DISS. ETH NO. 29385

IMPOSING AND UNCOVERING GROUP STRUCTURE IN WEAKLY-SUPERVISED LEARNING

A dissertation submitted to attain the degree of

DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

THOMAS MARCO SUTTER
MSc Information Technology and Electrical Engineering, ETH Zurich

born on 15 April 1989
citizen of Switzerland and Italy

accepted on the recommendation of

Prof. Dr. Julia E. Vogt (ETH Zürich), examiner
Prof. Dr. Stephan Mandt (UC Irvine), co-examiner
Prof. Dr. Gunnar Rätsch (ETH Zürich), co-examiner

2023

IMPOSING AND UNCOVERING GROUP STRUCTURE IN WEAKLY-SUPERVISED LEARNING

ABSTRACT

Humans naturally integrate various senses to understand our surroundings, enabling us to compensate for partially missing sensory input. On the contrary, machine learning models excel at harnessing extensive datasets but face challenges in handling missing data effectively.

While utilizing multiple data types provides a more comprehensive perspective, it also raises the likelihood of encountering missing values, underscoring the significance of proper missing data management in machine learning techniques.

In this thesis, we advocate for developing machine learning models that emulate the human approach of merging diverse sensory inputs into a unified representation, demonstrating resilience in the face of missing input sources. Generating labels for multiple data types is laborious and often costly, resulting in a scarcity of fully annotated multimodal datasets. On the other hand, multimodal data naturally possesses a form of weak supervision. We understand that these samples describe the same event and assume that certain underlying generative factors are shared among the group members, providing a form of weak guidance.

Our thesis focuses on learning from data characterized by weak supervision, delving into the interrelationships among group members. We start by exploring novel techniques for machine learning models capable of processing multimodal inputs while effectively handling missing data. Our emphasis is on variational autoencoders (VAE) for learning from weakly supervised data. We introduce a generalized formulation of probabilistic aggregation functions, designed to overcome the limitations of previous methods, and we show how this generalized formulation correlates with performance enhancements.

At a higher level, we investigate the impact of implicit assumptions regarding group structure on a model's learning behavior and efficacy. We find that the assumption of a single shared latent space is overly restrictive for generating coherent and high-quality samples. To overcome this limitation, we introduce modality-specific latent subspaces within multimodal VAEs, reflecting a more flexible modeling approach.

While we observe that greater flexibility in modeling assumptions, or assumptions aligned with the actual data generation process, leads to improved performance, we still depend on prior knowledge concerning the relationship of a group of multimodal or weakly supervised samples. As the number of group members grows, their underlying relationships become potentially more intricate, increasing the risk of overly rigid assumptions.

Therefore, in the final section, we shift our focus to minimizing the assumptions required when learning from weakly supervised data and simultaneously deducing the group structure during the learning process. In this context, we introduce a novel differentiable formulation of a random partition model, which follows a two-stage process. In the first step, we estimate the number of elements using a newly proposed differentiable formulation of the hypergeometric distribution. In the second step, we allocate the appropriate number of elements to each subset. We can demonstrate that our differentiable random partition model can learn shared and independent generative factors in the weakly supervised setting.

We aspire that this thesis and its contributions will enhance future applications in multimodal machine learning and reduce the assumptions necessary for learning from weakly supervised data in general.

ZUSAMMENFASSUNG

Um unsere Umwelt zu verstehen, nutzen Menschen auf natürliche und effiziente Weise verschiedene Sinneswahrnehmungen. Diese Fähigkeit erlaubt es uns, (teilweise) auf einen Teil unserer Sinne zu verzichten, ohne handlungsunfähig zu werden. Im Gegensatz dazu zeichnen sich maschinelle Lernmodelle durch ihre Fähigkeit aus, große Mengen an Daten zu verarbeiten und zu verstehen, haben jedoch Schwierigkeiten im Umgang mit fehlenden Daten.

Gleichermassen wie die unterschiedlichen menschlichen Sinne ein umfassenderes Verständnis der Umgebung ermöglichen, versprechen verschiedene Arten von Daten oder Modalitäten, Methoden des maschinellen Lernens zu verbessern. Allerdings steigt damit auch die Wahrscheinlichkeit, dass einige Datenpunkte fehlen, was die Notwendigkeit für Methoden des maschinellen Lernens verstärkt, mit fehlenden Daten umgehen können.

Die zentrale Fragestellung dieser Dissertation dreht sich um die Entwicklung von Methoden des maschinellen Lernens, die dem menschlichen Vorbild folgen, indem sie verschiedene Sensoren gemeinsam verarbeiten, und gleichzeitig robust gegenüber fehlenden Daten sind – ohne dabei die Hilfe von annotierten Datensätzen in Anspruch nehmen zu müssen. Dies ist von großer Bedeutung, da das Annotieren verschiedener Datentypen oft aufwändig und kostspielig ist, und vollständig annotierte Datensätze, die mehrere Modalitäten abdecken, selten sind. Verschiedene Datentypen, die gleichzeitig erfasst werden, liefern jedoch zusätzliche Informationen, die keine explizite Annotation erfordern.

Die Dissertation widmet sich Datensätzen, die eine solche schwache Form der Annotation aufweisen, wobei angenommen wird, dass eine Gruppe von Modalitäten dasselbe Phänomen beschreibt und einige zugrundeliegende generative Faktoren zwischen den Gruppenmitgliedern geteilt werden.

Im ersten Teil werden neue Ansätze für maschinelles Lernen untersucht, die die Verarbeitung verschiedener Datentypen ermöglichen und gleichzeitig mit fehlenden Werten umgehen können. Ein Schwerpunkt liegt auf der Verwendung von Variational Autoencodern (VAEs) zur Modellierung der schwach annotierten Daten und der Einführung einer generalisierten Formulierung einer probabilistischen Aggregationsmethode, die die Einschränkungen ähnlicher Methoden überwindet.

Eine weitere Ebene der Forschung betrifft die Auswirkungen von Annahmen über die Gruppenstruktur auf die Lernprozesse und Ergebnisse der Modelle. Wir stellten fest, dass die Annahme eines einzigen gemeinsamen latenten Raums zu restriktiv ist und Schwierigkeiten bei der Generierung kohärenter und qualitativ hochwertiger Datenbeispiele mit sich bringt. Daher werden VAEs vorgestellt, die sowohl einen gemeinsamen als auch modalitätsspezifische latente Räume nutzen.

Obwohl flexiblere Annahmen und eine genauere Modellierung des Datenentstehungsprozesses zu besseren Ergebnissen führen, bleibt die Abhängigkeit von a priori Wissen über die Gruppenstruktur bestehen, insbesondere bei komplexeren Strukturen mit einer größeren Anzahl von Gruppenmitgliedern.

Im letzten Abschnitt der Dissertation liegt der Fokus darauf, die Annahmen über die Datenstruktur zu minimieren, um die Gruppenstruktur während des Lernens zu inferieren. Hier wird eine neue Formulierung für ein differenzierbares zweistufiges zufälliges Partitionsmodell (RPM) präsentiert. Es schätzt zunächst die Anzahl der Elemente in jeder Untergruppe mit einer differenzierbaren Formulierung der hypergeometrischen Verteilung und weist dann die richtige Anzahl von Elementen jeder Untergruppe zu. Es wurde gezeigt, dass dieses differenzierbare RPM die geteilten und unabhängigen generativen Faktoren von schwach annotierten Datensätzen erlernen kann.

Wir hoffen, dass diese Dissertation und ihre Ergebnisse zukünftige Anwendungen im Bereich des multimodalen maschinellen Lernens verbessern können, indem sie die Abhängigkeit von Annahmen reduzieren und die Effektivität bei der Verarbeitung von schwach annotierten Daten steigern.

ACKNOWLEDGEMENTS

I want to thank everyone who supported me in the past few years, and I am grateful to all those who made my PhD an unforgettable and invaluable experience.

