Foundations of Data Science
Vol. 7, No. 1, March 2025, pp. 418-445
doi:10.3934/fods.2024019

UNSUPERVISED PHYSICS-INFORMED DISENTANGLEMENT OF MULTIMODAL DATA

Elise Walker[1], Nathaniel Trask[*2], Carianne Martinez[3,4], Kookjin Lee[4], Jonas A. Actor[1], Sourav Saha[5], Troy Shilt[6], Daniel Vizoso[7], Remi Dingreville[7] and Brad L. Boyce[7]

[1]Center for Computing Research, Sandia National Laboratories, Albuquerque, USA

[2]School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, USA

[3]Applied Information Sciences Center, Sandia National Laboratories, Albuquerque, USA

[4]School of Computing and Augmented Intelligence, Arizona State University, Tempe, USA

[5]Theoretical and Applied Mechanics, Northwestern University, Evanston, USA

[6]Center for Engineering Sciences, Sandia National Laboratories, Albuquerque, USA

[7]Center for Integrated Nanotechnologies, Sandia National Laboratories, Albuquerque, USA

Abstract. We introduce physics-informed multimodal autoencoders (PIMA) - a variational inference framework for discovering shared information in multimodal datasets. Individual modalities are embedded into a shared latent space and fused through a product-of-experts formulation, enabling a Gaussian mixture prior to identify shared features. Sampling from clusters allows cross-modal generative modeling, with a mixture-of-experts decoder that imposes inductive biases from prior scientific knowledge and thereby imparts structured disentanglement of the latent space. This approach enables cross-modal inference and the discovery of features in high-dimensional heterogeneous datasets. Consequently, this approach provides a means to discover fingerprints in multimodal scientific datasets and to avoid traditional bottlenecks related to high-fidelity measurement and characterization of scientific datasets.

1. **Introduction.** Many scientific and engineering datasets are multimodal, necessitating the fusion of disparate sources and datatypes for informed analysis. For example, cardiovascular disease research may consider multiple modalities, such as medical records, genetics, and radiology images [1], and materials science research may involve a myriad of process settings along with in-process and post-process measurements [32, 48]. Moreover, automated high-throughput characterization methods across various scientific domains are increasingly generating large, rich multimodal datasets, fueled by advances in robotics and automation [37, 5]. Many of these scientific datasets admit *fingerprints*: easily measurable signals which correlate with a difficult to measure underlying physical process. The hunt for exploitable fingerprints spans many scientific domains, including material science [19], quantum

2020 *Mathematics Subject Classification.* Primary: 62P35; Secondary: 68T07, 68T99.

Key words and phrases. Multimodal machine learning, physics-informed machine learning, variational inference, variational autoencoders, fingerprinting, product-of-experts, mixture-of-experts.

*Corresponding author: Nathaniel Trask.

mechanics [7], and climate change [15, 16]. Rapid datasets designed to detect fingerprints may potentially serve as a surrogate for, or in conjunction with, bespoke experiments capturing high-fidelity modalities. Accordingly, we aim to discover comprehensive fingerprints constructed from the weighted integration of multimodal datasets. Due to their size and complexity, parsing these datasets for exploitable fingerprints requires methods capable of representing high-dimensional, heterogeneous scientific data in a human-interpretable way [3, 5, 15, 39].

In representation learning, one seeks to discover interpretable, lower-dimensional representations of high-dimensional datasets. Finding these interpretable representations can be challenging, especially in an unsupervised setting where human-in-the-loop data labeling may render high-throughput processing intractable. For unsupervised learning, several works apply variational autoencoders (VAE) to seek latent *disentangled representations* of data which admit efficient separation into meaningful classes [6, 8, 23, 33]. While desirable from an interpretability and accuracy perspective, such representations are often challenging to reliably discover in the absence of labels [34]. However, the complementary information available in multimodal data has been shown to provide multiple pathways to disentanglement; for example, a human may be unable to distinguish an image of a one and a seven, but if the digit is read aloud there is no confusion. Multimodal datasets are often the subject of representation learning, where the modalities commonly considered in the literature are text/audio/video modalities [3]. Unfortunately, this treatment on text/audio/video modalities is unsatisfactory for the rapidly growing body of scientific and engineering datasets whose modalities include both physical and simulated data across several disparate data sources, each with unique fidelity, sparsity, and spatiotemporal resolution. In particular, these *multimodal scientific datasets* often come with exploitable physics-based inductive biases, which can provide the opportunity to move beyond purely data-driven techniques and constrain our models to physically consistent and interpretable representations. Indeed, the fact that scientific data is governed by physical models potentially allows an expert model to extract more information than purely data-driven encodings - i.e. known physics encodes the generative process, and therefore imposing even a low-fidelity physical model as an inductive bias may provide substantial disentanglement.

Accordingly, we construct an unsupervised, physics-amenable algorithmic framework that learns a joint lower-dimensional representation for multimodal scientific datasets with the ability to uncover hidden, underlying factors important for informed data analysis. Furthermore, our algorithmic framework enables *cross-modal inference*. Concretely, cross-modal inference corresponds to training jointly across modalities $X_1, ..., X_M$ in a manner that supports generative sampling of individual modalities, i.e. $p(X_i|X_j)$ for $i \neq j$. We achieve cross-modal inference across multimodal representations in a variational setting by combining the following algorithmic contributions (Figure 1): **1.** encoding data into unimodal embeddings $q(Z_m|X_m)$ and applying a product-of-experts (PoE) model to fuse data into a multimodal posterior $q(Z|X_1, \ldots, X_M) = \Pi_m q(Z_m|X_m)$; **2.** adopting a Gaussian mixture model (GMM) prior to determine latent clusters C shared across modalities; and **3.** decoding with a physics-informed mixture-of-experts (MoE) model $p(X_m|C, Z)$ to impose inductive biases. For scientific settings, the expert model provides a critical new means of fusing experimental audiovisual data with traditional scientific models; rather than considering generalized linear models commonly used in MoE [22], we may incorporate parameterized physical models, surrogates or simulators for the

system under consideration. These ingredients are designed to yield an evidence lower bound (ELBO) loss with closed form expressions for requisite integrals that is amenable to a novel optimization strategy similar to expectation-maximization to fit clusters and experts. In concert, this architecture produces fingerprints in the form of latent clusters spanning modalities, with cross-modal estimators allowing inference of cluster membership for a single modality. Our tests demonstrate that the combination of PoE, GMM, and expert models provides disentangled clusters whose positioning in the latent space reflects information shared across modalities. We anticipate that these physics-based disentangled representations of large, high-throughput datasets will uncover hidden information exploitable for downstream tasks in any number of fields, such as fingerprint detection in climate change data [15], and materials discovery [5].

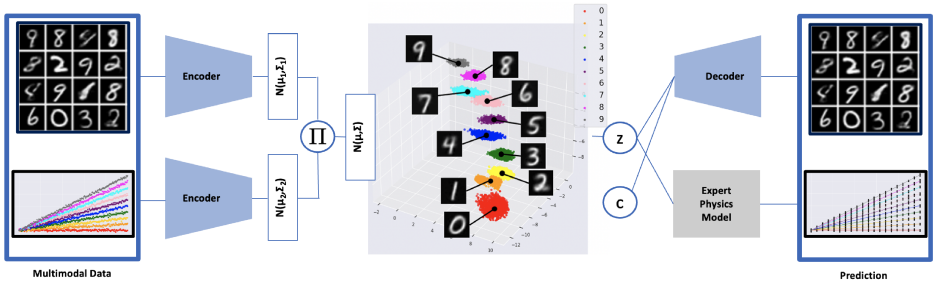


FIGURE 1. Individual modalities are encoded into Gaussian distributions in a shared latent space. During training, the posterior is sampled from a product-of-experts distribution fusing complementary information into a shared multimodal Gaussian distribution. A Gaussian mixture prior parameterizes clusters encoding cross-modal shared information. Sampling from mixture components provides generative models using either black-box decoders or expert physics models incorporating prior physics knowledge. To facilitate cross-modal generative inference, a random selection of modalities is used on each epoch of training to encourage unimodal embeddings to reproduce the multimodal embedding, allowing inference of $p(c|X_i)$. The MNIST dataset is shown here, where images are augmented with a synthetic, linear modality whose slope matches the digit label. A linear model for each cluster is used as the expert decoder for the synthetic modality, and successful unsupervised disentanglement implies multimodal fingerprint detection.

1.1. **Relationship to prior literature.** This work draws from several thematic bodies of literature. The non-exhaustive list below includes both works which have informed our approach, as well as recent state-of-the-art.

Gaussian mixture embeddings. For deep unsupervised clustering, several works replace the standard normal prior from [25, 46] with a GMM to facilitate disentanglement and provide an explicit parameterization of clusters [11, 21, 45, 29]. Each modality in the multimodal prior distribution is expected to provide disentangled

latent representations of data that admit an explicit parameterization of class distributions. The current work is most similar to Variational Deep Embedding (VaDE) [21] in its use of mean-field distributions to obtain a separable ELBO, and Bayesian estimator for $q(c|X)$. This work builds upon VaDE by incorporating multimodal data inputs while maintaining computational tractability of the ELBO, as well as employing clusters to decode into physics-informed MoE models.

Disentanglement. Another line of research is to extract latent disentangled representations into different factors of variations in data using VAEs. Earlier works such as β-VAE [17] and Annealed VAE [6] introduce additional weighting parameters to the KL divergence term of the original VAE ELBO loss. In Factor VAE [23] and Total Correlation Variational Autoencoder (β-TCVAE) [8] the ELBO is further decomposed to derive and penalize the total correlation to promote disentanglement in learned representations. For our purposes, however, these techniques lack an explicit parameterization of the cluster distributions so that it is not possible to conditionally define a physics-informed mixture of experts model.

Multimodal inference. Generative modeling from multimodal data can be broadly categorized into either conditional generative models [49, 42] which directly learn conditional cross-modal distributions $p(X_i|X_j)$, or joint models [51, 53, 59], which explicitly learn joint distributions that learn $p(Z, X_1, ..., X_M)$. We pursue the latter as [59] has been shown to provide a better description of the underlying data distribution. We pursue a strategy similar to works such as joint multimodal VAE [51] and joint VAE [53], where a joint inference network $q(Z|X_1, X_2)$ is trained, followed by training of two additional unimodal inference networks $q(Z|X_1)$ and $q(Z|X_2)$ which handle missing data at test time. The unimodal inference networks are trained to either match the joint inference network or to maximize an ELBO derived to perform unimodal variational inference. More recently, Multimodal Variational Autoencoder (MVAE) [59] and Mixture-of-Experts Multimodal Variational Autoencoder (MMVAE) [47] were proposed to model the joint posterior as a product-of-experts (PoEs) and a mixture-of-experts (MoEs). Most recently, Mixture-of-Products-of-Experts Variational Autoencoder (MoPoE-VAE) [50] proposed a new ELBO formulation, which generalizes ELBO formulations derived from PoEs and MoEs. Our encoder bears similarities to MMVAE, MoPoE-VAE, and PoE, while preserving a computationally tractable closed form ELBO when combined with the GMM prior.

Physics-informed ML and fingerprinting. Substantial works in recent years have focused on introducing physics into either solving partial differential equations (PDEs) or for building surrogates, typically introducing a PDE residual regularizer in *physics-informed neural networks* [27, 44] or by embedding physics directly into network architecture in *structure-preserving ML* [52]. Such tools can be combined to provide parametric surrogates of simulations which can perform real-time inference over a database of parameterized PDE solutions [35, 56, 36]. This paper provides a framework to fuse either these physics-informed surrogates or simpler empirical models together with experimental data. In contrast to traditional feature discovery tools, which rely on purely data-driven techniques like PCA [15, 16], the current framework provides a means to incorporate domain expertise into features, or fingerprints, tailored toward a scientific task.

