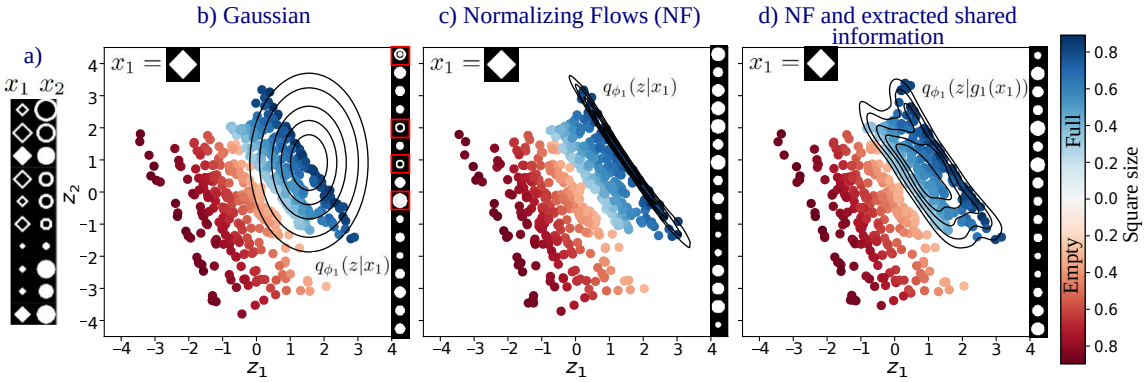


Figure 2: (a) Samples from the toy dataset. (b,c,d) Embeddings in the 2-dimensional latent space. Each point encodes a pair of images (x_1, x_2). The color of each point, indicates the size and class of the encoded *square*. We try to approximate the posterior $p_\theta(z|x_1)$ (for x_1 shown in the top left), that corresponds to *dark blue dots* in the latent space. In (b), we use a diagonal Gaussian distribution and in (c) we use normalizing flows that capture a more realistic posterior. The Gaussian's support is too large, leading to unrealistic samples framed in red. (d) Using DCCA, we extract the information shared across modalities, which is the shape class: full or empty. We learn $q_{\phi_1}(z|g_1(x_1))$ and see that it approximates well the part of the latent space which encodes full shapes. We present circles generated with the learned posterior on the right side of each plot. Both (c) and (d) produce relevant and diverse samples.

Using this well-structured latent space we then train $q_{\phi_1}(z|x_1)$ to approximate $p_\theta(z|x_1)$ using our objective $\mathcal{L}_1$ (Equation (7)). We display an example distribution $q_{\phi_1}(z|x_1)$ that we obtain for x_1 being a large full square. We compare results when modeling $q_{\phi_1}(z|x_1)$ with either a Gaussian or normalizing flows (NF). The latter provides a more realistic approximation and generate coherent and diverse samples of circles. In the last plot of Figure 2, we first extract the information shared across modalities (here the emptiness or fullness of the shape) with a projector $g_1(x_1)$ and then approximate $q_{\phi_1}(z|g_1(x_1))$. g_1 and g_2 are neural networks trained with the DCCA objective. We

see in the right panel of Figure 2, that $q_{\phi_1}(z|g_1(x_1))$ covers well the part of the latent space corresponding to full samples and generates coherent and diverse samples. This shows that we have been able to capture both the shared information and the posterior distribution. On this toy dataset, both $p_\theta(z|x_1)$ and $p_\theta(z|g_1(x_1))$ are well approximated but on benchmark datasets, it appears that the latter is often easier to approximate than the former because of its larger support.

4.2. Benchmark datasets and evaluation metrics

We evaluate JNF and JNF-Shared on four benchmark datasets:

- MNIST-SVHN introduced in (Shi et al. 2019) that contains paired images from MNIST (Lecun et al. 1998) and SVHN dataset (Netzer et al. 2011). The latter contains natural images of digits with diverse backgrounds and sometimes cropped distracting digits on the sides of the digit of interest.
- PolyMNIST introduced in (Sutter et al. 2021) with five image modalities built from MNIST images with complex backgrounds. This dataset allows to test the scalability of our method.
- Translated PolyMNIST introduced in (Daunhawer et al. 2022) to demonstrate the limitations of mixture-based models. It is made of downscaled and translated digits with the same backgrounds as PolyMNIST. In (Daunhawer et al. 2022) the authors point out that the generative performance is very degraded for mixture-based models on this dataset.
- Multimodal Handwritten Digits dataset (MHD) (Vasco et al. 2022) containing images, sound's spectrograms and trajectories.

We provide additional details and samples for each dataset in Appendix A. We focus on conditional and unconditional generation and we evaluate:

- the *coherence* of multimodal samples. Using pretrained classifiers, we evaluate whether the generated samples are consistent across modalities—that is, whether they are assigned the same label. A multimodal sample is considered *coherent* if all modalities receive the same predicted label. We report the proportion of coherent samples among the generated data for both conditional and joint generations.
- the *diversity* of generated samples. To assess sample diversity, we follow the evaluation protocols of (Palumbo et al. 2023) and (Vasco et al. 2022). For MNIST-SVHN and PolyMNIST, we compute the Fréchet Inception Distance (FID) (Heusel et al. 2017) between true and generated samples. For the MHD dataset—where modalities are not natural images and Inception features are unsuitable—we instead use pretrained, class-based, modality-specific autoencoders to extract features, and compute the Mean Fréchet Distance (MFD).

4.3. Comparison to previous work

We compare our method to several models: JMVAE (Suzuki et al. 2016) that also uses a joint encoder, mixture-based models such as MMVAE (Shi et al. 2019), MoPoE (Sutter et al. 2021), and MMVAE+(Palumbo et al. 2023), the MVTCAE model (Hwang et al. 2021) using a PoE aggregation, and two hierarchical models Nexus (Vasco et al. 2022) and CMVAE (Palumbo et al. 2024). We use implementations that were first validated by reproducing previous results. For a fair comparison, we use the same architectures and the same latent capacity across models except for the MMVAE, MMVAE+ and CMVAE for which we use smaller latent spaces due to memory limitations from the K-sampled objective. We detail all hyperparameters in Appendix E. We train all models with a β-weighted ELBO and keep the $\beta \in \{0.5, 1, 2.5\}$ that maximizes average coherence for each model. Each experiment is repeated with four different seeds. We try training the projectors (g_j) for our model JNF-Shared with CL or DCCA pre-training and report results for both.

4.4. Experimental results

In Figure 3, we present generated samples and in Table 1, we present quantitative results for the MNIST-SVHN dataset.

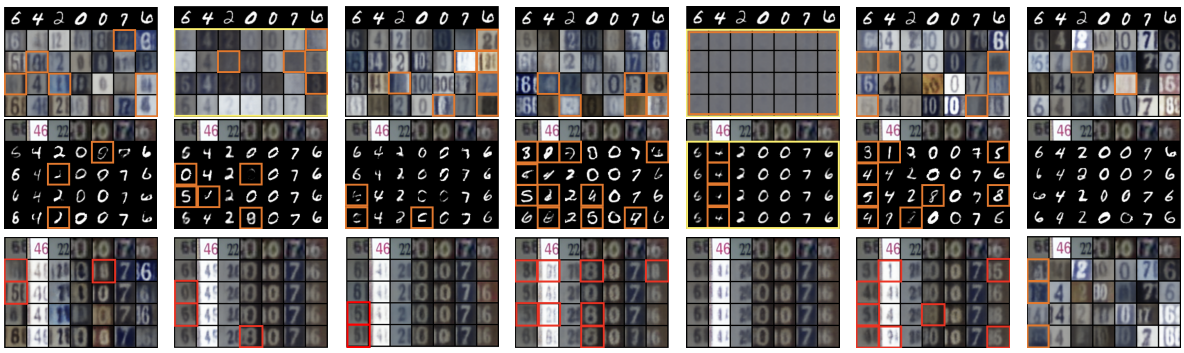


Figure 3: First row: generation from MNIST to SVHN. Second row: generation from SVHN to MNIST. Third row: generation from SVHN to SVHN (unimodal reconstruction). In orange, we frame samples containing unidentifiable or uncoherent digits. In yellow, we frame generations lacking diversity. In red, we frame reconstructions for the SVHN images where the background is well reconstructed but not the digit. JNF-CL refers to our model JNF-Shared with CL. Note that for this model, when reconstructing SVHN, we sample $z \sim q_{\phi_2}(z|g_2(x_2))$ and therefore the background information is filtered by the projector $g_2(x_2)$ and cannot be reconstructed. However, the digit is well preserved which is what is required for cross-modal generation.

Our model JNF-Shared (CL) is the only one to reach competitive values **for all metrics** on this dataset. As displayed in Figure 3, most models (except MoPoE and JNF-Shared) struggle to generate coherent MNIST images from SVHN images. We interpret this phenomenon by looking at reconstructed SVHN images; for many models, the background is well reconstructed but not the digit which is not well inferred using $q_{\phi_2}(z|x_2)$ (where x_2 is the SVHN modality). With JNF-Shared, the background is

Table 1: Results on MNIST-SVHN. We present coherence for joint generation, conditional generation from MNIST (M) to SVHN (S) and vice-versa. FID values are computed on 50,000 SVHN images generated from MNIST. Best values are in bold and second-best are underlined.

Model	**Joint**	**M $\longrightarrow$ S**	**S $\longrightarrow$ M**	**FID ($\downarrow$)**
JMVAE	0.43 ± 0.10	0.73 ± 0.07	0.53 ± 0.05	57 ± 3
MMVAE ($k = 10$)	0.35 ± 0.02	$\underline{0.80} \pm 0.01$	0.70 ± 0.01	130 ± 5
MVTCAE	0.44 ± 0.02	$\mathbf{0.81} \pm 0.01$	0.52 ± 0.02	$\mathbf{48} \pm 2$
MoPoE	0.36 ± 0.01	0.12 ± 0.01	$\underline{0.72} \pm 0.01$	359 ± 12
MMVAE+ ($k = 10$)	0.43 ± 0.05	0.60 ± 0.09	0.58 ± 0.04	63 ± 5
CMVAE ($k = 10$)	$\underline{0.48} \pm 0.18$	0.57 ± 0.18	0.57 ± 0.09	101 ± 54
JNF (Ours)	$\mathbf{0.51} \pm 0.01$	$\mathbf{0.82} \pm 0.01$	0.52 ± 0.01	$\underline{54} \pm 2$
JNF-Shared (DCCA) (Ours)	$\mathbf{0.51} \pm 0.01$	0.75 ± 0.03	0.69 ± 0.05	$\underline{53} \pm 2$
JNF-Shared (CL) (Ours)	$\mathbf{0.51} \pm 0.02$	$\mathbf{0.81} \pm 0.01$	$\mathbf{0.75} \pm 0.02$	$\mathbf{49} \pm 1$

tuned out by the projector g_2 and the digit information is therefore better preserved when sampling $z \sim q_{\phi_2}(z|g_2(x_2))$. We further notice that MMVAE and MoPoE both produce SVHN samples that look 'averaged' resulting from the quality gap analyzed in (Daunhawer et al. 2022).

In Figure 4, we present coherence and diversity results on PolyMNIST and Translated PolyMNIST. We observe that our models reach the best coherences while maintaining low FID values. In (Daunhawer et al. 2022), the authors observed that MMVAE and MoPoE show degraded coherence on TranslatedPolyMNIST. We extend their observations to the MMVAE+ and CMVAE that also have a conditional coherence close to 0.10, corresponding to random digit association. This is a direct consequence of the **mixture aggregation**, which limits generative quality on complex datasets (Daunhawer et al. 2022). However, all models fail on the unconditional generation task: MMVAE and MoPoE have high coherence but high FID because they only generate the first digit (see Appendix D). On the contrary, our models have low FID values but very low coherence. We present in Section (4.5) a simple direction to improve joint coherence. In Table 2, we present results on the MHD dataset. The performance gain is less noticeable on this dataset than on the previous ones but we still notice that our model JNF-Shared is the only one to obtain the best or second best value accross the four metrics showing that they offer a good compromise between coherence and diversity. The CMVAE model reaches higher joint coherence which is due to the fact that they use a Gaussian mixture (GM) prior where other models use a standard Gaussian. We show in Section (4.5) that we can improve the joint coherence of our model up to 0.81 by also sampling from a GM fitted on the training embeddings.