My deepest gratitude goes to my supervisor, Prof. Dr. Julia Vogt, who gave me the freedom and time to pursue my own research ideas. I am lucky that you considered my application and allowed me to do the PhD in your lab.

I would also like to thank Prof. Dr. Stephan Mandt and Prof. Dr. Gunnar Rätsch for agreeing to review my dissertation and acting as referees.

I want to thank all current and former members of the medical data science group, notably Imant Daunhawer, Kieran Chin-Cheong, Ričards Marcinkevičs, Laura Manduchi, Dr. Ece Özkan Elsen, Alice Bizeul, Dr. Alexander Marx, Alain Ryser, Emanuele Palumbo, Dr. Heike Leutheuser, Dr. Daphné Chopard, Moritz Vandenhirtz, Andrea Agostini, and Sonia Laguna for being great colleagues and collaborators. I have always enjoyed our discussions over lunch and coffee.

I would also like to thank my medical collaborators, notably Prof. Dr. Balthasar Hug, Dr. Jan Adam Roth, Prof. Dr. Nicole Ritz, Dr. Noemi Rebecca Meier, and Dr. Nora Fritschi.

I would also like to thank my former colleagues from the University of Basel, notably Dr. Maxim Samarin, Dr. Mario Wieser, Dr. Sebastian Keller, Prof. Dr. Sonali Parbhoo, Damian Murezzan, and Fabricio Arend Torres for providing help and feedback during my time at the University of Basel.

I want to thank Rita Klute, Patricia Kilchhofer, and Petra Lüthi for always providing help and never losing patience.

I would especially like to thank my former colleagues from Logitech, Dr. Fabian Nater and Dr. Helmut Grabner, and from MeteoSwiss, Dr. Christian Sigg, who all raised my research interest, supported me in pursuing a PhD and helped with my application.

Most importantly, none of this would have been possible without the love and support of my girlfriend Andrea, my sister Anina, my brother Fabrizio, and my parents, Laura and Toni.

PUBLICATIONS

The work presented in this thesis is based on the following publications, which were a collaborative effort together with my colleagues.

Chapter 3 is based on the following publications:

[Dau+22] Imant Daunhawer, Thomas M. Sutter, Kieran Chin-Cheong, Emanuele Palumbo, and Julia E Vogt. “On the Limitations of Multimodal VAEs”. In: *International Conference on Learning Representations* (2022).

[SDV21] Thomas M. Sutter*, Imant Daunhawer*, and Julia E Vogt. “Generalized Multimodal ELBO”. In: *International Conference on Learning Representations* (2021).

Chapter 4 is based on the following publications:

[Dau+20] Imant Daunhawer, Thomas Marco Sutter, Ricards Marcinkevics, and Julia E Vogt. “Self-supervised Disentanglement of Modality-specific and Shared Factors Improves Multimodal Generative Models”. In: *German Conference on Pattern Recognition* (2020).

[SDV20] Thomas M. Sutter, Imant Daunhawer, and Julia E Vogt. “Multimodal Generative Learning Utilizing Jensen-Shannon Divergence”. In: *Advances in Neural Information Processing Systems* (2020).

Chapter 5 is based on the following works (including a spotlight presentation at ICLR 2023):

[Sut+23a] Thomas M. Sutter, Laura Manduchi, Alain Ryser, and Julia E Vogt. “Learning Group Importance using the Differentiable Hypergeometric Distribution”. In: *International Conference on Learning Representations* (2023).

[Sut+23b] Thomas M. Sutter*, Alain Ryser*, Joram Liebeskind, and Julia E Vogt. “Differentiable Random Partition Models”. In: *Under Submission*. 2023.

[SV21] Thomas M. Sutter and Julia E Vogt. “Multimodal Relational VAE”. In: *Neurips Workshop on Bayesian Deep Learning*. 2021.

Furthermore, the following publications were part of my PhD, but are not part of this thesis:

[CCSV19] Kieran Chin-Cheong, Thomas M. Sutter, and Julia E Vogt. “Generation of Heterogeneous Synthetic Electronic Health Records using GANs”. In: *Workshop on Machine Learning for Health (ML4H) at the 33rd Conference on Neural Information Processing Systems*. 2019.

[CCSV20] Kieran Chin-Cheong, Thomas M. Sutter, and Julia E Vogt. “Generation of Differentially Private Heterogeneous Electronic Health Records”. In: *arxiv.org*. Vol. abs/2006.0. 2020.

[Hu+23] Yurong Hu, Thomas M. Sutter, Ece Ozkan, and Julia E Vogt. “Self-supervised Learning to Predict Ejection Fraction using Motion-mode Images”. In: *ICLR First Workshop on Machine Learning & Global Health*. 2023.

[KSV21] Hendrik J Klug, Thomas M. Sutter, and Julia E Vogt. “Multimodal Generative Learning on the MIMIC-CXR Database”. In: *Medical Imaging with Deep Learning (Abstract)*. 2021.

[Mei+21] Noëmi Rebecca Meier, Thomas M. Sutter, Marc Jacobsen, Tom H M Ottenhoff, Julia E Vogt, and Nicole Ritz. “Machine Learning Algorithms Evaluate Immune Response to Novel Mycobacterium tuberculosis Antigens for Diagnosis of Tuberculosis”. In: *Frontiers in Cellular and Infection Microbiology* 10 (2021), 821.

[Oez+23] Ece Oezkan*, Thomas M. Sutter*, Yurong Hu, Sebastian Balzer, and Julia E Vogt. “M(otion)-mode Based Prediction of Ejection Fraction using Echocardiiograms”. In: *Under Submission*. 2023.

[Sut+22] Thomas M. Sutter, Sebastian Balzer, Ece Oezkan, and Julia E Vogt. “M(otion)-mode Based Prediction of Cardiac Function on Echocardiograms”. In: *Medical Imaging meets Neurips*. 2022.

[Sut+21] Thomas M. Sutter, Jan A Roth, Kieran Chin-Cheong, Balthasar L Hug, and Julia E Vogt. “A comparison of general and disease-specific machine learning models for the prediction of unplanned hospital readmissions”. In: *J. Am. Medical Informatics Assoc.* 28.4 (2021), 868.

All code is avalaible under `https://github.com/thomassutter`.
* denotes a shared first authorship.

CONTENTS

1 INTRODUCTION

Machine Learning (ML) is the field of science trying to create machines that learn from experience [Mit07]. Methodological improvements in combination with enormous computational resources have led to various successful applications of ML algorithms, such as mastering the game of Go [Sil+16; Sil+17], accurate protein folding structure prediction [Jum+21], the recent success of text-to-image generative models such as Dall-E [Ram+22; Ram+21] or large language models and ChatGPT [Bro+20; Bub+23; Rad+18].

Using vast amounts of data, ML methods achieve exceptional and even super-human performances in different subfields of ML, such as computer vision [Den+09], natural language processing [Wan+19; Wan+18], and speech recognition [Pan+15].

The success of ML methods is often attributed to having good data representations, either by having a data collection process to that effect or by learning meaningful representations from raw data [BCV13]. The former involves a carefully designed data collection process with *a priori* knowledge about the problem to be tackled. What do we mean by that? We need detailed knowledge about the task we want to solve or the outcome we want to predict before collecting the data. Hence, the data collection process is designed with one specific task in mind. Biomarkers or biological markers, for example, are biological observations that provide information on a clinically relevant endpoint [AF17; Bio+01; ST10]. Therefore, combinations of biomarkers allow the prediction of diseases, but discovering new biomarkers requires research effort in various fields of science, such as biology and medicine [HW20; LHM19].

The latter approach, finding meaningful representations from raw or routinely collected data, aims to compress high-dimensional data into descriptive factors useful for various downstream tasks. Although meaningful is a vague term, for now, a good representation of the data can capture the factors of variations present in a dataset [BCV13]. Given enough labeled data, ML and Deep Learning (DL) have shown the advantage of this approach, as the results described in the beginning prove. DL is the sub-field of ML [GBC16], responsible for its recent advances and most headline-generating news. One of the drawbacks of DL is its hunger for data, specially labeled data, with data set sizes being in the millions, e. g.