Major contributions:

- Novel fusion of PoE with a Gaussian mixture to obtain parameterized cluster fingerprints for downstream data analysis and high-throughput diagnostic tasks.
- Multimodal embedding allows cross-modal inference while preserving closed form expressions for expectations in ELBO.
- Mixture-of-experts decoding allows incorporation of interpretable inductive biases by assuming an informative model form describing scientific processes. This allows the potential for embedding physics-informed surrogates or simulators alongside physics-agnostic encodings.
- Disentanglement of clusters into structured latent space exposing relationships across modalities.

2. **Framework construction.** We consider a dataset with D datapoints consisting of multiple modalities $X_1, \ldots, X_M$ where $X_m \subset \mathbb{R}^{d_m}$ and one data point $x^{(d)}$ is a tuple of the modalities, i.e. $x^{(d)} = (x_1^{(d)}, \ldots, x_M^{(d)})$ with $x_m^{(d)} \in \mathbb{R}^{d_m}$. The set of all modalities is denoted by $\mathcal{M} = \{X_1, \ldots, X_M\}$.

We seek a multimodal stochastic embedding of the data, $Z \in \mathbb{R}^l$, where the latent dimension $l << d_m$ for $m = 1, \ldots, M$. Assuming a categorical variable C clustering data into N clusters in latent space, we introduce parameterized prior p and posterior q distributions that maximize the following ELBO loss:

$$\mathcal{L} = \mathbb{E}_{q(Z,C|X_1,\ldots,X_M)}\left[\log \frac{p(X_1, ..., X_M, Z, C)}{q(Z, C|X_1, ..., X_M)}\right]. \tag{1}$$

We further assume a probabilistic chain rule decomposition of the prior and mean-field separability of the posterior:

$$\begin{aligned} p(X_1, ..., X_M, Z, C) &= \left(\prod_{m=1}^{M} p(X_m|Z, C)\right) p(Z|C)p(C), \\ q(Z, C|X_1, ..., X_M) &= q(Z|X_1, ..., X_M)q(C|X_1, ..., X_M). \end{aligned} \tag{2}$$

Our framework consists of three components: **1.** unimodal deep encodings with a product-of-experts (PoE) that fuses all modality encodings into one point in the latent space, **2.** a mixture of Gaussians prior which clusters the data into N Gaussians in the latent space, and **3.** a mixture of decoders, including deep decoders and also physics-informed decoders for modalities amenable to expert modeling. We introduce each component sequentially, derive a closed form expression for the ELBO, and introduce our optimization strategy for assignment of clusters and expert models. Table 2 contains a summary of the model distributions and parameters.

2.1. **Multimodal embedding.** We achieve a multimodal embedding by implementing a product-of-experts to fuse together unimodal embeddings. For the unimodal embeddings, each modality X_m is mapped into a common latent space $\mathbb{R}^l$ as a multivariate Gaussian Z_m with diagonal covariance. Indeed, these unimodal embeddings are represented by the posterior probabilities $q(Z_m|X_m) = \mathcal{N}(Z_m; \mu_m, \sigma_m^2\mathbf{I})$, where $\mu_m, \sigma_m^2 \in \mathbb{R}^l$ are the output of a set of neural networks F_m, and $\sigma_m^2\mathbf{I}$ denotes the diagonal matrix with diagonal entries given by σ_m^2. That is, for each modality m,

$$[\mu_m, \sigma_m^2] = F_m(X_m; \theta_m), \tag{3}$$

where θ_m denotes trainable weights and biases of the neural network F_m. Thus, for each $m = 1, \ldots, M$, the neural network F_m outputs the parameters for the unimodal Gaussian distribution $q(Z_m|X_m)$.

Distribution	Priors	Computation	Trainable Parameters
$p(X_m \mid Z, C=c)$	$\mathcal{N}(\hat{\mu}_{m,c}, \hat{\sigma}^2_{m,c}\mathbf{I})$	$[\hat{\mu}_{m,c}, \hat{\sigma}^2_{m,c}] = G_{m,c}(Z; \hat{\theta}_{m,c})$	$\hat{\theta}_{m,c}$, for $m=1,\ldots,M$, $c=1,\ldots,N$
$p(Z \mid C=c)$	$\mathcal{N}(\tilde{\mu}_c, \tilde{\sigma}^2_c\mathbf{I})$	$\tilde{\mu}_c$ = Equation 15 $\tilde{\sigma}^2_c$ = Equation 15	
$p(C)$	$\mathrm{Cat}(\pi)$	$\pi = \texttt{softmax}(\vec{v})$	$\vec{v} = (v_1, \ldots, v_N)$
$q(Z_m \mid X_m)$	$\mathcal{N}(\mu_m, \sigma^2_m\mathbf{I})$	$[\mu_m, \sigma^2_m] = F_m(X_m; \theta_m)$	θ_m, for $m=1,\ldots,M$
$q(Z \mid X_1, \ldots, X_M)$	$\mathcal{N}(\mu, \sigma^2\mathbf{I})$	σ^2 = Equation 5 μ = Equation 5	

TABLE 1. Distributions, priors, variables, and trainable parameters. Here the F_m and $G_{m,c}$ are each deep neural networks with respective weights and biases θ_m and $\hat{\theta}_{m,c}$. When suitable, each decoder network $G_{m,c}$ can optionally be replaced with an expert model $\mathcal{E}_{m,c}$.

We use a product-of-experts to fuse together the unimodal embeddings to obtain a multimodal distribution $q(Z|X_1,\ldots,X_M)$ in the latent space $\mathbb{R}^l$. Specifically, we assume that the multimodal distribution is the normalized product of the unimodal Gaussian distributions, i.e. $q(Z|X_1,...,X_M) \propto \Pi_{m=1}^M q(Z|X_m)$. As the product of Gaussian distributions is proportional to a Gaussian distribution, we obtain a new Gaussian distribution to represent the multimodal embedding:

$$q(Z|X_1,...,X_M) = \mathcal{N}(\mu, \sigma^2\mathbf{I}), \tag{4}$$

$$\sigma^{-2} = \sum_{m=1}^{M} \sigma_m^{-2}, \qquad \frac{\mu}{\sigma^2} = \sum_{m=1}^{M} \frac{\mu_m}{\sigma_m^2}. \tag{5}$$

The multimodal distribution may be sampled during training using the reparameterization trick: by sampling $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ and calculating $Z = \mu + \epsilon \odot \sigma$, we may back-propagate through the random node Z into the unimodal encoders, where $\odot$ denotes the Hadamard product.

The motivation for using the product-of-experts to obtain the multimodal distribution follows from an assumption that the modalities are pairwise independent. Indeed, under this assumption, one can show using Bayes' rule that

$$q(Z|X_1,...,X_M) = q(Z)^{1-M} \Pi_{m=1}^M q(Z|X_m), \tag{6}$$

so that the posterior $q(Z|X_1,...,X_M)$ is a scaled product of individual modalities, as in the product-of-experts formulation. The derivation of Equation 6 can be found in Appendix B.

2.2. **Gaussian mixture prior and expert decoding.** We adopt a simple Gaussian mixture prior of N clusters, where we let C denote the categorical random variable for the clusters and use c to denote a particular instance of C. Letting $\pi \in \mathbb{R}^N$ contain the probability of each cluster in the Gaussian mixture, our prior becomes

$$p(C) = Cat(\pi), \tag{7}$$

$$p(Z|C=c) = \mathcal{N}(\tilde{\mu}_c, \tilde{\sigma}_c^2\mathbf{I}), \text{ for } c = 1,\ldots,N, \tag{8}$$

where $\tilde{\mu}_c, \tilde{\sigma}_c^2 \in \mathbb{R}^l$ denote the mean and variances of cluster c, respectively. To ensure a positive π that sums to unity, we parameterize π as the softmax of a trainable vector $\vec{v}$.

Each modality is either decoded with a deep neural network or an expert model and can be a function of the latent space and the clusters. For tractability of the ELBO, the decodings are represented by Gaussian distributions, i.e.

$$p(X_m|Z, C = c) = \mathcal{N}(\hat{\mu}_{m,c}, \hat{\sigma}_{m,c}^2 \mathbf{I}), \text{ for } m = 1, \ldots, M \text{ and } c = 1, \ldots, N, \tag{9}$$

where $\hat{\mu}_{m,c}$, $\hat{\sigma}_{m,c}^2 \in \mathbb{R}^{d_m}$ are respectively the mean and variances of the reconstruction of modality X_m from a point in the latent space. To decode to a data-driven modality X_m, we employ a neural network with parameters $\hat{\theta}_{m,c}$ such that

$$[\hat{\mu}_{m,c}, \hat{\sigma}_{m,c}^2] = G_{m,c}(Z; \hat{\theta}_{m,c}). \tag{10}$$

To decode to a modality X_m with an expert model, we assume expert models of the form $\mathcal{E}_{m,c}(t, Z; \hat{\theta}_{m,c})$, such that

$$[\hat{\mu}_{m,c}, \hat{\sigma}_{m,c}^2] = \mathcal{E}_{m,c}(t, Z; \hat{\theta}_{m,c}), \tag{11}$$

where t is an independent variable and $\hat{\theta}_{m,c}$ denotes expert parameters associated with each cluster. The specific choice of $\mathcal{E}$ will be problem dependent. We specify expert models for our MNIST examples in the experiment section.

To facilitate postprocessing and uncertainty quantification, we note that $p(X_m|Z, C)$ admits interpretation as a mixture-of-experts model [22] and thus we obtain closed form expressions for the mean and variance:

$$\mathbb{E}[X_m|Z] = \sum_{c=1}^{N} \pi_c \hat{\mu}_{m,c} \quad \text{and} \quad \text{Var}[X_m|Z] = \left(\sum_{c=1}^{N} \pi_c (\hat{\sigma}_{m,c}^2 + \hat{\mu}_{m,c}^2) \right) - \mathbb{E}[X_m|Z]^2. \tag{12}$$

If X_m is decoded via an expert model that depends only upon the cluster, then $X_m \perp Z$ and Equation 12 computes means and variances of X_m independent of Z.

2.3. **ELBO loss and optimization of GMM parameters.** The parameters in PIMA include the weights and biases of the neural networks, the Gaussian mixture parameters (mixture probability and cluster means and variances), and any expert model parameters. These parameters are trained through maximizing the ELBO. We obtain a tractable ELBO through our extensive use of Gaussians for our model distributions. A modification of the derivation in [21] to account for multimodality yields the following analytic expression for the single sample ELBO:

$$\begin{aligned} \mathcal{L} = &- \sum_{m=1}^{M} \sum_{c=1}^{N} \sum_{j=1}^{d_m} \gamma_c \left(\log \hat{\sigma}_{m,c;j}^2 + \frac{(X_{m;j} - \hat{\mu}_{m,c;j})^2}{\hat{\sigma}_{m,c;j}^2} \right) \\ &- \sum_{c=1}^{N} \sum_{j=1}^{l} \gamma_c \left(\log \tilde{\sigma}_{c;j}^2 + \frac{\sigma_{;j}^2}{\tilde{\sigma}_{c;j}^2} + \frac{(\mu_{;j} - \tilde{\mu}_{c;j})^2}{\tilde{\sigma}_{c;j}^2} \right) \\ &+ 2 \sum_{c=1}^{N} \gamma_c \log \frac{\pi_c}{\gamma_c} + \sum_{j=1}^{l} \left(1 + \log \sigma_{;j}^2 \right), \end{aligned} \tag{13}$$

where the subscript $;j$ denotes the j^{th} coordinate of the vector, and γ_c is the posterior distribution

$$\gamma_c = \gamma_c(Z) = p(C{=}c|Z) = \frac{\pi_c p(Z|C{=}c)}{\sum_{c'} \pi_{c'} p(Z|C{=}c')}. \tag{14}$$

In particular, we estimate $q(C{=}c|X_1, ..., X_M)$ with $p(C{=}c|Z) = \gamma_c$, following [21]. The derivation of $\mathcal{L}$ may be found in Appendix B. We seek to maximize this loss over the entire data set. That is, letting $\mathcal{L}^{(d)}$ denote the loss function for one data point, then we seek to minimize $Loss = -\sum_d \mathcal{L}^{(d)}$. We note that the first line in $\mathcal{L}$ describes the reconstruction loss, while the remainder of the terms compute the KL-divergence $D_{KL}(q(Z, C|X_1, \ldots, X_M)||p(Z, C))$.