In the end, our models reach competitive performance on the four datasets presented, with significant gains on MNIST-SVHN, PolyMNIST, and in the more complex setting of TranslatedPolyMNIST. We note that, in general, CL pretraining appears to be more effective than DCCA for our method, JNF-Shared. Although our models have more parameters than aggregated approaches—since they model both joint and individual

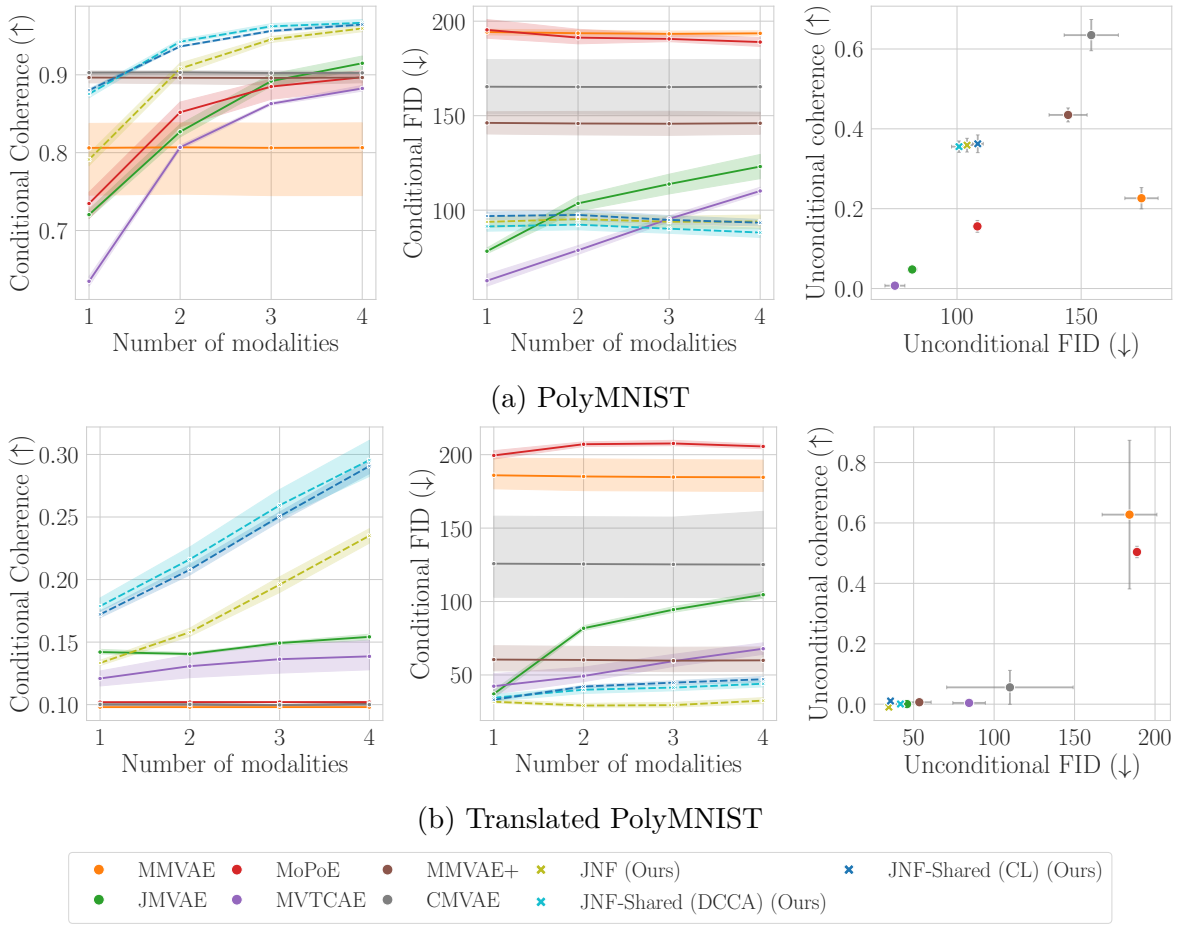


Figure 4: In the two left columns, we present results for conditional generation when varying the number of conditioning modalities. In the right column, we display coherence and FID for unconditional generation. Each point correspond to a different training seed. For these plots, best models having high coherence and low FID are in the top left corner. The FID is computed on 10,000 samples of the first modality.

posteriors separately—JNF-Shared remains relatively compact due to the use of lightweight projector architectures $(g_j)_j$. Additional experimental insights—including visualizations, a discussion on parameter counts, and training times—are provided in Appendix D. Ablation studies, presented in Appendix D.7, further explore the contribution of each component of our methods.

4.5. Improving the joint coherence with a posteriori sampling.

In all results above, joint generation quality is evaluated by sampling latent codes from the prior, as this aligns naturally with the probabilistic model. However, VAEs are known to produce latent codes that deviate from the prior distribution. Ghosh et al. (2020) argue that the standard Gaussian prior acts mostly as a regularizer and propose *ex post density estimation*, which involves fitting a simple distribution such as a Gaussian mixture (GM) to the training embeddings. We apply the same method to multimodal VAEs by fitting a GM to the joint embeddings to generate unconditional samples.

Table 2: Experimental results on the MHD dataset. We present average coherence and MFD results for each model, for conditional and unconditional generation. Best values are in bold and second-best values are underlined.

	Coherence (↑)		**MFD** (↓)	
	Joint	Conditional	Joint	Conditional
JMVAE	0.57 ± 0.02	0.86 ± 0.01	$\mathbf{1.32} \pm 0.01$	0.29 ± 0.02
MMVAE	0.63 ± 0.01	0.86 ± 0.01	1.63 ± 0.05	0.76 ± 0.01
MMVAE+	0.57 ± 0.01	$\underline{0.89} \pm 0.01$	1.58 ± 0.07	0.55 ± 0.08
MVTCAE	0.38 ± 0.01	0.87 ± 0.01	$\mathbf{1.31 \pm 0.02}$	$\mathbf{0.13 \pm 0.01}$
MoPoE	0.44 ± 0.02	0.74 ± 0.01	1.56 ± 0.03	2.17 ± 0.03
Nexus	0.13 ± 0.01	0.34 ± 0.01	2.98 ± 0.04	3.36 ± 0.03
CMVAE	$\mathbf{0.89} \pm 0.02$	$\mathbf{0.93} \pm 0.01$	$\mathbf{1.30} \pm 0.02$	0.46 ± 0.03
JNF(Ours)	$\underline{0.67} \pm 0.01$	$\underline{0.89} \pm 0.01$	$\mathbf{1.32} \pm 0.02$	$\underline{0.23} \pm 0.02$
JNF-Shared(CL)(Ours)	$\underline{0.65} \pm 0.02$	$\mathbf{0.93} \pm 0.01$	$\underline{1.35} \pm 0.04$	$\underline{0.21} \pm 0.03$
JNF-Shared(DCCA)(Ours)	$\underline{0.66} \pm 0.01$	$\mathbf{0.92} \pm 0.01$	$\underline{1.37} \pm 0.04$	$\underline{0.23} \pm 0.03$

In Table 3, we show how using this technique for generating unconditional samples greatly improves the joint coherence on all datasets. The same method can be applied on almost all models: additional results are presented in Appendix D.5.

5. Discussion and Perspectives

In this work, we introduced a novel VAE-based framework for modeling and generating multimodal data. Our approach relies on a multistage modeling strategy that enhances both the generative and inference models by employing *principled objective functions*. Additionally, we proposed a method for learning unimodal posteriors conditioned on a summary statistic capturing information shared across modalities. The proposed models demonstrate strong performance across four benchmark datasets, with notable gains in the particularly challenging Translated PolyMNIST setting.

Several components of our framework are modular and adaptable. For instance, the initial stage uses a standard VAE which can be seamlessly replaced with more expressive VAE variants (Tomczak and Welling 2018; Kingma et al. 2017; Burda et al. 2016). Similarly, the mechanism for extracting shared cross-modal information is general and

Table 3: Joint coherence of the JNF-Shared (CL) model when sampling from the prior distribution versus a GM fitted on the training embeddings.

	MNIST-SVHN	PolyMNIST	Translated PolyMNIST	MHD
prior	0.51 ± 0.02	0.36 ± 0.02	0.0005 ± 0.0003	0.67 ± 0.01
GM	$\mathbf{0.75 \pm 0.04}$	$\mathbf{0.56 \pm 0.04}$	$\mathbf{0.032 \pm 0.003}$	$\mathbf{0.81 \pm 0.02}$

can be substituted with domain-specific alternatives. For example, kernel canonical correlation analysis (KCCA) has shown effectiveness in areas like functional imaging and genomics (Alam et al. 2018). Finally, for complex applications, the normalizing flows used to model the posterior distributions could be replaced with more expressive methods such as conditional diffusion models (Ho et al. 2020), offering a promising direction for future work.

Computational details

For reproducibility, we provide a python implementation of our methods online: https://anonymous.4open.science/r/JNF_VAE/README.md. Our implementation uses **pytorch** (Paszke et al. 2019), **matplotlib** (Hunter 2007), **numpy** (Harris et al. 2020), **pandas** (Wes McKinney 2010), **seaborn** (Waskom 2021), **multivae** (Senellart et al. 2025).

Acknowledgments

Research partly supported by the European Union under the (2023-2030) ERC Synergy grant 101071601 (OCEAN) and the AI Cluster ANR program PRAIRIE-PSAI ANR-23-IACL-0008.

References

Alam, M. A., Calhoun, V. D., and Wang, Y.-P. (2018). Identifying outliers using multiple kernel canonical correlation analysis with application to imaging genetics. *Computational Statistics & Data Analysis*, 125:70–85, ISSN: 0167-9473, DOI: 10.1016/j.csda.2018.03.013.

Andrew, G., Arora, R., Bilmes, J., and Livescu, K. (2013). Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1247–1255. PMLR, ISSN: 1938-7228, https://proceedings.mlr.press/v28/andrew13.html.

Betancourt, M. (2018). A conceptual introduction to hamiltonian Monte Carlo. https://arxiv.org/abs/1701.02434.

Bounoua, M., Franzese, G., and Michiardi, P. (2024). Multi-modal latent diffusion. *Entropy*, 26(4), ISSN: 1099-4300, DOI: 10.3390/e26040320.

Brooks, S., Gelman, A., Jones, G., and Meng, X.-L. (2011). *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, ISBN: 9780429138508, DOI: 10.1201/b10905.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2016). Importance weighted autoencoders. DOI: 10.48550/arXiv.1509.00519.

Chadebec, C., Vincent, L. J., and Allassonniere, S. (2022). **Pythae**: Unifying generative autoencoders in Python - A benchmarking use case. https://openreview.net/forum?id=w7VPQWgnn3s.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In III, H. D. and Singh, A., editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, https://proceedings.mlr.press/v119/chen20j.html.

Daunhawer, I., Sutter, T. M., Chin-Cheong, K., Palumbo, E., and Vogt, J. E. (2022). On the limitations of multimodal VAEs. In *International Conference on Learning Representations*, https://openreview.net/forum?id=w-CPUXXrAj.

Daunhawer, I., Sutter, T. M., Marcinkevičs, R., and Vogt, J. E. (2021). Self-supervised disentanglement of modality-specific and shared factors improves multimodal generative models. In Akata, Z., Geiger, A., and Sattler, T., editors, *Pattern Recognition*, Lecture Notes in Computer Science, pages 459–473, Cham. Springer International Publishing, ISBN: 978-3-030-71278-5, DOI: 10.1007/978-3-030-71278-5_33.

Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M., and Schölkopf, B. (2020). From variational to deterministic autoencoders. http://arxiv.org/abs/1903.12436.

Girolami, M. and Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo Methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 73(2):123–214, ISSN: 1369-7412, DOI: 10.1111/j.1467-9868.2010.00765.x.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural Computation*, 16(12):2639–2664, ISSN: 0899-7667, DOI: 10.1162/0899766042321814.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). Array programming with **NumPy**. *Nature*, 585(7825):357–362, DOI: 10.1038/s41586-020-2649-2.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, https://api.semanticscholar.org/CorpusID:206594692.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, https://arxiv.org/abs/1706.08500.

Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A. (2017). beta-VAE: Learning basic visual concepts with a constrained variational framework. https://openreview.net/forum?id=Sy2fzU9gl.

Higgins, I., Sonnerat, N., Matthey, L., Pal, A., Burgess, C. P., Bosnjak, M., Shanahan, M., Botvinick, M., Hassabis, D., and Lerchner, A. (2018). SCAN: Learning hierarchical compositional visual concepts. DOI: 10.48550/arXiv.1707.03389.

