Mitigating Modality Collapse in Multimodal VAEs via Impartial Optimization

Adrián Javaloy [1] **Maryam Meghdadi** [1] **Isabel Valera** [1 2]

Abstract

A number of variational autoencoders (VAEs) have recently emerged with the aim of modeling multimodal data, e.g., to jointly model images and their corresponding captions. Still, multimodal VAEs tend to focus solely on a subset of the modalities, e.g., by fitting the image while neglecting the caption. We refer to this limitation as *modality collapse*. In this work, we argue that this effect is a consequence of *conflicting gradients* during multimodal VAE training. We show how to detect the sub-graphs in the computational graphs where gradients conflict (*impartiality blocks*), as well as how to leverage existing gradient-conflict solutions from multitask learning to mitigate modality collapse. That is, to ensure *impartial optimization* across modalities. We apply our training framework to several multimodal VAE models, losses and datasets from the literature, and empirically show that our framework significantly improves the reconstruction performance, conditional generation, and coherence of the latent space across modalities.

1. Introduction

Variational autoencoders (VAEs) (Kingma & Welling, 2014) enjoy great success in domains such as images, text, and temporal data (Vahdat & Kautz, 2020; Xu et al., 2017; Mehrasa et al., 2019). Their application to multimodal data, e.g., to model images and their captions, remains a challenge, since models tend to accurately fit only a subset of the modalities, neglecting the rest. We here refer to this problem as *modality collapse*.

To overcome this issue, a number of tailored VAE models for tabular (Nazabal et al., 2020; Ma et al., 2020) and multimodal data (Shi et al., 2019; Sutter et al., 2021) have emerged over the years. Interestingly, Nazabal et al. (2020) hypothesized that modality collapse is a result of the disparities between gradients across modalities during training.

Following this inkling, we study modality collapse as a result of gradient conflicts in specific blocks of the computational graph, which we here call impartiality blocks (§3). To address this problem, we propose a multimodal VAE training pipeline, which leverages existing multitask learning solutions (§3.1) to favor an impartial optimization process that does not favor a subset of modalities over the rest. We show the flexibility of our approach by applying our pipeline to several existing VAE models previously proposed in the literature to fit multimodal and tabular data (§4). Our empirical results on different datasets, models and training losses (§5) show that impartial optimization results in a more accurate fit of the marginal, joint and conditional distributions over all modalities.

Notation. We use the set indexing notation $\boldsymbol{x}_A$, where A is a set of indexes, e.g., $\mathbf{Z}_{1:K}$ denotes a sequence from 1 to K. We denote by $\mathbf{1}$ a vector full of ones, and by $[\cdot]$ the concatenation operator. $\mathcal{P}(D)$ denotes the power-set of D elements, and $|\mathcal{A}|$ the number of elements of the set $\mathcal{A}$.

2. Preliminaries

Multimodal data. In this work, we consider multimodal data, i.e., data coming from different sources and/or forms. Specifically, we consider as input data i.i.d. samples from a multimodal random variable (r.v.) $\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D\right]$, where the d-th modality is fully described by the r.v. $\mathbf{x}_d$. Note that we do not make any assumptions on the modalities, allowing for $\mathbf{x}_d$ of different sizes (e.g., images and their labels) and types (e.g., continuous vs. discrete).

We refer to $\mathbf{X}$ as heterogeneous when each modality $\mathbf{x}_d$ in $\mathbf{X}$ is unidimensional and can be of a different statistical type (e.g., normal or categorical). Thus, we consider heterogeneous data as a special case of multimodal data. Notice however that in the literature heterogeneous data is often studied independently (e.g., Nazabal et al. (2020); Ma et al. (2020)) of multimodal problems, and it is prominent in applications that deal with real-world tabular data.

[1]Department of Computer Science, Saarland University, Germany [2]MPI for Software Systems, Saarland, Germany. Correspondence to: Adrián Javaloy <ajavaloy@cs.uni-saarland.de>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

Variational autoencoders (VAEs) (Kingma & Welling, 2014) are probabilistic models that learn to model the data by assuming the existence of some latent variable $\mathbf{Z}$. Specifically, they learn the likelihood function that best approximates the input (decoder), $p_\theta(\mathbf{X}|\mathbf{Z})$, and an approximation to the posterior distribution of $\boldsymbol{Z}$ (encoder), $q_\phi(\mathbf{Z}|\mathbf{X})$. During learning, VAEs maximize a function of the following form:

$$L(\theta,\phi) = \mathbb{E}_{\mathbf{X}}\left[\mathbb{E}_{\mathbf{Z}_{1:K}\sim q_\phi}\left[\log \frac{1}{K}\sum_{k=1}^{K}\frac{p_\theta(\mathbf{X},\mathbf{Z}_k)}{q_\phi(\mathbf{Z}_k|\mathbf{X})}\right]\right] \quad (1)$$

where $p_\theta(\mathbf{X},\mathbf{Z}) = p_\theta(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})$, and $\mathbf{Z}_{1:K}$ is an i.i.d sequence of length K. This formulation includes the original ELBO (Kingma & Welling, 2014), as well as the importance weighted loss (IWAE) from Burda et al. (2016).

One important detail here is that the functional form of p_θ (and q_ϕ) is usually fixed beforehand—e.g., as a normal distribution—while a neural network determines its parameters $\boldsymbol{\eta}$. Importantly, when dealing with multimodal data, the usual practice is to assume that the likelihood fully factorizes across modalities, i.e.,

$$p_\theta(\mathbf{X}|\mathbf{Z}) = \prod_{d=1}^{D} p_d(\mathbf{x}_d;\ \boldsymbol{\eta}_d(\mathbf{Z};\theta)), \quad (2)$$

where p_d accounts for the statistical properties of $\mathbf{x}_d$.

2.1. State-of-the-art

Heterogeneous data: The most prominent VAE models found in the literature are probably HI-VAE (Nazabal et al., 2020) (see, §4.1), originally designed for missing data imputation tasks, and VAEM (Ma et al., 2020), designed instead for active data acquisition tasks. More recently, SHIVAE (Barrejón et al., 2021) has been introduced as an extension of HI-VAE to deal with temporal data.

Multimodal data: We focus in this work on mixture-based VAE models (see §4.2), which are at the moment an active area of research. While MVAE (Wu & Goodman, 2018), MMVAE (Shi et al., 2019), and MoPoE (Sutter et al., 2021) are the models to beat, different extensions compatible with our proposed framework keep coming up, e.g., using alternative training functions (Shi et al., 2021; Sutter et al., 2020). For a survey on other multimodal methods refer to the work of, e.g., Guo et al. (2019); Baltrušaitis et al. (2018).

3. Impartial Optimization in Multimodal VAEs

In this section, we investigate the standard assumptions and goals of multimodal VAEs, as well as discuss the optimization challenges that cause modality collapse. Then, we propose a flexible learning approach to palliate this issue.

First, let us bring multimodal modeling to context. When we think of multimodal applications (e.g., missing data imputation, or joint data generation) these are tasks that involve not only explaining the different modalities in the data, but jointly capturing the interactions and dependencies between each pair of modalities. That is, the main goal (often implicit) of multimodal learning is thus to accurately approximate the marginal, joint and conditional distributions over all modalities.

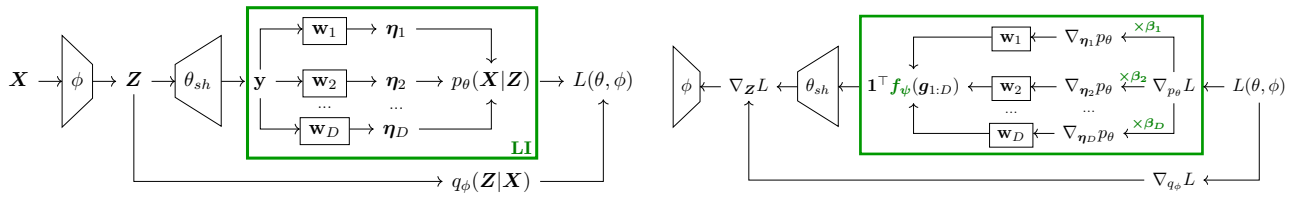
Likelihood Impartiality (LI): In order to do so, it is essential to accurately fit the likelihood of all modalities without neglecting any of them. *We thus aim for a learning process that does not prioritize the learning of, or equivalently, that is impartial to, the likelihood of the different modalities.*

We argue here that the reason why likelihood impartiality is often not satisfied by multimodal VAE training is the computational graph resulting from the likelihood factorization in Eq. 2. We illustrate this idea in Fig. 1a, where we highlight the problematic sub-graph in the computational graph, referred to as **impartiality block**.

As an example, assume here that the last layer of the decoder is a linear layer with parameter $\boldsymbol{W}$, and let us denote by θ_{sh} the rest of the decoder parameters, which are *shared* across all modalities. Then, we can write the likelihood parameters as $\boldsymbol{\eta} = \sigma(\boldsymbol{y}\boldsymbol{W})$, where $\boldsymbol{y}$ is the output of the decoder up to the last shared layer, and σ is an element-wise transformation to ensure that each parameter satisfies its distributional constraints (e.g., positive variance). Making now the modality dependency explicit we can write, $\left[\boldsymbol{\eta}_1,\boldsymbol{\eta}_2,\ldots,\boldsymbol{\eta}_D\right] = \sigma(\boldsymbol{y}\left[\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_D\right])$, where it is now clear that all modalities share $\boldsymbol{y}$, while the parameters $\boldsymbol{w}_d$ are exclusive of the likelihood for the d-th modality.

An impartiality block (green square in Fig. 1a) encloses a sub-graph in which a split-and-merge pattern across modalities appears, which we will recurrently observe later in §4. In the forward pass, the impartiality block takes a shared $\boldsymbol{y}$ as input, which is *independently* fed to each modality-specific "head" to compute $\boldsymbol{\eta}_d$. Then, these computations are collected to compute a common output, the total likelihood $p_\theta(\mathbf{X}|\mathbf{Z})$. Note that, outside this block, all computations are shared across modalities.

Impartiality blocks play an essential role on explaining modality collapse in multimodal VAEs. First, we need to understand the effect of the split-and-merge pattern on the update rule of the shared parameters during optimization. That is, we need to compute the gradient of $L(\theta,\phi)$ w.r.t. θ_{sh} (similar computations follow in the case of ϕ), *passing through the computational block*:



(a) Forward pass.

(b) Modified backward pass.

Figure 1. Schematic computational graph of a basic multimodal VAE: (a) forward pass, taking $\mathbf{X}$ as input and producing the training objective; (b) backward pass, modified to alleviate modality collapse (see §3.1). The impartiality block, which encloses conflicting gradients, as well as the modifications proposed in this work to tackle them, are highlighted in green.

$$\begin{aligned}\nabla_{\theta_{sh}} L(\theta,\phi) &= \nabla_{\theta_{sh}} \boldsymbol{y}\, \nabla_{\boldsymbol{y}} \boldsymbol{\eta}\, \nabla_{\boldsymbol{\eta}} p_\theta\, \nabla_{p_\theta} L \\ &= \nabla_{\theta_{sh}} \boldsymbol{y} \left(\sum_{d=1}^{D} \nabla_{\boldsymbol{y}} \boldsymbol{\eta}_d\, \nabla_{\boldsymbol{\eta}_d} p_\theta \right) \nabla_{p_\theta} L \\ &= \nabla_{\theta_{sh}} \boldsymbol{y} \sum_d \boldsymbol{g}_d, \end{aligned} \tag{3}$$

where $\boldsymbol{g}_d \coloneqq \nabla_{\boldsymbol{y}} \boldsymbol{\eta}_d\, \nabla_{\boldsymbol{\eta}_d} p_\theta\, \nabla_{p_\theta} L$ is the gradient of the loss w.r.t. $\boldsymbol{y}$ through the d-th modality, as it is computed during back-propagation (Rumelhart et al., 1986).

Equation 3 reveals why modality collapse may occur during training. Intuitively, each gradient $\boldsymbol{g}_d$ represents the update direction that the model should follow to better explain the d-th modality. However, if there exist large discrepancies between different gradients $\boldsymbol{g}_d$, i.e., in the presence of **conflicting gradients**, the overall gradient computation (namely, the sum $\sum_d \boldsymbol{g}_d$) can benefit some modalities over others, leading to an update of the shared parameters that prioritize a subset of the modalities.

Therefore, our goal is to ensure impartiality across modalities in the computations that output the impartiality block, such that no modality is neglected. Hence, its name. We remark here that the conflicting gradient problem is not exclusive to multimodal VAEs, and it has been studied in areas such as multitask learning (MTL). Refer to Appendix A for an overview of MTL.

3.1. Our Approach

In this section, we propose to modify the backward pass of the impartiality block during training (since all outer computations are shared across modalities). We do so by leveraging existing MTL solutions to enforce impartial optimization, and thus mitigate modality collapse.

We illustrate the proposed approach in Fig. 1b and Alg. 1, highlighting in green those parts that differ from usual back-propagation (see Appendix B for a general formulation). We propose two modifications within the impartiality block to bring impartiality with respect to the modalities:

- **Local step:** Backpropagating through the heads, we re-weigh the gradients with respect to the likelihood parameters $\boldsymbol{\eta}_d$ (which are local to each modality) by a factor of $\beta_d \in \mathbb{R}^+$ to keep them at a comparable scale. We choose β_d to be the number of dimensions of $\mathbf{x}_d$, similar to solutions in the literature (e.g., Shi et al. (2019)). Note, however, that in prior work re-weighing was an ad-hoc fix in the forward pass (rather than in the backward pass), despite breaking probabilistic assumptions.[1] This step is also similar to loss balance in MTL. Here, we opt for a simple approach as it works well in practice, but more complex approaches could be also adapted to our framework, e.g., those proposed by Kendall et al. (2018), Chennupati et al. (2019), and Liu et al. (2021b).
- **Global step.** Instead of propagating to the shared parameters (θ_{sh} and ϕ) the gradient with respect to (the shared or global representation) $\boldsymbol{y}$, we leverage existing MTL solutions to avoid conflicting gradients. These solutions can be described as a (parameterized) function f_ψ that takes a sequence of gradients $\boldsymbol{g}_{1:D}$, and returns another of equal length $\tilde{\boldsymbol{g}}_{1:D} \coloneqq f_\psi(\boldsymbol{g}_{1:D})$, where the function f_ψ is selected to mitigate conflicts (e.g., in magnitude or direction) in $\boldsymbol{g}_{1:D}$. We thus apply f_ψ to the gradients with respect to $\boldsymbol{y}$, and backpropagate $\sum_d \tilde{\boldsymbol{g}}_d$ instead of $\sum_d \boldsymbol{g}_d$. Note that the function f_ψ is determined by the specific MTL method that is applied.