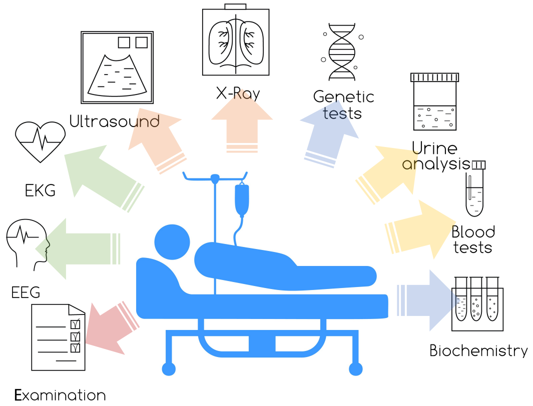


Figure 1.1: Modalities of different types, such as various tests and measurements of a patient in a hospital, surround us. In machine learning, we want to benefit from the additional information a set of modalities provides. We must rely on assumptions about how the modalities are connected because we do not know the underlying group structure of different data types describing a phenomenon. This thesis focuses on learning from grouped data under different assumptions – and how we can reduce the dependency on the assumptions regarding the group structure and instead understand the relation between multiple modalities. Icons © Adobe Stock.

the famous computer vision dataset ImageNet consists of 14′197′122 annotated images [Den+09]. While there are billions of natural images and texts describing everyday life stored on the internet, other areas do not have access to an almost infinite amount of data. Collecting data can be labor-intensive for healthcare and medical data, and annotating data requires highly skilled specialists, making labeling medical data expensive. Therefore, most labeled medical datasets tend to be small or not publicly available, even for (almost) routinely-collected data such as x-ray images [KL21][1]. The size of annotated medical datasets illustrates the need for ML, especially DL methods, to learn from unlabelled data, as the data in most areas cannot be as easily annotated as for natural images and language.

[1] Exceptions include the large x-ray databases from Johnson et al. [Mimic-Cxr, Joh+19] or Irvin et al. [CheXpert, Irv+19]

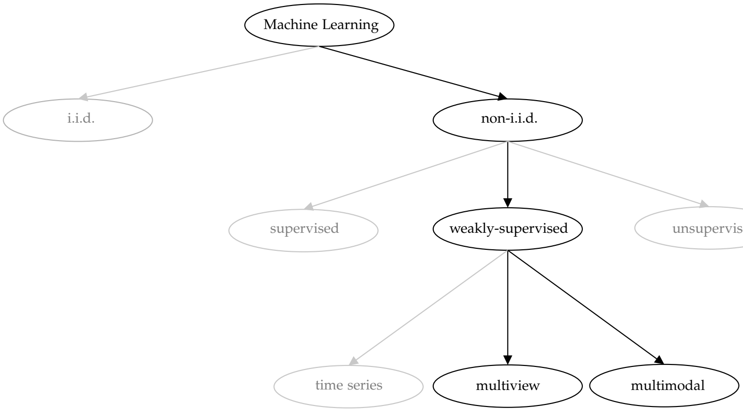


FIGURE 1.2: The topics of this thesis are centered around weakly-supervised learning. In weakly-supervised learning, where the weak supervision arises from the data collection process not being independently and identically distributed, i. e. *i.i.d.*, we are interested in leveraging the underlying group structure of the data. Examples are time-series data, where the temporal ordering between data points defines an unknown structure; multiview data, where multiple images depict the same scene; or multimodal data, where different data types describe the same phenomena. We focus on multimodal and multiview data in this thesis.

1.1 MULTIMODALITY AND WEAK SUPERVISION

While the focus in learning without explicit supervision is on learning from single data types [BCV13; GH10; KW14; OLV18], such as natural [Che+20] and medical images [Azi+21; Cha+20], text [Rad+18; Yan+17], and speech [Bae+20; VDO+16], many situations are inherently described using multimodal data. For example, assessing a patient's health in a hospital needs multiple data types. As depicted in Figure 1.1, the data collection process during a hospital visit involves a set of different tests and examinations to receive information on the patient's health status [Aco+22; Hua+20]. Hence, we need ML methods that leverage the additional information provided by the multitude of tests and measurements – similar to clinicians, who rely on all these tests when deciding on a diagnosis or procedure. Learning from multiple data types *without* the help of explicit supervision through labels is essential to advance ML in fields like healthcare and medicine.

Empirical risk minimization [ERM, Vap91] is the leading approach for training ML models. It assumes that all samples of a dataset are *independently and identically distributed* (i.i.d.) such that the loss function decomposes into a sum of individual terms for every sample. It is questionable whether the i.i.d. assumption is as reasonable for multimodal data, such as the tests and measurements collected per patient, as for unimodal datasets. In our patient example, we know a set of tests and measurements describes a single patient, and we can naturally group such a dataset by patients. Hence, we have additional knowledge about the structure of the data, which is based on the data collection process and, therefore, should be leveraged. The additional structure in the data provides a *weak* supervision signal, which arises in environments beyond the multimodal setting where multiple data samples form a group. Time-series data such as video [Dwi+19], electrocardiograms- [ECG, Rib+20], and electroencephalograms [EEG, CHCV19] belong to the class of weakly-supervised data, as well as multiview data where multiple cameras record the same scene from different angles [Luo+15]. Because multiple samples form a group in the weakly-supervised setting, the i.i.d. assumption is simplistic. Treating the different views in a multiview setting as independent does not capture the underlying data collection process and is neither done in practice [HZ03; XTX13].

Figure 1.2 illustrates how we embed data collected under weak supervision into the general ML world. We see multiple modalities in the multimodal, multiple images in the multiview, and multiple events in the time series setting as a group of samples. The connection between group members is different in every sub-category. It is the similarity on the time axis for time series data, the same scene recorded by multiple cameras for multiview data, and a single phenomenon described by different data types in the multimodal setting. So far, different weakly-supervised approaches require heuristics and specialized architectures trying to leverage the additional structure in the data instead of treating the inference of group structure as part of the learning objective.

1.2 DEEP GENERATIVE MODELING OF WEAKLY-SUPERVISED DATA

While the human brain is not yet fully understood [DZR12], we intuitively describe an image using high-level concepts. For example, we characterize a dog image using attributes like the dog’s size, fur color, and pattern, or the shape of its head. Given a dataset of dog images, a low-dimensional representation consists of all the necessary information to generate sample images of dogs.

We have already briefly touched on the importance of low-dimensional representations in ML. In ML, especially representation learning, we assume that some underlying and unknown probability distribution, which we call the data-generating process [GBC16], generates our dataset. We typically assume that a two-step process generates every data point. We first sample a low-dimensional latent variable $\boldsymbol{z}$, i. e. $\boldsymbol{z} \sim p(\boldsymbol{z})$, which ideally corresponds to high-level concepts [BCV13; Loc+19]. In a second step, we generate the data point $\boldsymbol{x}$ conditioned on the low-dimensional latent variable $\boldsymbol{z}$, i. e. $\boldsymbol{x} \sim p(\boldsymbol{x} \mid \boldsymbol{z})$. In our dog example, the sampled latent variable would ideally reflect high-level attributes (e. g., fur color, size). The data point would then correspond to an image reflecting these attributes. Hence, the goal of representation learning, extracting such factors directly from the data, stems from the assumption that a lower-dimensional set of explanatory factors can explain high-dimensional data [BCV13; Loc+19]. However, neither the generative factors nor their number are generally known for unimodal datasets. Nevertheless, many areas of science use latent variables, such as medicine, political science, or psychology [BMVH03; RHS08; RKF19; Thu27].