Ho, J., Jain, A., and Abbeel, P. (2020). Denoising diffusion probabilistic models. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Hunter, J. D. (2007). **Matplotlib**: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, DOI: 10.1109/MCSE.2007.55.

Hwang, H., Kim, G.-H., Hong, S., and Kim, K.-E. (2021). Multi-view representation learning via total correlation objective. In *Advances in Neural Information Processing Systems*, volume 34, pages 12194–12207. Curran Associates, Inc., https://proceedings.neurips.cc/paper/2021/hash/65a99bb7a3115fdede20da98b08a370f-Abstract.html.

Javaloy, A., Meghdadi, M., and Valera, I. (2022). Mitigating modality collapse in multimodal VAEs via impartial optimization. http://arxiv.org/abs/2206.04496.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1998). An introduction to variational methods for graphical models. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 105–161. Springer Netherlands, Dordrecht, ISBN: 978-94-010-6104-9, DOI: 10.1007/978-94-011-5014-9_5.

Kanatsoulis, C. I., Fu, X., Sidiropoulos, N. D., and Hong, M. (2019). Structured sumcor multiview canonical correlation analysis for large-scale data. *IEEE Transactions on Signal Processing*, 67(2):306–319, DOI: 10.1109/TSP.2018.2878544.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2017). Improving variational inference with inverse autoregressive flow. http://arxiv.org/abs/1606.04934.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. http://arxiv.org/abs/1312.6114.

Lawry Aguila, A., Chapman, J., and Altmann, A. (2023). Multi-modal variational autoencoders for normative modelling across multiple imaging modalities. In Greenspan, H., Madabhushi, A., Mousavi, P., Salcudean, S., Duncan, J., Syeda-Mahmood, T., and Taylor, R., editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2023*, pages 425–434, Cham. Springer Nature Switzerland, https://arxiv.org/abs/2303.12706.

Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, ISSN: 1558-2256, DOI: 10.1109/5.726791.

Lee, M. and Pavlovic, V. (2021). Private-shared disentangled multimodal vae for learning of latent representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 1692–1700, https://arxiv.org/abs/2012.13024

Liu, J. (2009). *Monte Carlo Strategies in Scientic Computing*. ISBN: 0387763694, DOI: 10.1007/978-0-387-76371-2.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. (2011). Reading digits in natural images with unsupervised feature learning. *NIPS*, https://www.researchgate.net/publication/266031774_Reading_Digits_in_Natural_Images_with_Unsupervised_Feature_Learning.

Palumbo, E., Daunhawer, I., and Vogt, J. E. (2023). MMVAE+: Enhancing the generative quality of multimodal VAEs without compromises. In *The Eleventh International Conference on Learning Representations*, https://openreview.net/forum?id=sdQGxouELX.

Palumbo, E., Manduchi, L., Laguna, S., Chopard, D., and Vogt, J. E. (2024). Deep generative clustering with multimodal diffusion variational autoencoders. In *The 12th International Conference on Learning Representations*, https://openreview.net/forum?id=k5THrhXDV3.

Papamakarios, G., Pavlakou, T., and Murray, I. (2017). Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., https://proceedings.neurips.cc/paper/2017/hash/6c1da886822c67822bcf3679d04369fa-Abstract.html.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). **Pytorch**: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Poklukar, P., Vasco, M., Yin, H., Melo, F. S., Paiva, A., and Kragic, D. (2022). Geometric multimodal contrastive representation learning. In *Proceedings of the 39th International Conference on Machine Learning*, pages 17782–17800. PMLR, ISSN: 2640-3498. https://proceedings.mlr.press/v162/poklukar22a.html.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *CoRR*, https://arxiv.org/abs/2103.00020.

Rezende, D. and Mohamed, S. (2015). Variational inference with normalizing flows. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1530–1538, Lille, France. PMLR, https://arxiv.org/abs/1505.05770.

Rezende, D. J. and Mohamed, S. (2016). Variational inference with normalizing flows. https://arxiv.org/abs/1505.05770.

Senellart, A., Chadebec, C., and Allassonnière, S. (2025). Multivae: A Python package for multimodal variational autoencoders on partial datasets. *Journal of Open Source Software*, 10(110):7996, DOI: 10.21105/joss.07996.

Shi, Y., Siddharth, N., Paige, B., and Torr, P. H. S. (2019). Variational mixture-of-experts autoencoders for multi-modal deep generative models. http://arxiv.org/abs/1911.03393.

Sutter, T., Daunhawer, I., and Vogt, J. (2020). Multimodal generative learning utilizing Jensen-Shannon-Divergence. In *Advances in Neural Information Processing Systems*, volume 33, pages 6100–6110. Curran Associates, Inc., https://proceedings.neurips.cc/paper/2020/hash/43bb733c1b62a5e374c63cb22fa457b4-Abstract.html.

Sutter, T. M., Daunhawer, I., and Vogt, J. E. (2021). Generalized multimodal ELBO. *ICLR*, https://arxiv.org/abs/2105.02470.

Suzuki, M. and Matsuo, Y. (2022). A survey of multimodal deep generative models. *Advanced Robotics*, 36(5-6):261–278, ISSN: 0169-1864, 1568-5535, DOI: 10.1080/01691864.2022.2035253.

Suzuki, M., Nakayama, K., and Matsuo, Y. (2016). Joint multimodal learning with deep generative models, http://arxiv.org/abs/1611.01891.

Tian, Y., Krishnan, D., and Isola, P. (2020). Contrastive Multiview Coding. In Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M., editors, *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pages 776–794, Cham. Springer International Publishing, ISBN: 978-3-030-58621-8, DOI: 10.1007/978-3-030-58621-8_45.

Tomczak, J. and Welling, M. (2018). Vae with a vampprior. In Storkey, A. and Perez-Cruz, F., editors, *Proceedings of the 21th International Conference on Artificial Intelligence and Statistics*, volume 84 of *Proceedings of Machine Learning Research*, pages 1214–1223. PMLR, https://proceedings.mlr.press/v84/tomczak18a.html.

Vasco, M., Yin, H., Melo, F. S., and Paiva, A. (2022). Leveraging hierarchy in multimodal generative models for effective cross-modality inference. *Neural Networks*, 146:238–255, ISSN: 08936080, DOI: 10.1016/j.neunet.2021.11.019.

Vedantam, R., Fischer, I., Huang, J., and Murphy, K. (2018). Generative models of visually grounded imagination. http://arxiv.org/abs/1705.10762.

Waskom, M. L. (2021). **seaborn**: Statistical data visualization. *Journal of Open Source Software*, 6(60):3021, DOI: 10.21105/joss.03021.

Wes McKinney (2010). Data structures for statistical computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56–61. DOI: 10.25080/Majora-92bf1922-00a.

Wu, M. and Goodman, N. (2018). Multimodal generative models for scalable weakly-supervised learning. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., https://proceedings.neurips.cc/paper/2018/hash/1102a326d5f7c9e04fc3c89d0ede88c9-Abstract.html.

Yin, H., Melo, F., Billard, A., and Paiva, A. (2017). Associate latent encodings in learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), ISSN: 2374-3468, DOI: 10.1609/aaai.v31i1.11040.

A. Details on the Datasets Used in the Experiments

A.1. The MNIST-SVHN dataset

To create this dataset, we paired images from the MNIST dataset (Lecun et al. 1998) and the SVHN dataset (Netzer et al. 2011). Previous work (Shi et al. 2019) paired each image in MNIST with 30 different images in SVHN to create a train set of 1,682,040 samples. To create a more challenging and realistic dataset, we only paired each image 5 times to have a smaller (yet still large) training dataset of 280,340 samples.

A.2. PolyMNIST and translated PolyMNIST dataset

In Figure 5, we plot example images of the PolyMNIST and Translated PolyMNIST dataset used in the experiments in Section (4). For the Translated PolyMNIST dataset, we downscale the digit by a factor 0.75 and add a random translation. Each dataset contains 60,000 training samples and 10,000 test samples.

(a) PolyMNIST. (b) Translated PolyMNIST.

Figure 5: Eight multimodal samples for the PolyMNIST and TranslatedPolyMNIST dataset: each row correspond to a modality.

A.3. Multimodal handwritten dataset

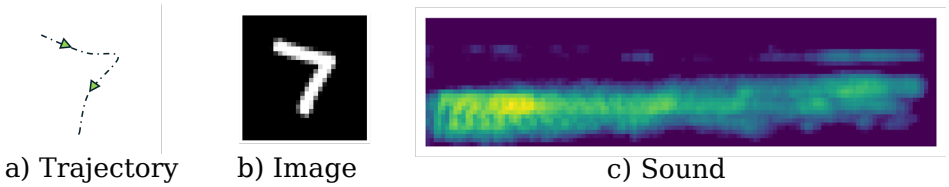


Figure 6: The MHD dataset that we use contains three modalities.

The "Multimodal Handwritten Digits" (MHD) introduced in (Vasco et al. 2022) contains 4 modalities (including label):

- Image: gray digit images of size (1,28,28)
- Audio: spectrograms images with shape (1,32,128)

- Trajectory: flat arrays with 200 values
- Label : 10 categorical values

In our experiments, we don't use the label as a modality to make the task more challenging. This dataset contains 50,000 samples for training and 10,000 for testing.

A.4. Toy dataset with circles and squares

The images of circles and squares used in the toy experiment are of size (32,32) with black and white pixels. All circles and squares are centered in the middle of the image with a minimum width of 10 pixels and a maximum width of 28 pixels. This dataset contains 200,000 pairs of circles and squares. Half are empty and half are full.

B. Methods to Learn Shared Information Across Multiple Modalities

Here, we detail two methods we have used to train the projectors $(g_j)_{1\leq j\leq M}$ to extract information *shared* across modalities. The projectors $(g_j)_{1\leq j\leq M}$ are trained *before* training our multimodal VAE JNF-Shared that uses them.

B.1. Deep canonical correlation analysis

Deep Canonical Correlation Analysis (Andrew et al. 2013) (DCCA) aims at finding correlated neural representations for two complex modalities such as images. It is based upon the classical Canonical Correlation Analysis (CCA) (Hardoon et al. 2004) which we briefly recall here. Let $(X_1, X_2) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ two random vectors, Σ_1, Σ_2 their covariances matrices and $\Sigma_{1,2} = \text{Cov}(X_1, X_2)$. CCA's objective is to find linear projections $a^T X_1$, $b^T X_2$ that are maximally correlated :

$$(a^*, b^*) = \underset{a^T\Sigma_1 a = b^T\Sigma_2 b = 1}{\arg\max} a^T \Sigma_{1,2} b.$$

Once these optimal projections are found, we can set $(a_1, b_1) = (a^*, b^*)$ and search for subsequent projections $(a_i, b_i)_{2\leq i\leq k}$ with the additional constraint that they must be uncorrelated with the previous ones. We can rewrite the problem of finding the first k optimal pairs of projection as finding matrices $A \in \mathbb{R}^{(n_1,k)}$, $B \in \mathbb{R}^{(n_2,k)}$ that solve:

$$(A^*, B^*) = \underset{A^T\Sigma_1 A = B^T\Sigma_2 B = I}{\arg\max} Tr(A^T \Sigma_{1,2} B). \tag{15}$$

If we further have $k = n_1 = n_2$ then the maximum value for $Tr(A^T\Sigma_{1,2}B)$ is $F(X_1, X_2) = Tr(T^\top T)^{1/2}$ with $T = \Sigma_1^{1/2}\Sigma_{1,2}\Sigma_2^{1/2}$. This value is the total CCA correlation of the random vectors X_1, X_2. It can also be seen as the sum of the singular values of T, each singular value representing the correlation of the embeddings along a direction. Note that this optimal value $F(X_1, X_2)$ only depends on the covariance matrices $(\Sigma_1, \Sigma_2, \Sigma_{1,2})$.

In the DCCA method, we consider two neural networks g_1, g_2 so as to optimize the total CCA correlation $F(g_1(X_1), g_2(X_2))$. The gradient of this objective with respect

to the parameters of g_1, g_2 can be derived in order to use gradient descent. In practice, to compute F we can use the singular value decomposition of T and sum the first k singular values of T. Furthermore the singular values are interesting since they give an information of how much correlation is contained in each projection. That information can be used to analyse the data and choose an optimal dimension k for the projection.