To sum up, we address modality collapse within each impartiality block by: i) scaling local gradients w.r.t. $\boldsymbol{\eta}_d$ by β_d to make them comparable; and ii) leveraging existing MTL solutions to modify the gradients w.r.t. $\boldsymbol{y}$ such that they do not conflict, propagating this impartial gradients to the shared parameters.

There are two important remarks to make here. First, the local character of impartiality blocks is in stark contrast with traditional MTL: we do not make any assumption on the outer computational graph, nor the number of blocks in the graph. Second, the optimal choice of f_ψ depends on the

[1] Specifically, that the likelihood integrates to one.

Algorithm 1 Backward pass within the impartiality block.

Input: Output gradient, $\nabla_{p_\theta} L$.
for $d = 1$ **to** D **do**
  $\boldsymbol{h}_d \leftarrow \boldsymbol{\beta_d} \nabla_{\eta_d} p_\theta \, \nabla_{p_\theta} L$
  $\nabla_{\omega_d} L \leftarrow \nabla_{\omega_d} \boldsymbol{\eta}_d \cdot \boldsymbol{h}_d$
  $\boldsymbol{g}_d \leftarrow \nabla_{\boldsymbol{y}} \boldsymbol{\eta}_d \cdot \boldsymbol{h}_d$
end for
$\tilde{\boldsymbol{g}}_{1:D} \leftarrow \boldsymbol{f_\psi}(\boldsymbol{g}_{1:D})$
return $\sum_d \tilde{\boldsymbol{g}}_d$

problem setting, with no clear winner among existing MTL solutions. Therefore, we treat the choice of algorithm f_ψ as a hyperparameter, which we need to cross-validate.

3.2. Conflicting Gradients Solutions

Here, we briefly discuss the MTL solutions for conflicting gradients considered in the global step from §3.1. Refer to Appendix A for a full description.

As explained above, these solutions modify the gradients through a function $f_\psi : \mathbb{R}^{D \times l} \rightarrow \mathbb{R}^{D \times l}$, where l is the dimension of the latent variable $\mathbf{Z}$, and the input are the gradients for each task, stacked on the first dimension.

Moreover, they can be classified into two main categories, depending on the way they deal with conflicting gradients:

- Scale-aware algorithms use a function f_ψ that scales each gradient $\boldsymbol{g}_d$ according to a given criterion, thus changing the magnitude of the gradient. That is, f_ψ replaces each $\boldsymbol{g}_d$ by $\omega_d \, \boldsymbol{g}_d$. This type of solutions usually deal therefore with disparities in magnitude.
- Direction-aware algorithms, instead, attempt to fully homogenize task gradients. As a consequence, f_ϕ deals also with issues related with gradients pointing towards different directions of the parameter space, thus cancelling out each other when added up.

Note that the contribution of our work is to identify *where* to modify gradients, rather than *how* to modify them. Thus, as mentioned at the end of §3.1, we cross-validate the choice of f_ψ between different magnitude-aware (Chen et al., 2018; Sener & Koltun, 2018; Liu et al., 2021a;b) and direction-aware (Chen et al., 2020; Yu et al., 2020) options. Moreover, our work is orthogonal to the choice of f_ψ, and therefore new algorithms can be easily included.

4. Extending Our Framework

Next, we revisit different VAE models proposed in the literature to handle multimodal data, and show how to apply them the ideas in §3 to avoid modality collapse.

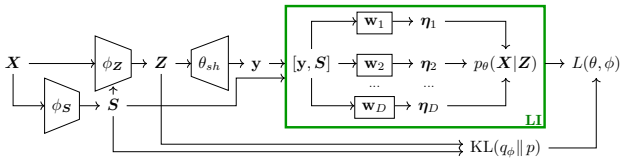


Figure 2. Forward pass of a HI-VAE and its impartiality block.

4.1. Heterogeneous VAE Models

The Heterogeneous-Incomplete VAE (HI-VAE) (Nazabal et al., 2020) is a model specialized on handling heterogeneous data. While it differs from a standard VAE in several aspects—e.g., including a data normalization layer, its hierarchical structure in the form of a Gaussian mixture prior is of especial interest to us. Quoting the original authors, this more expressive prior helps "overcoming the limitations of having assumed a generative model that fully factorizes for every dimension" (see Eq. 2).

We show the computational graph of the HI-VAE in Fig. 2. In short, HI-VAE introduces an additional latent variable, $\boldsymbol{S}$, and defines the encoder (and prior) to be of the form $q_\phi(\boldsymbol{Z}, \boldsymbol{S}|\boldsymbol{X}) = q_\phi(\boldsymbol{Z}|\boldsymbol{S}, \boldsymbol{X}) q_\phi(\boldsymbol{S}|\boldsymbol{X})$. Akin to the example in §3, the last layer of the model is a linear layer, $\boldsymbol{W}$, and the parameters are obtained as $\boldsymbol{\eta} = \sigma([\boldsymbol{y}, \boldsymbol{S}]\boldsymbol{W})$. Note that Eq. 3 remains valid in this case. Moreover, there are additional conflicting-gradient problems, this time w.r.t. $\boldsymbol{S}$:

$$\nabla_{\phi_{\mathbf{S}}} p_\theta \nabla_{p_\theta} L(\theta, \phi) = \nabla_{\phi_{\mathbf{S}}} \mathbf{S} \left(\sum_{d=1}^{D} \nabla_{\mathbf{S}} \boldsymbol{\eta}_d \, \nabla_{\boldsymbol{\eta}_d} p_\theta \right) \nabla_{p_\theta} L. \tag{4}$$

Equations 3 and 4 show that HI-VAE contains an impartiality block with two different inputs, $\boldsymbol{y}$ and $\mathbf{S}$. Hence, we propose to tackle modality collapse by applying our approach (§3.1), and thus Alg. 1, to both inputs. This implies using MTL twice, i.e., to learn $f_{\psi_{\boldsymbol{y}}}$ and $f_{\psi_{\mathbf{S}}}$.

4.2. Multimodal VAE Models

4.2.1. Mixture-based VAEs for Multimodal Data

One desirable property for multimodal VAEs is *conditional generation*, i.e., sampling a modality having observed a different one, representing the same underlying concept. For example, sample the caption for a given image, or vice-versa. However, when the encoder is shared across all modalities, accurate conditional generation is not straight-forward. Mixture-based multimodal VAEs solve this issue by introducing D modality-exclusive encoders (and decoders), using as variational distribution a mixture model of the form:

$$q_\phi(\mathbf{Z}|\mathbf{X}) = \frac{1}{|\mathcal{A}|} \sum_{A \in \mathcal{A}} q_A(\mathbf{Z}|\mathbf{X}_A), \tag{5}$$

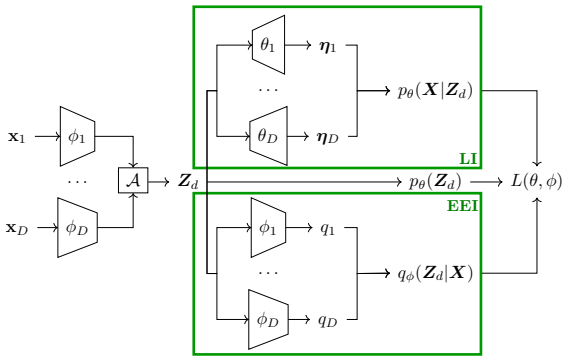


Figure 3. Forward pass of MMVAE, check Fig. 7 to see the blocks related with DEI. Note that MMVAE only has unimodal experts, and that we only show a single latent sample $\mathbf{Z}_d$.

where $\mathcal{A} \subset \mathcal{P}(D)$ is a subset of all the possible combinations of modalities, and $q_A(\mathbf{Z}|\mathbf{X}_A)$ is an *expert* composed of the modalities in $A \subset \{1, 2, \dots, D\}$,

$$q_A(\mathbf{Z}|\mathbf{X}_A) \propto \prod_{d \in A} q_{\phi_d}(\mathbf{Z}|\mathbf{x}_d). \tag{6}$$

We can recover existing models by selecting different values for $\mathcal{A}$ (Figure 3 illustrate the forward pass of the MMVAE):

MVAE (Wu & Goodman, 2018):	$\mathcal{A} = \{\{1, 2, \dots, D\}\}$,
MMVAE (Shi et al., 2019):	$\mathcal{A} = \{\{1\}, \dots, \{D\}\}$,
MoPoE (Sutter et al., 2021):	$\mathcal{A} = \mathcal{P}(D)$.

One setback of considering q_ϕ a mixture model is that we cannot longer differentiably sample from it. First introduced by Shi et al. (2019), and rediscovered by Morningstar et al. (2021), we can overcome this issue by employing stratified sampling, leading to the following objective:

$$L(\theta, \phi) = \sum_{A \in \mathcal{A}} \mathbb{E}_{\mathbf{X}, \mathbf{Z}^A_{1:K}} \left[\log \sum_{k=1}^{K} \frac{p_\theta(\mathbf{X}, \mathbf{Z}^A_k)}{q_\phi(\mathbf{Z}^A_k|\mathbf{X})} \right]. \tag{7}$$

We refer to Eq. 7 as loose since a tighter objective, SIWAE, can be derived (Shi et al., 2019; Morningstar et al., 2021):

$$\tilde{L}(\theta, \phi) = \mathbb{E}_{\mathbf{X}, \{\mathbf{Z}^A_{1:K}\}_{\mathcal{A}}} \left[\log \sum_{A \in \mathcal{A}} \sum_{k=1}^{K} \frac{p_\theta(\mathbf{X}, \mathbf{Z}^A_k)}{q_\phi(\mathbf{Z}^A_k|\mathbf{X})} \right]. \tag{8}$$

Despite being tighter, this objective is notoriously known for suffering from modality collapse. Shi et al. (2019) discarded its use, showing empirical evidence of modality collapse and arguing that "it leads to situations where the joint variational posterior collapses to one of the experts in the mixture."

4.2.2. Impartial Optimization

Recall that our main goal is to accurately approximate the marginal, joint and conditional distributions over all modalities. To achieve this objective, we now identify different impartiality blocks that may stray us from our goal.

Looking at Fig. 3, we find an upper impartiality block, which corresponds once again of evaluating the factorized likelihood (Eq. 2). For each expert A, we find such a impartiality block, having each decoder as a head and its latent variable $\mathbf{Z}_A$ as the common input. Hence, we can improve LI by applying Alg. 1 to each of these blocks. Next, we focus on the specific problems of mixture-based models that may also contribute to modality collapse. Just as in §3, we first describe the goals to pursue in order to achieve conditional generation. Then, we study the parts of the computational graph that may hinder achieving these goals.

Encoder Expert-Impartiality (EEI): In order to enable conditional generation, we need interchangeable encoders, so that we can replace them when modalities are missing. *In other words, we need the ability to generate encoder samples that are impartial to the expert.*

Given the latent samples from an expert, $\mathbf{Z}_A$, we can compute how likely these samples are of coming from any another expert A' by computing $q_{A'}(\mathbf{Z}_A|\mathbf{X}_{A'})$. Similar to the way $\boldsymbol{y}$ could receive gradients from $p_\theta(\mathbf{X}|\mathbf{Z})$ benefiting a subset of modalities (see §3), $\mathbf{Z}_A$ can receive gradients from the mixture $q_\phi(\mathbf{Z}|\mathbf{X})$ that favor a subset of modalities. This impartiality block can be observed in the bottom part of Fig. 3, as well as by computing the gradients of $L(\theta, \phi)$ w.r.t. $\mathbf{Z}_A$, passing through $q_\phi(\mathbf{Z}_A|\mathbf{X})$, i.e.:

$$\begin{aligned} &\nabla_{\phi_d} \mathbf{Z}_A \nabla_{\mathbf{Z}_A} q_\phi \nabla_{q_\phi} L(\theta, \phi) = \\ &\qquad = \nabla_{\phi_d} \mathbf{Z}_A \left(\sum_{A' \in \mathcal{A}} \nabla_{\mathbf{Z}_A} q_{A'} \right) \nabla_{q_\phi} L. \end{aligned} \tag{9}$$

Modality collapse can thus appear as a consequence of conflicting gradients in Eq. 9, having experts whose samples can only substitute a subset of other experts. We can prevent it by applying Alg. 1 to these impartiality blocks.

Decoder Expert-Impartiality (DEI): Similar to EEI, to have proper conditional generation, we need interchangeable decoders that can generate their modality using any latent sample. *That is, we aim for decoders that are impartial to the expert that generated the latent samples.*

DEI relates to the passive role of the latent samples, where the decoder parameters[2] are optimized taking these samples as input. In particular, each decoder $p_{\theta_d}(\mathbf{x}_d|\mathbf{Z})$ is optimized to explain the r.v. $\mathbf{x}_d$ *given the samples from each expert,* $\mathbf{Z}_A \sim q_A(\mathbf{Z}|\mathbf{X}_A)$, which is explicitly shown via stratification in Eqs. 7 and 8.

This time, modality collapse would lead to decoders that can

[2] We do not consider the encoder parameters here, since we use the STL estimator (Roeder et al., 2017).

only generate their modality based on a subset of experts. Building on the ideas from §3, we can find that, for each decoder $p_{\theta_d}(\mathbf{x}_d|\mathbf{Z})$, there exists an impartiality block:

$$\nabla_{\theta_d} L(\theta, \phi) = \left(\sum_{A \in \mathcal{A}} \nabla_{\theta_d} p^A_{\theta_d} \nabla_{p^A_{\theta_d}} L \right), \qquad (10)$$

where we denote $p^A_{\theta_d} \coloneqq p_{\theta_d}(\mathbf{x}_d|\mathbf{Z}_A)$ for the sake of brevity.

Note that the impartiality block in Eq. 10 (illustrated in Fig. 7 of Appendix B) has as input θ_d, the decoder parameters, and each sample $\mathbf{Z}_A$ as modality-specific head. However, due to the flexibility offered by the impartiality blocks, we can reason and tackle modality collapse just as we did in the other cases: applying to each impartiality block Alg. 1.

In total, there are $2|\mathcal{A}| + D$ impartiality blocks in a mixture-based VAE, for which we can use Alg. 1 to palliate modality collapse. Extra details on their application can be found in Appendix B.

5. Experiments

In this section, we assess the approaches shown in §3 and §4 for heterogeneous and multimodal settings. All results shown here are averaged over 5 different seeds and bold numbers represent statistically significant values according to a one-sided Student's t-test ($\alpha = 0.1$), unless stated otherwise. Additional details and results can be found in Appendices D and E.

5.1. Heterogeneous Data

We first turn our attention to heterogeneous data modeling. While the task may look simple at first, we need to deal with plenty of modalities, each one with unique properties. Moreover, models are comparatively simple, forming a breeding ground for modality collapse.

