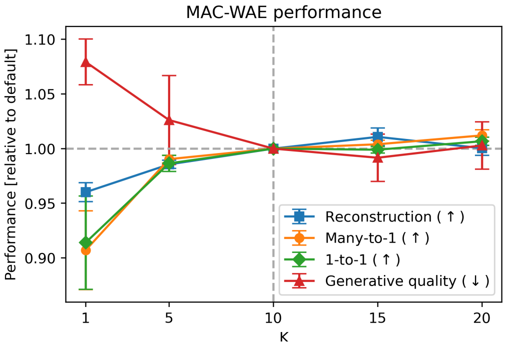
Supplementary Materials

Tomas Tokar, Scott Sanner

1 Supplementary results

1.1 Performance dependence on hyper-parameter K

We performed additional experiment in which we re-trained and re-tested MAC-WAE on PolyMNIST dataset (we found this dataset to be best suited for this experiment, as it contains only image modalities) under varying number of conditional indices being sampled (hyper-parameter K in the Algorithm 1), while keeping all the remaining parameters in their default setting. The results were normalized relative to the performance under the default value of K and then averaged across the modalities. The obtained results show that, across all tasks, the performance improves with the growing K, saturating between the values $K = 10$ and $K = 20$ (cf. Supplementary Figure 1).



Supplementary Figure 1: Performance of MAC-WAE across different tasks as a function of the number of conditioning indices sampled per batch (hyper-parameter K in the Algorithm 1), measured on the PolyMNIST dataset. The performance is reported relative to the default setting ($K = 10$).

1.2 Scalability with respect to hyper-parameter K

As can be seen from the MAC-WAE algorithm (Algorithm 1, cf. Section 3), MAC-WAE scales linearly with the number of conditional indices being sampled (K in the Algorithm 1). The favorable scaling is another advantage of our method.

1.3 Unconditional generative performance

We evaluated the non-conditioned generative performance of MAC-WAE and other models. Since comprehensive evaluation of the generative quality in the multimodal setup is difficult, as there are no established measures of generative quality for some types of modalities (e.g. trajectory in MHD), wo resorted to computing the FID (Frechet Inception Distance) scores [4] of the generated images only. While MAC-WAE was not able to consistently outperform all the baselines (RQ1, d), it stands above average (cf. Supplementary Table 5), indicating that its reconstruction and modality translation performance does not come at the expense of the generative qualities.

2 Possible extensions

The proposed model could possibly be enhanced, for instance, by incorporating a multimodal contrastive loss [19], or introducing modality-specific private and shared latent variables [9, 14] that have been previously reported beneficial. Additionally, while we refrained from employing regularizations such as dropout or batch normalization to isolate the model's core contributions, their inclusion presents another promising

3 Algorithms

Algorithm 1 MAC-WAE – single batch forward pass

```
Inputs:
$\mathbf{x} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(M)}\}$                {Data batch}
$f, g$                                                  {Encoder and decoder}
$C = \{c^{(1)}, \ldots c^{(M)}\}$                       {Reconstruction loss functions}
$\mathcal{D}, \beta$                                    {Regularization loss function and its strength}
Outputs: $L$                                            {Single batch loss}
Forward:
$L = 0$                                                 {Initialize loss}
for $i = 1$ to $K$ do
  $\mathbf{b} \sim p_B$                                 {Sample conditioning index}
  $\tilde{\mathbf{x}} \leftarrow \mathbf{x}$            {Initialize conditioned batch}
  for $m = 1$ to $M$ do
    if $b^{(m)} = 0$ then
      $\tilde{\mathbf{x}}^{(m)} \leftarrow \mathbf{0}$  {Set modality $m$ to zero}
    end if
  end for
  $q(\mathbf{z}|\mathbf{x}, \mathbf{b}) \leftarrow f(\tilde{\mathbf{x}}, \mathbf{b})$    {Encode}
  $\mathbf{z} \sim q(\mathbf{x}|\mathbf{z}, \mathbf{b})$                                 {Sample from posterior}
  $l_i \leftarrow 0$                                    {Init $i$-th conditioning loss}
  for $m = 1$ to $M$ do
    $\hat{\mathbf{x}}^{(m)} \leftarrow g^{(m)}(\mathbf{z})$                              {Decode}
    $l_i \leftarrow l_i + c^{(m)}(\mathbf{x}^{(m)}, \hat{\mathbf{x}}^{(m)})$             {Add recon. loss}
  end for
  $l_i \leftarrow l_i + \beta \cdot \mathcal{D}(\mathbf{z})$                             {Add regularization loss}
  $L \leftarrow L + l_i$                                {Update loss}
end for
$L \leftarrow L/K$                                      {Finalize loss}
```