2.4. **Training.** Throughout training, we use gradient descent and techniques borrowed from expectation-maximization (EM) to optimize the model parameters. Finding the optimal cluster centers and variances in the GMM parallels finding the maximum likelihood parameters of the GMM fitting the posterior distribution on Z. Consequently, we use an approach similar to expectation-maximization to optimize the GMM parameters to the latent space data. Like expectation-maximization, our hybrid approach has two steps. After embedding each point $x^{(d)}$ as a normal distribution $\mathcal{N}(\mu^{(d)}, \sigma^{2(d)}\mathbf{I})$ in Z, we sample $z^{(d)} \in \mathcal{N}(\mu^{(d)}, \sigma^{2(d)}\mathbf{I})$ and (1) for each sampled data point $z^{(d)}$ we compute the expectation of belonging to each cluster, $\gamma_c^{(d)} = \gamma_c(z^{(d)})$. Then, while π is fixed, we (2) maximize the log-likelihood of the GMM on the sampled points $z^{(d)}$ by computing new cluster centers ($\tilde{\mu}_c$) and cluster variances ($\tilde{\sigma}_c^2$):

$$\tilde{\mu}_c = \frac{\sum_{d=1}^D \gamma_c^{(d)} \mu^{(d)}}{\sum_{d=1}^D \gamma_c^{(d)}} \qquad \text{and} \qquad \tilde{\sigma}_{c;j}^2 = \frac{\sum_{d=1}^D \gamma_c^{(d)} (\mu_{;j}^{(d)} - \tilde{\mu}_{c;j})^2}{\sum_{d=1}^D \gamma_c^{(d)}}, \text{ for } j = 1, \ldots, l. \tag{15}$$

We update π via gradient descent in concert with the other model parameters, e.g. those coming from the encoders and decoders, while keeping the cluster means and variances fixed. We thus employ a streaming algorithm outlined in Algorithm 1 where we first perform an EM update for $\tilde{\mu}_c$ and $\tilde{\sigma}_c^2$, followed by an Adam update [24] that updates only the remaining variables. In our experiments, we found that a single EM update was sufficient, although one could optionally perform many EM update steps until a desired convergence is reached.

A weighted least squares problem for the optimal expert model parameters may be similarly obtained by taking the variation of the ELBO with respect to $\hat{\theta}_{m,c}$:

$$\hat{\theta}_{m,c} = \underset{\theta'}{\operatorname{argmin}} \sum_{d=1}^{D} \sum_{j=1}^{d_m} \gamma_c^{(d)} \left(X_{m;j} - \mathcal{E}_{m,c}(t^{(d)}, z^{(d)}; \theta')_{;j} \right)^2. \tag{16}$$

Efficient solution of this nonlinear least squares problem at each epoch will be dependent upon the problem-specific expert model and data stream, and performing batching may require a streaming technique such as recursive least squares or Kalman filtering [10]. For simplicity, we update all $\hat{\theta}_{m,c}$ with Adam in this work but note that solving Equation 16 at each epoch to ensure the expert model provides a best fit to the current partitions is likely to provide substantial improvement.

2.5. **Practical considerations and flexibility.** The PIMA framework admits flexibility to accommodate various applications and model choices. We outline the

Algorithm 1 Training with streaming EM for cluster centers

Input: data $X = \{X_1, ..., X_M\}$ in batches $\mathcal{B}$
for $i = 1$ **to** N_{epochs} **do**
  Calculate μ, σ s.t. $q(Z|X_1, \ldots, X_M) = \mathcal{N}(\mu, \sigma^2\mathbf{I})$ for all $x^{(d)} \in X$
  Sample $z \sim \mathcal{N}(\mu, \sigma^2\mathbf{I})$ for all μ, σ
  Calculate $\gamma_c = p(c|z)$ for all z and c
  Calculate $\tilde{\mu}_c$ via Equation 15 for all c
  Calculate $\tilde{\sigma}_c^2$ via Equation 15 for all c
  for $\mathbf{b} \in \mathcal{B}$ **do**
    Compute $Loss_{\mathbf{b}} = -\sum_{x \in \mathbf{b}} \mathcal{L}^{(x)}$
    Calculate Adam update on $Loss_{\mathbf{b}}$ for π and all $\theta_m, \hat{\theta}_{m,c}$
  end for
end for

options utilized in our experiments. A summary of the specific options used for each experiment is in Table 5 of Appendix A.

Fixed decoding variances. To prevent over-fitting, we follow [21] and use fixed variances for the decoders, i.e. $\hat{\sigma}_{m,c} = 1$ for all modalities $m = 1, \ldots, M$ and clusters $c = 1, \ldots, N$.

Weighting the reconstruction term. Within a multimodal dataset, often the dimensions $d_1, \ldots, d_M$ of the M modalities differ. As a result, the reconstruction term of Equation 13 (first line of $\mathcal{L}$) will favor higher-dimensional modalities. To equally weigh each modality X_m, we follow [50] and scale its reconstruction term by $d_{\max}/\dim d_m$, where $d_{\max} = \max\{d_1, \ldots, d_M\}$. The single sample reconstruction term then becomes:

$$-\sum_{m=1}^{M}\sum_{c=1}^{N}\sum_{j=1}^{d_m}\gamma_c \frac{d_{\max}}{d_m}\left(\log\hat{\sigma}_{m,c;j}^2 + \frac{(X_{m;j} - \hat{\mu}_{m,c;j})^2}{\hat{\sigma}_{m,c;j}^2}\right).$$

Note that, if variances $\hat{\sigma}_{m,c;j}^2$ are fixed, this reconstruction term is equivalent to assuming the prior $p(X_m|Z, C = c) = \mathcal{N}(\hat{\mu}_{m,c}, \frac{d_m}{d_{max}}\mathbf{I})$.

While we weight the reconstruction term to balance the importance of modalities of different dimensions, a user could optionally weight modalities according their interests. The flexibility in this weighting allows the user to influence the importance of the various modalities in the latent space representation. This is particularly useful to either encourage or discourage certain modalities from dominating the structure of the latent space.

One decoder per modality. Under the presented framework, each modality has N decoders. However, one may also consider only one decoder per modality, in which case the reconstruction term of the loss simplifies to:

$$-\sum_{m=1}^{M}\sum_{j=1}^{d_m}\left(\log\hat{\sigma}_{m;j}^2 + \frac{(X_{m;j} - \hat{\mu}_{m;j})^2}{\hat{\sigma}_{m;j}^2}\right).$$

This choice removes the decoders' dependencies on C. While potentially decreasing the model's expressivity, this choice has the advantage of simplifying the model by reducing the number of parameters while also being consistent with the γ estimation justifications in Appendix B.

Dropout. Sampling from the unimodal encoders $q(Z_m|X_m)$ provides embeddings far from the multimodal embedding $q(Z|X_1,\ldots,X_M)$, which is not conducive to cross-modal inference. To move unimodal embeddings closer to the multimodal embeddings, during training we optionally implement a form of modality dropout where we stochastically choose a subset of unimodal embeddings to compute the multimodal embedding. This is equivalent to multiplying each term in Equation (5) by a binomial random variable $w_m \sim B(1,p_m)$, where p_m is the probability that modality m is included:

$$\sigma^{-2} = \sum_{m=1}^{M} w_m \sigma_m^{-2}, \qquad \frac{\mu}{\sigma^2} = \sum_{m=1}^{M} w_m \frac{\mu_m}{\sigma_m^2}.$$

This effectively forces each individual encoder to better predict the whole, which prevents over-fitting and allows inference across modalities. In the present work we set $p_m = 0.5$, but other values of p_m could be used to encourage or discourage modality m from dominating the structure in the latent space. Furthermore, while we do not explore incomplete datasets in this work, we additionally note that this dropout capacity enables PIMA to train on incomplete datasets by setting $w_m = 0$ whenever modality m is missing for a specific datapoint. PIMA can also predict the missing data through its cross-modal capabilities.

Expert model flexibility. In practice, expert models $\mathcal{E}$ may take a variety of forms and its judicious selection imparts significant prior knowledge. In the experiment section, we consider simple generalized linear models and models from neural ODEs for the MNIST experiments, and we make suggestions for potential expert models for the vibrational density of states experiment. In general, these expert models could range from empirical engineering correlations obtained from e.g. dimensionless analysis or singular perturbation theory, to analytic parametric solutions to PDE based models, or to parametric physics-informed ML surrogates/reduced order models (see e.g. [35, 56, 36, 52]). While expert models can come from complex, high-fidelity simulations, such a choice can slow down the training time significantly. The aim with the expert models is not to produce exact reproductions with the decoder, but rather to constrain the decoder functional space to simpler, physically relevant and interpretable maps that can help separate the data of that modality into meaningful regimes. A high-fidelity expert model may not be necessary for this task of disentangling the latent space through relevant physics.

3. Cross-modal inference.

We identify two methods for cross-modal inference. While this work only investigates dropout inference, we include methods for Bayesian inference for completeness and use in future works.

3.1. Bayesian inference.

Here we only consider modalities where the decoders do not depend upon Z, i.e. $p(X_m|Z,C) = p(X_m|C)$. Given a modality $X_{m'}$ we can predict the mean and variance of X_m by identifying the probability of each cluster based on the modality $X_{m'}$. In particular, we predict X_m from $X_{m'}$ via the following:

$$\mathbb{E}[X_m] = \sum_c \kappa_c \hat{\mu}_{m,c} \qquad \text{and} \qquad \mathrm{Var}[X_m] = \left(\sum_c \kappa_c(\hat{\sigma}_{m,c}^2 + \hat{\mu}_{m,c}^2)\right) - \mathbb{E}[X_m]^2, \tag{17}$$

where via Bayes' Rule,

$$\kappa_c = p(C{=}c|X_{m'}) = \frac{\pi_c p(X_{m'}|C=c)}{\sum_{c'} \pi_{c'} p(X_{m'}|C{=}c')}. \tag{18}$$

In this framework, multiple modalities can be used to predict a given modality through redefining κ_c. That is, let $\mathcal{I} \subset \{1,\ldots,M\} \setminus \{m\}$. Then we can let

$$\kappa_c = p(C{=}c|X_{\mathcal{I}_1},\ldots,X_{\mathcal{I}_{|\mathcal{I}|}}) = \frac{\pi_c \prod_{k\in\mathcal{I}} p(X_k|C{=}c)}{\sum_{c'} \pi_{c'} \prod_{k\in\mathcal{I}} p(X_k|C{=}c')}$$

3.2. **Dropout inference.** By utilizing the dropout methods from Section 2.5, unimodal encoders may provide embeddings similar to the multimodal embedding $q(Z|X_1,\ldots,X_M)$. Sampling from $q(Z_m|X_m)$ allows generative modeling by decoding with $p(X_{m'}|Z,C)$ for $m' \neq m$ and an estimate of $p(C|Z)$ via Equation (14). This type of inference was explored in Experiment 4.1. In cases where dropout does not move unimodal embeddings close enough to the multimodal embeddings, one can train a new set of unimodal encoders, with architectures identical to the original unimodal encoders, to the multimodal latent space. In this sense, the unsupervised multimodal training provides labels which allows supervised training of the new unimodal encoders.

4. **PIMA Experiments.** We present results from PIMA applied to MNIST and a molecular simulation dataset. We perform two different experiments on the MNIST dataset, where each experiment augments the image modality with its own synthetic modality. All hyperparameters, hardware and training details for each experiment are provided in Appendix A.