We use as models VAEs as the one introduced in §3, using as objective the ELBO (Kingma & Welling, 2014), IWAE (Burda et al., 2016), and DReG (Tucker et al., 2019). Additionally, we include HI-VAE (Nazabal et al., 2020) as an example of tailored heterogeneous model (see §4.1).

We consider 12 datasets collected from the UCI (Dua & Graff, 2017) and R (R Core Team, 2021) repositories, covering a wide range of dataset sizes and likelihoods. We assign 4 likelihood types (normal, log-normal, Poisson, and categorical) depending on the modality domain. Since likelihoods are not comparable, we use as metric the normalized mean squared error (for numerical data) and error rate (for categorical data), similar to Nazabal et al. (2020).

Do we reconstruct better? Explaining the observed data explicitly appears in the objective function (Eq. 1). If our approach works, reconstruction error should be reduced as a result of impartially learning to explain all modalities. Table 1 (left) shows the reconstruction error for 9 heterogeneous datasets, for which the models trained with our approach improve over the vanilla case in a statistically significant manner in 30 out of 36 cases. Interestingly, our approach specially benefits the standard heterogeneous VAE model, outperforming the HI-VAE (trained with both vanilla and impartial optimization) in several datasets. Importantly, for the majority of datasets, the performance of HI-VAE is significantly improved by impartial optimization, outperforming the rest of VAE models, e.g., in *Adult* and *Diamonds*.

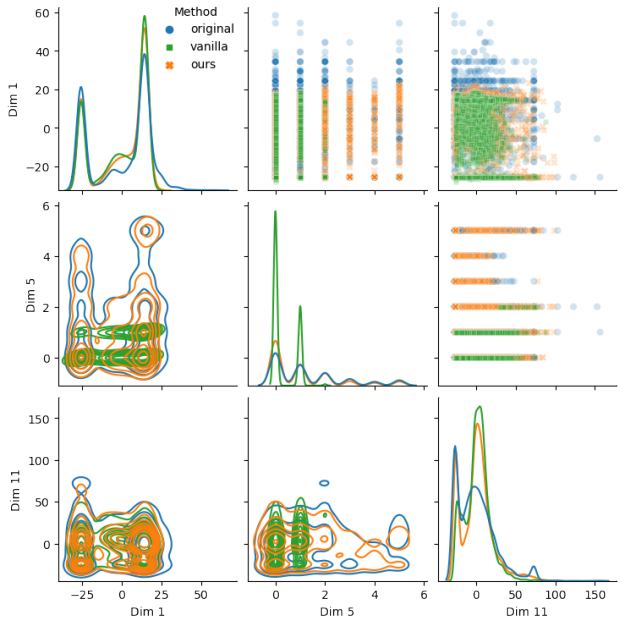


Figure 4. Pair plot of 3 dimensions of *HI*, generated from different VAE models. Diagonal show the marginals, upper-diagonals scatter plots, and lower-diagonals kernel density estimates. The VAE trained with our approach is able to generate faithful samples.

Where does the improvement come from? We investigate whether any likelihood type benefits from our framework. Table 2 shows again reconstruction error, this time aggregated by data type. Here we can observe that we improve over all data types—and specially in categorical variables—by slightly worsening reconstruction on Poisson likelihoods. In Appendix C, we argue that the gradients of Poisson likelihoods are comparatively big among likelihood types, and thus dominates the learning process under standard optimization. Essentially, the trade-off found by our framework in Table 2 is the result of preventing this dominance.

Does impartial optimization help in homogeneous settings? It is reasonable to suspect that modality collapse only appears when each modality uses a different likelihood type. Assigning now exclusively normal likelihoods, we show in

Table 1. Test reconstruction errors (median over five seeds) for different datasets and VAE models. Statistically different values according to a corrected paired t-test ($\alpha = 0.1$) are shown in bold. Models trained with our approach outperforms the baseline in most cases.

			Heterogeneous									Homogeneous		
			Adult	*Credit*	*Wine*	*Diam.*	*Bank*	*IMDB*	*HI*	*rwm5yr*	*labour*	*El Nino*	*Magic*	*BooNE*
Standard VAE	ELBO	vanilla	0.213	0.128	0.086	0.187	0.203	0.082	0.170	0.105	0.109	0.109	0.064	0.042
		ours	**0.104**	**0.041**	**0.071**	**0.139**	**0.043**	**0.032**	**0.041**	**0.026**	**0.063**	**0.068**	0.058	0.039
	IWAE	vanilla	0.226	0.134	0.075	0.185	0.199	0.090	0.155	0.094	0.098	0.086	0.053	0.037
		ours	**0.129**	**0.051**	**0.066**	**0.125**	0.076	**0.035**	**0.042**	**0.032**	**0.066**	**0.061**	0.048	**0.035**
	DReG	vanilla	0.234	0.132	0.077	0.176	0.191	0.088	0.153	0.094	0.096	0.085	0.050	0.037
		ours	0.168	**0.075**	**0.065**	**0.139**	0.103	**0.055**	**0.042**	**0.026**	**0.076**	**0.069**	0.046	0.036
	HI-VAE	vanilla	0.127	0.107	0.126	0.114	0.141	0.079	0.105	0.044	0.100	0.098	0.062	0.039
		ours	**0.081**	0.060	**0.117**	**0.011**	**0.095**	0.049	0.109	**0.024**	**0.069**	**0.015**	**0.033**	0.038

Table 2. Error on the heterogeneous experiments for the baseline and our framework, aggregated by type of likelihood.

	Poisson	Cat.	$\log \mathcal{N}$	$\mathcal{N}$
vanilla	**0.058**	0.158	0.064	0.041
ours	0.083	**0.065**	**0.057**	**0.039**

Table 3. Reconstruction coherence ($A = \{M, S, T\}$) for each modality and model, trained using SIWAE.

	$\mathbf{x}_d$	M	S	T
MVAE	vanilla	97.37	87.47	98.83
	ours	97.42	87.63	**99.20**
MMVAE	vanilla	58.95	61.27	63.27
	ours	**74.16**	68.93	**78.17**
MoPoE	vanilla	75.10	67.16	76.61
	ours	96.91	89.01	99.28

Table 1 (right) that modality collapse also occurs in homogeneous settings, and that our approach may significantly improve model training even if all modalities share the same data type.

Can we generate faithful data? A key aspect of heterogeneous modeling is data generation. As a qualitative example, we train on the *HI* dataset a VAE-ELBO, using vanilla and impartial optimization. We show generated samples by the two VAEs in Fig. 4 for three dimensions of the dataset, compared against the test data. While both models similarly reconstruct the two continuous marginals, only we properly generate the categorical variable (middle), which concurs with the previous analysis. More importantly, the VAE model trained with our framework is able to faithfully recreate the dependencies between modalities, as it can be observed in the off-diagonal figures.

5.2. Multimodal Data

We focus now on mixture-based multimodal VAE models. Besides the obvious architectural differences, these experiments are significantly more demanding and complex, involving millions of parameters and high-dimensional modalities. We use SIWAE (Eq. 8) for most results in the main paper, as it is specially prone to modality collapse.

We reproduce the setups of Sutter et al. (2021) and Shi et al. (2019), using the same architectures, and taking as dataset MNIST-SVHN-Text, which randomly matches positive pairs from MNIST (LeCun et al., 2010) and SVHN (Netzer et al., 2011), and generates a one-hot-encoded text representing the label in common. This is a well-suited dataset for our purposes, since the high disparity in number of dimensions should ease modality collapse during training. Note that in all experiments we divide the log-likelihood by the number of dimensions (local step, see §3.1), to offer fair comparisons, as it is a common practice in the field.

We consider MVAE, MMVAE, and MoPoE as models, which differ in the choice of experts ($\mathcal{A} \subset \mathcal{P}(D)$) for the posterior approximation, as explained in §4.2.

Do we reconstruct better? As a sanity check, we again check how well we are able to reconstruct each modality. Following the existing literature, we measure reconstruction capabilities in terms of *generative coherence*. Specifically, we generate latent samples using all the modalities as input, and reconstruct each modality $\mathbf{x}_d$. Then, we feed each of these samples into modality-specific digit classifiers, and compute the accuracy w.r.t. the ground-truth digit. Table 3 shows that our framework improves reconstruction coherence for all cases and models, sometimes by a statistically significant margin. It is also worth-noting that, in the case of MoPoE, the statistical test is inconclusive as the vanilla case has large variances.

Table 4. Self and cross generation coherence (%) results for different models on **M**NIST-**S**VHN-**T**ext, trained using SIWAE and averaged over 5 different seeds. Models trained with our framework are able to sample more coherent modalities.

		Self coherence			Cross coherence								
	$\boldsymbol{x}_d$	M	S	T	M			S			T		
	A	M	S	T	S	T	S,T	M	T	M,T	M	S	M,S
MVAE	vanilla	82.06	12.08	36.67	10.34	17.12	19.19	49.99	19.31	31.19	62.50	10.82	64.25
	ours	87.63	12.47	**78.88**	10.75	**25.99**	**27.85**	50.02	**33.13**	29.62	61.17	11.67	63.63
MMVAE	vanilla	**95.90**	48.30	53.02	28.43	52.52	40.45	84.44	51.08	67.77	**96.80**	39.96	68.38
	ours	95.90	58.20	**88.70**	**49.33**	**79.32**	**64.30**	**87.29**	**76.17**	**81.71**	96.70	57.86	77.28
MoPoE	vanilla	92.32	11.60	69.05	10.13	51.02	34.67	41.93	46.39	51.58	85.19	10.57	67.54
	ours	90.99	12.00	**83.82**	**10.63**	62.75	52.08	28.19	46.91	43.34	79.64	10.81	90.33

Do we improve conditional generation? One desirable property of a multimodal model is generating coherent samples based on another modalities. In our case, this translates to generating samples of the same digit as the input. We use again generative coherence as metric. This time, given an expert $A \subset \mathcal{P}(D)$, and an output modality $\boldsymbol{x}_d$, we impute $\mathbf{x}_d \sim p_{\theta_d}(\mathbf{x}_d|\mathbf{Z}_A)$ and check if the imputed value matches the original digit. Besides, for each modality d we distinguish between self coherence, where we compute the average accuracy of samples conditioned on that same modality ($A = \{d\}$); and cross coherence, where samples are instead conditioned on experts not containing that modality (every $A \in \mathcal{P}(D)$ such that $d \notin A$).

Table 4 shows the self and cross coherence results for all models and both approaches, trained with SIWAE. While there are trade-offs, we can observe that our framework in general improves both self and cross coherence across all models. For example, Text (T) and SVHN (S) were overlooked in MVAE and MMVAE, respectively, and the impartial VAE model increases self coherence for those modalities, as well as cross coherence when they appear in the expert A. As mentioned in §4.2, SIWAE is prone to modality collapse. However, all objectives benefit from our framework. Figure 5 shows a parallel coordinate plot with the generative coherence results for MMVAE, evaluated on all objectives. While SIWAE significantly improves with impartial optimization (as expected), we also improve all the different metrics for all losses.

Do we generate more informative latent spaces? One key aspect of latent space models is that the latent space should be rich and informative. Following the existing literature, we evaluate the quality of a latent space by training a linear classifier to predict the ground-truth label, taking samples of $\mathbf{Z}$ as input.

Another key aspect, this time of mixture-based multimodal VAE models, is that the encoders should be as similar as possible (EEI), and thus their latent spaces. Just as before, here we distinguish between self and cross latent classification accuracy. For each expert A, self latent classification refers to classifying test samples from the same expert the classifier was trained with, while cross latent classification refers to classifying test samples coming from an expert different from the one the classifier was trained with.

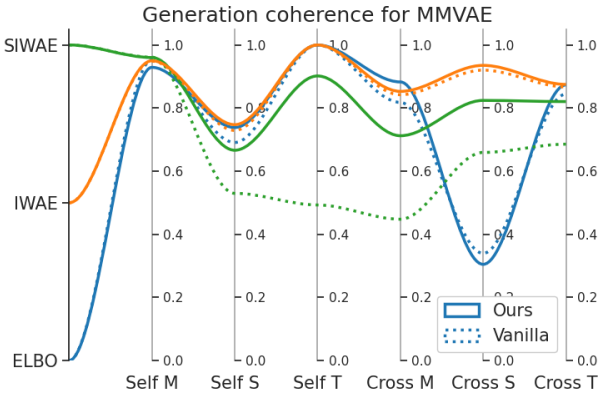


Figure 5. Generation coherence of MMVAE with ELBO, IWAE, and SIWAE. We improve most metrics w.r.t. the baseline.

We show in Table 5 the classification accuracies, averaged over experts. We can observe that MMVAE and MoPoE significantly improve self latent classification accuracy when they are trained with our framework. More importantly, all models significantly improve the cross latent classification accuracy, independently of the loss they were trained with, indicating that the latent spaces between experts are more similar between them (i.e., satisfy EEI).

Does impartial optimization add a lot of overhead? The inset table shows the training times for MMVAE as we change the number of blocks for which we apply Alg. 1. As expected, the training time increases as we ap-

LI	EEI	DEI	time (h)	#
☐	☐	☐	10.06	0
☑	☐	☐	11.42	D
☑	☑	☐	11.64	$2D$
☑	☑	☑	11.89	$3D$

Table 5. Self and cross latent classification accuracy (%) for different models and losses on MNIST-SVHN-Text.

		ELBO	IWAE	SIWAE
		Self latent classification		
MVAE	vanilla	69.68	69.14	68.58
	ours	69.95	69.06	**69.75**
MMVAE	vanilla	71.81	87.55	71.30
	ours	**87.83**	90.78	**85.55**
MoPoE	vanilla	89.85	87.23	67.58
	ours	**91.47**	**90.74**	**69.26**
		Cross latent classification		
MVAE	vanilla	33.60	39.15	38.36
	ours	35.25	**49.73**	**46.23**
MMVAE	vanilla	44.25	76.81	40.60
	ours	**71.42**	**84.80**	**60.50**
MoPoE	vanilla	66.14	83.71	40.36
	ours	**84.52**	**90.48**	**53.24**

ply more MTL algorithms to the training. In the case of MMVAE, we have 9 different impartiality blocks, and yet the training time increases only an $18\,\%$, going from $10\,\mathrm{h}$ of training to $11.89\,\mathrm{h}$. Each additional step increased in $25\,\mathrm{min}$ the training time, which makes us believe that the extra overhead in the first transition is due to our implementation to manipulate the backward pass for Alg. 1.

6. Conclusions

In this work, we have studied the problem of modality collapse in multimodal VAEs, showing that it can be understood as a consequence of the conflict between gradients of different modalities during training. We confined this conflict to a sub-graph of the computational graph, the impartiality block, and proposed a general pipeline to enforce impartial optimization across modalities. We have analyzed different tailored models, where several impartiality blocks may appear, proving the flexibility of our modular approach. Finally, we have empirically shown that our approach can significantly improve the performance of these models on a range of datasets, losses and metrics.