Following the concept of a data-generating process based on a set of explanatory factors, we define weakly-supervised data as groups of samples where some of the explanatory or generative factors are shared [see Loc+20]. Following this assumption, the set of shared factors determines this group structure. For example, group members sharing the same factors or only having pairwise shared factors both define a group of samples but with different underlying assumptions on the group structure. However, we do not know this underlying group structure.

In this thesis, we describe and investigate how different assumptions on group structure influence the learning process and performance of deep generative models of weakly supervised data. Please note that we only use the term group in the ordinary sense of a group of samples or modalities. We do not use it in its mathematical sense. Hence, we do not draw any connection or make any implications related to the mathematical theory of groups.

Intuitively, we would want to learn a representation that directly inverts the data-generating process, i. e., $\boldsymbol{z} \sim p(\boldsymbol{z} \mid \boldsymbol{x})$ [BLC13]. Unfortunately, the posterior distribution $p(\boldsymbol{z} \mid \boldsymbol{x})$ is often expensive or even intractable to compute [BKM17], which we discuss in more detail in Section 2.1. Based on the principles of variational inference [VI, JJ13; Jor+99], variational autoencoders [VAE, KW14; RMW14] offer an elegant solution to approximate the intractable $p(\boldsymbol{z} \mid \boldsymbol{x})$ by choosing a tractable variational posterior distribution $q(\boldsymbol{z} \mid \boldsymbol{x})$. VAEs are a probabilistic autoencoder architecture, where the encoder models the variational posterior distribution mapping a data point $\boldsymbol{x}$ to a latent representation $\boldsymbol{z} \sim q(\boldsymbol{z} \mid \boldsymbol{x})$ and the decoder the conditional data generating distribution mapping a latent variable $\boldsymbol{z}$ to a data point $\boldsymbol{x} \sim p(\boldsymbol{x} \mid \boldsymbol{z})$. Although problem-dependent, choosing the variational distribution $q(\boldsymbol{z} \mid \boldsymbol{x})$ is a trade-off between accuracy and efficiency. It has to be flexible enough to capture the variation in the data and efficient enough to evaluate and sample from. The VAE objective approximates the true posterior distribution. It maximizes the evidence lower bound (ELBO) of the data distribution $p(\boldsymbol{x})$, which we discuss in more detail in Section 2.2.

Unlike pure representation learning methods [e. g. Che+20] that want to learn the latent factors given the data, generative latent variable models [LVM, BN06], like the VAE, additionally learn to approximate the data distribution $p(\boldsymbol{x})$. In our dog image example, only some people who can describe a dog's (latent) attributes in an image can also draw a decent picture of a dog. Hence, we want to learn a more difficult task than just inferring representations. However, generative models offer additional beneficial properties over pure representation learning approaches [Tom22]. They hold promise for generating synthetic samples from the data distribution, increased interpretability [DCS18], and uncertainty estimation [CJ18; GHL17]. Additionally, generative models of weakly-supervised data can estimate missing group samples based on the available ones.

Text-to-image generative models made headlines recently (early 2020s). Methods like Dall-E [Ram+22; Ram+21], Imagen [Sah+22], or Stable Diffusion [Rom+22] achieve impressive performance in generating images based on text input. These models implicitly learn the conditional distribution of images given text to transform any text input into a visually appealing image. While this is a practical approach for a single conditional path, e. g., from a text to an image, we cannot use the learned distribution when transforming images to texts, resulting in unfeasible constraints for more than two modalities. There are 2^M different subsets of modalities in a multimodal dataset of M modalities. Hence, to generate all modalities conditioned on any subset of modalities, we need to learn 2^M different conditional models. Given current conditional multimodal models' size and training time [Ram+22; Rom+22; Sah+22], this is computationally not feasible. In contrast, directly mapping between modalities is a subtask of learning the joint data distribution of images and text, multimodal or weakly-supervised datasets. Hence, a single model that learns the joint data distribution optimizes for all the conditional distributions.

In the first part of this thesis (Chapters 3 and 4), we investigate how to learn from multimodal datasets efficiently. Following the data-generating process for weakly supervised data outlined before, the group structure of a multimodal dataset is unknown. In other words, we do not know *a priori* what connects a group of multimodal samples. Hence, multimodal methods rely on assumptions regarding the group structure. We evaluate the performance of multimodal methods and investigate how the performance of a method is related to the assumptions on the underlying group structure. We are interested in VAE-based methods that infer meaningful representations and generate missing modalities. We extend VAEs to the multimodal setting because they have proven successful in these tasks for unimodal datasets [VK20; VDOV017]. Additionally, we want to learn a joint multimodal distribution, unlike the current state-of-the-art conditional generation methods. We are interested in scalable approaches where we define a method as *scalable* if it only requires M encoders, one for every modality $m \in M$. In contrast, a straightforward implementation of a multimodal VAE would require an exponential number of encoders, one for every subset of modalities.

We first explore multimodal VAEs that assume a single joint latent space for all modalities (Chapter 3). This class of VAEs optimizes a multimodal ELBO of the joint multimodal distribution $p(\boldsymbol{X})$ where $\boldsymbol{X} = \{\boldsymbol{x}_1, \dots, \boldsymbol{x}_M\}$ is a set of M modalities $\boldsymbol{x}_m$ [Shi+19; SDV21; WG18]. Equal to the unimodal VAE, the variational posterior $q(\boldsymbol{z} \mid \boldsymbol{X})$ of a multimodal VAE models an encoder that maps $\boldsymbol{X}$ to a joint latent distribution $\boldsymbol{z} \sim q(\boldsymbol{z} \mid \boldsymbol{X})$, and the conditional generative distribution $\boldsymbol{X} \sim p(\boldsymbol{X} \mid \boldsymbol{z})$ a decoder, which generates multimodal samples $\boldsymbol{X}$ based on some latent variable $\boldsymbol{z}$. We discuss the subtleties of the multimodal ELBO in more detail in Section 2.3.3.

Scalable multimodal VAEs rely on the *late fusion* principle [BAM18; SM22]. Every modality $\boldsymbol{x}_m$ is mapped to the latent factors $\boldsymbol{z}$ using $q(\boldsymbol{z} \mid \boldsymbol{x}_m)$, its unimodal encoder. The joint latent distribution follows from a probabilistic aggregation function $\operatorname{agg}(\cdot)$

$$q(\boldsymbol{z} \mid \boldsymbol{X}) = \operatorname{agg}(q(\boldsymbol{z} \mid \boldsymbol{x}_1), \dots, q(\boldsymbol{z} \mid \boldsymbol{x}_M)) \tag{1.1}$$

The aggregation function $\operatorname{agg}(\cdot)$ needs to handle a varying number of inputs, i.e., be robust to missing values and invariant to its inputs' permutations. Additionally, we need to be able to sample from the resulting joint distribution $q(\boldsymbol{z} \mid \boldsymbol{X})$ and evaluate its likelihood.

In Chapter 3, we introduce a generalized aggregation function, which efficiently combines the information from all subsets of modalities. We show the improved performance of our generalized formulation compared to previous works and outline the limitations of multimodal VAEs following their restrictive assumptions.

The constraints for the aggregation function $\text{agg}(\cdot)$ limit the possible choices. Compared to non-linear and parameterized transformation functions, we rely on non-parametric formulations that aggregate independently of input values. Hence, a scalable multimodal VAE with a single joint latent space implicitly assumes that the modalities share all latent factors. We define weakly-supervised data as groups of samples with a subset of shared generative factors, and a learned representation should encode the factors of variation in the data. Hence, it seems over-restrictive to aggregate all latent factors of multimodal or weakly-supervised data. There is no incentive to aggregate the information of latent factors that do not encode the same information. However, if we infer the latent factors $\boldsymbol{z}$ using a variational posterior with a single joint latent space, this is the case. Hence, using a single joint latent space seems over-restrictive [BTN18; Hos18]. Chapter 4 investigates more flexible modeling assumptions of the underlying group structure for multimodal data. Using more flexible assumptions, multimodal VAEs overcome their limitations. However, we explain why the improved performance does not come for free.