When considering more than two modalities, a proposed extension to the CCA is to optimize the sum of the pairwise CCA objectives (Kanatsoulis et al. 2019). We adapt this idea to the DCCA framework and train DCCA encoders for m modalities by maximizing $\sum_{i<j\in[|1,m|]} F(g_i(X_i), g_j(X_j))$.

Our implementation is based upon `https://github.com/Michaelvll/DeepCCA`.

B.2. Multimodal contrastive learning

Contrastive learning methods have emerged as a powerful tool to learn descriptive, transferable representations of high dimensional data such as images or text (Radford et al. 2021).

In the two-modalities case, we aim at learning two embbeding functions $g_1(x_1)$, $g_2(x_2)$ that brings together "positive pairs" observed from the joint distribution $x_1, x_2 \sim p(x_1, x_2)$ and separates "negatives pairs" observed from the product of the marginal distributions $x_1, x_2 \sim p(x_1)p(x_2)$.

Formally, considering a batch of multimodal samples $(x_1^i, x_2^i)_{1\leq i\leq K}$, the loss function writes:

$$L = \sum_{i=1}^{K} L_{1,2}(i) + L_{2,1}(i) \tag{16}$$

$$L_{1,2}(i) = -\log\left(\frac{sim_\gamma(x_1^i, x_2^i)}{\sum_{j=1}^{K} sim_\gamma(x_1^i, x_2^j)}\right) \quad \forall 1 \leq i \leq K \tag{17}$$

$$L_{2,1}(i) = -\log\left(\frac{sim_\gamma(x_2^i, x_1^i)}{\sum_{j=1}^{K} sim_\gamma(x_2^i, x_1^j)}\right) \quad \forall 1 \leq i \leq K, \tag{18}$$

where $sim_\gamma(x_1, x_2) = \exp((1/\tau)(g_1(x_1))/(||g_1(x_1)||) \cdot (g_2(x_2)/||g_2(x_2)||))$ is the exponential cosine similarity between the embeddings,τ is a hyperparameter and γ parameterize the embedding functions g_1, g_2 that we aim to optimize. $\tau = 0.1$ in our experiments.

For any $1 \leq i \leq K$, the pair $(x_1^{(i)}, x_2^{(i)})$ is a *positive* pair which should have high similarity and the pairs $(x_1^{(i)}, x_2^{(j)})_{1\leq j\neq i\leq K}$, $(x_1^{(j)}, x_2^{(i)})_{1\leq j\neq i\leq K}$ are *negative* pairs that should have low similarity.

In order to bring together positive pairs in the embedding space and separate negative pairs, the projectors $(g_j)_{1\leq j\leq M}$ have to extract the information between modalities.

For a larger number of modalities: $M \geq 2$, we can compute the sum of all pairwise losses and minimize them jointly (Tian et al. 2020).

C. Interpretation of the Objective

In this appendix, we provide the proof of Proposition 1 as well a similar proposition for the $\mathcal{L}_j^{shared}$ objective.

C.1. Proof of Proposition 1.

We show that, for $j \in [|1, M|]$, $\mathcal{L}_j$ brings the unimodal encoder $q_{\phi_j}(z|x_j)$ close to an average distribution $q_{\text{avg}}(z|x_j) = \int_{X_{C_j}} q_\phi(z|X)\hat{p}(X_{C_j}|x_j)dX_{C_j}$ that is close to $p_\theta(z|x_j)$ provided that the joint encoder is well fit.

When maximizing $\mathcal{L}_j$ over the entire dataset, we actually optimize the expectation of this term over the empirical distribution $\hat{p}(X)$;

$$\mathbb{E}_{\hat{p}(X)}(\mathcal{L}_j) = \mathbb{E}_{\hat{p}(X)}\left(\mathbb{E}_{q_\phi(z|X)}\left(\log(q_{\phi_j}(z|x_j))\right)\right). \tag{19}$$

We can decompose $\hat{p}(X) = \hat{p}(x_j)\hat{p}(X_{C_j}|x_j)$ where we note $X_{C_j} = (x_i)_{1\leq i\neq j\leq M}$ the set of modalities from which we exclude x_j.

$$\mathbb{E}_{\hat{p}(X)}(\mathcal{L}_j) = \mathbb{E}_{\hat{p}(x_j)}\left(\mathbb{E}_{\hat{p}(X_{C_j}|x_j)}\left(\mathbb{E}_{q_\phi(z|X)}\left(\log(q_{\phi_j}(z|x_j))\right)\right)\right). \tag{20}$$

We suppose the density $q_{\phi_j}(z|x_j)$ bounded by a constant C, which allows us to use Fubini's theorem and exchange the expectations.

$$\begin{aligned}
\mathbb{E}_{\hat{p}(X)}(\mathcal{L}_j) &= \mathbb{E}_{\hat{p}(x_j)}\left(\mathbb{E}_{\hat{p}(X_{C_j}|x_j)}\left(\mathbb{E}_{q_\phi(z|X)}\left(\log\left(\frac{q_{\phi_j}(z|x_j)}{C}\right)\right)\right)\right) + \log(C) && (21)\\
&= \mathbb{E}_{\hat{p}(x_j)}\left(\int_{X_{C_j}}\int_z \log\left(\frac{q_{\phi_j}(z|x_j)}{C}\right) q_\phi(z|X)\hat{p}(X_{C_j}|x_j)dzdX_{C_j}\right) + \log(C) && (22)\\
&= \mathbb{E}_{\hat{p}(x_j)}\left(\int_z \log\left(\frac{q_{\phi_j}(z|x_j)}{C}\right)\int_{X_{C_j}} q_\phi(z|X)\hat{p}(X_{C_j}|x_j)dX_{C_j}dz\right) + \log(C) && (23)\\
&= \mathbb{E}_{\hat{p}(x_j)}\left(\mathbb{E}_{q_\phi^{(avg)}(z|x_j)}\left(\log\left(\frac{q_{\phi_j}(z|x_j)}{C}\right)\right)\right) + \log(C) && (24)\\
&= \mathbb{E}_{\hat{p}(x_j)}\left(-KL\left(q_\phi^{(avg)}(z|x_j)||q_{\phi_j}(z|x_j)\right)\right) + H(q_\phi^{(avg)}(z|x_j)) + \log(C), && (25)
\end{aligned}$$

where $q_\phi^{(avg)}(z|x_j) \coloneqq \int_{X_{C_j}} q_\phi(z|X)\hat{p}(X_{C_j}|x_j)dX_{C_j}$. We use Fubini's theorem at line (24) since all terms in the integral are positive.

Since $H(q_\phi^{(avg)}(z|x_j))$ is also a constant term, we see that maximizing $\mathcal{L}_j$ reduces to minimizing the Kullback-Leibler divergence between $q_{\phi_j}(z|x_j)$ and this average distribution $q_\phi^{(avg)}(z|x_j)$.

C.2. Interpretation of the loss when conditioning on the shared information

Once again, let's take $1 \leq j \leq M$. In the case, where we aim to approximate $p_\theta(z|g_j(x_j))$ instead of $p_\theta(z|x_j)$, we can easily adapt the proof above to show that maximizing $\mathcal{L}_j^{(shared)}$ brings $q_{\phi_j}(z|g_j(x_j))$ close to $q_\phi^{(avg)}(z|g_j(x_j)) = \int_X q_\phi(z|X)\hat{p}(X|g_j(x_j))dX$.

We can decompose $\hat{p}(X) = \hat{p}(g_j(x_j))\hat{p}(X|g_j(x_j))$.

$$\mathbb{E}_{\hat{p}(X)}(\mathcal{L}_j^{(shared)}) = \mathbb{E}_{\hat{p}(g_j(x_j))}\left(\mathbb{E}_{\hat{p}(X|g_j(x_j))}\left(\mathbb{E}_{q_\phi(z|X)}\left(\log(q_{\phi_j}(z|g_j(x_j)))\right)\right)\right). \tag{26}$$

We suppose the density $q_{\phi_j}(z|g_j(x_j))$ bounded by a constant C, which allows us to use Fubini's theorem and exchange the expectations.

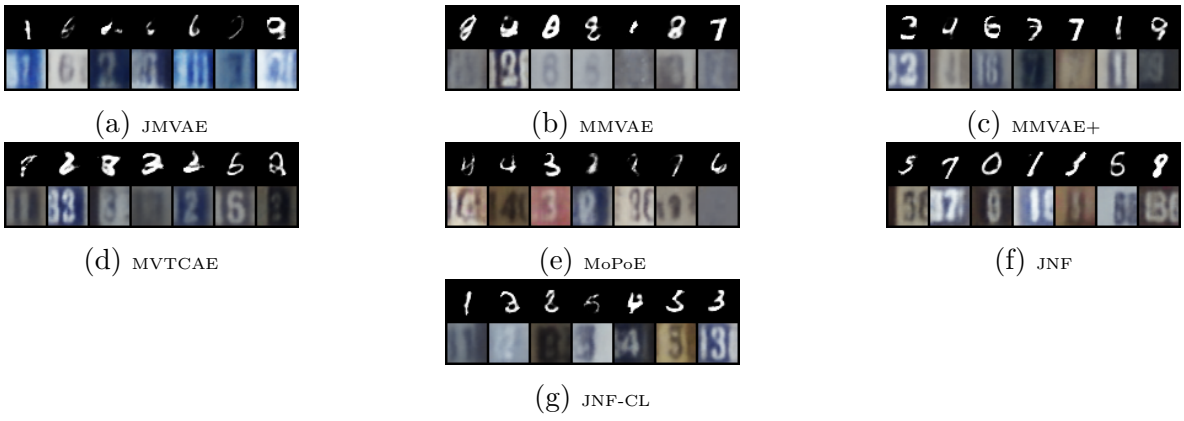
$$\begin{aligned}
\mathbb{E}_{\hat{p}(X)}(\mathcal{L}_j^{(shared)}) &= \mathbb{E}_{\hat{p}(g_j(x_j))}\left(\mathbb{E}_{\hat{p}(X|g_j(x_j))}\left(\mathbb{E}_{q_\phi(z|X)}\left(\log\left(\frac{q_{\phi_j}(z|g_j(x_j))}{C}\right)\right)\right)\right) + \log(C) && (27)\\
&= \mathbb{E}_{\hat{p}(g_j(x_j))}\left(\int_X\int_z \log\left(\frac{q_{\phi_j}(z|g_j(x_j))}{C}\right) q_\phi(z|X)\hat{p}(X|g_j(x_j))dzdX\right) && (28)\\
&\quad + \log(C)\\
&= \mathbb{E}_{\hat{p}(g_j(x_j))}\left(\int_z \log\left(\frac{q_{\phi_j}(z|g_j(x_j))}{C}\right)\int_X q_\phi(z|X)\hat{p}(X|g_j(x_j))dXdz\right) && (29)\\
&\quad + \log(C)\\
&= \mathbb{E}_{\hat{p}(g_j(x_j))}\left(\mathbb{E}_{q_\phi^{(avg)}(z|g_j(x_j))}\left(\log\left(\frac{q_{\phi_j}(z|g_j(x_j))}{C}\right)\right)\right) + \log(C) && (30)\\
&= \mathbb{E}_{\hat{p}(g_j(x_j))}\left(-KL\left(q_\phi^{(avg)}(z|g_j(x_j))||q_{\phi_j}(z|g_j(x_j))\right)\right) && (31)\\
&\quad + H(q_\phi^{(avg)}(z|g_j(x_j))) + \log(C),
\end{aligned}$$

where $q_\phi^{(avg)}(z|g_j(x_j)) := \int_X q_\phi(z|X)\hat{p}(X|g_j(x_j))dX$. We use Fubini's theorem at line (29) since all terms in the integral are positive. Note that if $q_\phi(z|X) = p_\theta(z|X)$ and $\hat{p}(X) = p_\theta(X)$, then $q_\phi^{(avg)}(z|g_j(x_j)) = p_\theta(z|g_j(x_j))$.

D. Additional Experimental Results

D.1. Additional results on MNIST-SVHN

In Figure 7, we present samples generated from the prior. JNF-CL refers to our model JNF-Shared using Constrastive Learning (CL) to extract the shared information. This method performed best on this dataset, to extract the shared information.



(a) JMVAE (b) MMVAE (c) MMVAE+

(d) MVTCAE (e) MoPoE (f) JNF

(g) JNF-CL

Figure 7: Unconditional generation: for each model, latent codes are sampled from the prior and decoded jointly.