We believe this work opens venues for future research. First, as our method relies on off-the-shelf solutions from MTL, it would be interesting to develop gradient-conflict solutions for the specifics of multimodal VAEs. Second, exploring variations of impartiality blocks for specific applications, e.g., non-modular designs that reduce the current overhead, or impartiality blocks that take into account missing patterns in real-world data, could lead to exciting future works.

7. Acknowledgements

We would like to thank Pablo Sánchez-Martín for providing useful feedback on the manuscript, as well as to the anonymous reviewers and meta-reviewer who helped to improve the quality of the paper during the review process.

References

Baltrušaitis, T., Ahuja, C., and Morency, L.-P. Multimodal machine learning: A survey and taxonomy. IEEE transactions on pattern analysis and machine intelligence, 41(2):423–443, 2018.

Barrejón, D., Olmos, P. M., and Artés-Rodríguez, A. Medical data wrangling with sequential variational autoencoders. arXiv preprint arXiv:2103.07206, 2021. URL https://arxiv.org/abs/2103.07206.

Burda, Y., Grosse, R. B., and Salakhutdinov, R. Importance weighted autoencoders. In Bengio, Y. and LeCun, Y. (eds.), 4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016. URL http://arxiv.org/abs/1509.00519.

Chen, Z., Badrinarayanan, V., Lee, C., and Rabinovich, A. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In Dy, J. G. and Krause, A. (eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research, pp. 793–802. PMLR, 2018. URL http://proceedings.mlr.press/v80/chen18a.html.

Chen, Z., Ngiam, J., Huang, Y., Luong, T., Kretzschmar, H., Chai, Y., and Anguelov, D. Just pick a sign: Optimizing deep multitask models with gradient sign dropout. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/16002f7a455a94aa4e91cc34ebdb9f2d-Abstract.html.

Chennupati, S., Sistu, G., Yogamani, S. K., and Rawashdeh, S. A. Multinet++: Multi-stream feature aggregation and geometric loss strategy for multi-task learning. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1200–1210, 2019.

Dua, D. and Graff, C. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M. J., and Schölkopf, B. From variational to deterministic autoencoders. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020. URL https://openreview.net/forum?id=S1g7tpEYDS.

Guo, W., Wang, J., and Wang, S. Deep multimodal representation learning: A survey. IEEE Access, 7:63373–63394, 2019.

Kendall, A., Gal, Y., and Cipolla, R. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pp. 7482–7491. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00781. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Kendall_Multi-Task_Learning_Using_CVPR_2018_paper.html.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings, 2014. URL http://arxiv.org/abs/1312.6114.

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. ATT Labs [Online], 2, 2010. URL http://yann.lecun.com/exdb/mnist.

Liu, B., Liu, X., Jin, X., Stone, P., and Liu, Q. Conflict-averse gradient descent for multi-task learning. Advances in Neural Information Processing Systems, 34, 2021a.

Liu, L., Li, Y., Kuang, Z., Xue, J.-H., Chen, Y., Yang, W., Liao, Q., and Zhang, W. Towards impartial multi-task learning. In International Conference on Learning Representations, 2021b. URL https://openreview.net/forum?id=IMPnRXEWpvr.

Ma, C., Tschiatschek, S., Turner, R., Hernández-Lobato, J. M., and Zhang, C. Vaem: a deep generative model for heterogeneous mixed type data. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 11237–11247. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/8171ac2c5544a5cb54ac0f38bf477af4-Paper.pdf.

Mehrasa, N., Jyothi, A. A., Durand, T., He, J., Sigal, L., and Mori, G. A variational auto-encoder model for stochastic point processes. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pp. 3165–3174. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00328. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Mehrasa_A_Variational_Auto-Encoder_Model_for_Stochastic_Point_Processes_CVPR_2019_paper.html.

Morningstar, W. R., Vikram, S. M., Ham, C., Gallagher, A. G., and Dillon, J. V. Automatic differentiation variational inference with mixtures. In Banerjee, A. and Fukumizu, K. (eds.), The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event, volume 130 of Proceedings of Machine Learning Research, pp. 3250–3258. PMLR, 2021. URL http://proceedings.mlr.press/v130/morningstar21b.html.

Nadeau, C. and Bengio, Y. Inference for the generalization error. Machine learning, 52(3):239–281, 2003.

Nazabal, A., Olmos, P. M., Ghahramani, Z., and Valera, I. Handling incomplete heterogeneous data using vaes. Pattern Recognition, 107:107501, 2020.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011.

R Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2021. URL https://www.R-project.org/.

Rainforth, T., Kosiorek, A. R., Le, T. A., Maddison, C. J., Igl, M., Wood, F., and Teh, Y. W. Tighter variational bounds are not necessarily better. In Dy, J. G. and Krause, A. (eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research, pp. 4274–4282. PMLR, 2018. URL http://proceedings.mlr.press/v80/rainforth18b.html.

Reddi, S. J., Kale, S., and Kumar, S. On the convergence of adam and beyond. In 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net, 2018. URL https://openreview.net/forum?id=ryQu7f-RZ.

Roeder, G., Wu, Y., and Duvenaud, D. Sticking the landing: Simple, lower-variance gradient estimators for variational inference. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 6925–6934, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/e91068fff3d7fa1594dfdf3b4308433a-Abstract.html.

Ruder, S. An overview of multi-task learning in deep neural networks. CoRR, abs/1706.05098, 2017. URL http://arxiv.org/abs/1706.05098.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning representations by back-propagating errors. Nature, 323:533–536, 1986.

Sener, O. and Koltun, V. Multi-task learning as multi-objective optimization. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pp. 525–536, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/432aca3a1e345e339f35a30c8f65edce-Abstract.html.

Shi, Y., Narayanaswamy, S., Paige, B., and Torr, P. H. S. Variational mixture-of-experts autoencoders for multi-modal deep generative models. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 15692–15703, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0ae775a8cb3b499ad1fca944e6f5c836-Abstract.html.

Shi, Y., Paige, B., Torr, P., and N, S. Relating by contrasting: A data-efficient framework for multimodal generative models. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=vhKe9UFbrJo.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research, 15(56):1929–1958, 2014. URL http://jmlr.org/papers/v15/srivastava14a.html.

Sutter, T. M., Daunhawer, I., and Vogt, J. E. Multimodal generative learning utilizing jensen-shannon-divergence. In NeurIPS, 2020.

Sutter, T. M., Daunhawer, I., and Vogt, J. E. Generalized multimodal ELBO. In International Conference on Learning Representations, 2021. URL https://openreview.net/forum?id=5Y21V0RDBV.

Tucker, G., Lawson, D., Gu, S., and Maddison, C. J. Doubly reparameterized gradient estimators for monte carlo objectives. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019. URL https://openreview.net/forum?id=HkG3e205K7.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 19667–19679. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e3b21256183cf7c2c7a66be163579d37-Paper.pdf.

Wu, M. and Goodman, N. D. Multimodal generative models for scalable weakly-supervised learning. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pp. 5580–5590, 2018. URL https://proceedings.neurips.cc/paper/2018/hash/1102a326d5f7c9e04fc3c89d0ede88c9-Abstract.html.

Xu, W., Sun, H., Deng, C., and Tan, Y. Variational autoencoder for semi-supervised text classification. In Singh, S. P. and Markovitch, S. (eds.), Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA, pp. 3358–3364. AAAI Press, 2017. URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14299.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), Advances in Neural Information Processing Systems, volume 33,

pp. 5824–5836. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/3fe78a8acf5fda99de95303940a2420c-Paper.pdf.

A. Multitask learning and conflicting gradients

The goal of multitask learning (MTL) is to simultaneously solve a set of D tasks. Suppose that all of them share the input data $\mathbf{X}$, but each task defines its own loss function L_d. To amortize parameters across tasks, one common choice is to have a shared backbone, $\boldsymbol{x} \mapsto \boldsymbol{y}$, parameterized by θ_{sh}, and a set of task-specific heads, $\boldsymbol{y} \mapsto \boldsymbol{\eta}_d$, where $\boldsymbol{\eta}_d$ is the prediction for its associated task. In order to learn the parameters, a common approach is to minimize the sum of losses, $\sum_d L_d$.

One main assumption in MTL is that of *task impartiality*, which assumes that all tasks are equally important to solve, i.e., we do not prefer learning one task over another (Liu et al., 2021b). MTL often suffers from *negative transfer*, which is defined as the negative effect that simultaneously learning some tasks can have on the final model performance (Ruder, 2017).

Akin to this work, one research direction in MTL studies conflicting gradients in order to explain the existence of negative transfer. Indeed, it is easy to observe that the gradient w.r.t. the shared parameters is of the form $\sum_d \nabla_{\theta_{sh}} L_d$, and thus gradient differences make the model lean toward prioritizing some tasks over others.

A.1. Conflicting-gradient solutions

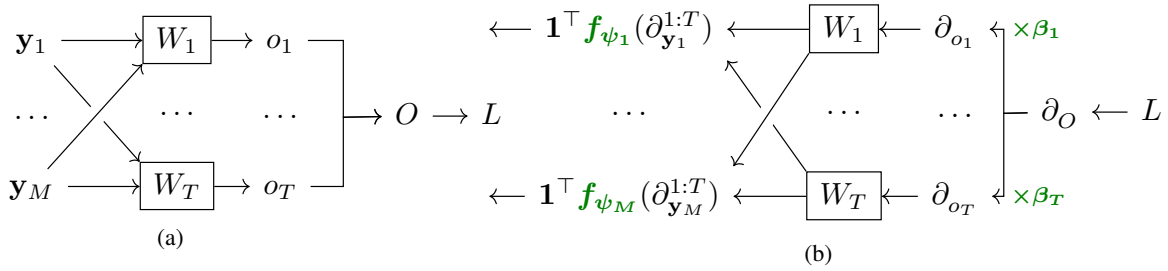
As explained in §3.1, we consider MTL solutions to conflicting gradients, f_ψ that modify the gradients during the backward pass. These solutions can be classified in two main categories.

- On the one hand, we have algorithms f_ψ that scale each gradient $\boldsymbol{g}_d$ according to a specific criterion, in order to deal with the disparities of gradients due to their magnitudes. That is, they replace each gradient $\boldsymbol{g}_d$ with $\omega_d \, \boldsymbol{g}_d$, where each algorithm f_ψ sets the value of ω_d in each step differently.
- Second, direction-aware algorithms. These algorithms attempt to solve issues related with gradients pointing towards different directions of the parameter space, thus cancelling out each other.

We consider for all our experiments the following existing algorithms from the MTL literature:

- Magnitude-aware:
  i) GradNorm (Chen et al., 2018) (GN) - Scales the gradients and try to normalize the magnitude of the gradients over time. Moreover, a hyperparameter α controls the intensity for which to normalize these gradients, using the ratio between task losses as a measure of the task convergence.
  In this work we have slightly modified GradNorm, such that instead of using the task losses, we use the magnitude of the gradients as a criterion to identify the "task" convergence.
  ii) MGDA-UB (Sener & Koltun, 2018) - Scales the gradient by finding the convex sum of the gradients that results in the minimum norm, such that advancing in that direction reduces all task losses.
  iii) IMTL-G (Liu et al., 2021b) - Scales the gradients by optimizing the scaling factors via a closed-form solution, such that the aggregated gradient (sum of raw gradients weighted by the scaling factors) has equal projections onto individual tasks.
  iv) CAGrad (Liu et al., 2021a) (CA) - Generalization of MGDA-UB that introduces a hyperparameter α to control how much the resulting gradient direction differs from the one followed by SGD. In the multimodal experiments, we approximate MGDA-UB using CAGrad with $\alpha = 10$.
- Direction-aware:
  i) GradDrop (Chen et al., 2020) (GD) - Randomly drops elements of the task gradients based on how much they conflict in direction with the aggregated gradient, such that those directions "self-correct" themselves and align with the rest of the gradients.
  ii) PCGrad (Yu et al., 2020) (PG) - Randomly projects task gradients between them, thus removing the orthogonal parts that would cancel out when computing the aggregated gradient.

For the heterogeneous experiments, we find the best f_ψ by combining magnitude-aware solutions followed by direction-aware solutions, since they are cheaper to compute, and we can run more experiments. For the multimodal experiments, we do not consider combinations of algorithms, but the algorithms by themselves.



(a) (b)

Figure 6. Generic sketches of a (a) forward and a (b) modified backward pass of an impartiality block. Note that here we are using the shorthand $\partial_x^{1:K}$ for a sequence of K gradients w.r.t. x (not to be confused with the partial derivative).

Algorithm 2 Generic impartial backward pass within the impartiality block.

Definition IMPARTIALBACKWARD(inputs: $\boldsymbol{y}_{1:M}$; heads: $W_{1:T}$; output: O)
Input: Output gradient, ∂_O.
for $t = 1$ **to** T **do**
  $\partial_{o_t} \leftarrow \boldsymbol{\beta_t} \nabla_{o_t} O \, \partial_O$ ▷ Re-weigh the gradient at the heads.
  $\nabla_{W_t} L \leftarrow \nabla_{W_t} o_t \cdot \partial_{o_t}$ ▷ Gradient of the head parameters (if any).
  for $m = 1$ **to** M **do**
    $\partial^t_{\boldsymbol{y}_m} \leftarrow \nabla_{\boldsymbol{y}_m} o_t \cdot \partial_{o_t}$ ▷ Per-task gradients w.r.t. the common inputs.
  end for
end for
for $m = 1$ **to** M **do**
  backpropagate $\mathbf{1}^\top \boldsymbol{f}_{\psi_m}(\partial^{1:T}_{\boldsymbol{y}_m})$ **through** $\boldsymbol{y}_m$ ▷ Apply MTL methods and backpropagate the changes.
end for

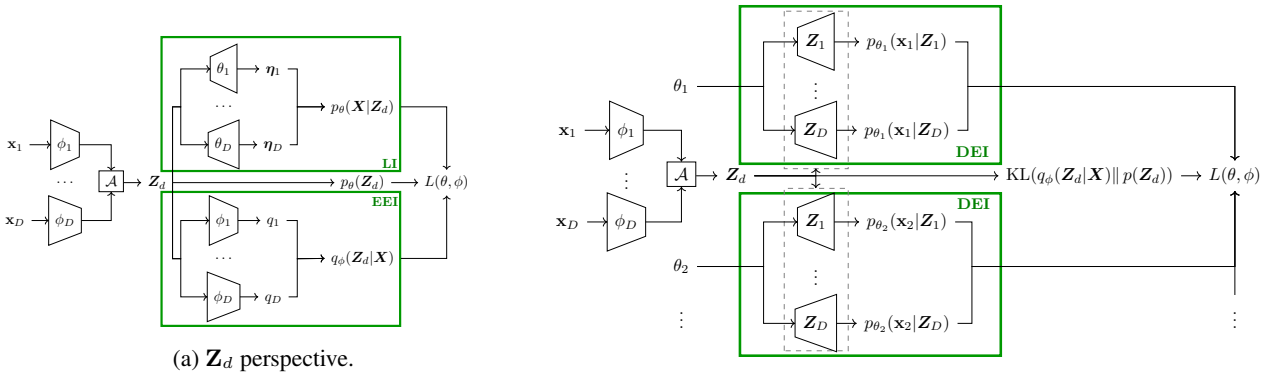
B. Alleviating modality collapse

In §3 of the main paper, we have introduced the impartiality block gradually, starting with a simple example, and showing how to adapt it as we were facing different challenges. Here, we introduce the impartiality block in a generic and flexible way, so that it could be easier for the reader to understand how to apply it to the tailored models explained in the main manuscript, as well as how to use the impartiality block for their own use-cases.