4 Proofs

Definition 4.1. A functional $F[p(x)]$ is convex with respect to probability density function $p(x)$ if for any two valid probability density functions $p_1(x)$ and $p_2(x)$, and for any $\lambda \in [0,1]$, the following holds:

$$F[\lambda p_1(x) + (1-\lambda)p_2(x)] \leq \lambda F[p_1(x)] + (1-\lambda)F[p_2(x)]$$

Definition 4.2. The expected value of a function $f(x)$, given a probability density function $p(x)$, is defined as:

$$\mathbb{E}_{p(x)}[f(x)] = \int f(x)p(x)dx$$

Corollary 4.3. *The expected value of a function $f(x)$ is convex with respect to the probability density function $p(x)$.*

Proof.

$$\begin{aligned}\lambda \mathbb{E}_{p_1(x)}[f(x)] + (1-\lambda)\mathbb{E}_{p_2(x)}[f(x)] &= \lambda \int f(x)p_1(x)dx + (1-\lambda)\int f(x)p_2(x)dx \\ &= \int f(x)[\lambda p_1(x) + (1-\lambda)p_2(x)]dx \\ &= \mathbb{E}_{\lambda p_1(x)+(1-\lambda)p_2(x)}[f(x)]\end{aligned}$$

□

Lemma 4.4. *For any fixed $p(x)$ the maximum mean discrepancy $\mathcal{D}_{MMD}(p(x), q(x))$ is convex with respect to probability density function $q(\mathbf{x})$.*

Proof.

$$\mathcal{D}_{\text{MMD}}(p(x), q(x)) = \mathbb{E}_{x,x' \sim p(x)}[k(x,x')] + \mathbb{E}_{x,x' \sim q(x)}[k(x,x')] - 2\mathbb{E}_{x\sim p(x), x' \sim q(x)}[k(x,x')]$$

Since, as we just proved (Corollary 4.3), the expected values are convex with respect to their sampling probability density functions and *the sum of convex functions is also convex*, if follows that $\mathcal{D}_{\text{MMD}}(p(x), q(x))$ itself is convex with respect to probability density function $q(x)$. □

Theorem 4.5. *For any fixed prior probability density function $p(\mathbf{z})$, the loss $\mathcal{L}_{MACWAE}$ incurred by the model is a subject to the following inequality:*

$$\mathcal{L}_{MACWAE} \geq \mathbb{E}_{p(\mathbf{b})}\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{b})}[c(\mathbf{x}, g(\mathbf{z}))] + \beta \cdot \mathcal{D}_{MMD}(p(\mathbf{z}), q(\mathbf{z}))$$

where $q(\mathbf{z}) = \mathbb{E}_{p_\mathbf{b}}[q(\mathbf{z}|\mathbf{b})]$.

Proof.

$$\begin{aligned}\mathcal{L}_{\text{MACWAE}} &= \mathbb{E}_{p(\mathbf{b})}\Big[\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{b})}[C(\mathbf{x}, g(\mathbf{z}))] + \beta \cdot \mathcal{D}_{\text{MMD}}(p(\mathbf{z}), q(\mathbf{z}|\mathbf{b}))\Big] \\ &= \mathbb{E}_{p(\mathbf{b})}\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{b})}[C(\mathbf{x}, g(\mathbf{z}))] + \beta \cdot \mathbb{E}_{p(\mathbf{b})}[\mathcal{D}_{\text{MMD}}(p(\mathbf{z}), q(\mathbf{z}|\mathbf{b}))] \\ &\geq \mathbb{E}_{p(\mathbf{b})}\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{b})}[C(\mathbf{x}, g(\mathbf{z}))] + \beta \cdot \mathcal{D}_{\text{MMD}}(p(\mathbf{z}), \mathbb{E}_{p(\mathbf{b})}[q(\mathbf{z}|\mathbf{b})]) \\ &= \mathbb{E}_{p(\mathbf{b})}\mathbb{E}_{p(\mathbf{x})}\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{b})}[C(\mathbf{x}, g(\mathbf{z}))] + \beta \cdot \mathcal{D}_{\text{MMD}}(p(\mathbf{z}), q(\mathbf{z}))\end{aligned}$$

where the inequality in the third line comes from Jensen's inequality, provided $\mathcal{D}_{\text{MMD}}$ is convex function with respect to $q(\mathbf{z})$, for any fixed prior probability density $p(\mathbf{x})$ [3]. The validity of this condition was just proven (Lemma 4.4). □

5 Experimental Details

Models implementation All experiments were conducted using PyTorch (v2.2.1) implementations of the proposed method, its ablated version, and baseline models. Baseline implementations were sourced from the MultiVAE library [1], a publicly available Python library containing a validated collection of multimodal VAE models [17]. Unless specified otherwise, all model-specific parameters were kept at their default values. To ensure a fair comparison, all models utilized the same modality-specific encoder and decoder architectures, as detailed in Table 3.