In Table 4, we report all coherences results for different values of the parameter β. For each model, we kept the value of β that maximises the mean coherence for the results presented in Table 1.

Table 4: All coherences results for different values of β for each model. We indicate in bold, the value of β that maximises average (conditional and joint) coherence for each model and that we kept for Table 1. In 1, we presented results for our model JNF-Shared using Constrastive Learning (CL). Here, we present additional results with the DCCA used instead of Constrastive Learning.

		Joint		$M \longrightarrow S$		$S \longrightarrow M$	
		mean	std	mean	std	mean	std
Model	β						
JMVAE	0.5	0.27	0.02	0.67	0.03	0.57	0.03
	1	0.34	0.07	0.69	0.05	0.54	0.03
	2.5	0.43	0.10	0.73	0.07	0.53	0.05
MMVAE	**0.5**	0.35	0.02	0.80	0.01	0.70	0.02
	1	0.35	0.02	0.80	0.02	0.68	0.02
	2.5	0.33	0.01	0.80	0.02	0.68	0.03
MMVAE+	0.5	0.24	0.04	0.55	0.04	0.62	0.02
	1	0.27	0.03	0.50	0.03	0.59	0.06
	2.5	0.43	0.05	0.60	0.09	0.58	0.05
MVTCAE	0.5	0.29	0.01	0.74	0.02	0.36	0.02
	1	0.35	0.02	0.75	0.05	0.44	0.02
	2.5	0.44	0.02	0.81	0.01	0.52	0.02
MoPoE	0.5	0.27	0.02	0.13	0.01	0.77	0.00
	1	0.32	0.01	0.12	0.00	0.75	0.01
	2.5	0.36	0.01	0.12	0.00	0.72	0.01
CMVAE	0.5	0.30	0.01	0.63	0.02	0.63	0.03
	1	0.30	0.02	0.49	0.02	0.63	0.04
	2.5	0.48	0.18	0.57	0.18	0.57	0.09
JNF	0.5	0.37	0.01	0.80	0.01	0.47	0.01
	1	0.43	0.01	0.81	0.01	0.48	0.02
	2.5	0.51	0.01	0.82	0.01	0.52	0.01
JNF-Dcca	0.5	0.36	0.02	0.76	0.01	0.71	0.02
	1	0.42	0.02	0.76	0.01	0.71	0.02
	2.5	0.51	0.01	0.75	0.03	0.69	0.05
JNF-CL	0.5	0.36	0.03	0.78	0.02	0.79	0.01
	1	0.42	0.01	0.81	0.01	0.78	0.02
	2.5	0.51	0.02	0.81	0.01	0.75	0.02

D.2. Additional results on PolyMNIST

In Figure 8, we present samples generated by conditioning on a subset of two modalities and in Figure 9, we present samples generated from the prior (unconditional generation). Our models produce diverse and coherent images, while the MoPoE and MMVAE models produce images that look "averaged" from using a mixture based aggregation (Daunhawer et al. 2022).

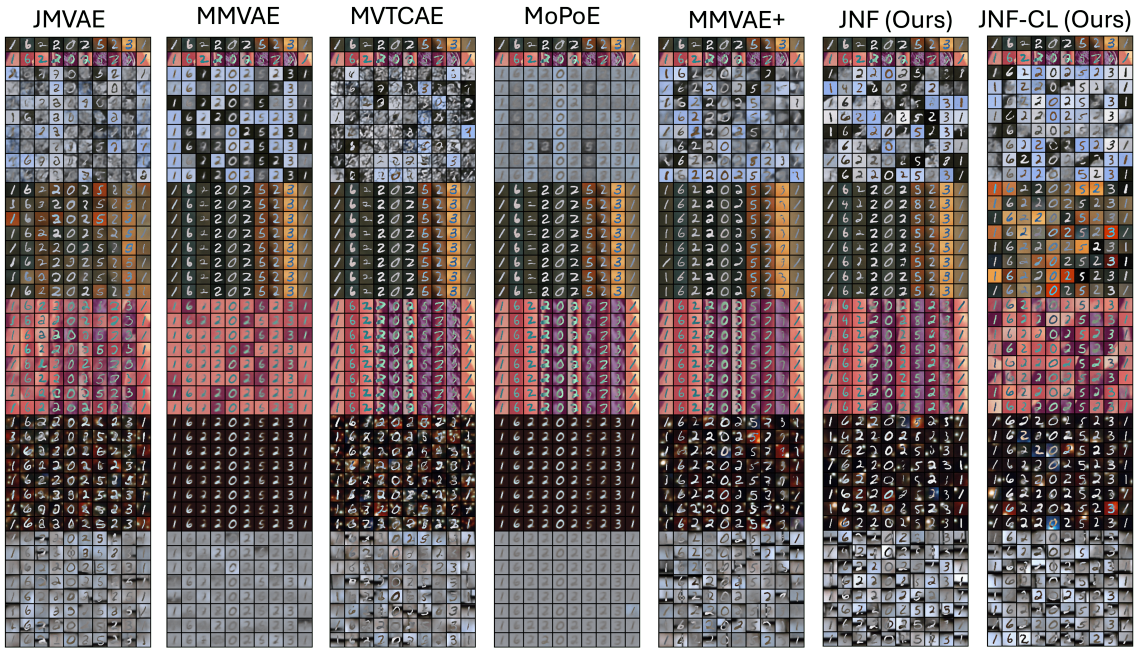
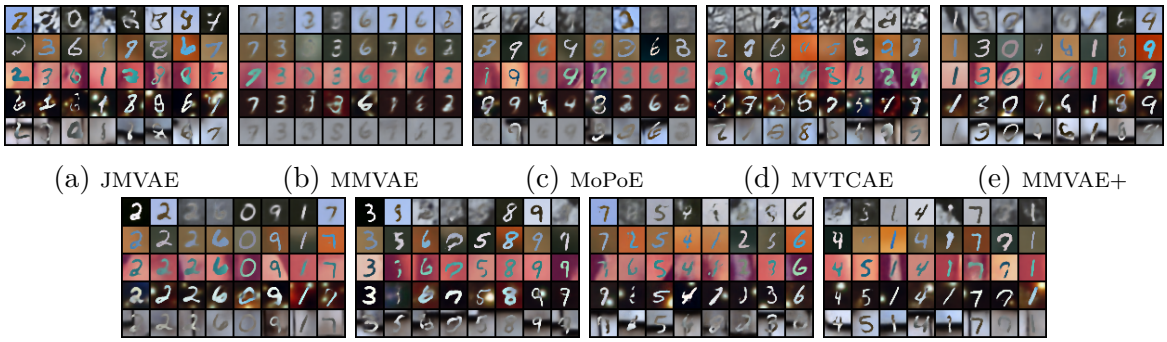


Figure 8: We present generated samples when conditioning on the first two modalities. The first two rows are the samples we condition on and the rest of the rows are generated samples in each modality.

In Figure 9, we present multimodal samples generated from the prior distribution for each model. We see that our models reach a good visual quality with most digits being readable and coherent.

D.3. Additional results on Translated PolyMNIST

Figure 10 shows examples of generated images on TranslatedPolyMNIST and in Figure 11, we present samples generated from the prior. MMVAE and MoPoE reach a high joint coherence on this dataset but if we look at the generated images, we realize the generated images all look averaged, resembling a small "1" digit. The FID is very high since the generation is not diverse.



(a) JMVAE (b) MMVAE (c) MoPoE (d) MVTCAE (e) MMVAE+

(f) CMVAE (g) JNF (h) JNF-CL (i) JNF-DCCA

Figure 9: Joint generation in all five modalities when sampling a latent code from the prior. In each image, each row corresponds to a modality. JNF-CL (resp. DCCA) correspond to our method JNF-Shared with CL (resp. DCCA).

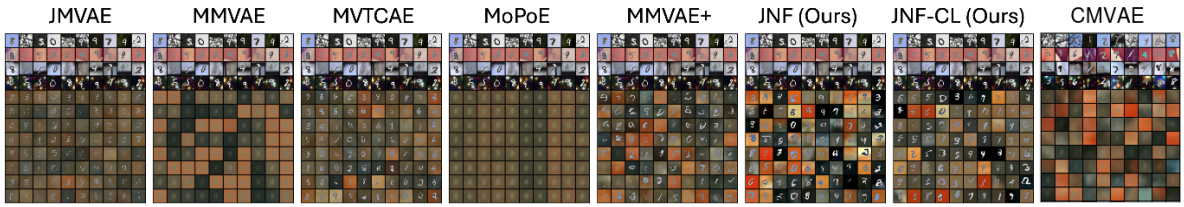
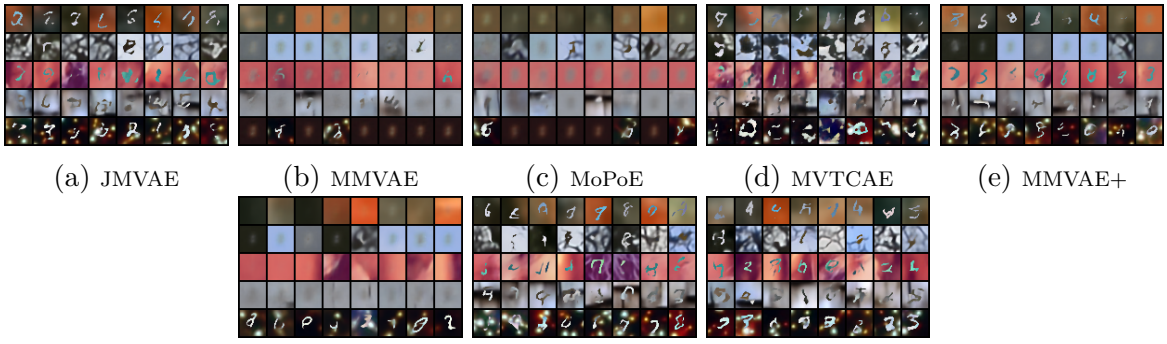


Figure 10: Conditional generation on Translated PolyMNIST. The first four rows are the images we condition on and the newt rows are generated samples in the first modality. JNF-CL refers to our model JNF-Shared with CL.



(a) JMVAE (b) MMVAE (c) MoPoE (d) MVTCAE (e) MMVAE+

(f) CMVAE (g) JNF(Ours) (h) JNF-CL (Ours)

Figure 11: Unconditional generation on Translated PolyMNIST when sampling a latent code from the prior.

In Table 5, we present coherences and FID results for different values of the parameter β for each model. We used this table for selecting the value of β. For all models we observe inverse tendencies between joint and conditional coherence with the value of β. We chose to favor conditional coherence to select the best value of β for each model for the results presented in Table 4. In this table, we also test two values for the number of flows $n_{flows} \in \{2, 3\}$ for our models. *After* selecting β, we varied and selected the optimal parameter n_{flows}.

Table 5: Coherences and FID results for different values of β. Here, we average over all possible subsets for the conditional coherence. For almost all models, we observe inverse tendencies for joint and conditional coherence with the value of β. For the results presented in the main text, we chose to favor conditional generation to select the value of β for each model. The chosen β is set in bold. For the JNF-Shared (DCCA) we used the same $\beta = 0.5$ and 2 flows as for JNF-Shared (CL) because of the similarity between models.

			Coherence (↑)		FID (↓)
			Joint	Conditional	1 modality to m_0
Model	β	n_{flows}			
JMVAE	**0.5**		0.00	0.15	37.06
	1.0		0.00	0.14	43.93
	2.5		0.00	0.12	55.09
MMVAE+	**0.5**		0.006	0.10	60.48
	1		0.005	0.10	69.80
	2.5		0.15	0.10	206.13
MVTCAE	**0.5**		0.004	0.13	42.35
	1		0.08	0.11	121.86
	2.5		0.23	0.11	178.49
MMVAE	**0.5**		0.63	0.10	185.97
	1.0		0.49	0.10	172.44
	2.5		0.58	0.10	181.08
MoPoE	0.5		0.26	0.10	195.53
	1.0		0.50	0.10	199.48
	2.5		0.50	0.10	199.94
CMVAE	0.5		0.42	0.10	159.62
	1.0		0.05	0.10	125.78
	2.5		0.24	0.10	176.98
JNF (Ours)	**0.5**	3	0.0004	0.17	30.91
	0.5	**2**	0.0002	0.18	31.76
	1.0	3	0.0007	0.17	33.82
	2.5	3	0.06	0.12	218.75
JNF-Shared (CL) (Ours)	**0.5**	3	0.0002	0.21	32.09
	0.5	**2**	0.0005	0.23	33.17
	1	3	0.0008	0.20	35.30
	2.5	3	0.06	0.13	217.02

D.4. Additional results on the MHD dataset

In Figure 12, we display images and spectrograms obtained when conditioning on a given trajectory (that is not displayed here) drawing a zero digit. Our models generate diverse and constrasted images.