Alg. 2 shows the new algorithm, and Fig. 6 the forward and backward pass. To detach the block from its original presentation, we have adopted here a generic notation for the different elements of the block, as well as allow for multiple entries. In this way, we would like to emphasize that the key aspect of the impartiality block is its structure, and not the variables that appear within it. In other words, Alg. 2 can be applied to any impartiality block, independently of whether the input is an intermediate feature (such as in the blocks related with LI, see §3), or the features of a neural network (such as in the blocks related with DEI, see §4.2.2). As for the last example, we could not show a computational block for the mixture-base models that introduces the three types of impartiality blocks at once. To help the reader, we present in Fig. 7 two different ways of drawing the computational graph of mixture-based models that unveil all the impartiality blocks.

With the re-formulation of the impartiality block, we provide here a summary of the impartiality blocks presented in the models of the main paper:

Model	Goal	#	Backward call
VAE	LI	1	IMPARTIALBACKWARD($\boldsymbol{y}; \omega_{1:D}; \boldsymbol{\eta}$)
IWAE	LI	1	IMPARTIALBACKWARD($\boldsymbol{y}; \omega_{1:D}; \boldsymbol{\eta}$)
DReG	LI	2	IMPARTIALBACKWARD($\boldsymbol{y}; \omega_{1:D}; \boldsymbol{\eta}$)
HI-VAE	LI	1	IMPARTIALBACKWARD($[\boldsymbol{y}, \boldsymbol{S}]; \omega_{1:D}; \boldsymbol{\eta}$)
mixture-based	LI	$\lvert\mathcal{A}\rvert$	IMPARTIALBACKWARD($\mathbf{Z}_A; \theta_{1:D}; p_\theta(\mathbf{X}\lvert\mathbf{Z}_A)$)
mixture-based	EEI	$\lvert\mathcal{A}\rvert$	IMPARTIALBACKWARD($\mathbf{Z}_A; \phi_\mathcal{A}; q_\phi(\mathbf{Z}_A\lvert\mathbf{X})$)
mixture-based	DEI	D	IMPARTIALBACKWARD($\theta_d; \mathbf{Z}_\mathcal{A}; L$)



(a) $\mathbf{Z}_d$ perspective.

(b) θ_d perspective.

Figure 7. Forward pass of MMVAE from two different points of view: as in §3, (a) shows the perspective where $\mathbf{Z}$ is the active role, where for each $\mathbf{Z}_A$ we find two impartiality blocks; in (b) we show the perspective of the decoder parameters, where the decoder parameters play the active role, and now we can observe that there is an impartiality block result of evaluating each decoder in each expert samples. Here, the dashed block simply indicates that the sample $\mathbf{Z}_d$ gets distributed into that set of blocks to be evaluated. Note that we explicitly show only 2 out of D DEI blocks.

C. Dominance of Poisson likelihoods

In this section, we attempt to mathematically sketch the results obtained in Table 2 of the main paper. To do that, we are simply going to compute the expected value of the squared norm of the gradient with respect to each of the likelihoods, that is, we estimate $\mathbb{E}_{\mathbf{x}_d}\left[||\nabla_{\boldsymbol{\eta}_d} \log p_{\theta_d}(\mathbf{x}_d;\boldsymbol{\eta}_d)||^2\right]$. We further simplify things by making the assumption that $\mathbf{x}_d$ actually follows the distribution $p_{\theta_d}(\mathbf{x}_d;\boldsymbol{\eta}_d)$. While unrealistic, this assumption should become more and more real as the training progresses. We break down this informal proof in two steps:

Computing the expected squared norms. We first take advantage that all considered distributions are part of the exponential distribution, and find a general formula valid for all of them. As a reminder, the exponential family, with natural parameters $\boldsymbol{\eta} \in \mathbb{R}^I$, is a family of distributions which is characterized by having a density function of the form

$$p(\mathbf{x};\boldsymbol{\eta}) = h(\mathbf{x}) \exp\left(\boldsymbol{T}(\mathbf{x})^\top \boldsymbol{\eta} - A(\boldsymbol{\eta})\right), \tag{11}$$

where each member of the family defines the values for: $h(\mathbf{x})$, the base measure; $\boldsymbol{T}(\mathbf{x})$ the sufficient statistics; and $A(\boldsymbol{\eta})$, the log-partition function. Using this general expression, we can compute the value of $\mathbb{E}_{\mathbf{x}_d}\left[||\nabla_{\boldsymbol{\eta}_d} \log p_{\theta_d}(\mathbf{x}_d;\boldsymbol{\eta}_d)||^2\right]$:

$$\ln p(\mathbf{x};\boldsymbol{\eta}) = \boldsymbol{T}(\mathbf{x})^\top \boldsymbol{\eta} - A(\boldsymbol{\eta}) + C(\mathbf{x}), \tag{12}$$

$$\partial_{\eta_i} \ln p(\mathbf{x};\boldsymbol{\eta}) = T_i(\mathbf{x}) - \partial_{\eta_i} A(\boldsymbol{\eta}) = T_i(\mathbf{x}) - \mathbb{E}\left[T_i(\mathbf{x})\right], \tag{13}$$

$$\mathbb{E}_{\mathbf{x}_d}\left[||\nabla_{\boldsymbol{\eta}_d} \log p_{\theta_d}(\mathbf{x}_d;\boldsymbol{\eta}_d)||^2\right] = \sum_i \mathbb{E}_{\mathbf{x}_d}\left[\left(T_i(\mathbf{x}) - \mathbb{E}\left[T_i(\mathbf{x})\right]\right)^2\right], \tag{14}$$

where we have used the fact that $\partial_{\eta_i} A(\boldsymbol{\eta}) = \mathbb{E}\left[T_i(\mathbf{x})\right]$.

We can now simply plug in the specific values for the sufficient statistics for each of the likelihoods:

	$\partial_{\eta_1} \ln p(\mathbf{x};\boldsymbol{\eta})$	$\partial_{\eta_2} \ln p(\mathbf{x};\boldsymbol{\eta})$	$\mathbb{E}\left[||\nabla_{\boldsymbol{\eta}} \ln p(\mathbf{x};\boldsymbol{\eta})||^2\right]$
Normal	$\mathbf{x} - \mu$	$\mathbf{x}^2 - (\mu^2 + \sigma^2)$	$\sigma^2 + 4\mu^2\sigma^2$
Log-normal	$\ln \mathbf{x} - \mu$	$(\ln \mathbf{x})^2 - (\mu^2 + \sigma^2)$	$\sigma^2 + 4\mu^2\sigma^2$
Poisson	$\mathbf{x} - \lambda$		λ
Categorical	$[\mathbf{x} = i] - \pi_i$		$\sum_{i=1}^I \mathbb{E}\left[([\mathbf{x} = i] - \pi_i)^2\right]$

For each likelihood above, we have used the usual notation for their normal parameters. Moreover, notice that the moments

are not well-defined for the categorical distribution. Instead, we just compute the average over the entire dataset. Here, $[\mathbf{x} = i]$ denotes the Iverson brackets (whether $\mathbf{x}_i$ pertains to the i-th class).

Bounding the norms under our working pipeline. Once that we have rough estimates of the expected squared norms of the gradients for each likelihood, we need to come down to earth and connect it with the experiments in §5.1. Specifically, we need to take into account the preprocessing and the datasets themselves. We use the *Adult* dataset as an example:

- Normal: We standardize normal data, such that $\mu = 0$ and $\sigma = 1$. Therefore, $\mathbb{E}\left[||\nabla_{\boldsymbol{\eta}} \ln p(\mathbf{x}; \boldsymbol{\eta})||^2\right] \approx 1$.
- Log-normal: We standardize (without shifting) in log-space. In *Adult*, the biggest log-normal distribution lies in the range $[15, 22]$, such that $\mu \approx 1$ and $\sigma < 1$ in log-scale, and $\mathbb{E}\left[||\nabla_{\boldsymbol{\eta}} \ln p(\mathbf{x}; \boldsymbol{\eta})||^2\right] \approx 1$.
- Poisson: Since data is discrete, we do not standardize it. Count data can be quite large, reaching in *Adult* a maximum value of 100. Thus, $\mathbb{E}\left[||\nabla_{\boldsymbol{\eta}} \ln p(\mathbf{x}; \boldsymbol{\eta})||^2\right] >> 1$ in *Adult*.
- Categorical: Again, we do not standardize categorical data, as it is discrete. However, it is relatively simple to see that $0 \leq \mathbb{E}\left[||\nabla_{\boldsymbol{\eta}} \ln p(\mathbf{x}; \boldsymbol{\eta})||^2\right] \leq I$ since $0 \leq \pi_i \leq 1$ and $[\mathbf{x} = i] \in \{0, 1\}$. However, the number of classes I is usually small, and the gradient is bounded by I during the entire training, while in the other cases they are not (we just considered the cases where we have the ground-truth parameters).

Therefore, using these rough calculations, we can expect the values of $\mathbb{E}\left[||\nabla_{\boldsymbol{\eta}} \ln p(\mathbf{x}; \boldsymbol{\eta})||^2\right]$ to lie in the following order:

$$\text{Categorical} < \text{Normal} \approx \text{Log-normal} << \text{Poisson}.$$

And, if we compute the difference between normalized errors in Table 2, we obtain that our approach improves the error across types in an order similar to the reverse of the one shown above:

	Cat.	$\log \mathcal{N}$	$\mathcal{N}$	Poisson
vanilla	0.158	0.064	0.041	**0.058**
ours	**0.065**	**0.057**	**0.039**	0.083
improvement	0.092 $>$	0.008 $\approx$	0.002 $>$	–0.025

D. Model descriptions

In this section we explain the implementation details for each model, please refer to the original papers for a detailed explanation of each model. We use the following notation to describe the models:

D	Number of features.
D'	Total number of likelihood parameters.
l	Latent size.
h	Hidden size.
`[Linear-`h`]`	Linear layer with output of size h.
`[Conv-`k`-`s`-`p`]`	Convolutional layer with kernel size k, stride s and padding p.
`[ConvT-`k`-`s`-`p`]`	Transposed convolutional layer with kernel size k, stride s and padding p.
`[Dropout-10%]`	Dropout Srivastava et al. (2014) with $10\,\%$ of dropping probability.
`[ReLU]`	Rectified linear unit activation function.
`[Tanh]`	Hyperbolic tangent activation function.
`[Sigmoid]`	Sigmoid activation function.

D.1. Variational autoencoder (VAE)

We implement the original VAE (Kingma & Welling, 2014) assuming the following probabilistic model:

$$\begin{aligned}
&\text{Prior:} && p(\mathbf{Z}) = \mathcal{N}(0, I) \\
&\text{Likelihood:} && p_\theta(\mathbf{X}|\mathbf{Z}) = \textstyle\prod_d p_d(\mathbf{x}_d|\eta_d(\mathbf{Z};\theta)) \\
&\text{Variational approx.:} && q_\phi(\mathbf{Z}|\mathbf{X}) = \mathcal{N}(\mu(\mathbf{X};\phi), \sigma(\mathbf{X};\phi))
\end{aligned}$$

Here μ and σ are modelled by the encoder, and all η_d are jointly modelled by the decoder.

These two neural networks are of the following form:

Encoder: `[Dropout-10%][BN][Linear-`h`][Tanh][Linear-`h`][Tanh][Linear-`h`][Tanh][Linear-2`l`]`
Decoder: `[Linear-`h`][ReLU][Linear-`h`][ReLU][Linear-`h`][ReLU][Linear-`D'`]`

Additionally, we make sure that each parameter fulfills its distributional constraints (e.g., the variance has to be positive) by passing it through a softplus function when necessary. It is also important to note that, while we parametrize the latent space using the mean and standard deviation, we parametrize the parameters of the likelihoods using their natural parameters.

Loss. We use the negative ELBO as training loss:

$$\mathrm{ELBO}(\mathbf{X}, \theta, \phi) \coloneqq \mathbb{E}_{q_\phi}\left[\log p_\theta(\mathbf{X}|\mathbf{Z})\right] - \mathrm{KL}(q_\phi(\mathbf{Z}) \| \, p(\mathbf{Z})). \tag{15}$$

Imputation. We impute data by taking the modes of $q_\phi(\boldsymbol{z}|\mathbf{X})$ and $p_d(\mathbf{x}_d; \eta_d(\mathbf{Z};\theta))$.

D.2. Importance weighted autoencoder (IWAE)

Importance weighted autoencoder (IWAE) (Burda et al., 2016) differs from VAE only on the training loss.

Loss. Instead of maximizing the ELBO, IWAE maximizes a tighter loss that makes use of K i.i.d. samples from $\mathbf{Z}$:

$$\mathrm{IWAE}(\mathbf{X}, \theta, \phi) \coloneqq \mathbb{E}_{\mathbf{Z}_1,\ldots,\mathbf{Z}_K \sim q_\phi}\left[\log \frac{1}{K} \sum_k \frac{p_\theta(\mathbf{X}|\mathbf{Z}_k) p(\mathbf{Z}_k)}{q_\phi(\mathbf{Z}_k|\mathbf{X})}\right]. \tag{16}$$

For all the results shown in Table 1 we set the number of importance samples to $K = 20$.

D.3. Doubly reparametrized gradient estimator (DReG)

Rainforth et al. (2018) showed that the gradient estimators produced by IWAE have some undesired properties that could hamper properly learning the inference parameters (encoder). A strict improvement over this negative result was later provided by Tucker et al. (2019), as they provide a simple way of addressing these issues by applying the reparametrization trick a second time. As a result, we obtain again a model structurally identical to VAE, but which is optimized with two different losses: one for the encoder, and one for the decoder. We use $K = 20$ importance samples as for IWAE.