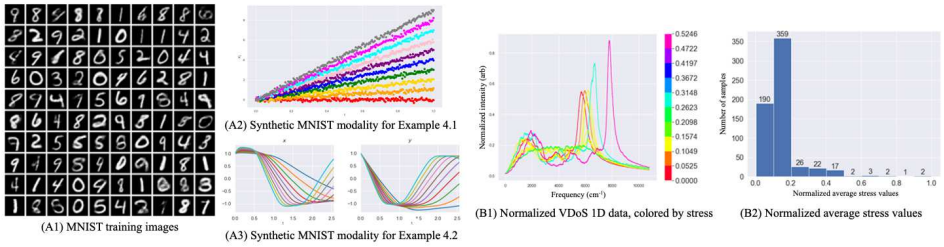


FIGURE 2. Experimental setup for test examples. For unsupervised multimodal MNIST, we use training images (A1) and we replace the labels on digits $c \in \{0,\ldots,9\}$ by a sample of the function (A2) $X_2 = ct + \epsilon$, for $t \in [0,1]$ and Gaussian noise ϵ. For neural ODE MNIST, we use the training images (A1) and replace the label on digits by solutions to an ODE system (A3). For VDoS, we use the 1D VDoS data (B1) and the corresponding 0D average stress (B2).

Method	CNN SotA*	VAE+GMM†	DEC†	VaDE†	GMVAE††	GMVAE††
Notes	Supervised				10 clusters	16 clusters
Acc. (max)	99.91%	72.94%	84.30%	94.46%	88.54%	96.92%
Acc. (mean±stdev)	n/a	n/a	n/a	n/a	82.31% (3.75%)	87.82% (5.33%)

Method	PIMA	PIMA	PIMA	PIMA	PIMA
Notes	multimodal dropout	multimodal no dropout	X_1 only -	X_2 only -	multi., no expert dropout
Acc. (max)	99.79%	99.59%	14.84%	53.37%	58.36%
Acc. (mean±stdev)	90.31% (14.81%)	87.95% (11.70%)	-	-	-
X_1 Acc. (max)	39.15%	11.26%	-	-	50.34%
X_2 Acc. (max)	99.92%	32.39%	-	-	56.22%

TABLE 2. Unsupervised classification accuracy for MNIST. Results gathered from [2], [21] and [11] denoted by $*$,† and ††, respectively. If statistics were not provided we assume maximum accuracy was reported. While the data augmentation offered by X_2 is not incorporated in comparisons to unimodal unsupervised benchmarks, a comparison to the supervised setting is valid. For all experiments we do not overparameterize and keep clusters equal to the number of digits. The PIMA results are reported on the standard 10,000 test samples. Averages and standard deviation results are reported over 9 runs with different random seeds.

4.1. **Unsupervised multimodal MNIST.** For our first MNIST experiment, we use a 90/10 train/validation split of the training data and report on the standard 10,000 held out test examples [28]. To test with multiple modalities, we augment the traditional MNIST images X_1 and labels $c \in \{0, ..., 9\}$ with a manufactured synthetic 1D modality $X_2 = ct + \epsilon$, where $t \in [0, 1]$ and $\epsilon \sim \mathcal{N}(0, 0.01)$. We adopt the affine expert model $\mathcal{E}(t; \theta_c) = \theta_c t$, and perform unsupervised clustering of the multimodal dataset (X_1, X_2) as well as cross-modal inference. For this artificial problem, the labels are thinly veiled as the slope of X_2, and so we expect that if we successfully perform multimodal inference we should obtain accuracy comparable to a supervised MNIST benchmark.

We define unsupervised clustering accuracy (acc) as in [60], [21]:

$$acc = \max_{\phi \in \Phi} \frac{\sum_{i=1}^{N} \mathbb{1}\{l_i = \phi(c_i)\}}{N}, \tag{19}$$

where N is the number of examples, Φ is the set of all possible mappings from a cluster to a label assignment, l_i is the true label and c_i is the cluster assignment by the model.

We aim to develop a multimodal model on (X_1, X_2) with maximal accuracy such that we can also perform cross-modal inference. As such, we perform a hyperparameter search over learning rate to maximize accuracy for a multimodal dropout model. We then fixed the hyperparameters and compared against unimodal models, a multimodal model without dropout, and a multimodal model but with a 1D convolutional neural network in place of the expert model. The latent spaces and confusion matrices for all four models with the expert 1D decoder are given in Figure 3. Confusion matrices for the multimodal models reveal an approximately banded structure, whereby misclassified modalities primarily occur between adjacent digits, suggesting that X_2 dominates the latent space structure. The dropout multimodal model performed the best, exceeding the accuracies of the non-dropout multimodal model, demonstrating the effectiveness of modality dropout for training.

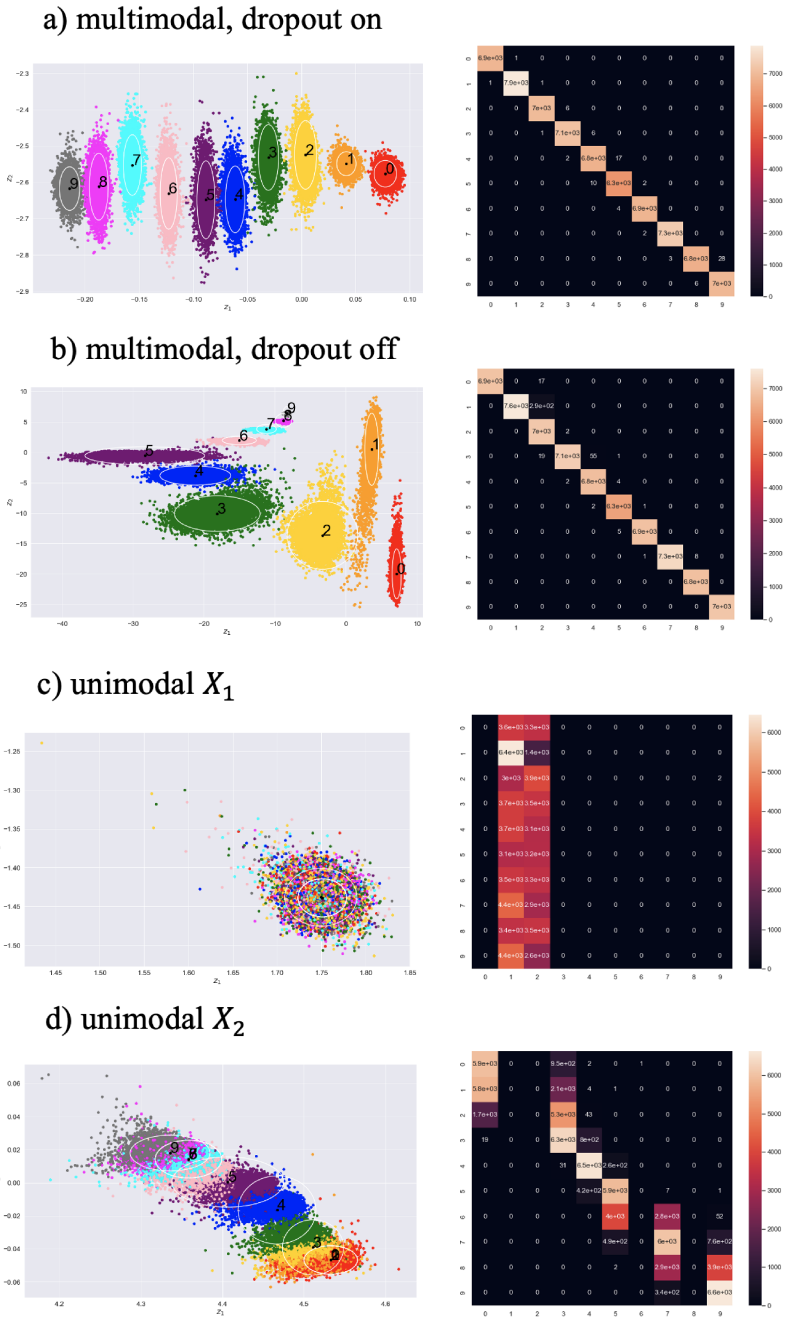


FIGURE 3. MNIST clusters in the latent space and resulting confusion matrices for a) multimodal dataset with dropout, b) multimodal dataset without dropout, c) unimodal X_1 image dataset, d) unimodal X_2 1D dataset. The white ellipses in the latent space represent two standard deviations of each cluster in the GMM. The approximate banding of the matrix in a) and b) illustrates that the sequential embedding of clusters limits misclassified digits to numbers with similar values in X_2.

Furthermore, the dropout multimodal model obtained model performance comparable to the current state-of-the-art in supervised classification and outperformed the state-of-the-art in unsupervised learning. We discovered that the unimodal models did not train well and consequentially gave the worst model accuracies. As GMM and VAE frameworks are known to perform well with (un)supervised learning on MNIST images (see Table 2), this indicates that further efforts, e.g. pre-training as in [21] or additional hyperparameter tuning, would be necessary given our fixed architectures. The multimodal model without the 1D expert model performed marginally better than the unimodal models, but with an accuracy of only 58.36% whereas the multimodal, dropout model with 1D expert model achieved a maximum accuracy of 99.79%. In Table 2 we provide a comparison to classification accuracy against state of the art supervised and unsupervised models trained on images only. We also performed cross-modal inference on our multimodal models and saw that the dropout model outperformed the model without dropout. Finally, we repeated the multimodal experiments for eight other random seeds and reported the overall statistics in Table 2.

4.2. **Neural ODE expert model.** Next we test PIMA on a different augmentation of the MNIST dataset. In this experiment, the manufactured dataset consists of measurements of simulated dynamical systems. The parameters of these simulations are functions of the class labels c of the MNIST datasets. We then use a different expert model, namely a neural ordinary differential equations (NODEs) model [9], to model the dynamical system data. The aim of this expert model is to learn the parameters from the simulated dynamical systems, much like the aim of the expert model in Subsection 4.1 was to learn the slopes of the lines of X_2. Specifically, we pair the images of the MNIST dataset (X_1) with simulations of a parametrized cubic oscillator (X_2), which is governed by a system of ODEs,

$$\begin{aligned} \tfrac{\mathrm{d}x}{\mathrm{d}t} &= \alpha(c)x^3 + \beta(c)y^3, \\ \tfrac{\mathrm{d}y}{\mathrm{d}t} &= \gamma(c)x^3 + \delta(c)y^3, \end{aligned} \tag{20}$$

where $\alpha(c), \beta(c), \gamma(c)$, and $\delta(c)$ denote the ODE parameters that are dependent on the class labels c of the MNIST dataset. That is, there is one set of parameters associated with each target label. The goal of this experiment is to check the capability of accurately identifying the ODE parameters via training of PIMA and demonstrate the efficacy of using an expert model decoder over a black-box decoder.

Preliminaries on NODEs. NODEs are a class of deep neural network models that learn the depth-continuous dynamics of hidden states $\boldsymbol{h}(s)$ of a feed-forward network as a form of ODEs:

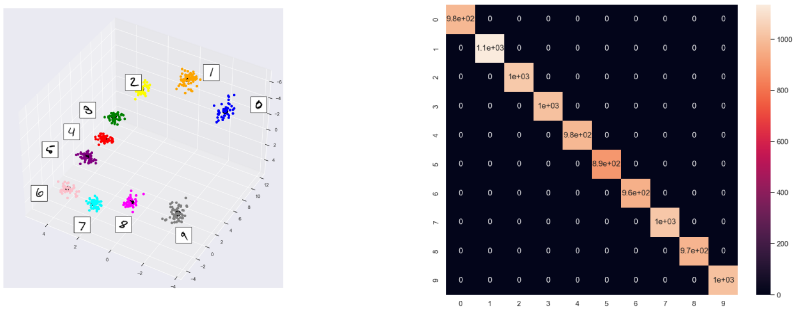
$$\frac{\mathrm{d}\boldsymbol{h}}{\mathrm{d}s} = \boldsymbol{f}(\boldsymbol{h}; \Theta), \tag{21}$$

where s denotes the depth in a continuous representation, $\boldsymbol{h}(s)$ is a depth-continuous representation of a state, $\boldsymbol{f}$ is a parameterized (trainable) velocity function, and Θ is a set of neural network weights and biases. For the parameterization of NODEs (i.e., the right-hand side of Eq. (21)), there can be many alternatives including a multilayer perceptron (i.e., a 'black-box' approach) or a dictionary-based parameterization. In this work, we choose the dictionary-based parameterization for our expert model, where we assume that the exact governing equations are known a priori, but not the coefficients, which should be recovered from the training.