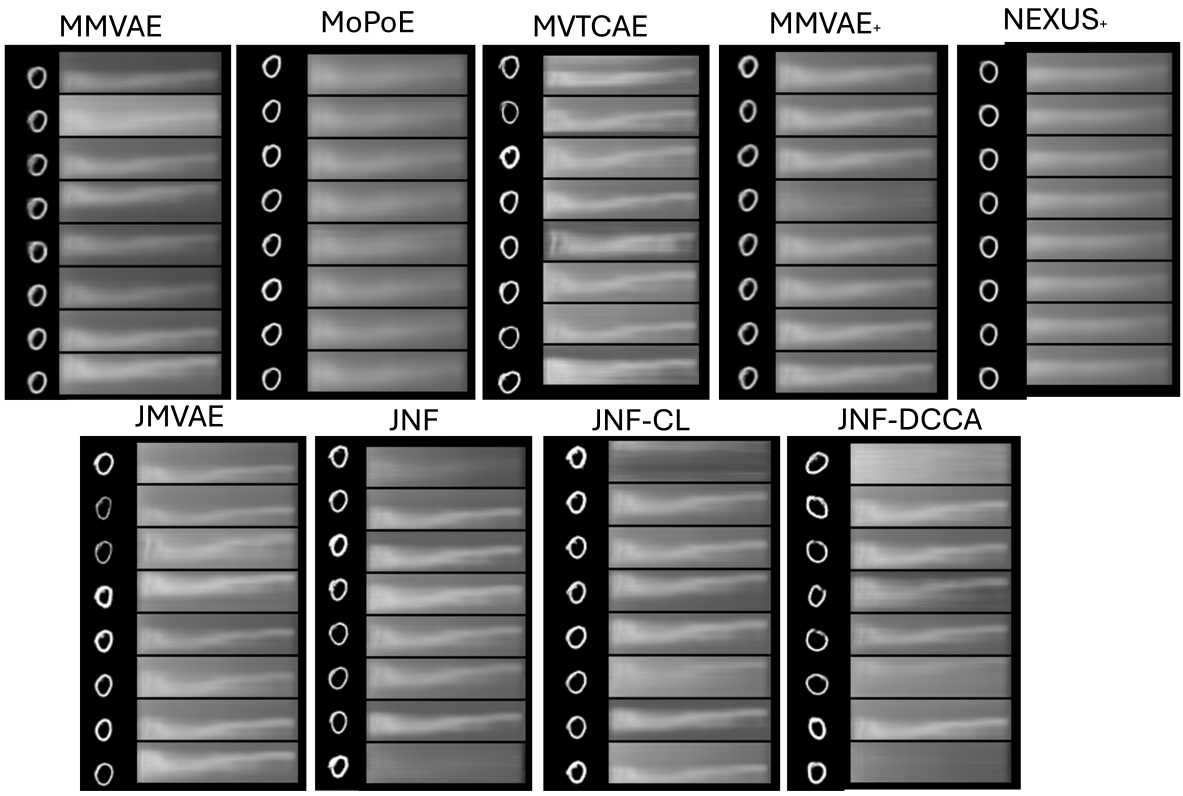


Figure 12: Samples generated when conditioning on a given trajectory.

In Table 6, we present all coherence results for different values of the parameter β for each model. We used this table to chose β for each model. For all models we observe inverse tendencies between joint and conditional coherence with the value of β. We chose to favor conditional coherence to select β for each model for the results presented in Table 2.

D.5. Improving the joint coherence of multimodal VAEs with a posteriori sampling

In this appendix, we apply the method described in Section (4.5) to all models trained on the MHD dataset. Table 7 show that this technique can be used to improve the joint coherence of almost all models.

D.6. Comparison of training times

In Table 8, we present a comparison between the different methods regarding the total number of trainable parameters and training times. The training times presented here are all obtained on GPUs Quadro RTX 6000.

Regarding the number of parameters. Because our approach models both the joint and individual posteriors using separate neural networks, it results in a higher parameter count compared to aggregated models. For JNF, the total number of parameters is approximately twice that of the aggregated baselines when using the same latent dimension and architectures. In the case of JNF-Shared, the parameter count

Table 6: All coherence results for different values of β for each model. We indicate in bold, the value of β that maximises conditional coherence for each model and that we kept for Table 2. For the JNF-Shared (CL) we use the same β as for JNF-Shared (DCCA) since the model are very similar.

		Joint		Conditional	
		mean	std	mean	std
Model	β				
JMVAE	**0.5**	0.57	0.02	0.86	0.01
	1.0	0.15	0.02	0.79	0.01
	2.5	0.15	0.04	0.75	0.02
MMVAE	**0.5**	0.63	0.01	0.86	0.01
	1.0	0.60	0.07	0.84	0.04
	2.5	0.65	0.02	0.86	0.01
MMVAEPlus	**0.5**	0.58	0.03	0.89	0.02
	1.0	0.64	0.05	0.82	0.03
	2.5	0.47	0.15	0.50	0.08
MVTCAE	**0.5**	0.38	0.01	0.87	0.01
	1.0	0.48	0.01	0.85	0.00
	2.5	0.54	0.02	0.79	0.01
MoPoE	**0.5**	0.44	0.02	0.74	0.01
	1.0	0.50	0.01	0.72	0.02
	2.5	0.45	0.02	0.62	0.01
CMVAE	0.5	0.85	0.01	0.925	0.0031
	1.0	0.89	0.01	0.93	0.02
	2.5	0.77	0.03	0.83	0.01
JNF	**0.5**	0.67	0.01	0.89	0.01
	1.0	0.71	0.02	0.86	0.01
	2.5	0.72	0.02	0.81	0.01
JNF-Shared (DCCA)	**0.5**	0.66	0.01	0.92	0.01
	1.0	0.71	0.02	0.90	0.01
	2.5	0.72	0.02	0.80	0.01
JNF-Shared (CL)	**0.5**	0.65	0.02	0.93	0.01

depends on the architectural complexity of the projector networks $(g_i)_i$. On PolyMNIST, we use lightweight convolutional architectures for these projectors, resulting in a parameter count that remains comparable to MVTCAE and MoPoE.

In both variants, the number of parameters scales linearly with the number of modalities—unlike earlier joint models such as JMVAE Suzuki et al. (2016) and TELBO Vedantam et al. (2018), where this scalability was not preserved. Note that on PolyMNIST, we adapted the JMVAE model using the same technique described in Section (3.3) for modelling the subset posteriors. Otherwise the JMVAE model would

Table 7: Unconditional coherence on the MHD dataset when sampling from the prior distribution and when sampling from a GM (10 components) fitted after training. We see that the GM sampling greatly improve unconditional coherence for almost all models. Note the CMVAE already has a GM as prior distribution.

Model	**Joint Coherence Prior**	**Joint Coherence GM**
MMVAE	0.63 ± 0.01	0.63 ± 0.02
MoPoE	0.44 ± 0.02	0.84 ± 0.01
MMVAEPlus	0.57 ± 0.01	0.80 ± 0.02
MVTCAE	0.38 ± 0.01	0.91 ± 0.02
CMVAE	0.89 ± 0.02	
JNF (Ours)	0.67 ± 0.01	0.81 ± 0.02
JNF-Shared (CL) (Ours)	0.65 ± 0.02	0.82 ± 0.02

Table 8: Comparison of number of parameters (P) and training times. t_1 refers to the total training time of one run on a GPU Quadro RTX 6000. t_2 refers to the training time until the best model is reached on this run. We report the results on only one seed here.

	MNIST-SVHN			PolyMNIST		
Model	P	t_1 (min)	t_2 (min)	P	t_1 (min)	t_2 (min)
MMVAE	1,106,987	151	151	18,805,903	1,135	900
JMVAE	1,988,715	113	113	70,645,923	427	175
MoPoE	1,106,947	128	76	42,515,195	593	150
MVTCAE	1,106,947	112	112	42,515,195	508	93
MMVAE+	1,586,627	168	162	22,156,943	1,140	1,020
CMVAE	1,586,817	231	215	22,157,529	1,260	1,140
JNF	2,152,203	132	120	71,710,843	375	226
JNF-Shared (DCCA)	2,574,733	175	172	47,511,903	384	320
JNF-Shared (CL)	2,574,733	195	195	48,919,429	400	337

not scale to this dataset of 5 modalities as it would require instantiating $2^5 = 32$ different encoder architectures (one for each possible subset) i.e $\approx$ 700,000,000 trainable parameters.

Regarding training times. On MNIST-SVHN, the training times of all models are rather similar. The MoPoE and MVTCAE are the fastest models to train. On PolyMNIST, the MMVAE, MMVAE+ and CMVAE become much longer to train. We hypothesize that this comes from the loss formulation which implies $m = 5$ different reconstructions for each modality so 25 reconstructions per datapoint where other models only compute one reconstruction per modality i.e 5 reconstructions per datapoint. We note that, while our method uses a multi-stage training, the training times remain similar to those of other methods and shorter than those of MMVAE, MMVAE+ and

CMVAE.

D.7. Ablation studies

Without normalizing flows

In this ablation study, we look at the impact of using normalizing flows for the encoder distributions $(q_{\phi_j}(z|x_j))_j$ and $(q_{\phi_j}(z|g_j(x_j)))_j$ instead of simple diagonal Gaussian distributions.

In Table 9, we present results on the MNIST-SVHN dataset, with and without using normalizing flows. We see that using normalizing flows increase the coherence of both the JNF and JNF-Shared models. For the JNF-Shared model, using normalizing flows results in a gain of 12–13 % in conditional coherence.

Table 9: Ablation study on our method on the MNIST-SVHN dataset. β is set to 2.5.

Model	Joint	$M \to S$	$S \to M$	FID
JNF - no flows	52 ± 1	78 ± 1	46 ± 1	52 ± 1
JNF- flows	51 ± 1	82 ± 1	52 ± 2	54 ± 2
JNF-Shared (CL)- no flows	51 ± 1	68 ± 2	63 ± 1	54 ± 1
JNF-Shared (CL)-flows	51 ± 1	81 ± 1	75 ± 1	49 ± 1

JNF-Shared with Independents Projectors

In the JNF-Shared method that we propose, the projectors $(g_i)_{1 \leq i \leq M}$ are jointly trained with a multimodal contrastive approach or a DCCA approach to extract the information shared accross modalities.

In this ablation study, we train each modality projector $(g_i)_i$ independently of the others with a SimCLR (Chen et al. 2020) method to extract compact (but uncorrelated) representations of each modality. We choose SimCLR as it is also a contrastive learning method with a similar objective as the one we use in our multimodal contrastive learning approach. The main difference lies in how the positive pairs are defined: in multimodal contrastive learning, two paired modalities define a positive pair, whereas in SimCLR, two random augmentations of a single modality define a positive pair.

For the architectures, we consider two configurations:

1. We use the same architectures as the one used in our method with multimodal contrastive learning. The MNIST projector is a two-layer Linear network, whereas the SVHN projector is a simple convolutional network. See the detailed architectures in Table 11.

2. To obtain better representations for the SVHN modality, we replace the convolutional network with a resnet18 network as is done in the original SimCLR paper (Chen et al. 2020).

The networks are trained for 200 epochs with a learning rate of 0.003 and a cosine scheduler following the guidelines of Chen et al. (2020).

Table 10: Coherence results for the JNF-Shared method with independent projectors $(g_i)_i$ versus with jointly trained projectors. The architecture column specifies whether we are using simple convolution or resnet architectures for the SVHN projector.