Encoder loss. For one importance sample $\mathbf{Z}_k$, let us define

$$\omega_k \coloneqq \frac{p_\theta(\mathbf{X}|\mathbf{Z}_k) p(\mathbf{Z}_k)}{q_\phi(\mathbf{Z}_k)}, \text{ and } \quad \widetilde{\omega}_k \coloneqq \frac{\omega_k}{\sum_i \omega_i} \text{ such that } \quad \sum_k \widetilde{\omega}_k = 1. \tag{17}$$

Then, we optimize the parameters of the encoder by maximizing

$$\mathrm{DReG}^{\mathrm{enc}}(\mathbf{X}, \theta, \phi) \coloneqq \mathbb{E}_{\mathbf{Z}_1,\ldots,\mathbf{Z}_K \sim q_\phi}\left[\sum_k \widetilde{\omega}_k^2 \log \omega_k\right], \tag{18}$$

where we consider $\widetilde{\omega}_k$ to be a constant value (i.e., we do not backpropagate through it), and we compute the derivative w.r.t. ϕ only through $\mathbf{Z}$ (i.e., we do not compute the partial derivative w.r.t. ϕ).

Decoder loss. Similarly, we optimize the parameters of the decoder by maximizing the following loss (same assumptions on $\widetilde{\omega}_k$ and ϕ):

$$\mathrm{DReG}^{\mathrm{dec}}(\mathbf{X}, \theta, \phi) \coloneqq \mathbb{E}_{\mathbf{Z}_1,\ldots,\mathbf{Z}_K \sim q_\phi}\left[\sum_k \widetilde{\omega}_k \log \omega_k\right]. \tag{19}$$

D.4. HI-VAE

We have faithfully re-implemented the original version of HI-VAE (Nazabal et al., 2020), this includes implementing their architecture with the same number of parameters, as well as implementing their methods (such as the proposed normalization and denormalization layers). Regarding the architecture, we have maintained the same one as the original authors used in their experiments. Therefore, results between HI-VAE and the rest of the models in Table 1 are not completely comparable.

HI-VAE assumes a hierarchical latent space. Thus, we assume the following probabilistic model:

$$\begin{aligned}
&\text{Prior:} && p(\mathbf{Z},\mathbf{s}) && = p(\mathbf{s})p(\mathbf{Z}|\mathbf{s}) \\
& && && = \text{Cat}(\tfrac{1}{d_s},\tfrac{1}{d_s},\dots,\tfrac{1}{d_s})\,\mathcal{N}(\mu_0(\mathbf{s}),I) \\
&\text{Likelihood:} && p_\theta(\mathbf{X}|\mathbf{Z}) && = \textstyle\prod_d p_d(\mathbf{x}_d|\eta_d(\mathbf{Z};\theta)) \\
&\text{Variational approx.:} && q_\phi(\mathbf{Z},\mathbf{s}|\mathbf{X}) && = q_\phi(\mathbf{s}|\mathbf{X})q_\phi(\mathbf{Z}|\mathbf{X},\mathbf{s}) \\
& && && = \text{Cat}(\pi(\mathbf{X}))\,\mathcal{N}(\mu(\mathbf{X},\mathbf{s};\phi),\sigma(\mathbf{X},\mathbf{s};\phi)).
\end{aligned}$$

Similar to VAE, μ_0, μ, and σ are all neural networks, and all likelihood parameters η_d are jointly modelled by the decoder. Note also the introduction of new variables to describe the size of each latent variable, d_z and d_s.

We set in our experiments $d_z = d_s = 10$, and the hidden size to $h = 5D$, just as in the original paper.

Loss. We maximize the ELBO as originally proposed by Nazabal et al. (2020):

$$\text{ELBO}(\mathbf{X},p_\theta,q_\phi) \coloneqq \mathbb{E}_{\mathbf{Z},\mathbf{s}\sim q_\phi}\left[\log p_\theta(\mathbf{X}|\mathbf{Z},\mathbf{s})\right] - \text{KL}(q_\phi(\mathbf{Z},\mathbf{s})\|\,p(\mathbf{Z},\mathbf{s})). \tag{20}$$

D.5. Mixture-based VAEs

For the mixture-based models, we have followed the same architecture and setups as the ones used by Shi et al. (2019); Sutter et al. (2021). When it comes to different models, we only have changed the way we sample the modalities $\mathbf{Z}_A$ by changing the selection of $\mathcal{A}$, but the architectures remain the same as the ones used in previous literature.

Therefore, we here describe the architecture for all the models at once, as they differ on the loss function and the experts, which does not modify the underlying network. We assume the following probabilistic model for the MNIST-SVHN-Text experiments:

$$\begin{aligned}
&\text{Prior:} && p(\mathbf{Z}) = \mathcal{N}(0,I) \\
&\text{Likelihood:} && p_\theta(\mathbf{X}|\mathbf{Z}) = \text{Laplace}(\mathbf{x}_M|\mu(\mathbf{Z};\theta),0.75)\ \text{Laplace}(\mathbf{x}_S|\mu(\mathbf{Z};\theta),0.75)\ \text{Cat}(\mathbf{x}_T|\pi(\mathbf{Z};\theta)) \\
&\text{Variational approx.:} && q_\phi(\mathbf{Z}|\mathbf{X}) = \mathcal{N}(\mu(\mathbf{X};\phi),\sigma(\mathbf{X};\phi))
\end{aligned}$$

where variables are properly transformer to meet their constraints, e.g., we use a softmax to model the class probabilities of the likelihood of the text modality. We consider the following encoders and decoders for each modality:

MNIST:

Encoder: `[Linear-`h`][ReLU][Linear-`h`][ReLU][Linear-`$2l$`]`
Decoder: `[Linear-`h`][ReLU][Linear-`h`][ReLU][Linear-`$2D$`][Sigmoid]`

SVHN:

Encoder: `[Conv-4-2-1][ReLU][Conv-4-2-1][ReLU][Conv-4-2-1][ReLU][Conv-4-1-0]`
Decoder: `[ConvT-4-1-0][ReLU][ConvT-4-2-1][ReLU][ConvT-4-2-1][ReLU][Conv-4-2-1][Sigmoid]`

where the last convolutional layer of the encoder is repeated twice, one for each parameter of the variational approximation.

Text:

Encoder: `[Conv-1-1-0][ReLU][Conv-4-2-1][ReLU][Conv-4-2-0][ReLU][Linear-`$2l$`]`
Decoder: `[Linear-`D`][ConvT-4-1-0][ReLU][ConvT-4-2-1][ReLU][Conv-1-1-0]`

Experimental setup. For each experiment, we train the model for 30 epochs and a batch size of 128. We use AMSGrad (Reddi et al., 2018) with a learning rate of 0.001. Regarding the variational loss, we use $K = 30$ importance samples for all losses (when using the ELBO, we instead use those samples for the Monte Carlo estimator of the outer expectation). For evaluation, we take the model parameters with the highest validation error (10 % of the training data) during training, and report all the metrics with respect to a test set.

E. Experimental details

E.1. Heterogeneous experiments

E.1.1. Dataset descriptions

Likelihood selection. Choosing the proper likelihood is a hard task which requires expert-domain knowledge for each specific setting. We attempt to simplify this process, and instead automatize likelihood selection based on basic properties of the data that can be programmatically verified. Specifically, we use the following criteria:

Real-valued:	$x_d \sim \mathcal{N}(\mu, \sigma)$
Positive real-valued:	$x_d \sim \log\mathcal{N}(\mu, \sigma)$
Count:	$x_d \sim \text{Poiss}(\lambda)$
Binary:	$x_d \sim \text{Bern}(p)$
Categorical:	$x_d \sim \text{Cat}(\pi_1, \pi_2, \ldots, \pi_K)$

Datasets. For the experiments shown in §5.1, we use 12 different heterogeneous and homogeneous datasets. First, we took *Adult*, *defaultCredit*, *Wine*, *Bank marketing*, *El Nino*, *Magic*, and *MiniBooNE* datasets from the UCI repository (Dua & Graff, 2017). Then, we included from the R package datasets (R Core Team, 2021) the following datasets: *Diamonds*, *Movies (IMDB)*, *Health Insurance (HI)*, *German health registry (rwm5yr)*, and *labour*. Table 6 provides the statistics per dataset in terms of sizes and number of likelihoods. It is important to remark that the *IMDB* and *Adult* datasets contain NaNs values (each only in two of the features). We replace them by non-NaNs values and ignore them during training and evaluation using boolean masks (similar to what Nazabal et al. (2020) do).

Table 6. Datasets description. The first two columns describe number of instances, N, and number of features, D. The next columns describe the number of data types per dataset. Note that the last three datasets are homogeneous, and thus only have real variables.

Dataset	N	D	*Real*	*Positive*	*Count*	*Categorical*
Adult	32561	12	0	3	1	7
Credit	30000	24	6	7	1	10
Wine	6497	13	0	11	1	1
Diamonds	53940	10	7	0	0	3
Bank	41188	21	10	0	0	11
IMDB	28819	23	4	1	10	8
HI	22272	12	5	1	0	6
rwm5yr	19609	16	0	2	3	11
labour	15992	9	3	0	2	4
El Nino	178080	12	12	0	0	0
Magic	19020	11	11	0	0	0
BooNE	130065	43	43	0	0	0

Preprocessing. When parsing the dataset, we center all real-valued features by removing their mean. We further standardize real-valued features, computing their (training) standard deviation and dividing the data by this quantity. We also divide by the standard deviation for positive real-valued features (but in the log-space, as we assume a log-normal likelihood). These last two steps are omitted for HI-VAE, since it uses its own normalization layer as described by Nazabal et al. (2020). We also treat non-negative as positive real-valued features by adding a negligible value of 1×10^{-20}. Finally, we make sure that the support of count, binary, and categorical features are in accordance to that of the library used during implementation by removing their minimum value in the case of binary and categorical features, and 1 in the case of count features.

Additionally, we performed some extra preprocessing to the *IMDB* and *Bank* datasets. In the *IMDB* dataset, there are ten features that contain rating percentages of users to the movies, ranging from 0 to 100, at intervals of 0.5. We convert each of them into discrete features starting from one by performing $\mathbf{x}'_d = 2\mathbf{x}_d + 1$ to each of these features, treating them afterwards as count data. As for the *Bank* dataset, we remove the uninformative dimension 12-th as a data cleaning step.

E.1.2. Experimental settings

We train all experiments using Adam as optimizer, with a learning rate of 0.001 for all models. For all models (except HI-VAE) we set the batch size to 128, and train for 400 epochs for the all datasets (except for *Wine* with 2000 epochs). For HI-VAE, we set the batch size to 1000 and the number of epochs to 2000 as in the original paper. We randomly split the data into training ($70\,\%$), validation ($10\,\%$), and testing ($20\,\%$).

We set the latent size of $\mathbf{z}$, d, to $50\,\%$ of the number of features of the dataset, D, and the hidden size of each layer to 50 for all the experiments, except for those of the *Bank* dataset which are set to 100.

Metric. Since we deal with heterogeneous data, where each feature has different type and range, we compute the reconstruction error using metrics that account for these differences. For numerical features (real, positive, and count data) we compute the normalized root mean squared error:

$$\text{err}(d) = \frac{1}{N}\frac{||x_d - \hat{x}_d||_2}{\max(x_d) - \min(x_d)}, \tag{21}$$

where $\hat{x}$ is the model prediction. For the case of nominal features (categorical and binary data) we use the error rate as reconstruction error:

$$\text{err}(d) = \frac{1}{N}\sum_{n=1}^{N} I(x_{n,d} \neq \hat{x}_{n,d}). \tag{22}$$

The final metric shown in Table 1 is the average across dimensions, $\text{err} = \frac{1}{D}\sum_d \text{err}(d)$.

Model selection. In order to make fair comparisons, for each model and dataset we first tuned the hyperparameters (for example, hidden/latent/batch size, number of epochs, etc.) for the vanilla implementations (i.e., without modifying the backward pass). To this end, we ran grid searches and averaged the validation metric over five random seeds, just as in Table 1, choosing the set of hyperparameters that performed the best in terms of reconstruction error during validation. Note that all these hyperparameters (including optimization hyperparameters such as learning rate) are shared across all methods of the same setting. Additionally, we verified that the vanilla models were performing well by visually inspecting the marginal reconstructions.

Selecting the algorithm f_ψ. For the heterogeneous experiments we trained all the possible combinations between the following magnitude-aware algorithms: {nothing, GradNorm (Chen et al., 2018), MGDA-UB (Sener & Koltun, 2018), IMTL-G (Liu et al., 2021b)} and direction-aware algorithms:{nothing, GradDrop (Chen et al., 2020), PCGrad (Yu et al., 2020)} on the training data. This amounts to a total of 12 combinations, plus the hyperparameter of specific algorithms. In this case, we only tune the α parameter from GradNorm between the values zero and one. Then, similar to model selection, we chose the best algorithm by averaging over five random seeds and taking the combination of methods that performed the best in terms of reconstruction error in validation (see Table 7). In general, it was enough to focus on the median to select the best combination. However, some combinations had outliers, and we chose those having a good balance between median, mean, and standard deviation.

Statistical test. In order to compare the performance of the proposed method with the baseline, we employ the corrected paired t-test (Nadeau & Bengio, 2003). The usual paired t-test assumes that the data used to perform the test is independently sampled, which usually does not hold in the machine learning as we sample the training and test data from the same distribution. As a consequence, paired t-test might suggest statistical significance between the compared models, whereas there is no such significance (type I error). Corrected paired t-test considers the dependency of the sampled data, correcting the variance of the differences of the paired samples in the two testing models.

Data Generation. To generate the data for the experiments in §5.1, we followed the same approach as Ghosh et al. (2020) and made use of post-hoc Gaussian Mixture Models (GMMs) to approximate the aggregated posterior, $\mathbb{E}_{\mathbf{X}}\left[q_\phi(\mathbf{Z}|\mathbf{X})\right]$. After

Table 7. The best MTL methods chosen by cross-validation. GN, GD, PG, and MGDA stand for GradNorm, GradDrop, PCGrad, and MGDA-UB, respectively.

Dataset	VAE-ELBO	VAE-IWAE	VAE-DReG	HI-VAE
Adult	IMTL-G	IMTL-G	IMTL-G-PG	GN-PG ($\alpha = 1$)
Credit	IMTL-G	IMTL-G-GD	IMTL-G-GD	GN ($\alpha = 1$)
Wine	GN ($\alpha = 0$)	GN-PG ($\alpha = 0$)	GN ($\alpha = 0$)	GN ($\alpha = 0$)
Diamonds	IMTL-G	IMTL-G	IMTL-G-PG	GN ($\alpha = 0$)
Bank	GN ($\alpha = 0$)	GN-GD ($\alpha = 0$)	GN ($\alpha = 0$)	MGDA-PG
IMDB	GN-GD ($\alpha = 0$)	GN ($\alpha = 0$)	GN-PG ($\alpha = 0$)	GN-PG ($\alpha = 0$)
HI	GN-GD ($\alpha = 0$)	GN ($\alpha = 0$)	GN-PG ($\alpha = 0$)	MGDA
rwm5yr	GN ($\alpha = 1$)	GN-GD ($\alpha = 1$)	GN ($\alpha = 1$)	MGDA-PG
labour	GN ($\alpha = 0$)	GN ($\alpha = 1$)	GN-PG ($\alpha = 0$)	GN ($\alpha = 0$)
El Nino	IMTL-G	IMTL-G-PG	IMTL-G-GD	GN ($\alpha = 0$)
Magic	GN ($\alpha = 1$)	IMTL-G	GN ($\alpha = 1$)	IMTL-G
BooNE	IMTL-G-PG	IMTL-G	GN-PG ($\alpha = 0$)	MGDA-PG

training the VAE models, we use the latent space Z generated from the training data and fit a GMM (with 100 components) on that data. Next, we use this GMM to sample a dataset with as many samples as the test data.