Training & Testing Training was performed under a consistent set of hyper-parameters, with values drawn from the literature and summarized in Table 2. The models were trained on the pre-specified training subsets of the respective datasets. For each dataset we performed three experimental replicates, each time using different initialization seeds. The reported results are average values across the replicates. For each model, the value of the hyper-parameter β that resulted in the lowest loss incurred across the validation subset was selected. The reported performance indicates the models' evaluation on the testing set, under the selected value of β. The raw results files can be found in the Supplementary Materials. The experiments were performed across 4 NVIDIA RTX A6000 GPUs, and 28 AMD EPYC-Rome Processors, on Ubuntu 20.04.6 LTS (Focal Fossa). The project code is publicly available at: `https://github.com/tomastokar/MACWAE`

Performance metrics Performance was measured in terms of the quality of the reconstructions and modality translation. The quality of the reconstruction and modality translation was measured using the modality-specific evaluation metrics, namely: accuracy (ACC) for the categorical modalities, root-mean-squared error (RMSE) for the numerical modalities, structural similarity index (SSIM) [27] for the images and BLEU-3 score (BLEU3) [15] for text. All the metrics were computed using the package `torcheval` (v0.0.7, nightly)[2]. The modality translation was evaluated in *many-to-1* fashion, i.e. modeling the conditional $p(\mathbf{x}^{(\backslash e)}|p(\mathbf{x}^{(e)}))$, and *1-to-1* fashion, i.e. modeling the conditionals $p(\mathbf{x}^{(q)}|p(\mathbf{x}^{(e)}))\,, q \neq e$. In the case of 1-to-1 translation, for each *query* modality q we report the average calculated across all the remaining modalities, which thus served as for the evidence e. In the case of the `CelebA` data, the values obtained across the 40 binary attributes were averaged, and are reported as a single modality.

Average performance rank Let $r^v_{t,m}$ denote the ranking of the model m in the task t and the modality m, so that $r^v_{t,m} = 1$ means that the model v achieved the best performance among the six models, while $r^v_{t,m} = 0$ means the worst performance. For each task, we aggregated the obtained ranks into *average performance rank* by arithmetic mean computed across all datasets and modalities: $r^v_t = \frac{1}{N}\sum_m r^v_{t,m}$; where N is the total number of modalities across the datasets ($N = 14$). In addition, we computed the standard error of the obtained values. The results are visualized as a bar plot with error bars, providing a comparative overview of the models performance.

Win-loss matrix The models were compared in pair-wise fashion and the results were conveyed as a win-loss matrix, summarizing the total number of outcomes where the model in the given row outperformed the model in the given column, normalized by the total number of outcomes the models participated in (cf. Supplementary Table 4).

Binomial test To assess the statistical significance of the win proportions in the win-loss matrix, a one-tailed binomial test was conducted for each entry. The test was designed to evaluate whether the observed proportion of wins for a given model is greater than what would be expected by chance. The null hypothesis thus says that observed proportion of the wins r is not greater than 0.5 – the expected proportion under random chance:

$$H_0 : r \leq 0.5$$
$$H_1 : r > 0.5$$

[1] `https://github.com/AgatheSenellart/MultiVae`

[2] `https://github.com/pytorch/torcheval`

The entries with the p–$value < 0.05$ were considered significant (*), and those with p–$value < 0.01$ were considered highly significant (**).

6 Supplementary Tables

Supplementary Table 1: Table summarizing the works related to our research.

Domain	Model class	Model name	Ref.
Multimodal VAEs	Surrogate unimodal inference models	JMVAE	[24]
		M2VAE	[8]
		VAEVAE	[30]
		JNF–DCCA	[16]
	Mixture-based posterior models	MVAE	[29]
		MMVAE	[18]
		mmJSD	[22]
		MoPoE	[23]
		MVTCAE	[5]
		MMVAE+	[14]
	Hierarchical models	VAEM	[12]
		Hi-VAE	[13]
		Nexus	[26]
Wasserstein regularization	AEs	WAE	[25]
		SWAE	[7]
		SWD-AE	[28]
	GANs	WGAN	[1]
		SWD-GAN	[28]
Arbitrary conditioning	VAEs	PM-VAE	[21]
		VAEAC	[6]
		EDDI	[11]
	GANs	NC	[2]
	NFs	AC-Flow	[10]
	AR	ACE	[20]

Supplementary Table 2: Hyper-parameters as used in our experiments. The β indicates parameter controlling the strength of regularization, p_0 is probability of the modality to be set as non-observed and K is the number of conditioning indices being sampled per batch. All the remaining parameters were set as per default.