	$M \to S$	$S \to M$	*Joint*	*FID*
Architecture				
1. (conv)	0.76 ± 0.01	0.26 ± 0.01	0.52 ± 0.02	57 ± 2
2. (resnet)	0.74 ± 0.01	0.36 ± 0.02	0.5 ± 0.01	57 ± 2
JNF-Shared (CL)	$\mathbf{0.81 \pm 0.01}$	$\mathbf{0.75 \pm 0.02}$	$\mathbf{0.51 \pm 0.02}$	$\mathbf{49 \pm 1}$

We present the results in Table 10. We see that the method using projectors independently trained do not reach as good performances as JNF-Shared (CL) using jointly pre-trained projectors. This confirms the intuition that using pretrained projectors is beneficial not only because they extract compact representations of the modalities but because they extract compact representations of the *shared information* only.

E. Architectures and Hyperparameters Used in the Experiments

In Figure 17, we summarize the general architectures of most models used in our experiments. For the Nexus model, we refer the reader to (Vasco et al. 2022).

We describe in the following sections the encoders/decoders architectures for all experiments. Note that for the JNF-Shared, the projectors $(g_j)_{1\leq j\leq M}$ have the same architectures as the encoders of other models, and the encoders that parameterize $q_{\phi_j}(z|g_j(x_j))$ are simple two-layers MLPs taking the projections $(g_j)_{1\leq j\leq M}(x_j)$ as inputs.

Our implementations of normalizing flows rely on the opensource library **Pythae** (Chadebec et al. 2022).

Code and data needed for reproducing the experiments are available at `https://anonymous.4open.science/r/JNF_VAE/README.md`.

E.1. On MNIST-SVHN

In Table 11, we indicate all architectures and training parameters used in the MNIST-SVHN experiments. All models are trained until convergence. For all models, we test three values for $\beta \in \{0.5, 1.0, 2.5\}$ and for each model we kept the value that maximized average coherence (joint and conditional). Extensive results for all values of β are presented in Table 4. For the MMVAE and MMVAE+ model, we use Laplace distributions for modeling prior and posterior distribution following (Shi et al. 2019). For all others models, we use Gaussian distributions for prior and posteriors. For the decoders distributions $p_\theta(X|z)$ we use Laplace distributions. Following previous work

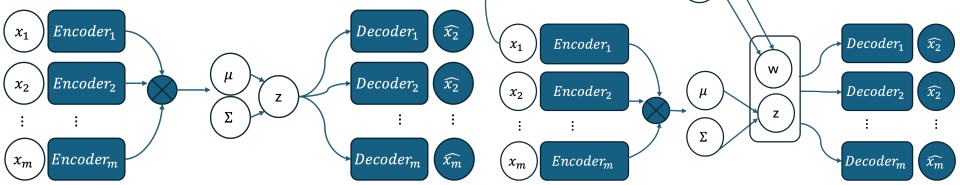


Figure 13: Architectures of MM-VAE, MoPoE and MVTCAE

Figure 14: Architecture of MM-VAE+

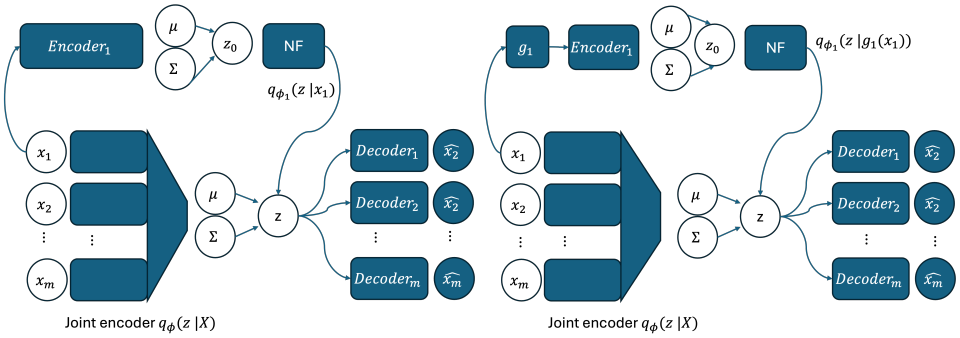


Figure 15: Architecture of JNF and JMVAE model.

Figure 16: Architecture of JNF-Shared

Figure 17: Architectures for most models used. For the JMVAE model, it is the same architecture as the JNF model except without the normalizing flows.

(Shi et al. 2019; Sutter et al. 2021) we rescale the likelihoods of each modality with factors $\lambda_{MNIST}, \lambda_{SVHN}$ in order to compensate for the different sizes of the modalities and mitigate conflictual gradients (Javaloy et al. 2022).The values for $\lambda_{MNIST}, \lambda_{SVHN}$ are indicated in Table 11. Intuitively, we need to put more weight on the smaller modalities so that they are also well reconstructed.

We give specific details for each model:

- MVTCAE: we set $\alpha = 0.9$ following their recommandations in the supplemental material in (Hwang et al. 2021).
- MMVAE: we set K=10 for the number of samples in the ELBO.
- MMVAE+: we set K=10 for the number of samples in the ELBO. The shared latent space as well as the modality-specific latent spaces have a dimension of 10.
- JMVAE model, we set $\alpha = 0.1$ as it appears as a good compromise value in (Suzuki et al. 2016). We use annealing as in the original paper with a 100 epochs for warmup. The joint encoder is made up of separate heads and a common merging part where the separate heads have the same architecture as the unimodal

encoders in Table 11. The merging part is a simple two-layer MLP with 512 neurons in each layer.

- JNF: we used Masked Autoregressive Flows with two MADE blocks (Papamakarios et al. 2017). We use the same joint encoder as for the JMVAE model.
- JNF-Shared: We use the same flows and joint encoder as JNF. The projectors used for CL or DCCA have the same architectures as the encoders in Table 11. The encoders $q_{\phi_j}(z|g_j(x_j))$ are simple networks with two linear hidden layers.

E.2. On PolyMNIST

For the PolyMNIST experiments, we used the same Resnet (He et al. 2015) architectures as used in (Palumbo et al. 2023). These architectures are summarized in Figure 18. Following (Palumbo et al. 2023), we train all models as β-VAE and set $\beta = 2.5$. Each model is trained until convergence with a batchsize of 128 and learning rate of 1e-3. The latent dimension is set to 190 to match the total capacity of the MMVAE+ model in (Palumbo et al. 2023).

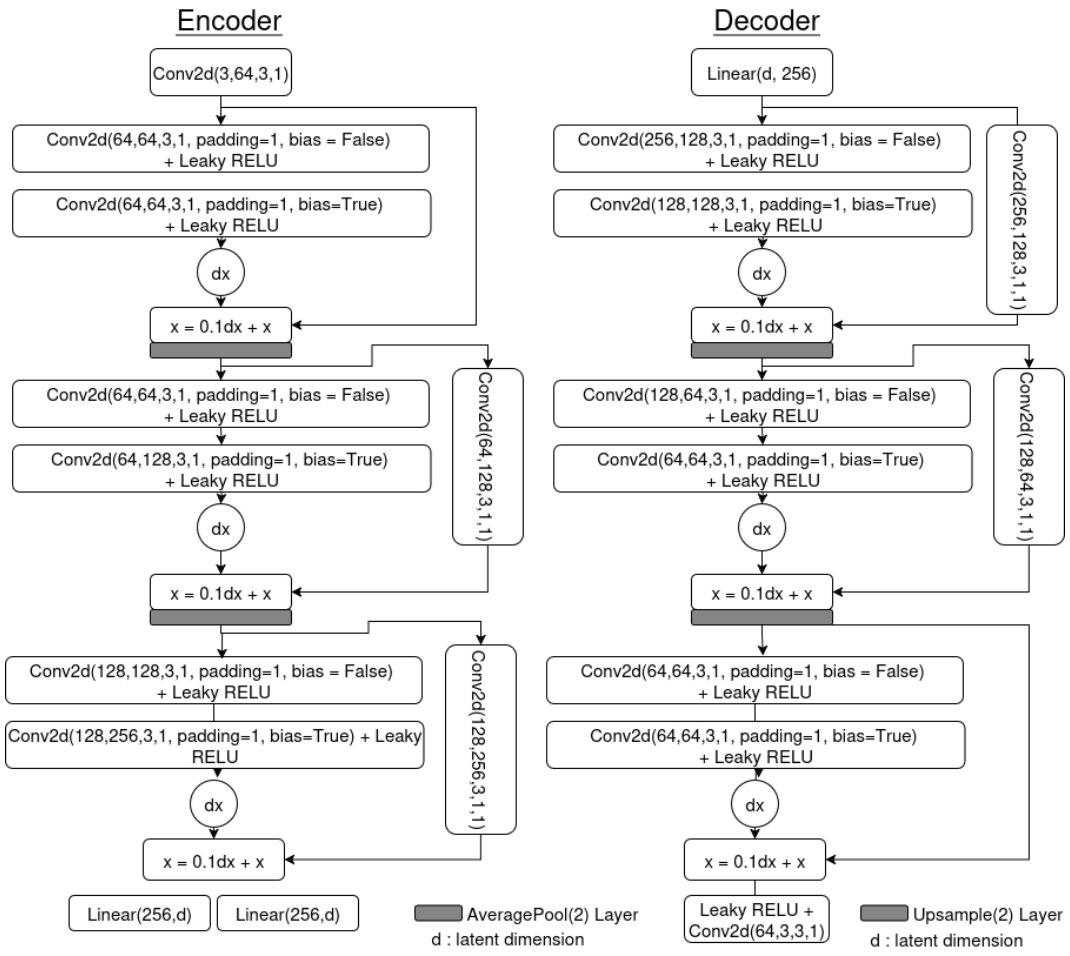


Figure 18: Encoder and decoder architectures used for the experiments on the PolyMNIST dataset.

We give specific details for each model:

- MMVAE: Due to memory limitations, we set the latent dim to 64 and used K=10 for the number of samples in the ELBO.
- MVTCAE: we set $\alpha = 5/6$.
- JMVAE: we set $\alpha = 0.1$ and annealing with a warmup of 100 epochs. In the original JMVAE model, a new encoder network needs to be introduced for each subset of modalities. In our experiments, we didn't choose that solution since it represents a very large number of parameters. Instead, we use for the JMVAE model, the PoE sampling solution that we also use for our models (Equation (11)). The joint encoder is made-up of separate heads with the same architectures as in Figure 18 and a merging neural networks with two hidden linear layers of 512 neurons.
- MMVAE+: We use 32 dimensions for the shared latent space and 32 dimension for each modality specific space as in (Palumbo et al. 2023).
- JNF: Same joint encoder as JMVAE. We use Masked Autoregressive flows with 2 MADE blocks.
- JNF-Shared: Same joint encoder and Normalizing flows as JNF. The projectors (g_j) are simple convolutional networks similar to the SVHN encoders in 11 and the encoders $q_{\phi_j}(z|g_j(x_j))$ are simple linear encoders as for the MNIST-SVHN experiments: see Table 11.

E.3. On Translated PolyMNIST

For the TranslatedPolyMNIST experiments, we used similar architectures as in the PolyMNIST experiments with a latent dimension of 200 (except for MMVAE and MMVAE+ whose parameters are specified below). We performed experiments with $\beta \in \{0.5, 1., 2.5\}$. For all models, we kept the value of β that maximized average conditional coherence. In Table 5, we present results for different values of β and the selected values for each model. We use a latent dimension of 200 for all models but the MMVAE+ that has multiple latent spaces (see below). All models are trained until convergence with learning rate 1e-3 and batchsize 128.

We give specific details for each model:

- MMVAE: Due to memory limitations, we used a latent dimension of 100 for the MMVAE model and used K=10 for the number of samples in the ELBO.
- MVTCAE: we set $\alpha = 5/6$ as in PolyMNIST.
- JMVAE: we set $\alpha = 0.1$ and a warmup of 100 epochs. PoE sampling is applied for JMVAE as in the other experiments. The joint encoder is made of separate heads with the same architectures as in Figure 18 and we concatenate the outputs of each head to form the joint representation. This concatenation instead of a merging network allows to avoid conflictual gradient issues and modality collapse (Javaloy et al. 2022).

Table 11: Architectures and training parameters used for the Mnist-SVHN Experiments.