Training NODEs. In training NODEs, an initial value problem (IVP) is typically solved given the initial condition $\boldsymbol{h}(0)$, to obtain the status of the hidden states at an arbitrary depth:

$$\tilde{\boldsymbol{h}}(t_1), \ldots, \tilde{\boldsymbol{h}}(t_m) = \text{ODESolve}(\boldsymbol{h}(0), \boldsymbol{f}_\Theta, t_1, \ldots, t_m). \tag{22}$$

Although NODEs can be effective in data-driven dynamics modeling or system identification [30], it is often the case that solving initial value problems in the forward/backward pass causes long computation time. To avoid such numerical issues and also to be consistent with the experimental setting used in the previous MNIST experiment, we consider spectral NODEs (SNODEs) for the construction of the NODEs expert model [43].



1D Model	# Decoder Parameters	Mean Acc.	Min Acc.	Max Acc.
NODEs Expert Model	10	90.1% +/- 10.3%	70.5%	100.0%
Large Data-driven Model	32,954	82.3% +/- 11.2%	60.5%	90.2%
Small Data-driven Model	4,230	62.6% +/- 11.1%	50.0%	80.5%

FIGURE 4. MNIST clusters, confusion matrix, and accuracy for the best PIMA run on the multimodal ODE dataset with an encoding dimension of size 3. With a NODEs expert model, maximum test accuracy achieved is 100.0%. With data-driven models, the average test accuracies are 82.3% and 62.6% for large and small 1D convolutional architectures, respectively. Statistics are computed from 10 independent runs.

SNODEs first approximate the solution of target ODEs spectrally, as a linear combination of a set orthogonal polynomials, e.g., Legendre polynomials or Fourier basis, such that

$$u(t) = [x(t), y(t)] \approx \tilde{u}(t) = [\tilde{x}(t), \tilde{y}(t)] = \sum_{i=1}^{n} \omega_i \psi_i(t), \tag{23}$$

where $\{\psi_i(t)\}$ denotes a set of orthogonal polynomials and $\{\omega_i\}$ denotes coefficients with the $\dim(\omega_i) = \dim(u(t)) = 2$. The coefficients can be computed by fitting the model to data, which results in the learned coefficients $\{\omega_i\}$. The next step is to learn dynamics (i.e., to identify the ODE parameters). Replacing the solution of the ODEs with the spectrally approximated quantity yields a residual

$$r(\tilde{u}) = \frac{\mathrm{d}u}{\mathrm{d}t} - f(\tilde{u}) = \begin{cases} \frac{\mathrm{d}x}{\mathrm{d}t} - (\alpha(c)\tilde{x}(t)^3 + \beta(c)\tilde{y}(t)^3) \\ \frac{\mathrm{d}y}{\mathrm{d}t} - (\gamma(c)\tilde{x}(t)^3 + \delta(c)\tilde{y}(t)^3) \end{cases}. \tag{24}$$

Then the ODE parameters can be identified by solving the optimization problem

$$\arg\min_{\alpha,\beta,\gamma,\delta} \|r(\tilde{u})\|. \tag{25}$$

In PIMA, we use POUNets [31] and automatic differentiation to estimate solutions $\tilde{x}_c(t)$ and $\tilde{y}_c(t)$ from the dataset. Then our NODE expert model for X_2 is

$$\mathcal{E}(t;\alpha_c,\beta_c,\gamma_c,\delta_c) = \begin{cases} \alpha_c\tilde{x}_c(t)^3 + \beta_c\tilde{y}_c(t)^3 \\ \gamma_c\tilde{x}_c(t)^3 + \delta_c\tilde{y}_c(t)^3 \end{cases}, \tag{26}$$

where each cluster c has trainable parameters $\alpha_c, \beta_c, \gamma_c, \delta_c$. These parameters are trained through the ELBO (Equation (13)), which consequently minimizes Equation (25) through the ELBO reconstruction term.

Experimental results. For the experiment, we consider the same setting used in the previous MNIST experiments (Section 4.1). The dataset consists of two modalities: MNIST images (X_1) and measurements on time-derivatives of solutions of the ODEs in Equation (20) (X_2), which are obtained numerically from synthetic 2D trajectories. To manufacture the synthetic 2D trajectories, we solve the IVPs of Equation (20) with the ODE parameters, $\alpha(c) = -0.1$, $\gamma(c) = -2$, $\delta(c) = -0.1$ and $\beta(c) = c/3 + 0.5$, and with the initial condition $x^0 = [1, 1]$. The simulation time is set as 2.56 seconds and the measurement step size is 0.005, resulting in 10 different 2D state trajectories collected at 512 time steps. Then for each collected trajectory, we numerically approximate $u \approx \tilde{u}$ via a regression technique called POUNets [31], which builds a meshfree partition of space and in each partition, there is an associated polynomial space with learnable coefficients. We use these POUNets to compute the time derivative of $\tilde{u}$ (i.e., $\frac{\mathrm{d}\tilde{u}}{\mathrm{d}t}$) via automatic differentiation. Then for each datapoint in X_2 an additive noise sampled from a normal distribution $\epsilon \sim \mathcal{N}(0, 0.1)$ is injected to the numerical time-derivatives of $\tilde{u}$.

For the performance metric, we again measure the unsupervised cluster accuracy (acc). The latent dimension is set as $l = 3$ with a learning rate of 10^{-3}. We perform 10 independent runs with varying random initializations. Figure 4 reports the results of the best run, which yields 100% test accuracy and statistics computed from the 10 runs; (1) the left panel visualizes the learned clusters illustrating that PIMA learned disentangled latent representations, which leads to the 100% classification accuracy, (2) the right panel reports the confusion matrix showing 100% True positive rate and 0% False negative rate, and (3) finally, the bottom table reports mean, min, and max accuracy that are obtained from the 10 runs. To show the efficacy of the expert model in this experiment, we additionally ran PIMA on this dataset with convolutional 1D neural networks in place of the NODEs expert model. Specifically, we performed 10 runs on a large 1D convolutional neural network containing 32,954 parameters, as well as a small 1D convolutional neural network of 4,230 parameters. Statistics over 10 runs are reported in Figure 4, where both neural network architectures, on average, performed worse than the NODEs expert model, demonstrating the advantage of implementing an expert model when possible.

label (c)	0	1	2	3	4
$\beta(c)$	0.5	.833	1.166	1.5	1.833
identified β	0.4997	0.8339	1.1661	1.4999	1.8339
label (c)	5	6	7	8	9
$\beta(c)$	2.166	2.5	2.833	3.166	3.5
identified β	2.1669	2.5010	2.8330	3.1673	3.4989

TABLE 3. Ground-truth ODE parameters $\beta(c)$ and identified ODE parameters.

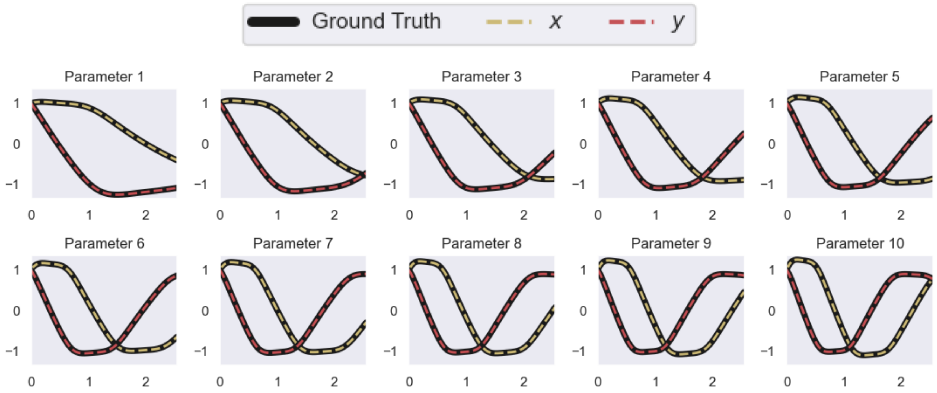


FIGURE 5. Ground-truth trajectories (black solid lines) and reconstructed trajectories (dashed lines) by solving IVPs with the learned ODE parameters.

Next we assess the performance of the expert model in identifying the ODE parameters. Table 3 reports the ground-truth ODE parameters and the identified ODE parameters. The identified ODE parameters match well with the ground-truth ODE parameters (i.e., the first 3-4 significant digits match the ground-truth values). Figure 5 depicts the reconstructed trajectories by solving IVPs with the learned ODE parameters. We observe that the reconstructed trajectories match well with the ground-truth trajectories (black solid lines).

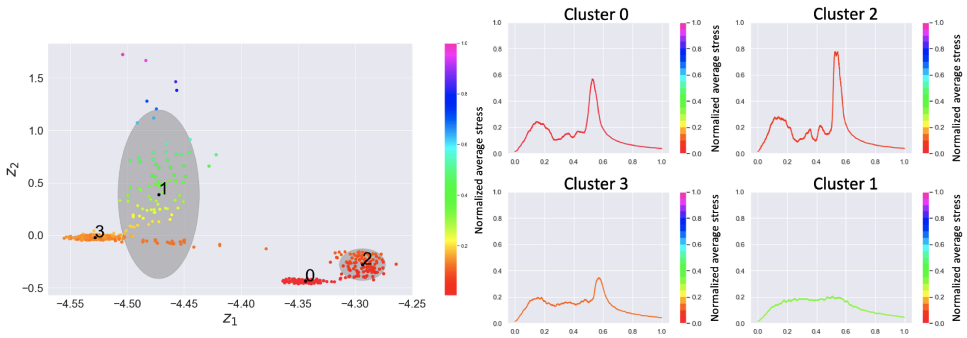


FIGURE 6. PIMA results on the VDoS dataset. VDoS latent space (left) with four clusters, where embedded data points are colored by the normalized true stress values. (Right) Nearest true VDoS data to the means of each cluster in the latent space, colored by the normalized stress. Results show a latent space organized by stress and VDoS profiles; when referencing the data generation information in Figure 7, we also see clusters 0 and 2 are differentiated by compression type, indicating discovery of hidden features.

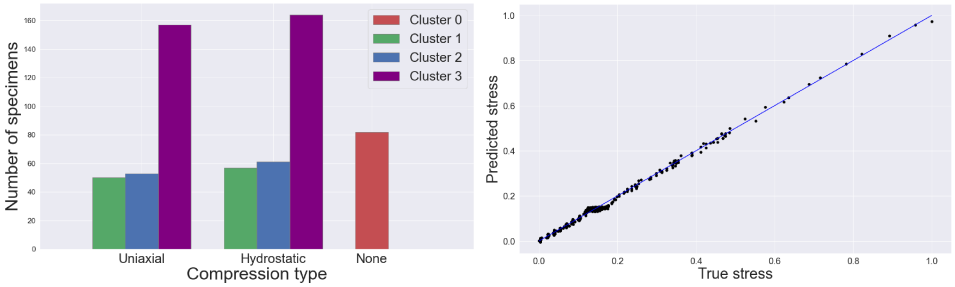


FIGURE 7. (Left) Number of specimens per cluster from the training dataset that underwent uniaxial compression, hydrostatic compression, or no compression. (Right) Predicted vs. true normalized average stress values over entire VDoS dataset.