Dataset	Learning rate	Latent dimension	# Epochs	β	p_0	K
PolyMNIST	5.0×10^{-4}	128	150	$\{0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0\}$	0.40	10
MHD	1.0×10^{-3}	128	150	$\{0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0\}$	0.40	10
CUB	5.0×10^{-4}	128	200	$\{0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0\}$	0.25	5
CelebA	1.0×10^{-4}	128	150	$\{1.0\}$	0.40	10

Supplementary Table 3: Summary of encoder and decoder architectures used for individual modalities across the given datasets. Note, dim indicates latent space dimension and $|M|$ is the number of modalities in the dataset.

Dataset	Modality	Encoder	Decoder
PolyMNIST	m0–m4	Input: $\mathbb{R}^{3\times 28\times 28}$ Conv 32, kernel 3×3, stride 2, pad 1 + ReLU Conv 64, kernel 3×3, stride 2, pad 1 + ReLU Conv 128, kernel 3×3, stride 2, pad 1 + ReLU FC dim + ReLU FC dim	Input: $\mathbb{R}^{dim}$ FC 2048 + ReLU ConvTrans 64, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 32, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 3, kernel 3×3, stride 2, pad 1 + Sigmoid
MHD	image	Input: $\mathbb{R}^{1\times 28\times 28}$ Conv 32, kernel 3×3, stride 2, pad 1 + ReLU Conv 64, kernel 3×3, stride 2, pad 1 + ReLU Conv 128, kernel 3×3, stride 2, pad 1 + ReLU FC dim + ReLU FC dim	Input: $\mathbb{R}^{dim}$ FC 2048 + ReLU ConvTrans 64, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 32, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 1, kernel 3×3, stride 2, pad 1 + Sigmoid
	label	Input: $\mathbb{R}^{10}$ FC 32 + ReLU FC dim	Input: $\mathbb{R}^{dim}$ FC 32 + ReLU FC 10 + Sigmoid
	trajectory	Input $\mathbb{R}^{200}$ FC 512 + ReLU FC 512 + ReLU FC dim	Input $\mathbb{R}^{dim}$ FC 512 + ReLU FC 512 + ReLU FC 200 + Sigmoid
	audio	Input $\mathbb{R}^{1\times 32\times 128}$ Conv 32, kernel 1×128, stride 1, pad 0 + ReLU Conv 64, kernel 4×1, stride 2, pad 1 + ReLU Conv 128, kernel 4×1, stride 2, pad 1 + ReLU FC dim + ReLU FC dim	Input $\mathbb{R}^{dim}$ FC 1024 + ReLU ConvTrans 64, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 32, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 1, kernel 1×128, stride 2, pad 1 + Sigmoid
CUB	img	Input: $\mathbb{R}^{3\times 64\times 64}$ Conv 32, kernel 3×3, stride 2, pad 1 + ReLU Conv 64, kernel 3×3, stride 2, pad 1 + ReLU Conv 128, kernel 3×3, stride 2, pad 1 + ReLU Conv 256, kernel 3×3, stride 2, pad 1 + ReLU FC dim + ReLU FC dim	Input: $\mathbb{R}^{dim}$ FC 4096 + ReLU ConvTrans 128, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 64, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 32, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 3, kernel 3×3, stride 2, pad 1 + Sigmoid
	text	Input: $\mathbb{R}^{18} + \mathbb{R}^{18}$ (tokens, padding mask) Embedding 512 PosEncoder 512 Transformer 512, heads 4, dropout 0.5 + ReLU Transformer 512, heads 4, dropout 0.5 + ReLU Transformer 512, heads 4, dropout 0.5 + ReLU Transformer 512, heads 4, dropout 0.5 + ReLU FC dim	Input: $\mathbb{R}^{dim}$ FC 512 + ReLU FC $18 \times$ vocab. size + Sigmoid
	label	Input: $\mathbb{R}^{200}$ FC 32 + ReLU FC dim	Input: $\mathbb{R}^{dim}$ FC 32 + ReLU FC 200 + Sigmoid
CelebA	image	Input: $\mathbb{R}^{3\times 64\times 64}$ Conv 32, kernel 3×3, stride 2, pad 1 + ReLU Conv 64, kernel 3×3, stride 2, pad 1 + ReLU Conv 128, kernel 3×3, stride 2, pad 1 + ReLU Conv 256, kernel 3×3, stride 2, pad 1 + ReLU FC dim + ReLU FC dim	Input: $\mathbb{R}^{dim}$ FC 4096 + ReLU ConvTrans 128, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 64, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 32, kernel 3×3, stride 2, pad 1 + ReLU ConvTrans 3, kernel 3×3, stride 2, pad 1 + Sigmoid
	attributes	Input: $\mathbb{R}^{2}$ FC 32 + ReLU FC dim	Input: $\mathbb{R}^{dim}$ FC 32 + ReLU FC 2 + Sigmoid
All datasets	**b**	Input: $\{0,1\}^{\|M\|}$ FC 512 + ReLU FC dim	–