Mnist Encoder	Mnist Decoder
Linear($1 \times 28 \times 28$,400) RELU	Linear(20, 400) RELU
Linear(400,20), Linear(400,20)	Linear(400, $1 \times 28 \times 28$), Sigmoid
SVHN Encoder	**SVHN Decoder**
Conv2d(3,32,4,2,1,bias=True), RELU	Conv2dTranspose(d,128,4,4,1,0,bias=True), RELU
Conv2d(32,64,4,2,1,bias=True), RELU	Conv2dTranspose(128,64,4,4,1,0,bias=True), RELU
Conv2d(64,128,4,2,1,bias=True), RELU	Conv2dTranspose(64,32,4,4,1,0,bias=True), RELU
Conv2d(128,20,4,1,0,bias=True) $\times 2$	Conv2dTranspose(32,3,4,4,1,0,bias=True), Sigmoid
Training parameters	**Joint Encoder for JMVAE, JNF, JNF-Shared**
Batchsize = 128	Separates head for each modality
Learning rate = 1e-3	Linear(512), RELU
Optimizer = Adam	Linear(512), RELU
Latent dimension = 20	Linear(20), Linear(20)
$\lambda_{MNIST}, \lambda_{SVHN} = \frac{3\times32\times32}{28\times28}, 1$	
Dimension for the projection $(g_j) = 10$	
Normaling Flows	**JNF-Shared encoders for $q_{\phi_j}(z\|g_j(x_j))$**
Masked Autoregressive with two MADE blocks	Linear(10,512) RELU
	Linear(512,512) RELU
	Linear(512,20), Linear(512,20)

- MMVAE+: We use 32 dimensions for the shared latent space and 32 dimension for each modality specific space as in (Palumbo et al. 2023). We use K=10 for the number of samples in the ELBO.
- JNF: Same joint encoder as JMVAE. We use Masked Autoregressive flows with 3 MADE blocks.
- JNF-Shared (CL): Same joint encoder and Normalizing flows as JNF. The projectors (g_j) have the encoder architectures in Figure 18 and the encoders $q_{\phi_j}(z|g_j(x_j))$ are simple linear networks as for the MNIST-SVHN experiments: see Table 11. We use Masked Autoregressive flows with 2 MADE blocks.
- JNF-Shared (DCCA): when using DCCA to extract the shared information, we used more simple architectures for the projectors (g_j) for instability reasons. We used simple convolutional networks similar to the SVHN encoders in Table 11. Precise architectures are given in the code. We use Masked Autoregressive flows with 2 MADE blocks.

E.4. On MHD

Table 12 contains all relevant architectures and general training parameters.

We use the same architectures than the ones used in (Vasco et al. 2022) except that we don't pretrain the sound encoder and decoder. All models with a β term weighing the Kullback-Leibler divergence in (3) and for all models, the $\beta = 0.5$ gives the best average conditional coherence. We present additional results for all values of β in Table 6. We used Gaussian distributions to model all posterior, prior and decoding distributions. We use a latent dimension of 64 for all models but the MMVAE+ that has multiple latent spaces (see below).

We use rescaling for the likelihoods of each modality following (Shi et al. 2019). It has been shown that this limits the phenomenons of conflictual gradients and modality collapse (Javaloy et al. 2022). The rescaling factors $\lambda_{image}, \lambda_{audio}, \lambda_{trajectory}$ are given in Table 12

We train all models until convergence. We give specific details for each model:

- MMVAE: We used K=10 for the number of samples in the ELBO.
- MVTCAE: we tried $\alpha \in \{0.75, 0.9\}$ and kept best results obtained for $\alpha = 0.9$.
- JMVAE: we set $\alpha = 0.1$ and a warmup of 100 epochs. In the original JMVAE model, a new encoder network needs to be introduced for each subset of modalities. In our experiments, we didn't choose that solution since it represents a very large number of parameters. Instead, we use for the JMVAE model, the PoE sampling solution that we also use for our models (Equation (11)). The joint encoder is made-up of separate heads with the same architectures as in Table 12 and a merging neural networks with two hidden linear layers of 512 neurons.
- MMVAE+: We use 32 dimensions for the shared latent space and 32 dimension for each modality specific space. We used K=10 for the number of samples in the ELBO.

- JNF: Same joint encoder as JMVAE. We use Masked Autoregressive flows with 2 MADE blocks.
- JNF-Shared: Same joint encoder and Normalizing flows as JNF. The projectors (g_j) have the encoder architectures in Figure 18 and the encoders $q_{\phi_j}(z|g_j(x_j))$ have the same architectures as for the MNIST-SVHN experiments: see Table 11.
- NEXUS : we use the same hyperparameters as used in (Vasco et al. 2022).

F. Hamiltonian Monte Carlo Sampling

In this appendix, we recall the principles of Hamiltonian Monte Carlo Sampling and detail how we apply it in our model. The Hamiltonian Monte Carlo (HMC) sampling belongs to the larger class of Markov Chain Monte Carlo methods (MCMC) that allow to sample from any distribution $f(z)$ known up to a constant (Brooks et al. 2011). The general principle is to build a Markov Chain that will have our target $f(z)$ as stationary distribution. More specifically, the HMC is an instance of the Metropolis-Hasting Algorithm (see 1) that uses a physics-oriented proposal distribution.

Algorithm 1 Metropolis-hasting algorithm.

1: **Initialization :** $z \leftarrow z_0$**.**
2: **for** $i := 0 \rightarrow$ N **do**
3: Sample z' from the proposal $g(z'|z)$.
4: With probability $\alpha(z', z)$ accept the proposal $z \leftarrow z'$.
5: **end for**

Sampling from the proposal distribution $g(z'|z_0)$ is done by integrating the Hamiltonian equations :

$$\begin{cases} \dfrac{\partial z}{\partial t} = \dfrac{\partial H}{\partial v}, \\ \dfrac{\partial v}{\partial t} = -\dfrac{\partial H}{\partial z}, \\ z(0) = z_0 \\ v(0) = v_0 \sim \mathcal{N}(0, I), \end{cases} \tag{32}$$

where the Hamiltonian is defined by $H(z, v) = -\log f(z) + (1/2)v^t v$. In physics, Eq. (32) describes the evolution in time of a physical particle with initial position z and a random initial momentum v. The leap-frog numerical scheme is used to integrate Eq. (32) and is repeated l times with a small integrator step size ϵ :

$$\begin{aligned} v\left(t + \frac{\epsilon}{2}\right) &= v(t) + \frac{\epsilon}{2} \cdot \nabla_z(\log f(z)(t)), \\ z(t + \epsilon) &= z(t) + \epsilon \cdot v\left(t + \frac{\epsilon}{2}\right), \\ v(t + \epsilon) &= v\left(t + \frac{\epsilon}{2}\right) + \frac{\epsilon}{2}\nabla_z \log f(z(t + \epsilon)). \end{aligned} \tag{33}$$

Table 12: Architectures and training parameters used for the MHD Experiments.

Image Encoder	Image Decoder
Conv2d(1,128,4,2), Batchnorm, RELU Conv2d(128,256,4,2), Batchnorm, RELU Conv2d(256,512,4,2), Batchnorm, RELU Conv2d(512,1024,4,2), Batchnorm, RELU Linear(d), Linear(1024,d)	Linear(1024*4*4) ConvTranspose2d(1024,512,3,2,padding = 1), Batchnorm,RELU ConvTranspose2d(512,256,3,2,padding = 1),Batchnorm, RELU ConvTranspose2d(256,1,3,2,padding = 1),Sigmoid
Trajectory Encoder	**Trajectory Decoder**
Linear(512), Batchnorm, LeakyRELU Linear(512), Batchnorm, LeakyRELU Linear(512), Batchnorm, LeakyRELU Linear(d), Linear(d)	Linear(512), Batchnorm, LeakyRELU Linear(512), Batchnorm, LeakyRELU Linear(512), Batchnorm, LeakyRELU Linear(128), Sigmoid
Sound Encoder	**Sound Decoder**
Conv2d(1,128, kernel = (1,128), stride = (1,1)), Batchnorm, RELU Conv2d(128,128, kernel=(4,1), stride = (2,1)), Batchnorm, RELU Conv2d(128,256, kernel=(4,1),stride = (2,1)), Batchnorm, RELU Linear(d), Linear(d)	Linear(2048), Batchnorm, RELU ConvTranspose2d(256, 128, kernel=(4,1), stride=(2,1), padding=(1,0)) ConvTranspose2d(128, 128, kernel=(4,1), stride=(2,1), padding=(1,0)) ConvTranspose2d(128, 1, kernel=(1,128), stride=(1,1), padding=0), Sigmoid
Training parameters	**Joint Encoder**
Batchsize = 64 Learning rate = 1e-3 Optimizer = Adam $\lambda_{image} = \frac{32\times128}{3\times28\times28} \approx 1.7$ $\lambda_{audio} = 1.0$ $\lambda_{trajectory} = \frac{32\times128}{200} = 20.48$	Separates head for each modality with same architectures as encoders Linear(512), RELU Linear(512), RELU Linear(d), Linear(d)
normalizing flows	**JNF-Shared encoders for $q_{\phi_j}(z\|g_j(x_j))$**
Masked Autoregressive with two MADE blocks	Linear(10,512) RELU Linear(512,512) RELU Linear(512,20), Linear(512,20)

After l integration steps, we obtain the proposal position $z' = z(t + l \cdot \epsilon)$ that corresponds to step 3 in Algorithm 1. The acceptance ratio is then defined as $\alpha(z', z_0) = \min(1, (\exp(-H(z_0, v_0)))/(\exp(-H(z', v(t + l \cdot \epsilon)))))$. This procedure is repeated to produce an ergodic Markov chain (z^n) converging to the target distribution f (Liu 2009; Brooks et al. 2011; Girolami and Calderhead 2011). In this work, we use HMC sampling to sample from the PoE of unimodal posteriors in Eq. (11). To do so we need to compute and derivate the (log) of the target distribution given by the PoE of the unimodal distributions:

$$\log q(z|(x_i)_{i\in S}) = -\log p(z) + \sum_{i\in S} \log q_{\phi_i}(z|x_i). \tag{34}$$

We can use autograd to automatically compute the gradient $\nabla_z \log q(z|(x_i)_{i\in S})$ that is needed in the leapfrog steps.

In our experiments, we use 100 steps per sampling.

G. Information on the Classifiers Used for Evaluation

G.1. MNIST-SVHN

In Table 13 we provide the architectures and the accuracies for the classifiers that we use to evaluate coherence on the MNIST-SVHN dataset.

Table 13: Classifiers used for the MNIST-SVHN experiments.

SVHN	**MNIST**
Conv2d(3,10,5)	Conv2d(1,10,5)
MaxPool2d,RELU	MaxPool2d,RELU
Conv2d(10,20,5), Dropout(0.5)	Conv2d(10,20,5), Dropout(0.5)
MaxPool2d,RELU	MaxPool2d,RELU
Linear(500,50), RELU, Dropout	Linear(350,50), RELU, Dropout
Linear(50,10), Softmax	Linear(50,10), Softmax
Accuracies on test	
0.87	0.99

G.2. Classifiers on PolyMNIST

We use the architectures and the pretrained models available at `https://github.com/thomassutter/MoPoE` (Sutter et al. 2021).

The accuracies of the classifiers for the five modalities of the test set are respectively: 0.95, 0.99, 0.99, 0.97, 0.95.

G.3. Classifiers on Translated PolyMNIST

We pretrain classifiers on this dataset having similar architectures as in Figure 18 with a output size of 10.

The accuracies of the trained classifiers for the five modalities of the test set are respectively: 0.98, 0.97, 0.98, 0.97, 0.98.

G.4. Classifiers on MHD

We use the pretrained classifiers available at https://github.com/miguelsvasco/nexus_pytorch/.

The accuracies of the trained classifiers on the test set are: 0.95 for the audio modality, 0.99 for the image modality and 0.99 for the trajectory modality.

Affiliation:

Agathe Senellart
UMR1346, Université de Paris Cité, Inria Paris, Inserm
HeKA team
PariSanté Campus
3ème étage, aile ouest
2-10 rue d'Oradour-sur-Glane
75015 Paris, France
E-mail: agathe.senellart@inria.fr

Stéphanie Allassonnière
UMR1346, Université de Paris Cité, Inria Paris, Inserm
HeKA team
PariSanté Campus
3ème étage, aile ouest
2-10 rue d'Oradour-sur-Glane
75015 Paris, France
E-mail: stephanie.allassonniere@u-paris.fr

Journal of Data Science, Statistics, and Visualisation https://jdssv.org/
published by the International Association for Statistical Computing
http://iasc-isi.org/

ISSN 2773-0689
November 2025, Volume V, Issue IX *Submitted:* 2025-07-10
doi:10.52933/jdssv.v5i9.160 *Accepted:* 2025-10-27