E.1.3. Additional experimental results

In addition to the results presented in the main paper, we present in Table 8 the same table as Table 1 but showing also the standard deviation of the results. Moreover, we show in Figs. 8 and 9 the full pair plot for the *HI* dataset, as well as another full pair plot of the *labour* dataset.

E.2. Multimodal experiments

E.2.1. Experiment details

For the multimodal experiments on MNIST-SVHN-Text, we have followed the same setup (including hyper-parameters) as Shi et al. (2019) and Sutter et al. (2021). We differ from their setups in that, in order to provide a fair comparison between losses, we always employ $K = 30$ samples from $\mathbf{Z}$, whether they are used as importance samples (IWAE, SIWAE) or used for the Monte-Carlo approximation of the expected value w.r.t. $\mathbf{Z}$. Also, we do model selection using a validation dataset (10% of the training data), and use a test set to obtain all the results presented in this work. Following Shi et al. (2019), we use the Sticking-The-Landing estimator (STL) (Roeder et al., 2017) for all losses. In short, this estimator simply omits the partial derivatives of the variational approximation w.r.t. the encoder parameters. Note that Shi et al. (2019) did not mention this estimator, but they rather talk about the DReG loss (Tucker et al., 2019). However, due to a bug in their code, they effectively compute the STL estimator in their experiments.

Selecting algorithm f_ψ. Since the number of impartiality blocks is large, and the training times are considerably longer than for the heterogeneous experiments, here we keep performing cross-validation, but this time we substitute grid-selection by hand-picked hyperparameters options that we observed to perform better than others (for example, we replaced IMTL-G (Liu et al., 2021b) by CAGrad (Liu et al., 2021a), as it was really clear by looking at the logs that IMTL-G was not working at all). Instead of looking for a specific algorithm for each of the impartiality blocks, we assume the same algorithm for all of them (same hyperparameters, but different parameters) and only cross-validate by using the modified backward pass on the blocks associated with the different goals in an incremental way (i.e., as presented in the inset table of §5 in the main paper).

Choosing the best algorithm in the multimodal setup is more complicated, as we care about different metrics (coherence and latent classification) at different levels (self and cross metrics), for each modality. We group all metrics in metric-type pairs (e.g., latent-classification-self), and within each group, we group them by the expert/modality they are testing (e.g., cross latent classification for the first expert tests all other latent samples in the classifier of the first expert). For each metric, we compute a value $I(x, y)$, where x is the value obtained by the algorithm, and y the value obtained by the baseline, and take the average of each sequence recursively until obtaining a single number. We use the relative improvement

Table 8. Test reconstruction errors (mean and standard deviation) of different models and losses for the baseline and our framework.

Dataset	*Method*	VAE-ELBO	VAE-IWAE	VAE-DReG	HI-VAE
Adult	vanilla	0.21 ± 0.01	0.22 ± 0.02	0.24 ± 0.01	0.13 ± 0.00
	ours	$\mathbf{0.11} \pm 0.02$	$\mathbf{0.12} \pm 0.02$	0.19 ± 0.08	$\mathbf{0.09} \pm 0.02$
defaultCredit	vanilla	0.13 ± 0.00	0.14 ± 0.02	0.14 ± 0.01	0.15 ± 0.09
	ours	$\mathbf{0.04} \pm 0.00$	$\mathbf{0.05} \pm 0.01$	$\mathbf{0.08} \pm 0.01$	0.06 ± 0.01
Wine	vanilla	0.09 ± 0.00	0.08 ± 0.00	0.08 ± 0.00	0.13 ± 0.01
	ours	$\mathbf{0.07} \pm 0.01$	$\mathbf{0.07} \pm 0.00$	$\mathbf{0.07} \pm 0.00$	$\mathbf{0.11} \pm 0.02$
Diamonds	vanilla	0.19 ± 0.01	0.18 ± 0.01	0.18 ± 0.00	0.11 ± 0.02
	ours	$\mathbf{0.13} \pm 0.02$	$\mathbf{0.12} \pm 0.01$	$\mathbf{0.14} \pm 0.01$	$\mathbf{0.01} \pm 0.01$
Bank	vanilla	0.20 ± 0.00	0.20 ± 0.00	0.19 ± 0.00	0.13 ± 0.02
	ours	$\mathbf{0.04} \pm 0.00$	0.10 ± 0.05	0.11 ± 0.04	$\mathbf{0.10} \pm 0.01$
IMDB	vanilla	0.09 ± 0.02	0.10 ± 0.02	0.10 ± 0.02	0.08 ± 0.00
	ours	$\mathbf{0.05} \pm 0.04$	$\mathbf{0.05} \pm 0.04$	$\mathbf{0.06} \pm 0.04$	0.10 ± 0.09
HI	vanilla	0.17 ± 0.01	0.16 ± 0.00	0.15 ± 0.00	0.11 ± 0.00
	ours	$\mathbf{0.04} \pm 0.00$	$\mathbf{0.04} \pm 0.00$	$\mathbf{0.04} \pm 0.00$	$\mathbf{0.11} \pm 0.01$
rwm5yr	vanilla	0.11 ± 0.01	0.09 ± 0.00	0.10 ± 0.00	0.04 ± 0.01
	ours	$\mathbf{0.03} \pm 0.00$	$\mathbf{0.03} \pm 0.01$	$\mathbf{0.03} \pm 0.00$	$\mathbf{0.02} \pm 0.00$
labour	vanilla	0.11 ± 0.00	0.10 ± 0.00	0.10 ± 0.00	0.10 ± 0.00
	ours	$\mathbf{0.06} \pm 0.00$	$\mathbf{0.07} \pm 0.00$	$\mathbf{0.08} \pm 0.01$	$\mathbf{0.07} \pm 0.00$
EL Nino	vanilla	0.10 ± 0.01	0.09 ± 0.00	0.08 ± 0.00	0.10 ± 0.01
	ours	$\mathbf{0.07} \pm 0.01$	$\mathbf{0.06} \pm 0.01$	$\mathbf{0.07} \pm 0.00$	$\mathbf{0.02} \pm 0.00$
Magic	vanilla	0.06 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	0.06 ± 0.00
	ours	0.06 ± 0.00	0.05 ± 0.00	0.05 ± 0.00	$\mathbf{0.03} \pm 0.00$
BooNE	vanilla	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.00	0.04 ± 0.00
	ours	0.04 ± 0.00	$\mathbf{0.04} \pm 0.00$	0.04 ± 0.00	0.04 ± 0.00

Table 9. Reconstruction coherence ($A = \{M, S, T\}$) for each modality, model, and dataset.

		ELBO			IWAE			SIWAE		
	$\mathbf{x}_d$	M	S	T	M	S	T	M	S	T
MVAE	vanilla	97.53	88.26	99.30	97.27	87.19	98.76	97.37	87.47	98.83
	ours	**97.85**	**89.65**	**99.64**	**98.28**	**89.01**	**99.93**	97.42	87.63	**99.20**
MMVAE	vanilla	86.01	45.59	89.17	85.25	84.03	88.66	58.95	61.27	63.27
	ours	**89.42**	45.83	**91.54**	87.55	**86.87**	90.93	**74.85**	**73.89**	**81.09**
MoPoE	vanilla	95.72	85.86	98.01	95.82	87.55	97.93	75.10	67.16	76.61
	ours	**96.50**	**93.60**	**99.14**	**97.29**	**92.93**	**99.00**	96.91	89.01	99.28

Table 10. Self and cross generation coherence (%) results for different models on **M**NIST-**S**VHN-**T**ext, trained using ELBO and averaged over 5 different seeds. Models trained with our framework are able to sample more coherent modalities.

					Self coherence			Cross coherence									
	$\boldsymbol{x}_d$				M	S	T	M			S			T			Time
	A	LI	EEI	DEI	M	S	T	S	T	S,T	M	T	M,T	M	S	M,S	h
MVAE	vanilla	☐	☐	☐	80.30	12.63	25.77	11.12	18.22	19.49	43.29	18.50	16.90	51.82	11.65	54.71	3.82
	CA ($\alpha = 0.4$)	☑	☐	☐	85.12	12.34	34.64	10.70	12.51	16.83	44.94	22.36	**29.45**	61.41	12.26	69.77	4.82
MMVAE	vanilla	☐	☐	☐	**95.23**	68.25	99.99	62.93	99.92	81.43	31.06	37.42	34.24	96.27	71.29	83.79	10.07
	CA ($\alpha = 10$)	☑	☑	☑	92.62	**73.84**	99.99	**76.01**	99.59	**87.80**	29.11	34.46	31.78	95.27	**79.34**	**87.31**	13.06
MoPoE	vanilla	☐	☐	☐	94.52	71.21	99.99	66.80	99.98	98.12	**19.70**	33.25	31.55	**96.79**	77.04	**97.15**	23.24
	CA ($\alpha = 10$)	☑	☑	☑	94.64	**73.36**	**100.00**	**74.76**	99.98	98.42	15.84	32.84	31.03	96.03	**78.61**	96.67	29.41

$I(x, y) = \Delta(x, y) = \frac{x-y}{y}$ to compare the different metrics, choosing the method that obtains the best improvement, averaged across experts/modalities and metrics. For MVAE, we noticed that the metrics tend to oscillate and there are important trade-offs in performance. Therefore, for this model we adopt a more conservative approach and use $I(x, y) = \mathbf{1}_{x \geq y}$, to choose the algorithm that, on average, improves the most number of metrics.

E.3. Additional experimental results

In this section, we have included the complete results for the MNIST-SVHN-Text experiments. Specifically, we present: the reconstruction coherence results for all the three losses (Table 9); the self and cross coherence results in tabular form for the three losses (Tables 10 to 12), including extra information like the training times, the specific MTL algorithms used, and the goals for which we apply them; and the log-likelihoods conditioned on different modalities (Tables 13 to 15), showing standard deviations as space permits, thus showing the high variance that the vanilla approach shows at times (for example, MoPoE in Table 15). Finally, we present the parallel coordinate plots for the three models (Fig. 10).

Table 11. Self and cross generation coherence (%) results for different models on **M**NIST-**S**VHN-**T**ext, trained using IWAE and averaged over 5 different seeds. Models trained with our framework are able to sample more coherent modalities.

					Self coherence			Cross coherence									
	$\boldsymbol{x}_d$				M	S	T	M			S			T			Time
	A	LI	EEI	DEI	M	S	T	S	T	S,T	M	T	M,T	M	S	M,S	h
MVAE	vanilla	☐	☐	☐	87.33	11.70	37.49	10.76	**26.63**	29.48	54.10	25.16	29.51	**70.24**	11.38	72.33	3.80
	GN ($\alpha = 0.0$)	☑	☐	☐	85.73	12.68	**79.19**	11.18	19.45	22.05	49.37	**55.83**	**54.48**	59.56	11.79	63.69	4.24
MMVAE	vanilla	☐	☐	☐	94.70	68.13	99.99	62.34	98.79	80.55	86.72	97.29	92.01	**96.83**	69.23	83.02	10.08
	CA ($\alpha = 0.4$)	☑	☐	☐	94.96	73.66	99.99	68.49	**99.25**	83.85	**88.99**	**97.97**	**93.49**	96.27	76.55	86.40	11.96
MoPoE	vanilla	☐	☐	☐	94.44	63.16	99.97	58.20	99.01	99.24	80.75	94.76	88.62	96.51	66.13	96.03	23.27
	CA ($\alpha = 10.0$)	☑	☑	☑	**95.41**	**69.54**	**99.99**	61.75	99.06	99.07	81.23	**96.47**	**92.98**	96.64	**73.33**	96.16	29.35

Table 12. Self and cross generation coherence (%) results for different models on **M**NIST-**S**VHN-**T**ext, trained using SIWAE and averaged over 5 different seeds. Models trained with our framework are able to sample more coherent modalities.

					Self coherence			Cross coherence									
	$\boldsymbol{x}_d$				M	S	T	M			S			T			Time
	A	LI	EEI	DEI	M	S	T	S	T	S,T	M	T	M,T	M	S	M,S	h
MVAE	vanilla	☐	☐	☐	82.06	12.08	36.67	10.34	17.12	19.19	49.99	19.31	31.19	62.50	10.82	64.25	3.81
	CA ($\alpha = 0.4$)	☑	☐	☐	86.89	12.89	**59.15**	10.99	20.92	**30.34**	54.83	**31.82**	24.97	71.87	11.10	72.74	4.77
MMVAE	vanilla	☐	☐	☐	**95.90**	48.30	53.02	28.43	52.52	40.45	84.44	51.08	67.77	**96.80**	39.96	68.38	10.07
	CA ($\alpha = 0.4$)	☑	☐	☐	95.90	58.20	**88.70**	**49.33**	**79.32**	**64.30**	**87.29**	**76.17**	**81.71**	96.70	57.86	77.28	12.30
MoPoE	vanilla	☐	☐	☐	92.32	11.60	69.05	10.13	51.02	34.67	41.93	46.39	51.58	85.19	10.57	67.54	23.29
	GN ($\alpha = 0.5$)	☑	☐	☐	90.99	12.00	**83.82**	**10.63**	62.75	52.08	28.19	46.91	43.34	79.64	10.81	90.33	24.58

Table 13. Log-likelihood of the joint generative model, conditioned on the variational posterior of subsets of the modalities. Results report on ELBO as loss function and are averaged over 5 different seeds. Each log-likelihood is divided by the dimensionality of its modality before adding them up, to better reflect improvement across modalities.