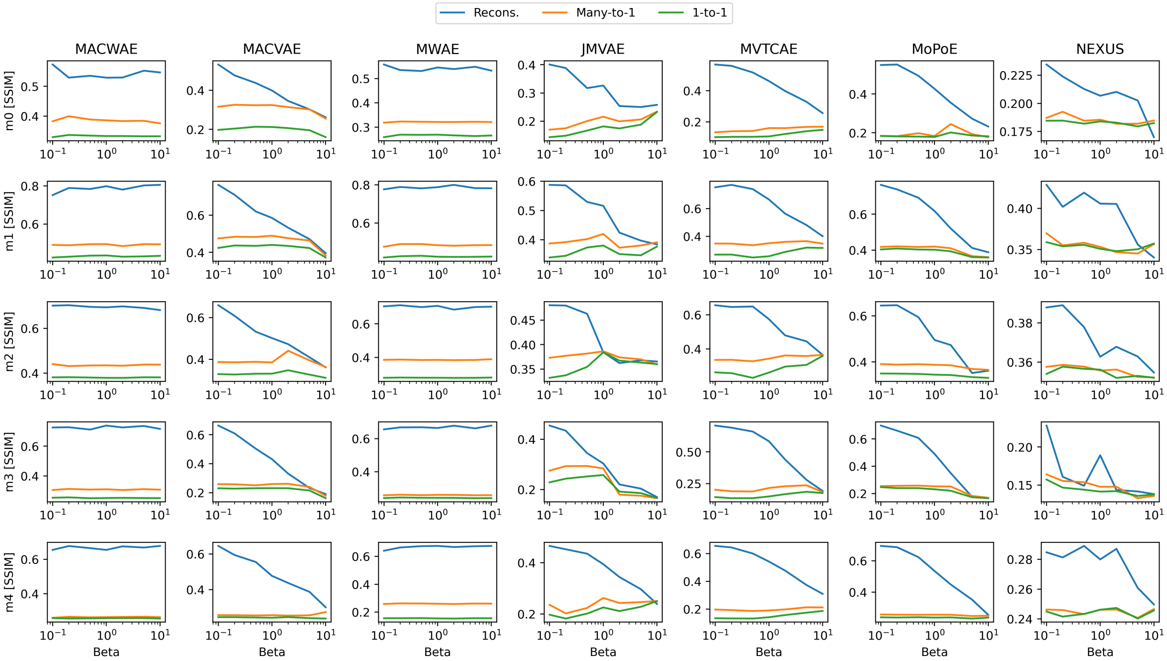
Supplementary Table 4: The table summarizes the models performance in three tasks across the four datasets, averaged across the experimental replicates (associated standard deviations are listed in the Technical Supplement). The column "Dir." indicates whether higher (↑), or lower (↓) values mean better performance. The best outcomes are highlighted in bold. Note that, due to computational reasons, MoPoE is not applicable to CelebA dataset under the given setup, resulting in the missing values.