4.3. **Vibrational density of states.** Finally, we test PIMA on a large, synthetic multimodal materials dataset with the intent to discover shared, cross-modal information. Specifically, we consider a synthetic dataset of [55] in which molecular dynamic simulations of single crystal silicon atomic structures were performed with varying degrees of disorder, deformation (varying strains for uniaxial compression or hydrostatic compression, or no compression), and combinations of both disorder and deformation. The resulting dataset consists of 772 atomic structures, each generated by different disorder and deformation process parameters, where the vibrational density of states (VDoS) spectroscopy profile and average stress values are computed for each structure. Both disorder and deformation alter the VDoS of a structure in various ways, including peak shifts due to changes in interatomic distances, peak broadening from localization effects that change the vibrational modes, the emergence of new peaks from local phase transitions, and even the overall VDoS shape can change as complex combinations of deformation and disorder can transform the vibrational landscape of the system [38]. Generally, relationships between stress and VDoS are inferred by peak analysis (peak location, peak width), but there is not a clear relationship between stress and VDoS [58]. In contrast, our goal is to use PIMA to find unequivocal correlations between the average stress of an atomic structure and its VDoS.

We consequently implement a data-driven PIMA model with two modalities: the 1D VDoS data as X_1 and the scalar 0D average stress data as X_2. We performed a sweep over the learning rate and the number of clusters, and we selected the model yielding the lowest RMS error of the predicted average stress. Our selected model gave RMS errors for stress predictions of 0.85%/0.96%/0.88% on the train/validation/test data. While our model does give low RMS errors, the aim of PIMA is not to just give good predictions on quantities of interest, but rather to learn additional, hidden information from considering multimodal data jointly. Indeed in our learned model, the latent space is organized by the different modalities, where latent points are arranged by the average stress modality and four distinct clusters highlight different VDoS profiles, as illustrated in Figure 6. Specifically, high values of the Z_2 axis of the latent space correspond to high average stress values. Furthermore, while clusters 0 and 2 display similar VDoS with low stress values, they are separated in latent space, indicating a hidden feature influencing the separation. When referencing the disorder and deformation process parameters for the atomic structures in these two clusters

(see Figure 7), we see that cluster 0 corresponds to those atomic structures that underwent disorder with no compression deformation, while cluster 2 corresponds to atomic structures that have been deformed. Similarly, cluster 1 corresponds to atomic structure that have been highly deformed (and therefore with high stress values) and for which the form of the VDoS profile deviates significantly from the baseline VDoS profile of non-deformed atomic structures. The ability of the PIMA model to simultaneously parse the VDoS profiles and stress values with no *a priori* knowledge or label on the state of the atomic structure generation demonstrates its capacity to detect hidden features within the multimodal VDoS and stress data that would be challenging to uncover with classical machine learning techniques.

In terms of implications, this PIMA model shows that it is possible to fingerprint complex states of the atomic systems based on an observed VDoS. This ability goes beyond current state of the art, which is based on peak analysis and reduces the VDoS to a handful of scalars with limited predictions on the state of the atomic system [58]. Here the PIMA model is able to cluster various classes of states of the atomic systems, even when the VDoS profile changes drastically – an ability not doable with classical peak width analysis. Moving forward, this ability suggests that PIMA models could directly use an observed VDoS (either in the form of a Raman spectra or neutron scattering spectra) and infer a set of materials descriptors that are otherwise impossible to extract from those spectra with current peak analysis techniques. This is important because it opens the door for a fast, deeper characterization of complex materials based solely on this type of 1D spectral data. This capacity to find deeper characterizations from 1D spectral data has a range of applications, including materials science for complex solid state problems [13], biophysics for the study of protein functions and their dynamics [12], and chemistry for the study of complex phase transitions seen in chemical reactions [14].

Our model here did not use an expert model decoder to demonstrate the value of exploiting multimodal data, even without expert models. One could, however, surmise developing an expert model for the VDoS data. While the data was generated using molecular dynamics software, the size and complexity of such simulations precludes them from backpropagation and thus would not be suitable as an expert model. Alternatively, one could construct an expert model of VDoS using PCA to determine the important peak structures within the data. This option would require careful analyses in the material science domain, which falls outside the scope of this introductory paper. While we did not construct such a model here, we postulate that such a VDoS expert model could further structure the latent space meaningfully, as seen in the MNIST experiments.

5. **Discussion, conclusions, and future work.** The present approach provides an abstract variational inference algorithm for unsupervised feature discovery in multimodal datasets, while incorporating physical model biases. We demonstrated that our framework is capable of representing multiple modalities for fingerprint detection (see Experiment 4.3), performing cross-modal inference (see Experiment 4.1), and exploiting expert models to enhance model performance and training (see Experiments 4.1 and 4.2). We also described some flexible aspects of our framework. For example, our framework has the flexibility to weight modality importance in the loss function, as discussed in Section 2.5, which means latent space embeddings can be influenced by the user's preference of modalities. The flexibility of the expert decoding, including replacement by a neural network, also allows this framework

to be widely applicable. While in this work we have focused our expert models on a simple MNIST example to probe dynamics for an easily replicable and understandable dataset, we will employ more sophisticated physics-informed surrogates as expert models in future work. A brief discussion of potential expert models is given in Section 2.5.

While our experiments only contained two relatively small modalities, the model can, in theory, handle any number of modalities, and each modality can have any dimension. In fact, the framework can even handle missing data through the dropout methods explained in Section 2.5. This makes our model flexible so it can handle a slew of various multimodal scientific datasets. In practice, however, we do anticipate a few limitations to our framework. First, as disparate modalities have different complexities, it is possible for one modality to dominate other modalities in the latent space structure. If the dominating modality is the most informative modality, as is the case with our MNIST examples, then this behavior is perhaps desired. If undesired dominating modalities appear, then one can balance the modality contributions by weighting the reconstruction terms, as described in Section 2.5 and exhibited in the VDoS experiment (Experiment 4.3). The quality and properties of the modalities can also affect the model's capacity for cross-modal inference. Indeed for cross-modal applications, there must be some correlation across the modalities and each modality must contain enough information for PIMA's encoders to learn a mapping to the relevant region of latent space in order for the decoders to accurately reconstruct data across modalities.

We also anticipate limitations to result from the amount of time and computing resources available. The training times do scale with the size of the data, the number of epochs, and the number of parameters in the encoders and decoders. To give reference, the multimodal MNIST models in Experiment 4.1 had on the order of 20,000 parameters and trained with 40,000 epochs over the course of four days on a single GPU, without any attempts at parallelization (see Appendix A for more details on training for each experiment). Depending on size of datasets and GPU availability, we expect PIMA to readily handle a few multi-dimensional modalities accompanied by a number of 1D modalities and 0D modalities. The expert model of choice could also affect the capacity of PIMA; expert models likely contain fewer parameters than black-box neural networks, but may increase the computational time. For our purposes, we anticipate that bottlenecks in our future applications will come from the amount of data that we can fit onto a single GPU, but such issues can be mitigated through downsampling of data. Another limitation lies within the training. VAEs are notoriously difficult to train, and we do see some evidence of this in our unimodal MNIST experiments (Experiment 4.1). We note that our GMM prior is well-suited for classification problems where the number of classes in the data provides a natural choice for the number of clusters hyperparameter, but regression problems may require more extensive hyperparameter tuning to train PIMA effectively. We used Weights and Biases [4] to perform our hyperparameter tuning, where we usually terminated the sweeps after 50 agents or less.

In conclusion, this framework is widely applicable to a range of scientific disciplines where feature detection is crucial for tasks, ranging from predicting and attributing climate change to designing biochemical pathways at a molecular level. In addition to feature detection, this framework may be used for a variety of general purpose downstream tasks based on multimodal processing of scientific data. This framework provides an exciting platform for discovering data-driven scientific fingerprints which

may be combined with advances in automated experimentation to accelerate scientific discovery. Accordingly, the future of the present PIMA approach will involve extensions into other fields of study, broadening the impact of this multimodal approach.

Appendix A. Architecture, hyperparameters, and implementation.

A.1. **Model Architectures. For Experiment 4.1 (MNIST):** We employ relatively small convolutional architectures to serve as encoders for both modalities. The image modality encoder consists of 2 2D convolutional layers with 32 and 64 channels respectively, each with 3x3 kernels. We apply the exponential linear unit (ELU) activation function as well as batch normalization after each convolutional layer, then pass the output to a fully connected layer of size $encoding\_dim \times 2$ to enable an embedding into a representation of the mean and standard deviation of the input in the latent space. For the 1D modality encoder, in place of 2D convolutional layers, we use 1D convolutions with 8 and 16 channels in the respective layers, but with an otherwise identical architecture. The image decoder begins with a fully connected layer of appropriate size to be reshaped into 32 channels of 2D arrays, with each dimension having a length $\frac{1}{4}$ of the length of the number of pixels per side of the original image. The reshaped output of the initial dense layer is passed into a series of 3 deconvolutional layers with 64, 32, and 1 channel respectively, each with a kernel size of 3. The first 2 deconvolutional layers use a stride of 3 and a Rectified Linear Unit (ReLU) activation function, and the final deconvolutional layer uses a stride of 1. Zero padding is used to retain the input shape while traversing these layers.

For Experiment 4.2 (MNIST-NODEs): We employ the convolutional encoder architectures for both modalities that have the same specifications used in the MNIST experiments: (1) the image encoder with two 2D convolutional layers with 32 and 64 channels and 3×3 kernels, followed by ELU activation and batch normalization and (2) the 1D modality encoder with two 1D convolutional layers with 8 and 16 channels with 3-dimensional 1D kernel, followed by ELU and batch normalization. The image decoder also has the same specification as the previous experiments: a fully-connected layer, which converts a latent dimension to a 2D array, with each dimension having a length of $\frac{1}{4}$ of the width and the heights of the original input image and with 32 channels. Then three 2d transposed convolutional layers with channels 64, 32, and 1 are sequentially applied. For all the transposed convolutional layers, the kernel size is 3×3. The nonlinear activation for the two internal transposed convolutional layers is ReLU and the last layer has no nonlinear activation.

The solutions of ODEs are approximated by two separate POUNets for each of ten classes (20 in total): $\tilde{u}_1(t;c) = \sum_{i=1}^{n_{\text{part}}} \phi_i^{(1)}(t;c,\pi) \sum_{j=1}^{\dim(V)} \alpha_{i,j}^{(1)}(c)\psi_j^{(1)}(t;c)$ and $\tilde{u}_2(t;c) = \sum_{i=1}^{n_{\text{part}}} \phi_i^{(2)}(t;c,\pi) \sum_{j=1}^{\dim(V)} \alpha_{i,j}^{(2)}(c)\psi_j^{(2)}(t;c)$, where a partition of unity can be defined as $\Phi(t) = \{\phi_i(t)\}_{i=1}^{n_{\text{part}}}$ satisfying $\sum_i \phi_i(t) = 1$ and $\phi_i \leq 0$ for all t. Also, $\psi_j(\boldsymbol{s}) \in \mathbb{R}$ denotes a polynomial basis and $V = \text{span}(\{\psi_j\})$. In the experiments, we use 32 partitions ($n_{\text{part}} = 32$, 7 Taylor basis polynomials ($\dim(V) = 7$), and for the partition functions, we use a radial basis function (RBF) network, $\phi_i(t) = \exp\left(-\frac{|t-m^{(i)}|}{b^{(i)}}\right) / \sum_k \exp\left(-\frac{|t-m^{(k)}|}{b^{(k)}}\right)$, where $\{(m^{(i)}, b^{(i)}\}_{i=1}^{n_{\text{part}}}$ is a set of learnable parameters.

For Experiment 4.3 (VDoS): The VDoS 1D encoder is identical to the 1D modality encoder used for Experiment 4.1. The encoder for the 0D stress data

	learning rate	encoding dim	number of clusters	number of epochs
Experiment 4.1	3.125×10^{-7}	2	10	40,000
Experiment 4.2	1×10^{-3}	3	10	10,000
Experiment 4.3	9×10^{-5}	2	4	40,000

TABLE 4. Hyperparameters for each experiment.

consists of two fully connected layers, each of size 100 and followed by the rectified linear unit (ReLU) activation function. The output is passed to a fully connected layer of size $encoding\_dim \times 2$, representing the mean and standard deviation of the input embedded in the latent space. The decoders for both the 1D and 0D modalities consist of one full connected layer of size 64 followed by the ELU activation function and a final fully connected layer with an output size equal to the input dimension of each modality.