1.3 LEARNING GROUP STRUCTURE UNDER WEAK SUPERVISION

Section 1.2 defines weakly-supervised data as a group of samples sharing a set of generative factors. The weak supervision follows from the knowledge that all samples describe the same phenomenon. However, the number of generative factors and the subset of shared factors are unknown [Loc+20], and such detailed knowledge requires human annotation or a controlled data collection process [Loc+20]. Hence, we require additional assumptions on the set of shared factors, e. g., the complete set or a pre-defined subset of generative factors are shared (see Section 1.2). Assumption-based models lead to well-performing methods for specific datasets [BTN18; Hos18]. However, the assumptions must be re-evaluated every time the data collection slightly changes, hindering development, and acquiring expert annotation is time-consuming and expensive (Section 1.1). Hence, we must reduce the assumptions required for learning from weakly-supervised data. In

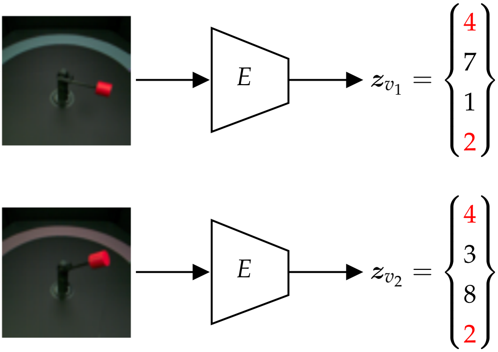

E
$z_{v_1} = \begin{Bmatrix} 4 \\ 7 \\ 1 \\ 2 \end{Bmatrix}$
E
$z_{v_2} = \begin{Bmatrix} 4 \\ 3 \\ 8 \\ 2 \end{Bmatrix}$

FIGURE 1.3: In weakly-supervised learning, we assume that a set of generative factors is shared between group members, e. g., between the two images of a robot arm. We are interested in learning from the paired data and inferring the relationship between the views, which is equivalent to learning the shared and independent latent factors. The robot arm images are taken from Locatello et al. [Loc+20]. Originally, they are from the *mpi3d* dataset. See `https://github.com/rr-learning/disentanglement_dataset`.

other words, we need to formulate probabilistic models, which infer the underlying group structure in parallel to learning from the data.

Figure 1.3 shows a weakly-supervised multiview example: Two images of a robot arm with an unknown subset of shared generative factors (highlighted in red). If we can discover shared and independent generative factors, we can infer the relationship of a group of samples. Please note that we are not interested in learning fully disentangled latent representations, which is impossible [Loc+19]. Instead, we want to define probability distributions that allow finding latent factors that depend on different subsets of group members [Dau+23; Gre+20; Loc+20; Shu+19].

Learning the underlying group structure in the weakly supervised setting is equivalent to learning a group's shared and independent latent factors. Hence, inference of the shared and independent factors has to be incorporated into the learning objective instead of being replaced by heuristics and simplified assumptions.

The second part of this thesis (Chapter 5) discusses how to learn the relationship between group members, which we can reformulate as finding independent and shared generative factors. Hence (see Figure 1.3), we learn which and how many generative factors are shared and which and how many are independent *without* a priori knowledge. In addition, we can extend the approach to having more than just two subsets of factors, which would reflect a different group structure. Hence, we need to formulate the problem of learning the group structure as finding a probability distribution that assigns every generative factor to precisely one of the K groups.

Partitioning a set of elements into subsets is a classical mathematical problem that attracted much interest over the last few decades [MS16]. A partition model assigns every element of a set to precisely one of K subsets, and a random partition model [RPM, Har90] defines a probability distribution over the space of partitions. While there are many well-studied combinatorial partitioning problems [GKP89; Rot64], most existing RPMs either lack a reparameterization scheme or are computationally intractable for large datasets, prohibiting their use in ML pipelines [Mac67; Pit96; Pla75].

Reparameterization schemes [KW14; RMW14; TLG15] enable low-variance gradients of probability distributions such that we can learn the distribution parameters using gradient-based optimization. However, discrete probability distributions are non-differentiable in general. Only the Gumbel-Softmax trick [GST, JGP16; MMT17] has enabled the gradient-based optimization of categorical distributions. In contrast, most discrete distributions cannot be efficiently reparameterized. Theoretically, we can model every discrete distribution as a categorical distribution by defining every distribution state as one category. In practice, however, the number of states quickly becomes computationally infeasible, e. g., when modeling RPMs as categorical distributions. Therefore, naively using the GST to make RPMs differentiable is unattainable. The increase in possible states for partition problems shows the need for specialized formulations to optimize discrete distributions.

Chapter 5 describes a new differentiable formulation for a random partition model based on a two-stage procedure. First, we infer the number of elements per subset. Second, we assign the according number of elements to every subset. We use a newly proposed differentiable formulation of the hypergeometric distribution to infer the number of elements per subset. We then apply the newly introduced RPM to the weakly-supervised setting by modeling the generative factors as a set of elements. Using a differentiable RPM, we can infer the group structure in the weakly-supervised setting and overcome the need for *(over-)*restrictive assumptions.

This thesis proposes novel approaches to learning from multimodal data, a complex case of weakly-supervised data. We show the sensitivity of multimodal VAEs regarding the assumptions on the underlying group structure of a multimodal dataset. To tackle this limitation, we propose a novel formulation of differentiable random partition models that enables the inference of the group structure. Inferring the group structure while learning from data overcomes the need for restrictive assumptions or precise a priori knowledge.

1.4 CONTRIBUTIONS OF THE THESIS

In the previous sections, we outlined the motivation for and challenges in learning from weakly-supervised data, such as multimodal and multiview data, and the need for scalable and generative approaches. We provide the following contributions to improve learning from weakly supervised data based on *a priori* group definitions and to include the *learning of the group structure* in the training objective.

A GENERALIZED PROBABILISTIC AGGREGATION METHOD We propose a new probabilistic aggregation function for scalable multimodal VAEs, which generalizes the aggregation methods of previous work. We show that a multimodal VAE equipped with the new mixture of products of experts posterior approximation outperforms existing works. The results show that the proposed formulation can generate coherent and high-quality samples. In evaluating previous works, we can relate the strengths and weaknesses of every method to its respective particular case regarding our generalized formulation.

LIMITATIONS OF JOINT LATENT SPACE MULTIMODAL VAES We uncover the limitations of multimodal VAEs using a single joint latent space, which results in inferior generative quality compared to unimodal VAEs. We attribute the limitations to a combination of the formulation of the variational posterior distribution and over-restrictive assumptions regarding the underlying group structure.

MULTIMODAL VAES USING SHARED AND MODALITY-SPECIFIC LATENT SPACES We propose a new formulation for multimodal VAEs following a more flexible modeling assumption. Instead of a single joint latent space, we equip multimodal VAEs with shared and modality-specific latent spaces. This simple yet effective change enables scalable multimodal VAEs to encode shared and modality-specific latent factors. We show that the more flexible modeling assumptions increase generative quality for scalable multimodal VAEs without a trade-off concerning the quality of learned latent representations and coherence of generated samples. Hence, we show that assumptions regarding the underlying group structure are critical to learning from multimodal data.

DIFFERENTIABLE HYPERGEOMETRIC DISTRIBUTION We propose a novel differentiable formulation for the multivariate hypergeometric distribution. Our differentiable and reparameterizable formulation enables the integration of the hypergeometric distribution into deep learning pipelines as a stochastic node. We compare the new distribution against a non-differentiable reference distribution using the Kolmogorov-Smirnov test, which shows the correctness of our formulation.

DIFFERENTIABLE RANDOM PARTITION MODEL We propose a two-stage differentiable random partition model. We first estimate the number of elements for every subset, and second, we assign the according number of elements to the respective subsets. We model the number of elements per subset using the proposed differentiable hypergeometric distribution. Assigning a given number of elements to subsets follows a Plackett-Luce ranking model. Given our fully-differentiable random partition model, we highlight its versatility in three popular machine learning applications: clustering, multitask learning, and weakly-supervised learning.