Dataset	Modality	Metric	Dir.	**MAC-WAE**	MAC-VAE	M-WAE	MVTCAE	JMVAE	MoPoE	Nexus
Reconstruction										
PolyMNIST	m0	SSIM	↑	0.535	0.531	0.533	**0.567**	0.401	0.549	0.224
	m1	SSIM	↑	**0.784**	0.763	0.782	0.753	0.588	0.763	0.402
	m2	SSIM	↑	0.696	0.659	**0.702**	0.657	0.48	0.645	0.389
	m3	SSIM	↑	**0.712**	0.663	0.679	0.708	0.454	0.698	0.161
	m4	SSIM	↑	0.665	0.646	0.675	0.655	0.466	**0.695**	0.281
MHD	image	SSIM	↑	0.966	0.922	**0.967**	0.939	0.479	0.933	0.487
	trajectory	RMSE	↓	**0.016**	0.027	0.021	0.026	0.068	0.025	0.087
	label	ACC	↑	**99.993**	99.667	99.98	99.56	47.84	99.82	65.6
	audio	RMSE	↓	**0.043**	0.052	0.048	0.048	0.068	0.049	0.093
CUB	img	SSIM	↑	0.492	0.467	0.488	0.544	0.458	**0.563**	0.418
	text	BLEU	↑	37.686	**73.773**	38.103	68.044	23.095	46.206	9.949
	label	ACC	↑	2.314	4.867	3.038	**5.937**	0.932	3.624	0.0
CelebA	image	SSIM	↑	**0.757**	0.667	0.725	0.674	0.509		0.431
	attributes	ACC	↑	**99.606**	86.594	94.397	86.773	81.658		81.516
Many-to-1 translation										
PolyMNIST	m0	SSIM	↑	**0.389**	0.316	0.321	0.133	0.17	0.184	0.193
	m1	SSIM	↑	**0.494**	0.475	0.485	0.349	0.387	0.418	0.355
	m2	SSIM	↑	**0.434**	0.387	0.39	0.336	0.373	0.39	0.359
	m3	SSIM	↑	**0.308**	0.259	0.258	0.201	0.275	0.255	0.155
	m4	SSIM	↑	**0.262**	0.257	0.261	0.197	0.236	0.257	0.246
MHD	image	SSIM	↑	0.883	0.849	**0.887**	0.587	0.326	0.808	0.281
	trajectory	RMSE	↓	**0.022**	0.029	0.023	0.028	0.074	0.029	0.088
	label	ACC	↑	**99.98**	99.64	99.96	99.56	35.66	99.7	53.6
	audio	RMSE	↓	**0.098**	0.108	0.108	0.125	0.116	0.107	0.117
CUB	img	SSIM	↑	0.411	0.408	0.411	0.331	0.4	**0.413**	0.409
	text	BLEU	↑	5.058	6.593	4.745	3.343	6.01	**6.66**	6.159
	label	ACC	↑	1.829	4.119	2.692	**4.591**	0.828	2.819	0.0
CelebA	image	SSIM	↑	**0.466**	0.395	0.436	0.394	0.391		0.375
	attributes	ACC	↑	**88.876**	86.232	88.751	86.202	80.592		81.2
1-to-1 translation										
PolyMNIST	m0	SSIM	↑	**0.335**	0.197	0.265	0.102	0.144	0.184	0.185
	m1	SSIM	↑	**0.434**	0.424	0.425	0.269	0.339	0.404	0.354
	m2	SSIM	↑	**0.38**	0.329	0.281	0.263	0.332	0.349	0.358
	m3	SSIM	↑	**0.25**	0.23	0.241	0.143	0.229	0.247	0.147
	m4	SSIM	↑	**0.256**	0.245	0.156	0.135	0.197	0.239	0.242
MHD	image	SSIM	↑	0.586	**0.605**	0.591	0.451	0.39	0.581	0.286
	trajectory	RMSE	↓	**0.038**	0.048	0.049	0.065	0.076	0.05	0.09
	label	ACC	↑	90.391	**96.098**	89.787	62.6	27.533	92.8	27.6
	audio	RMSE	↓	**0.107**	0.108	0.112	0.151	0.133	0.108	0.117
CUB	img	SSIM	↑	0.41	0.409	**0.411**	0.276	0.378	**0.411**	0.408
	text	BLEU	↑	4.945	**6.707**	5.362	2.934	6.437	6.503	5.937
	label	ACC	↑	1.41	**2.318**	1.985	1.657	0.656	1.415	0.0
CelebA	image	SSIM	↑	**0.403**	0.37	0.395	0.173	0.355		0.367
	attributes	ACC	↑	**80.747**	79.028	80.729	78.29	79.732		80.191