A.2. **Hyperparameters.** We used the Weights and Biases tool [4] to perform a hyperparameter search over learning rates and encoding dimensions for all datasets. We also used Weights and Biases to search over the number of clusters for Experiment 4.3 (VDoS).

Table 4 provides details of the final hyperparameters for each experiment.

A.3. **Dataset implementation.** For Experiment 4.1 (MNIST), we used a 90/10 train/validation split of the training data and the standard 10,000 held out test samples. Each line in the manufactured synthetic 1D modality contains 20 coordinates across $[0, 1]$ and each coordinate is normalized to a unit Gaussian across the entire dataset before encoding. For the expert decoder, we have one trainable parameter per cluster representing the digit label, which is used to generate a line over the same 20 coordinates for each cluster. Each coordinate is likewise normalized to a unit Gaussian.

For Experiment 4.2 (MNIST-NODE), we use the same training/validation/test split with the previous experiments: 90/10 split of the training set for training and validation and 10,000 held out test samples. For manufacturing the trajectories, we solve the cubic oscillator ODEs given in Eq. (20) with the step size 0.005 for 512 time steps. The initial condition is given as $[u_1, u_2] = [1.0, 1.0]$ and the TORCHDIFFEQ library [9] is used to solve the initial value problems.

For the VDoS dataset, we used an 81/9/10 train/validation/test data split. Each 1D VDoS data is an array length of 10794. All stress values and VDoS arrays were min-max normalized so values are between 0 and 1.

Each experiment used a different set of the model options outlined in Section 2.5. The choices for each experiment are summarized in Table 5. Our models are implemented in Python using PyTorch [40], and we leverage the Scikit-learn library [41] for data preparation and accuracy metrics, Scipy [54] for data preparation and the `linear_sum_assignment` implementation of the Hungarian method [26] for efficient computation of the unsupervised cluster accuracy. We visualize our results using the Matplotlib [18] and Seaborn [57] libraries. Training was performed on NVIDIA DGX-2 machines with each run executed on 1 graphics processing unit (GPU). A100 GPUs were used in this work. Training duration was a function of the number of epochs and the size of the datasets. In particular, Experiments 4.2 (MNIST-NODEs) and 4.3 (VDoS) ran for a matter of hours, whereas Experiment 4.1

	DOFs Rescaling	Single Decoder per Modality	Dropout	Expert Model
Experiment 4.1	✓	✓	✓	✓
Experiment 4.2		✓		✓
Experiment 4.3	✓	✓		

TABLE 5. Model decisions for each experiment.

(MNIST) ran for a couple days. We made no attempt to optimize parallel training to improve run time or training efficiency in this work.

Appendix B. Mathematical Derivations.

B.1. **Motivating the product-of-experts.** In Equation 6, we claim that

$$q(Z|X_1, ..., X_M) = q(Z)^{1-M}\Pi_{m=1}^{M} q(Z|X_m),$$

under the assumption that the modalities are independent of each other. We show Equation 6 by the following:

$$\begin{aligned}
q(Z|X_1, \ldots, X_M) &= \frac{q(X_1, \ldots, X_M|Z)q(Z)}{q(X_1, \ldots, X_M)} \\
&= \frac{q(X_1|X_2, \ldots, X_M, Z)q(X_2|X_3, \ldots, X_M, Z)\cdots q(X_M|Z)q(Z)}{q(X_1, \ldots, X_M)} \\
&= q(X_1|Z)q(X_2|Z)\cdots q(X_M|Z)\frac{q(Z)}{q(X_1)\cdots q(X_M)} \\
&= \frac{q(Z|X_1)q(X_1)}{q(Z)}\frac{q(Z|X_2)q(X_2)}{q(Z)}\cdots\frac{q(Z|X_M)q(X_M)}{q(Z)}\frac{q(Z)}{q(X_1)\cdots q(X_M)} \\
&= q(Z)^{1-M}\Pi_{m=1}^{M} q(Z|X_m)
\end{aligned}$$

B.2. **Derivation of ELBO.** To derive a closed form expression for the single sample ELBO

$$ELBO = \mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log \frac{p(X_1, ..., X_M, Z, C)}{q(Z, C|X_1, ..., X_M)}\right] \tag{27}$$

we apply the separability assumptions in Equation 2. These separability assumptions provide an additive decomposition of the loss function,

$$\begin{aligned}
ELBO &= \mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log p(X_1, ..., X_M, Z, C)\right] \\
&\quad - \mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log q(Z, C|X_1, ..., X_M)\right] \\
&= \sum_{m=1}^{M} \mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log p(X_m|Z, C)\right] + \mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log p(Z|C)\right] \\
&\quad + \mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log p(C)\right] - \mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log q(Z|X_1, ..., X_M)\right] \\
&\quad - \mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log q(C|X_1, ..., X_M)\right].
\end{aligned} \tag{28}$$

For convenience we denote $\mathbb{E}_{q(Z,C|X_1,...,X_M)} = \mathbb{E}_q$. The separability assumptions therefore decompose the remainder of the ELBO terms into constituent expectations of the form

$$\mathbb{E}_q\left[\log f(Z, C)\right] = \sum_C \int_{\mathbb{R}^l} f(Z, C)\log g(Z, C)\, dZ \tag{29}$$

which may be integrated exactly for the Gaussian/categorical f and g appearing in the ELBO. The only term which may not be immediately computed is $\mathbb{E}_q\left[\log q(C|X_1, ..., X_M)\right]$. The lack of a reparameterization trick for the categorical distribution precludes backpropagation into the encoder, forcing us to consider an encoder which only provides predictions for Z. While there are options to use

e.g. a regularized Gumbel-softmax approximation to the categorical distribution [20], we would lose the tractability of the closed form expression for the ELBO. Instead we follow [21] and approximate $q(C|X_1,...,X_M) = p(C|Z)$ using the following justification.

If given Z we assume that $X_1,\dots,X_M$ are independent of C, i.e. we assume that $p(X_1,\dots,X_M|Z,C) = p(X_1,\dots,X_M|Z)$, then we can rewrite the ELBO:

$$\begin{aligned} ELBO &= \mathbb{E}_{q(Z,C|X_1,\dots,X_M)}\left[\log \frac{p(X_1,...,X_M,Z,C)}{q(Z,C|X_1,...,X_M)}\right] \\ &= \mathbb{E}_{q(Z,C|X_1,\dots,X_M)}\left[\log \frac{p(X_1,...,X_M|Z)p(Z)}{q(Z|X_1,...,X_M)} + \log \frac{p(C|Z)}{q(C|X_1,...,X_M)}\right] \\ &= \int_{\mathbb{R}^l} q(Z|X_1,\dots,X_M)\log \frac{p(X_1,...,X_M|Z)p(Z)}{q(Z|X_1,...,X_M)} \\ &- q(Z|X_1,\dots,X_M)D_{KL}(q(C|X_1,...,X_M)||p(C|Z))dZ, \end{aligned} \tag{30}$$

where we seek extremal points with respect to C. The first term is independent of C, and the second term takes zero value when $q(C|X_1,...,X_M) = p(C|Z)$, providing the desired maximum. We caution however that this holds only at local minima of the loss landscape and requires the additional assumption that $X_1,\dots,X_M$ are independent of C given Z, but empirically has been shown to perform well as an estimator.

For completeness, we gather from [21] the various integral formulas required to compute the expectations in closed form with modifications for our multimodal setting.

Lemma B.1. *Given Gaussian distributions* $f(Z) = \mathcal{N}(Z;\mu_1,\sigma_1\mathbf{I})$ *and* $g(Z) = \mathcal{N}(Z;\mu_2,\sigma_2\mathbf{I})$

$$\int_{\mathbb{R}^l} f(Z)\log g(Z)\,dZ = -\frac{1}{2}\sum_j \log 2\pi\sigma_{2;j}^2 + \frac{\sigma_{1;j}^2}{\sigma_{2;j}^2} + \frac{(\mu_{1;j}-\mu_{2;j})^2}{\sigma_{2;j}^2}. \tag{31}$$

Lemma B.2.

$$\mathbb{E}_{q(Z,C|X_1,\dots,X_M)}\left[\log p(Z|C)\right] = \sum_{c=1}^{N} q(C{=}c|X_1,...,X_M)\int_{\mathbb{R}^l} q(Z|X_1,...,X_M)\log p(Z|C{=}c)dZ, \tag{32}$$

where the integrand may be computed from Lemma B.1.

Lemma B.3.

$$\begin{aligned} \mathbb{E}_{q(Z,C|X_1,\dots,X_M)}\left[\log p(C)\right] &= \int_{\mathbb{R}^l} q(Z|X_1,...,X_M)dZ\sum_{c=1}^{N} q(C=c|X_1,...,X_M)\log \pi_c \\ &= \sum_{c=1}^{N} q(C{=}c|X_1,...,X_M)\log \pi_c. \end{aligned} \tag{33}$$

Lemma B.4.

$$\begin{aligned} \mathbb{E}_q\left[\log q(Z|X_1,...,X_M)\right] &= \sum_{c=1}^{N} q(C{=}c|X_1,...,X_M)\int_{\mathbb{R}^l} q(Z|X_1,...,X_M)\log q(Z|X_1,...,X_M)dZ, \\ &= \int_{\mathbb{R}^l} q(Z|X_1,...,X_M)\log q(Z|X_1,...,X_M)dZ, \end{aligned} \tag{34}$$

where the integrand may be computed from Lemma B.1.

Lemma B.5.

$$\mathbb{E}_q\left[\log q(C|X_1,...,X_M)\right] = \int_{\mathbb{R}^l} q(Z|X_1,...,X_M)dZ \sum_{c=1}^{N} q(C{=}c|X_1,...,X_M)\log q(C{=}c|X_1,...,X_M),$$

$$= \sum_{c=1}^{N} q(C{=}c|X_1,...,X_M)\log q(C{=}c|X_1,...,X_M). \tag{35}$$

where the integrand may be computed from Lemma B.1.

The only term not addressed by the above lemmas is the reconstruction term, that is $\mathbb{E}_{q(Z,C|X_1,...,X_M)}\left[\log p(X_1,\ldots,X_M|Z,C)\right]$ (denoted below as $\mathbb{E}_q\left[\log p(\mathbf{X}|Z,C)\right]$). Using our separability assumptions and Monte Carlo methods, we reduce this term via the following:

$$\begin{aligned}
&\mathbb{E}_q\left[\log p(\mathbf{X}|Z,C)\right]\\
&= \sum_{m=1}^{M} \mathbb{E}_{q(Z,C|X_1,\ldots,X_M)}\left[\log p(X_m|Z,C)\right]\\
&= \sum_{m=1}^{M} \mathbb{E}_{q(Z|X_1,\ldots,X_M)}\left[\sum_{c=1}^{N} q(C{=}c|X_1,\ldots,X_M)\log p(X_m|Z,C{=}c)\right]\\
&= \sum_{m=1}^{M} \mathbb{E}_{q(Z|X_1,\ldots,X_M)}\left[\sum_{c=1}^{N} \gamma_c\left(\frac{-d_m}{2}\log 2\pi - \frac{1}{2}\sum_{j=1}^{d_m}\left(\log\hat{\sigma}^2_{m,c;j} + \frac{(X_{m;j}-\hat{\mu}_{m,c;j})^2}{\hat{\sigma}^2_{m,c;j}}\right)\right)\right]\\
&\approx \sum_{m=1}^{M}\sum_{d=1}^{D}\sum_{c=1}^{N} \gamma_c^{(d)}\left[\frac{-d_m}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{d_m}\left(\log\hat{\sigma}^{2(d)}_{m,c;j} + \frac{(x^{(d)}_{m;j}-\hat{\mu}^{(d)}_{m,c;j})^2}{\hat{\sigma}^{2(d)}_{m,c;j}}\right)\right]
\end{aligned}$$

For all terms, $q(C{=}c|X_1,...,X_M)$ is calculated via the posterior estimator γ_c given in Equation (14). Furthermore, constant terms do not affect the minimization of the loss and are consequently dropped in $\mathcal{L}$. Similarly the entire loss ($Loss$) is scaled by 2.