LEARNING THE GROUP STRUCTURE UNDER WEAK SUPERVISION Based on our findings from learning from multimodal data and the importance of assumptions regarding the underlying group structure, we propose to model the relationship between group members with a random partition model. We partition the latent factors of a weakly-supervised dataset into subsets and show that the subsets reflect shared and independent generative factors. Using our differentiable random partition model, we can eliminate over-restrictive assumptions regarding underlying group structure in the weakly-supervised setting and directly learn the relationship between group members from the data.

1.5 OVERVIEW OF THE THESIS

In Chapter 2, we introduce the essential concepts this thesis is based on. Compared to Chapter 1, it is a technical introduction and motivation for the research done during my Ph.D.

In Chapter 3, we discuss multimodal VAEs using a joint latent space. We derive our generalized formulation, the basis of the proposed MoPoE-VAE, and we connect it to MoPoE-VAE's performance improvements in two experiments. Towards the end of this chapter, we uncover the limitations of using a single joint latent space in the multimodal setting and the over-restrictive assumptions underlying this concept.

Chapter 4 presents a more flexible scalable multimodal VAE class that models the latent factors as a combination of shared and modality-specific spaces. On two different datasets, we can show that the changes in the underlying assumptions lead to better generative quality of multimodal VAEs. At the end of the chapter, we demonstrate its sensitivity to the choice of hyperparameters when introducing modality-specific latent subspaces.

Chapter 5 introduces two new differentiable formulations of important discrete distributions: the multivariate noncentral hypergeometric distribution and random partition models. We leverage these new formulations to learn the shared and independent factors in a weakly supervised setting and eliminate poorly motivated heuristics and over-restrictive assumptions. We show the versatility of the proposed differentiable random partition model (DRPM) in three popular ML applications, the already discussed weakly-supervised learning, clustering, and multitask learning.

At the end of this thesis (Chapter 6), we summarize, analyze, and discuss the contributions and limitations of this thesis. Additionally, we provide an outlook on the potential next steps in research on weakly supervised learning.

2 PRELIMINARIES

This chapter provides information on the underlying concepts and methods for this thesis. It should equip the reader with the required knowledge to understand this dissertation and further motivate the thesis topic based on previous work.

2.1 LATENT VARIABLE MODELS

The aim of latent variable models (LVM) is a simplified description of the structure of high-dimensional data or observations by a model [BN06; Eve84]. These latent variables explain the factors of variation in the data and describe the underlying concepts of the high-dimensional and hard-to-interpret observed data [BCV13]. Let $X = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N} \in \mathbb{R}^{N \times p}$ describe a dataset of N *i.i.d* samples. Let $Z = \{\boldsymbol{z}^{(i)}\}_{i=1}^{N} \in \mathbb{R}^{N \times d}$ be the corresponding set of latent variables, and $p(X, Z)$ the joint distribution of X and Z. We assume $p \geq d$ without loss of generality.

The inference problem is to compute the conditional density of the latent variables given the observations, $p(Z \mid X)$. We rewrite the posterior distribution as [BKM17]

$$p(Z \mid X) = \frac{p(Z, X)}{p(X)} \tag{2.1}$$

The marginal density of observed variables $p(X)$ in the denominator, also called *evidence*, is calculated by marginalizing out the latent variables $\boldsymbol{z}$ from the joint density

$$p(X) = \int_{Z} p(Z, X) dZ \tag{2.2}$$

For many models, the integral in Equation (2.2) is unavailable in closed form or requires exponential time to compute [BKM17]. We need the evidence to compute the conditional posterior distribution $p(Z \mid X)$. This is why inference in latent variable models is difficult.

2.2 VARIATIONAL INFERENCE AND VARIATIONAL AUTOENCODERS

Variational Inference [VI, Jor+99; WJ08] is a method for approximating intractable probability distributions. VI is an alternative strategy to Markov Chain Monte Carlo [MCMC, Has70] in approximating posterior densities for Bayesian models. Despite constant progress and landmark developments such as the Metropolis-Hastings algorithm [Has70; Met+53], the Gibbs sampler [GG84], and its many applications to Bayesian statistics [GS90], MCMC sampling is not well suited for large datasets [BKM17].

2.2.1 *Basic Variational Inference*

Unlike MCMC, a sampling-based approach to approximate posterior distributions, the main idea behind VI is optimization. First, we posit a family of approximate distributions $\mathcal{D}$, a set of densities over the latent variables. Then we try to find the member of that family that minimizes the Kullback-Leibler divergence [KL, KL51] to the exact posterior,

$$q^*(Z) = \underset{q(Z)\in\mathcal{D}}{\arg\min}\, D_{\mathrm{KL}}[q(Z) \mid\mid p(Z \mid X)], \tag{2.3}$$

where the KL divergence between two distributions $q(Z)$ and $p(Z \mid X)$ is given as

$$\begin{aligned} D_{\mathrm{KL}}[q(Z) \mid\mid p(Z \mid X)] &= \int_Z q(Z) \log\left(\frac{q(Z)}{p(Z \mid X)}\right) dZ && (2.4)\\ &= \mathbb{E}_{q(Z)}\left[\log\left(\frac{q(Z)}{p(Z \mid X)}\right)\right] && (2.5) \end{aligned}$$

Hence, VI turns the inference problem into an optimization problem, where the reach of the family of distributions $\mathcal{D}$ controls the complexity of this optimization. One of the key ideas behind VI is to choose $\mathcal{D}$ to be flexible enough to capture a density close to the true $p(Z \mid X)$ but simple enough for efficient optimization. We write the variational distribution as $q(Z; \Lambda)$ where $\Lambda \in \mathbb{R}^{N\times p_v}$ are the variational parameters, and p_v is the number of variational parameters per latent variable. Every latent variable $\boldsymbol{z}^{(i)} \in Z$ has its own variational parameters $\boldsymbol{\lambda}^{(i)} \in \mathbb{R}^{p_v}$ such that $\Lambda = [\boldsymbol{\lambda}^{(1)}, \ldots, \boldsymbol{\lambda}^{(N)}]^T$. Please note that, in general, p_v can vary across different data points and their latent variables. For the rest of this work, we restrict ourselves to p_v

being the same for all data points. Following the introduction of variational parameters Λ, we write Equation (2.3) as

$$q^*(Z;\Lambda^*) = \arg\min_{\Lambda} D_{\mathrm{KL}}[q(Z;\Lambda) \mid\mid p(Z \mid X)], \tag{2.6}$$

where Λ^* are the variational parameters minimizing the right hand side of Equation (2.6).

However, the direct computation of Equation (2.6) is generally not possible as it involves the intractable calculation of $p(X)$ (see Equation (2.2)). Because we cannot directly compute Equation (2.6), we optimize an alternative objective, which is equivalent up to an additive constant. Minimizing the KL divergence equals finding the variational distribution, which is as close to the true posterior as the variational family of distributions $\mathcal{D}$ and its parameters Λ allow.

$$\begin{aligned} D_{\mathrm{KL}}(q(Z;\Lambda) \mid\mid p(Z \mid X)) &= \mathbb{E}_{q(Z;\Lambda)}\left[\log q(Z;\Lambda) - \log p(Z \mid X)\right] \\ &= \mathbb{E}_{q(Z;\Lambda)}\left[\log q(Z;\Lambda) - \log \frac{p(Z,X)}{p(X)}\right] \\ &= \mathbb{E}_{q(Z;\Lambda)}\left[\log q(Z;\Lambda) - \log p(Z,X) + \log p(X)\right] \\ &= \mathbb{E}_{q(Z;\Lambda)}\left[\log q(Z;\Lambda) - \log p(Z,X)\right] \\ &\quad + \log p(X) \end{aligned} \tag{2.7}$$

Equation (2.7) reveals the dependence of Equation (2.3) on $\log p(X)$, but also the evidence lower bound (ELBO).