		$\mathbb{X}$|M	$\mathbb{X}$|S	$\mathbb{X}$|T	$\mathbb{X}$|MS	$\mathbb{X}$|MT	$\mathbb{X}$|ST	$\mathbb{X}$|MST
MVAE	vanilla	-8.61 ± 1.18	-10.26 ± 1.14	-8.17 ± 1.30	-8.18 ± 1.26	-1.51 ± 0.11	-7.41 ± 1.38	-1.02 ± 0.11
	ours	-8.07 ± 0.88	-9.89 ± 0.62	$\mathbf{-6.94 \pm 1.19}$	-7.44 ± 0.64	-1.42 ± 0.05	-6.12 ± 1.69	-0.95 ± 0.00
MMVAE	vanilla	-2.30 ± 0.21	$\mathbf{-2.18 \pm 0.09}$	-1.19 ± 0.00	$\mathbf{-2.24 \pm 0.11}$	-1.75 ± 0.10	$\mathbf{-1.68 \pm 0.04}$	$\mathbf{-1.89 \pm 0.07}$
	ours	-2.59 ± 0.40	-2.31 ± 0.13	$\mathbf{-1.19 \pm 0.00}$	-2.45 ± 0.23	-1.89 ± 0.20	-1.75 ± 0.06	-2.03 ± 0.15
MoPoE	vanilla	-1.93 ± 0.01	-2.06 ± 0.04	-1.19 ± 0.00	-1.76 ± 0.00	-1.18 ± 0.00	-1.04 ± 0.00	-1.03 ± 0.05
	ours	-1.92 ± 0.03	-2.09 ± 0.05	$\mathbf{-1.19 \pm 0.00}$	-1.75 ± 0.02	-1.17 ± 0.01	$\mathbf{-1.03 \pm 0.00}$	-1.00 ± 0.01

Table 14. Log-likelihood of the joint generative model, conditioned on the variational posterior of subsets of the modalities. Results report on IWAE as loss function and are averaged over 5 different seeds. Each log-likelihood is divided by the dimensionality of its modality before adding them up, to better reflect improvement across modalities.

		$\mathbb{X}$|M	$\mathbb{X}$|S	$\mathbb{X}$|T	$\mathbb{X}$|MS	$\mathbb{X}$|MT	$\mathbb{X}$|ST	$\mathbb{X}$|MST
MVAE	vanilla	-8.62 ± 0.40	-10.68 ± 0.98	-6.78 ± 1.66	-8.28 ± 0.47	-1.83 ± 0.14	-6.08 ± 1.36	-1.15 ± 0.03
	ours	$\mathbf{-8.26 \pm 0.29}$	$\mathbf{-9.51 \pm 0.25}$	$\mathbf{-2.83 \pm 0.54}$	$\mathbf{-7.74 \pm 0.15}$	$\mathbf{-1.35 \pm 0.03}$	$\mathbf{-1.96 \pm 0.33}$	$\mathbf{-0.94 \pm 0.00}$
MMVAE	vanilla	$\mathbf{-1.98 \pm 0.06}$	-2.74 ± 0.32	-1.27 ± 0.00	-2.36 ± 0.15	$\mathbf{-1.62 \pm 0.03}$	-2.00 ± 0.16	-1.99 ± 0.10
	ours	-2.44 ± 0.03	$\mathbf{-2.45 \pm 0.12}$	$\mathbf{-1.26 \pm 0.00}$	-2.44 ± 0.06	-1.85 ± 0.01	$\mathbf{-1.86 \pm 0.06}$	-2.05 ± 0.04
MoPoE	vanilla	$\mathbf{-1.97 \pm 0.00}$	-2.55 ± 0.12	-1.27 ± 0.00	$\mathbf{-5.85 \pm 0.52}$	-1.25 ± 0.00	-1.05 ± 0.00	-1.02 ± 0.00
	ours	-2.01 ± 0.01	-2.61 ± 0.11	$\mathbf{-1.27 \pm 0.00}$	-7.07 ± 0.06	$\mathbf{-1.24 \pm 0.01}$	-1.05 ± 0.00	$\mathbf{-1.01 \pm 0.00}$

Table 15. Log-likelihood of the joint generative model, conditioned on the variational posterior of subsets of the modalities. Results report on SIWAE as loss function and are averaged over 5 different seeds. Each log-likelihood is divided by the dimensionality of its modality before adding them up, to better reflect improvement across modalities.

		$\mathbb{X}$|M	$\mathbb{X}$|S	$\mathbb{X}$|T	$\mathbb{X}$|MS	$\mathbb{X}$|MT	$\mathbb{X}$|ST	$\mathbb{X}$|MST
MVAE	vanilla	-8.82 ± 0.63	-10.13 ± 0.29	-7.17 ± 1.73	-8.52 ± 0.74	-2.23 ± 1.31	-5.94 ± 1.35	-1.15 ± 0.02
	ours	-8.41 ± 0.14	-10.52 ± 0.53	-5.65 ± 1.76	-8.13 ± 0.17	-1.46 ± 0.07	-5.49 ± 1.87	$\mathbf{-1.01 \pm 0.01}$
MMVAE	vanilla	$\mathbf{-2.01 \pm 0.08}$	-4.04 ± 0.43	-3.25 ± 0.94	-3.02 ± 0.20	-2.63 ± 0.49	-3.64 ± 0.25	-3.10 ± 0.18
	ours	-2.34 ± 0.36	-3.22 ± 1.28	-3.03 ± 1.07	-2.78 ± 0.59	-2.69 ± 0.54	$\mathbf{-3.12 \pm 0.43}$	$\mathbf{-2.86 \pm 0.22}$
MoPoE	vanilla	-6.29 ± 2.75	-10.02 ± 0.43	-3.80 ± 1.48	-7.51 ± 0.93	-1.59 ± 0.35	-4.97 ± 2.68	-3.15 ± 3.78
	ours	-8.01 ± 0.42	-10.16 ± 0.55	-2.96 ± 1.11	-7.26 ± 0.28	-1.34 ± 0.01	$\mathbf{-2.30 \pm 0.91}$	-0.99 ± 0.02

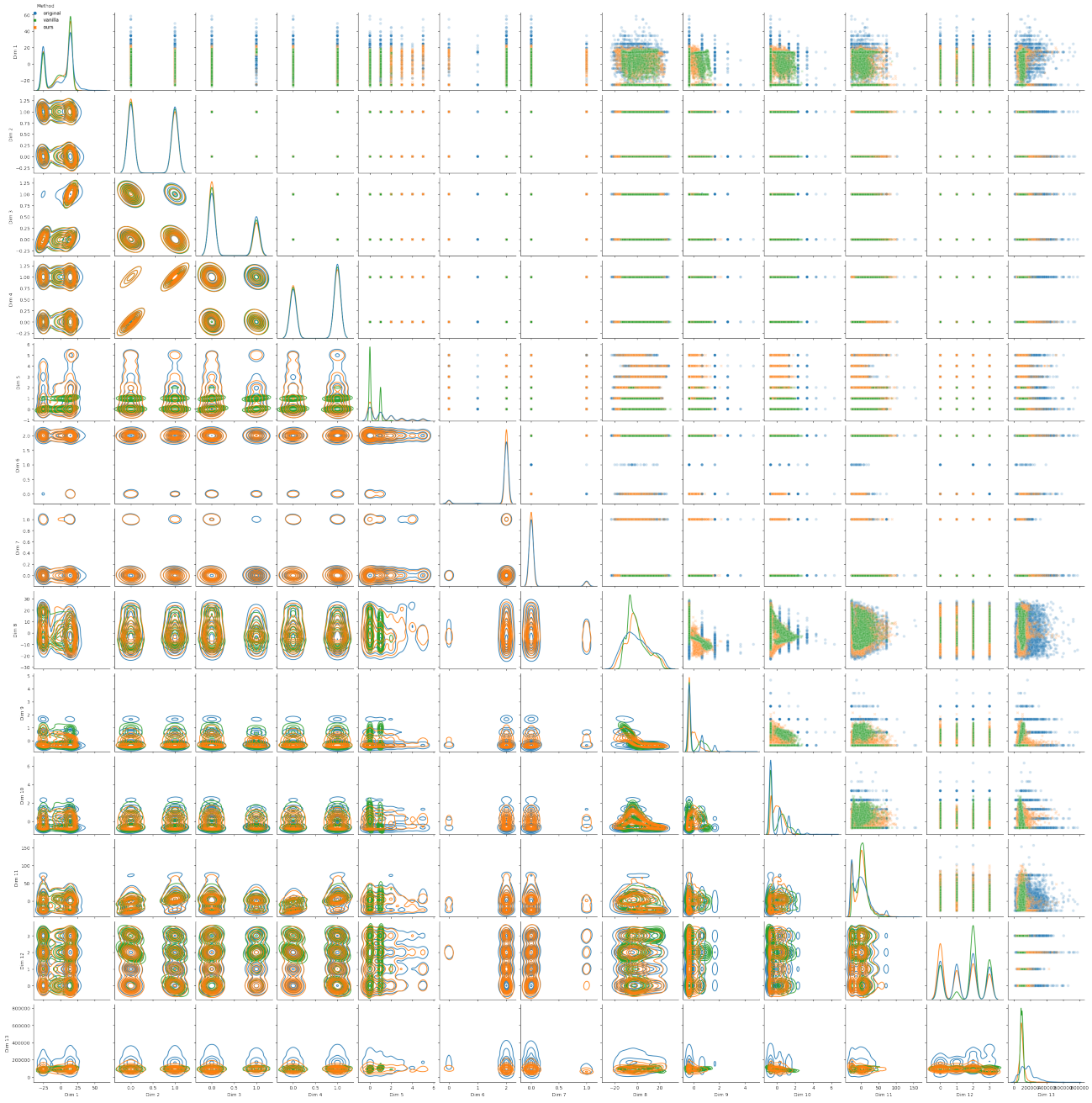


Figure 8. Pair plot of all the dimensions of *HI*, generated from different VAE models. Diagonal show the marginals, upper-diagonals scatter plots, and lower-diagonals kernel density estimates. The VAE trained with our approach is able to generate faithful samples.

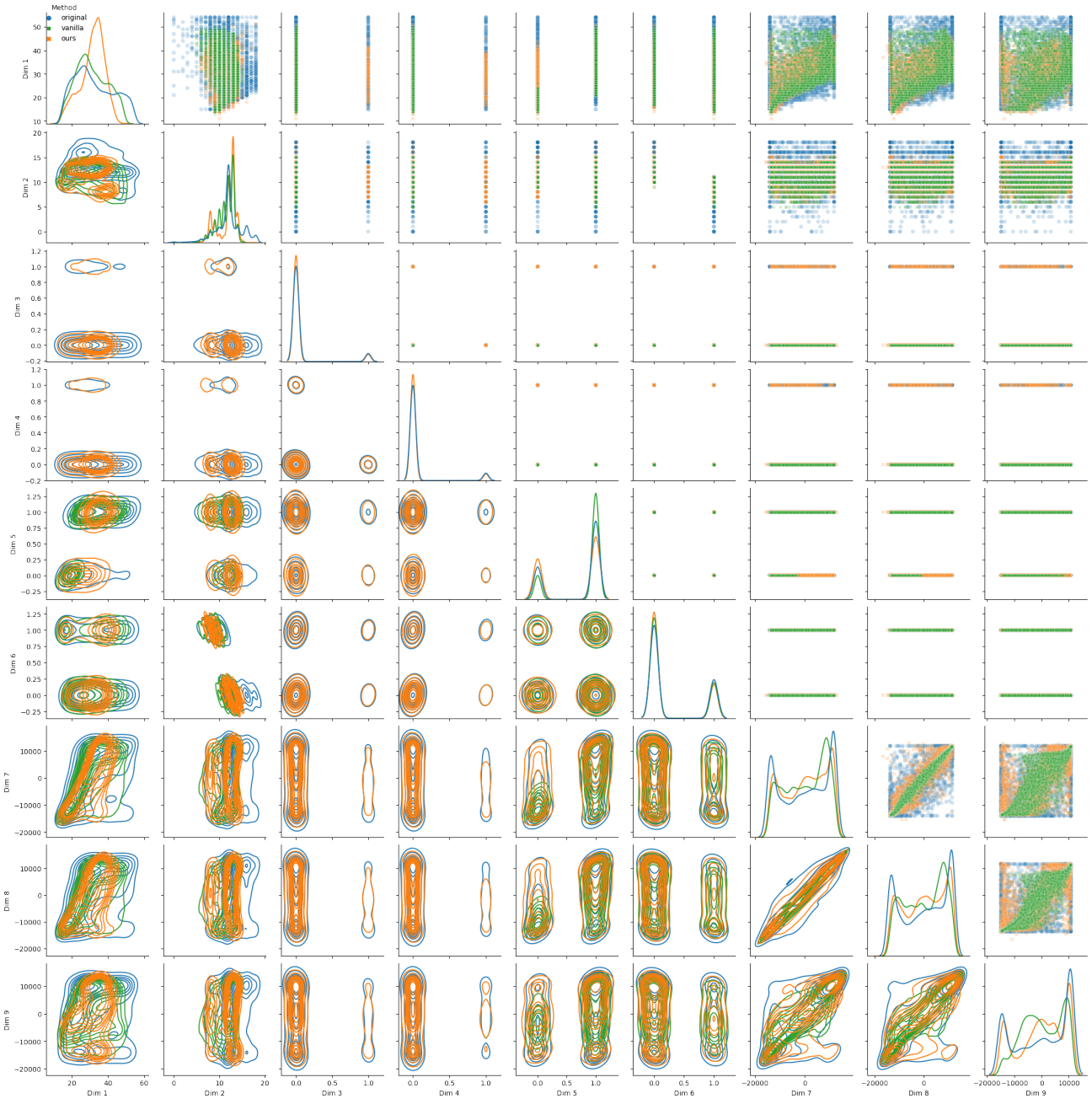
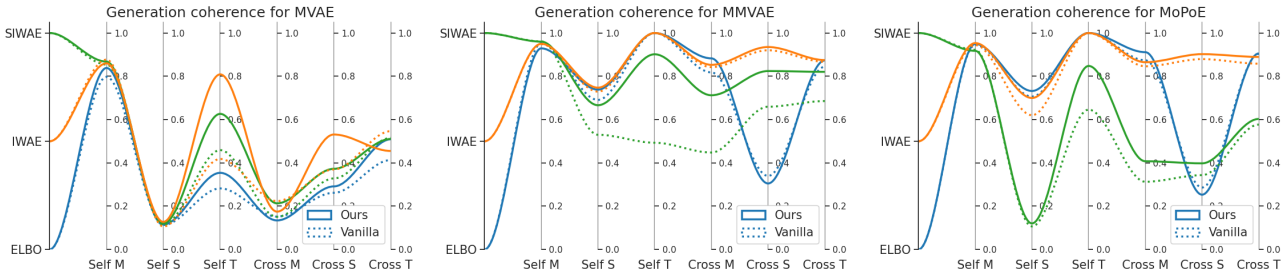


Figure 9. Pair plot of all the dimensions of *labour*, generated from different VAE models. Diagonal show the marginals, upper-diagonals scatter plots, and lower-diagonals kernel density estimates. The VAE trained with our approach is able to generate faithful samples.



(a) MVAE. (b) MMVAE. (c) MoPoE.

Figure 10. Generation coherence results for all models and losses. In general, improve all metrics w.r.t. the baseline.