Acknowledgments. The authors thank Warren Davis, Anthony Garland and Lekha Patel for providing guidance on the variational inference framework and review of the manuscript and Kat Reiner and Greg Geller for providing computing support. All authors acknowledge funding under the Beyond Fingerprinting Sandia Grand Challenge Laboratory Directed Research and Development program. S. Saha acknowledges supplemental funding from CMMI-1934367 to conduct this research. The work of N. Trask, E. Walker, and J. Actor is supported by the U.S. Department of Energy, Office of Advanced Computing Research under the "Scalable and Efficient Algorithms - Causal Reasoning, Operators and Graphs" (SEA-CROGS) project. This work was supported in part by the Center for Integrated Nanotechnologies, an Office of Science user facility operated for the U.S. Department of Energy. This article has been co-authored by employees of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employees co-own all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan https://www.energy.gov/downloads/doe-public-access-plan. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. SAND number: SAND2023-08755O

REFERENCES

[1] S. Amal, L. Safarnejad, J. A. Omiye, I. Ghanzouri, J. H. Cabot and E. G. Ross, Use of multi-modal data and machine learning to improve cardiovascular disease care, *Frontiers in Cardiovascular Medicine*, **2** (2022).

[2] S. An, M. Lee, S. Park, H. Yang and J. So, An ensemble of simple convolutional neural network models for MNIST digit recognition, arXiv preprint, arXiv:2008.10400, 2020.

[3] T. Baltrušaitis, C. Ahuja and L.-P. Morency, Multimodal machine learning: A survey and taxonomy, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **41** (2018), 423-443.

[4] L. Biewald, *Experiment Tracking with Weights and Biases*, Software available from wandb.com, 2020.

[5] B. L. Boyce and M. D. Uchic, Progress toward autonomous experimental systems for alloy development, *MRS Bulletin*, **44** (2019), 273-280.

[6] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins and A. Lerchner, Understanding disentangling in β-VAE, arXiv preprint, arXiv:1804.03599, 2018.

[7] A. Chakraborty, P. Nandi and B. Chakraborty, Fingerprints of the quantum space-time in time-dependent quantum mechanics: An emergent geometric phase, *Nuclear Phys. B*, **975** (2022), Paper No. 115691, 27 pp.

[8] R. T. Chen, X. Li, R. Grosse and D. Duvenaud, Isolating sources of disentanglement in vaes, in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018, 2615-2625.

[9] R. T. Chen, Y. Rubanova, J. Bettencourt and D. K. Duvenaud, Neural ordinary differential equations, *Advances in Neural Information Processing Systems*, **31** (2018).

[10] J. Cioffi and T. Kailath, Fast, recursive-least-squares transversal filters for adaptive filtering, *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **32** (1984), 304-337.

[11] N. Dilokthanakul, P. A. Mediano, M. Garnelo, M. C. Lee, H. Salimbeni, K. Arulkumaran and M. Shanahan, Deep unsupervised clustering with Gaussian mixture variational autoencoders, arXiv preprint, arXiv:1611.02648, 2016.

[12] F. Dos Santos Rodrigues, G. Delgado, T. Santana de Costa and L. Tasic, Applications of fluorescence spectroscopy in protein conformational changes and intermolecular contacts, *BBA Advances*, **3** (2023).

[13] M. El Hariri El Nokab and K. Sebakhy, Solid state nmr spectroscopy a valuable technique for structural insights of advanced thin film materials: A review, *Nanomaterials (Basel)*, **11** (2021).

[14] D. Gao, J. Huang, X. Lin, D. Yang, Y. Wang and H. Zheng, Phase transitions and chemical reactions of octahydro-1,3,5,7-tetranitro-1,3,5,7-tetrazocine under high pressure and high temperature, *RSC Advances*, **9** (2019).

[15] K. Hasselmann, Multi-pattern fingerprint method for detection and attribution of climate change, *Climate Dynamics*, **13** (1997), 601-611.

[16] G. Hegerl, F. Zwiers, P. Braconnot, N. P. Gillett, Y. M. Luo, J. M. Orsini, N. Nicholls, J. E. Penner and P. A. Stott, *Understanding and Attributing Climate Change*, 2007.

[17] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed and A. Lerchner, beta-vae: Learning basic visual concepts with a constrained variational framework, in *5th International Conference on Learning Representations, ICLR*, **2017** (2017).

[18] J. D. Hunter, Matplotlib: A 2d graphics environment, *Computing in Science & Engineering*, **9** (2007), 90-95.

[19] O. Isayev, D. Fourches, E. N. Muratov, C. Oses, K. Rasch, A. Tropsha and S. Curtarolo, Materials cartography: Representing and mining materials space using structural and electronic fingerprints, *Chemistry of Materials*, **27** (2015), 735-743.

[20] E. Jang, S. Gu and B. Poole, Categorical reparameterization with gumbel-softmax, arXiv preprint, arXiv:1611.01144, 2016.

[21] Z. Jiang, Y. Zheng, H. Tan, B. Tang and H. Zhou, Variational deep embedding: An unsupervised and generative approach to clustering, in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, 1965-1972.

[22] M. I. Jordan and R. A. Jacobs, Hierarchical mixtures of experts and the em algorithm, *Proceedings of 1993 International Conference on Neural Networks*, **6** (1993), 181-214.

[23] H. Kim and A. Mnih, Disentangling by factorising, in *International Conference on Machine Learning, PMLR,* 2018, 2649-2658.

[24] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint, arXiv:1412.6980, 2014.
[25] D. P. Kingma and M. Welling, *Auto-Encoding Variational Bayes*, in 2nd International Conference on Learning Representations, ICLR 2014, 2014.
[26] H. W. Kuhn, The hungarian method for the assignment problem, *Naval Research Logistics Quarterly*, **2** (1955), 83-97.
[27] I. E. Lagaris, A. Likas and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, *IEEE Transactions on Neural Networks*, **9** (1998), 987-1000.
[28] Y. LeCun, C. Cortes and C. Burges, Mnist handwritten digit database, ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist, 2 (2010).
[29] D. B. Lee, D. Min, S. Lee and S. J. Hwang, *Meta-GMVAE: Mixture of Gaussian VAE for Unsupervised Meta-Learning*, in International Conference on Learning Representations, 2020.
[30] K. Lee, N. Trask and P. Stinis, Structure-preserving sparse identification of nonlinear dynamics for data-driven modeling, in *Mathematical and Scientific Machine Learning, PMLR*, 2022, 65-80.
[31] K. Lee, N. A. Trask, R. G. Patel, M. A. Gulian and E. C. Cyr, Partition of unity networks: Deep hp-approximation, arXiv preprint, arXiv:2101.11256, 2021.
[32] A. Liu, W. Zhu, D. Tsai and N. I. Zheludev, Micromachined tunable metamaterials: A review, *Journal of Optics*, **14** (2012), p. 114009.
[33] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf and O. Bachem, Challenging common assumptions in the unsupervised learning of disentangled representations, in *International Conference on Machine Learning, PMLR*, 2019, 4114-4124.
[34] F. Locatello, S. Bauer, M. Lucic, G. R atsch, S. Gelly, B. Schölkopf and O. Bachem, *Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations*, in International Conference on Machine Learning, PMLR, 2019.
[35] L. Lu, P. Jin and G. E. Karniadakis, Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators, arXiv preprint, arXiv:1910.03193, 2019.
[36] Z. Mao, L. Lu, O. Marxen, T. A. Zaki and G. E. Karniadakis, Deepm&mnet for hypersonics: Predicting the coupled flow and finite-rate chemistry behind a normal shock using neural-network approximation of operators, *Journal of Computational Physics*, **447** (2021), p. 110698.
[37] S. M. Mennen and et al, The evolution of high-throughput experimentation in pharmaceutical development and perspectives on the future, *Organic Process Research & Development*, **23** (2019), 1213-1242.
[38] E. J. Miittemeijer and P. Scardi, *Diffraction Analysis of the Microstructure of Materials*, Springer-Verlag, Berlin, 2004.
[39] P. Nikolaev, D. Hooper, F. Webbed, R. Rao, K. Decker, M. Krein, J. Poleski, R. Barto and B. Maruyama, Autonomy in materials research: a case study in carbon nanotube growth, *Npj Computational Materials*, **2** (2016).
[40] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, *Advances in Neural Information Processing Systems*, **32** (2019).
[41] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al., Scikit-learn: Machine learning in python, *The Journal of Machine Learning Research*, **12** (2011), 2825-2830.
[42] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens and L. Carin, Variational autoencoder for deep learning of images, labels and captions, *Advances in Neural Information Processing Systems*, **29** (2016), 2352-2360.
[43] A. Quaglino, M. Gallieri, J. Masci and J. Koutník, *SNODE: Spectral Discretization of Neural ODEs for System Identification*, in International Conference on Learning Representations, 2020.
[44] M. Raissi, P. Perdikaris and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *Journal of Computational Physics*, **378** (2019), 686-707.
[45] D. Rao, F. Visin, A. Rusu, R. Pascanu, Y. W. Teh and R. Hadsell, Continual unsupervised representation learning, *Advances in Neural Information Processing Systems*, **32** (2019), 7647-7657.

[46] D. J. Rezende, S. Mohamed and D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in *International Conference on Machine Learning, PMLR*, 2014, 1278-1286.
[47] Y. Shi, B. Paige, P. Torr, et al., Variational mixture-of-experts autoencoders for multi-modal deep generative models, *Advances in Neural Information Processing Systems*, **32** (2019).
[48] R. D. Sochol, E. Sweet, C. C. Glick, S.-Y. Wu, C. Yang, M. Restaino and L. Lin, 3d printed microfluidics and microelectronics, *Microelectronic Engineering*, **189** (2018), 52-68.
[49] K. Sohn, H. Lee and X. Yan, Learning structured output representation using deep conditional generative models, *Advances in Neural Information Processing Systems*, **28** (2015), 3483-3491.
[50] T. M. Sutter, I. Daunhawer and J. E. Vogt, Generalized multimodal ELBO, in *9th International Conference on Learning Representations*, ICLR, 2021.
[51] M. Suzuki, K. Nakayama and Y. Matsuo, Joint multimodal learning with deep generative models, in *5th International Conference on Learning Representations*, ICLR 2017, 2017.
[52] N. Trask, A. Huang and X. Hu, Enforcing exact physics in scientific machine learning: A data-driven exterior calculus on graphs, *J. Comput. Phys.*, **456** (2022), Paper No. 110969, 19 pp.
[53] R. Vedantam, I. Fischer, J. Huang and K. Murphy, *Generative Models of Visually Grounded Imagination*, in 6th International Conference on Learning Representations, ICLR, 2018.
[54] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al., Scipy 1.0: fundamental algorithms for scientific computing in python, *Nature Methods*, **17** (2020), 261-272.
[55] D. Vizoso, G. Subhash, K. Rajan, and R. Dingreville, Connecting vibrational spectroscopy to atomic structure via supervised manifold learning: Beyond peak analysis, *Chem. Mater.*, **35** (2023), 1186-1200.
[56] S. Wang, H. Wang and P. Perdikaris, Learning the solution operator of parametric partial differential equations with physics-informed DeepONets, *J. Comput. Phys.*, **475** (2023), Paper No. 111855, 18 pp.
[57] M. L. Waskom, Seaborn: Statistical data visualization, *Journal of Open Source Software*, **6** (2021), p3021.
[58] C. Weidenthaler, Pitfalls in the characterization of nanoporous and nanosized materials, *Nanoscale*, **3** (2011), 792-810.
[59] M. Wu and N. Goodman, Multimodal generative models for scalable weakly-supervised learning, *Advances in Neural Information Processing Systems*, **31** (2018).
[60] J. Xie, R. Girshick and A. Farhadi, Unsupervised deep embedding for clustering analysis, in *International Conference on Machine Learning, PMLR*, 2016, 478-487.

Received November 2023; 1st and 2nd revision February 2024; early access April 2024.