Definition 2.2.1 (Evidence Lower Bound).
The evidence lower bound $\mathcal{L}(\Lambda; X)$ is defined as

$$\mathcal{L}(\Lambda; X) = \mathbb{E}_{q(Z;\Lambda)}\left[p(X,Z) - \log q(Z;\Lambda)\right] \tag{2.8}$$

We want to look closer at two interesting properties of the ELBO. The first property follows directly from its name: it lower-bounds the log-evidence of $\log p(X)$.

Lemma 2.2.1 (The ELBO is a lower bound).
The ELBO defined in Definition 2.2.1 is a lower bound to the marginal log-probability of the data $\log p(X)$.

$$\log p(X) \geq \mathcal{L}(\Lambda; X) \tag{2.9}$$

Proof. Rearranging Equation (2.7), it follows

$$\log p(X) = D_{\mathrm{KL}}[q(Z;\Lambda) \mid\mid p(Z \mid X)] - \mathbb{E}_{q(Z;\Lambda)}\left[\log q(Z;\Lambda) - \log p(Z,X)\right]$$

As $D_{\mathrm{KL}}(\cdot) \geq 0$ [KL51], we have

$$\begin{aligned}\log p(X) &\geq \; - \mathbb{E}_{q(Z;\Lambda)}\left[\log q(Z;\Lambda) - \log p(Z,X)\right] \\ &= \mathbb{E}_{q(Z;\Lambda)}\left[\log p(Z,X) - \log q(Z;\Lambda)\right] \\ &= \mathcal{L}(\Lambda;X)\end{aligned}$$

□

From Lemma 2.2.1 it follows that minimizing the KL divergence in Equation (2.6) is equivalent to maximizing the ELBO. In addition, the ELBO defines an objective that does not depend on $p(X)$.

The second property of the ELBO follows from examining its terms:

$$\begin{aligned}\mathcal{L}(\Lambda;X) &= \mathbb{E}_{q(Z;\Lambda)}\left[\log p(Z,X) - \log q(Z;\Lambda)\right] && (2.10) \\ &= \mathbb{E}_{q(Z;\Lambda)}\left[\log p(X \mid Z) + \log p(Z) - \log q(Z;\Lambda)\right] && (2.11) \\ &= \mathbb{E}_{q(Z;\Lambda)}\left[\log p(X \mid Z) + \log p(Z) - \log q(Z;\Lambda)\right] && (2.12) \\ &= \mathbb{E}_{q(Z;\Lambda)}\left[\log p(X \mid Z)\right] - D_{\mathrm{KL}}[q(Z;\Lambda) \mid\mid p(Z)] && (2.13)\end{aligned}$$

The first term of the ELBO is the expected log-likelihood, which favors configurations of latent variables that explain the data. The second term is the negative KL divergence between the variational and the prior distribution. It encourages the variational approximation to remain similar to the prior distribution. The two terms mirror the balance between likelihood and prior [BKM17].

2.2.2 *Stochastic Variational Inference*

For many large datasets, we assume the data samples to be *i.i.d.* distributed. Hence, the joint probability of the dataset $\log p(X)$ factorizes into the following form

$$\log p(X) = \sum_{i=1}^{N} \log p\left(\boldsymbol{x}^{(i)}\right) = \sum_{\boldsymbol{x} \in X} \log p\left(\boldsymbol{x}\right) \qquad (2.14)$$

We again assume variational parameters $\boldsymbol{\lambda}^{(i)}$ per latent variable $\boldsymbol{z}^{(i)}$. To improve readability and reduce clutter in the notation, we write $\boldsymbol{x}$ instead of

$\boldsymbol{x}^{(i)}$ for a random sample of the dataset X and $\boldsymbol{z}$ instead of $\boldsymbol{z}^{(i)}$, respectively. Rewriting Equation (2.8), it follows

$$\begin{aligned} \mathcal{L}(\Lambda; X) &= \sum_{\boldsymbol{x} \in X} \mathcal{L}(\boldsymbol{\lambda}; \boldsymbol{x}) && (2.15) \\ &= \sum_{\boldsymbol{x} \in X} \mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\lambda})} \left[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}; \boldsymbol{\lambda})\right] && (2.16) \end{aligned}$$

The variational objective in Equation (2.15) is the sum of contributions from N data points.

Stochastic Variational Inference [SVI, Hof+13; HV03; Sat01] applies stochastic optimization to the ELBO objective function. Stochastic optimization efficiently solves problems, which sum individual contributions [Bot10; RM51]. Hence, we can optimize the ELBO in Equation (2.15) using gradient descent. In SVI, we randomly select mini-batches X_S to obtain a stochastic estimate of the ELBO

$$\hat{\mathcal{L}}(\Lambda; X_S) = \frac{N}{|S|} \sum_{\boldsymbol{x} \in X_S} \mathbb{E}_{q(\boldsymbol{z};\boldsymbol{\lambda})} \left[\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z}; \boldsymbol{\lambda})\right] \qquad (2.17)$$

where $|S|$ is the number of samples in the mini-batch X_S. Calculating the gradient in Equation (2.17) returns a noisy estimator. The noisy estimator points toward the steepest ascent of the true ELBO (Equation (2.15)). When $|S| = N$, SVI equals the basic batch VI. However, we only obtain computational savings for $|S| \ll N$. The optimal mini-batch size $\mid S \mid$ is a trade-off between computational speed-up and gradient noise [Zha+18].

2.2.3 *Black-box Variational Inference*

Basic and stochastic VI (Sections 2.2.1 and 2.2.2) are limited to families of distributions for which we can analytically compute the ELBO [Hof+13; Zha+18].

Black-box variational inference methods [BBVI, RGB14] remove the need for analytic solutions of the ELBO. It only requires the generative process of the data to be specified. The idea behind BBVI is to represent the gradient of the ELBO as an expectation and to estimate it using Monte Carlo (MC) sampling [Zha+18]. Sampling from the variational distribution provides an unbiased gradient estimator without having to compute the ELBO analytically [PBJ12; RGB14]. We write the gradient of the ELBO as an expectation of the variational distribution [RGB14; Zha+18]:

$$\nabla_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{\lambda}; \boldsymbol{x}) = \mathbb{E}_q \left[\nabla_{\boldsymbol{\lambda}} \log q(\boldsymbol{z} \mid \boldsymbol{\lambda}) (\log p(\boldsymbol{x}, \boldsymbol{z}) - \log q(\boldsymbol{z} \mid \boldsymbol{\lambda}))\right] \qquad (2.18)$$

$\nabla_{\boldsymbol{\lambda}}\mathcal{L}$ can now be approximated by a stochastic gradient estimator $\nabla_{\boldsymbol{\lambda}}\hat{\mathcal{L}}_{\text{sgd}}$ by sampling from q:

$$\nabla_{\boldsymbol{\lambda}}\hat{\mathcal{L}}_{\text{sgd}}(\boldsymbol{\lambda};\boldsymbol{x}) = \frac{1}{L}\sum_{l=1}^{L}\nabla_{\boldsymbol{\lambda}}\log q(\boldsymbol{z}_l \mid \boldsymbol{\lambda})(\log p(\boldsymbol{x},\boldsymbol{z}_l) - \log q(\boldsymbol{z}_l \mid \boldsymbol{\lambda})) \quad (2.19)$$

where $\boldsymbol{z}_l \sim q(\boldsymbol{z} \mid \boldsymbol{\lambda})$. Hence, BBVI provides a black box gradient estimator for VI using $\nabla_{\boldsymbol{\lambda}}\log q(\boldsymbol{z}_l \mid \boldsymbol{\lambda})$ as a score function. Score function-based gradient estimators, such as REINFORCE [Wil92], suffer from high variance when e. g. directly optimizing Equation (2.19). Only the improvement in stability via reducing the variance through Rao-Blackwellization [Bla47; Kol50; RR45] and control variates (e. g. Lemieux [Lem14]) made the success of BBVI possible [RGB14].