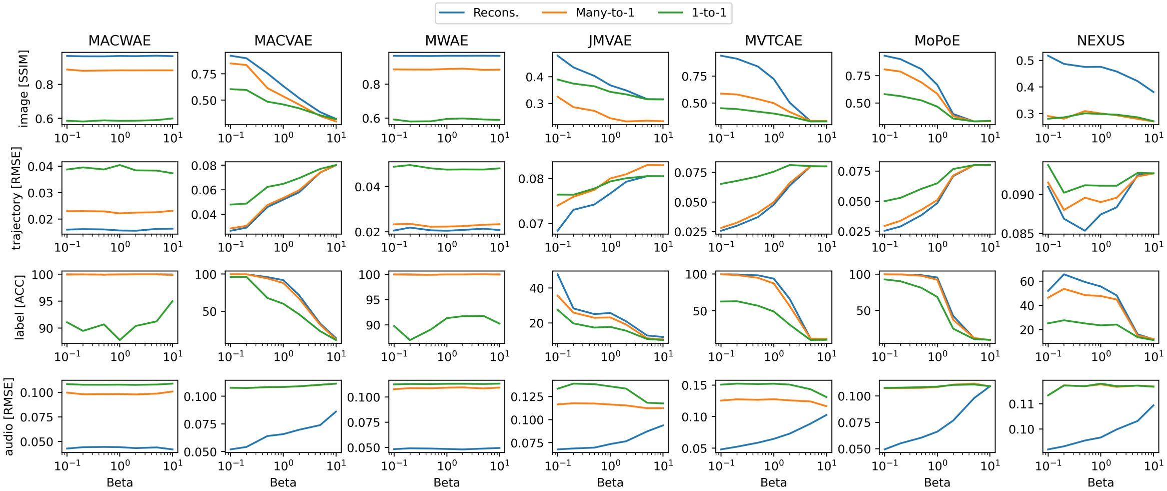
Supplementary Table 5: Generative performance of the models, measured by the FID score (lower is better), calculated from the images in the generated samples. MAC-WAE does not surpass all baselines, but scores above the average, suggesting that its performance in reconstruction and modality translation does not come at the expense of generative capabilities.

	PolyMNIST	MHD	CUB	CelebA
MAC-WAE	64.7	49.0	151.6	74.7
MAC-VAE	71.5	37.6	212.0	79.3
M-WAE	83.9	44.1	**148.8**	78.6
MVTCAE	**49.7**	**25.5**	169.5	**46.8**
JMVAE	89.8	105.2	195.4	101.5
MoPoE	88.9	52.7	175.7	
NEXUS	114.7	109.5	240.9	110.8

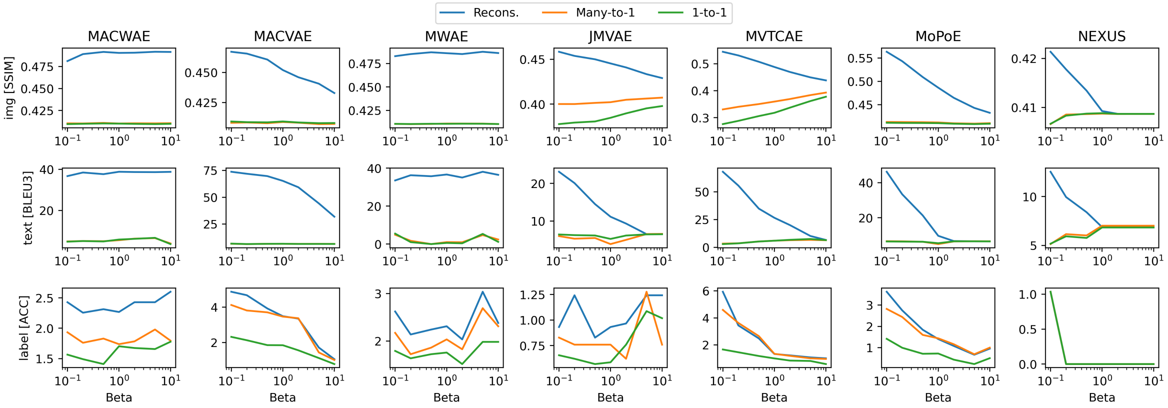
7 Supplementary Figures



Supplementary Figure 2: Performance of different models across three tasks as a function of the hyperparameter β. Rows represent individual data modalities in the `PolyMNIST` dataset, while columns correspond to specific models.



Supplementary Figure 3: Performance of different models across three tasks as a function of the hyperparameter β. Rows represent individual data modalities in the `MHD` dataset, while columns correspond to specific models.



Supplementary Figure 4: Performance of different models across three tasks as a function of the hyperparameter β. Rows represent individual data modalities in the `CUB-Captions` dataset, while columns correspond to specific models.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[2] Mohamed Belghazi, Maxime Oquab, and David Lopez-Paz. Learning about an exponential amount of conditional distributions. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Allal Guessab and Gerhard Schmeisser. Necessary and sufficient conditions for the validity of jensen's inequality. *Archiv der Mathematik*, 100:561–570, 2013.

[4] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

[5] HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. Multi-view representation learning via total correlation objective. *Advances in Neural Information Processing Systems*, 34:12194–12207, 2021.

[6] Oleg Ivanov, Michael Figurnov, and Dmitry Vetrov. Variational autoencoder with arbitrary conditioning. *arXiv preprint arXiv:1806.02382*, 2018.

[7] Soheil Kolouri, Phillip E Pope, Charles E Martin, and Gustavo K Rohde. Sliced wasserstein autoencoders. In *International Conference on Learning Representations*, 2018.

[8] Timo Korthals, Daniel Rudolph, Jürgen Leitner, Marc Hesse, and Ulrich Rückert. Multi-modal generative models for learning epistemic active sensing. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3319–3325. IEEE, 2019.

[9] Mihee Lee and Vladimir Pavlovic. Private-shared disentangled multimodal vae for learning of latent representations. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 1692–1700, 2021.

[10] Yang Li, Shoaib Akbar, and Junier B Oliva. Flow models for arbitrary conditional likelihoods. In *International Conference on Machine Learning*, pages 5831–5841. PMLR, 2020.

[11] Chao Ma, Sebastian Tschiatschek, Konstantina Palla, Jose Miguel Hernandez-Lobato, Sebastian Nowozin, and Cheng Zhang. Eddi: Efficient dynamic discovery of high-value information with partial vae. In *International Conference on Machine Learning*, pages 4234–4243. PMLR, 2019.

[12] Chao Ma, Sebastian Tschiatschek, Richard Turner, José Miguel Hernández-Lobato, and Cheng Zhang. Vaem: a deep generative model for heterogeneous mixed type data. *Advances in Neural Information Processing Systems*, 33:11237–11247, 2020.

[13] Alfredo Nazabal, Pablo M Olmos, Zoubin Ghahramani, and Isabel Valera. Handling incomplete heterogeneous data using vaes. *Pattern Recognition*, 107:107501, 2020.

[14] Emanuele Palumbo, Imant Daunhawer, and Julia E Vogt. Mmvae+: Enhancing the generative quality of multimodal vaes without compromises. In *The Eleventh International Conference on Learning Representations*, 2023.

[15] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[16] Agathe Senellart, Clément Chadebec, and Stéphanie Allassonnière. Improving multimodal joint variational autoencoders through normalizing flows and correlation analysis. *arXiv preprint arXiv:2305.11832*, 2023.

[17] Agathe Senellart, Clement Chadebec, and Stephanie Allassonniere. MultiVae: A Python library for Multimodal Generative Autoencoders, 2023.

[18] Yuge Shi, Brooks Paige, Philip Torr, et al. Variational mixture-of-experts autoencoders for multi-modal deep generative models. *Advances in neural information processing systems*, 32, 2019.

[19] Yuge Shi, Brooks Paige, Philip Torr, and N Siddharth. Relating by contrasting: A data-efficient framework for multimodal generative models. In *International Conference on Learning Representations (ICLR 2022)*, 2022.

[20] Ryan Strauss and Junier B Oliva. Arbitrary conditional distributions with energy. *Advances in Neural Information Processing Systems*, 34:752–763, 2021.

[21] Ryan Strauss and Junier B Oliva. Posterior matching for arbitrary conditioning. *Advances in Neural Information Processing Systems*, 35:18088–18099, 2022.

[22] Thomas Sutter, Imant Daunhawer, and Julia Vogt. Multimodal generative learning utilizing jensen-shannon-divergence. *Advances in neural information processing systems*, 33:6100–6110, 2020.

[23] Thomas M Sutter, Imant Daunhawer, and Julia E Vogt. Generalized multimodal elbo. In *International Conference on Learning Representations (ICLR 2021)*, 2021.

[24] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. Joint multimodal learning with deep generative models. In *International Conference on Learning Representations*, 2017.

[25] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. In *International Conference on Learning Representations*, 2018.

[26] Miguel Vasco, Hang Yin, Francisco S Melo, and Ana Paiva. Leveraging hierarchy in multimodal generative models for effective cross-modality inference. *Neural Networks*, 146:238–255, 2022.

[27] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[28] Jiqing Wu, Zhiwu Huang, Dinesh Acharya, Wen Li, Janine Thoma, Danda Pani Paudel, and Luc Van Gool. Sliced wasserstein generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3713–3722, 2019.

[29] Mike Wu and Noah Goodman. Multimodal generative models for scalable weakly-supervised learning. *Advances in neural information processing systems*, 31, 2018.

[30] Mike Wu and Noah Goodman. Multimodal generative models for compositional representation learning. *arXiv preprint arXiv:1912.05075*, 2019.