2.2.3.1 *BBVI using reparameterization gradients*

An alternative to the REINFORCE gradients is the reparameterization gradients. Transforming random variables with a deterministic function of a noise distribution enables reparameterization gradients of the ELBO using MC samples [KW14; RMW14; TLG15; Zha+18]. The random variable $\boldsymbol{z}$ is given by

$$\boldsymbol{z} = g(\varepsilon,\boldsymbol{\lambda}) \quad \text{where} \quad \varepsilon \sim r(\varepsilon), \quad (2.20)$$

for some deterministic transformation $g(\cdot)$ and an independent noise distribution $r(\cdot)$ such that $q(\boldsymbol{z};\boldsymbol{\lambda})$ and $g(\varepsilon,\boldsymbol{\lambda})$ share the same parameters $\boldsymbol{\lambda}$.

According to the change of variables in integrals, which follows from the fundamental theorem of calculus [e. g., Spi19], we can compute any expectation over $\boldsymbol{z}$ as an expectation over ε [KW14; RMW14; TLG15; Zha+18]. Using reparameterization gradients, the reformulated ELBO is given as

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}}\hat{\mathcal{L}}_{\text{rep}}(\boldsymbol{\lambda};\boldsymbol{x}) = \frac{1}{L}\sum_{l=1}^{L}\nabla_{\boldsymbol{\lambda}}\left(\log p(\boldsymbol{x},g(\varepsilon_l,\boldsymbol{\lambda})) - \log q(g(\varepsilon_l,\boldsymbol{\lambda}) \mid \boldsymbol{\lambda}\right), \quad (2.21)\\ \varepsilon_l \sim r(\varepsilon_l),\end{aligned}$$

where L is the number of MC samples. Although a theoretical guarantee is missing [Gal16], empirically, reparameterization gradients do not suffer from the high variance issue of their REINFORCE counterpart. A major drawback of reparameterization tricks is that they do not trivially extend to many distributions, especially discrete distributions. We discuss reparameterizations for discrete distributions in more detail in Section 2.4.

2.2.4 *Amortized Variational Inference*

In Sections 2.2.1 to 2.2.3, every latent variable $\boldsymbol{z}$ has its variational parameters $\boldsymbol{\lambda}$. Hence, optimizing $\boldsymbol{\lambda}$ for every data point $\boldsymbol{x}$ is necessary, which is computationally expensive. Amortized inference replaces the local variational parameters $\boldsymbol{\lambda}$ with a single function whose parameters are shared across all data points, $\boldsymbol{z} = f(\boldsymbol{x})$ [Day+95; GG14; Zha+18]. The basic assumption of amortized VI is that a parameterized function of the data $q_{\boldsymbol{\phi}}(\cdot \mid \boldsymbol{x})$ can predict the local variational parameters $\boldsymbol{\lambda}$. The function $q_{\boldsymbol{\phi}}(\cdot \mid \boldsymbol{x})$ can be arbitrarily complex and $\boldsymbol{\phi} = [\phi_1, \ldots]$ is the vector of its parameters. Amortized VI combines the probabilistic formulation of VI with the advantages of DL [Zha+18].

We use two sets of neural networks: a generative model mapping from a latent variable $\boldsymbol{z}$ to a data point $\boldsymbol{x}$ and an inference model for the variational posterior $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ projecting the data point $\boldsymbol{x}$ to the latent variable $\boldsymbol{z}$. Hence, we write the generative model as

$$p_\theta(X \mid Z) = \prod_{\boldsymbol{x} \in X} p_\theta\left(\boldsymbol{x} \mid \boldsymbol{z}\right) \tag{2.22}$$

and the inference model as

$$q_\phi(Z \mid X) = \prod_{\boldsymbol{x} \in X} q_\phi\left(\boldsymbol{z} \mid \boldsymbol{x}\right) \tag{2.23}$$

where θ are the learnable parameters of the generative, and ϕ of the inference model.

Definition 2.2.2 (Amortized ELBO).
The amortized ELBO $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ of a single data point $\boldsymbol{x}$ is defined as

$$\mathcal{L}\left(\theta, \phi; \boldsymbol{x}\right) = \mathbb{E}_{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}\left[\log p_\theta\left(\boldsymbol{x}, \boldsymbol{z}\right) - \log q_\phi\left(\boldsymbol{z} \mid \boldsymbol{x}\right)\right] \tag{2.24}$$

where $q_\phi\left(\boldsymbol{z} \mid \boldsymbol{x}\right)$ is the posterior approximation.

The ELBO of the full dataset $\mathcal{L}(\theta, \phi; X)$ is the sum of the data point specific ELBOs, i. e.

$$\mathcal{L}(\theta, \phi; X) = \sum_{\boldsymbol{x} \in X} \mathcal{L}\left(\theta, \phi; \boldsymbol{x}\right), \tag{2.25}$$

which directly follows from the assumptions in eqs. (2.22) and (2.23). However, it is important to remember that the inference and generative model parameters, ϕ and θ, are shared between all data points $\boldsymbol{x}$. The ELBO

$\mathcal{L}(\theta, \phi; \boldsymbol{x})$ in Equation (2.24) can be rewritten to include the KL divergence [KL51]

$$\begin{aligned}\mathcal{L}(\theta, \phi; \boldsymbol{x}) &= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) - \log \frac{q_\phi(\boldsymbol{z} \mid \boldsymbol{x})}{p_\theta(\boldsymbol{z})}\right] \\ &= \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right] - D_{\mathrm{KL}}\left[q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p_\theta(\boldsymbol{z})\right] \end{aligned} \tag{2.26}$$

The ELBO objective $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ in Equation (2.26) shows that during optimization the conditional log-likelihood $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ is maximized. In parallel, the variational distribution $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ is regularized with the prior distribution $p_\theta(\boldsymbol{z})$.

If we assume a Gaussian distribution $\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}, \Sigma)$ with parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the variational approximation $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$, we can write the inference model as

$$q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}(\boldsymbol{x}), \Sigma(\boldsymbol{x})) \tag{2.27}$$

where $\boldsymbol{\mu}(\cdot)$ and $\Sigma(\cdot)$ are two (non-linear) functions that map the data points $\boldsymbol{x}$ to the parameters describing the variational distribution. The same holds for the generative model. For $p_\theta(\boldsymbol{x} \mid \boldsymbol{z})$ being a Gaussian distribution, we map a reparameterized latent variable $\boldsymbol{z}$ to the parameters of a Gaussian distribution using two non-linear functions $\boldsymbol{\mu}(\cdot)$ and $\Sigma(\cdot)$

$$p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}(\boldsymbol{z}), \Sigma(\boldsymbol{z})) \tag{2.28}$$

2.2.5 *Variational Autoencoders*

The concept of amortized VI resulted in variational autoencoders [VAE, KW14; RMW14; TLG15]. Compared to basic amortized VI, VAEs jointly train the inference and generative model while optimizing the ELBO [Zha+18].

To approximate $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ in Equation (2.26), we draw L MC samples $\varepsilon_l \sim p(\varepsilon)$ from a noise distribution. Using a reparameterization function $g_\phi(\cdot)$ such that $z_l = g_\phi(\varepsilon_l, \boldsymbol{x})$ reflect samples from the variational distribution $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$. We are able to approximate the amortized ELBO $\mathcal{L}(\theta, \phi; \boldsymbol{x})$ in Definition 2.2.2 using L MC samples

$$\begin{aligned}\hat{\mathcal{L}}(\theta, \phi; \boldsymbol{x}) = &\frac{1}{L}\sum_{l=1}^{L} \log p_\theta\left(\boldsymbol{x}, g_\phi(\varepsilon_l, \boldsymbol{x})\right) \\ &- \frac{1}{L}\sum_{l=1}^{L} \log q_\phi\left(g_\phi(\varepsilon_l, \boldsymbol{x}) \mid \boldsymbol{x}\right)\end{aligned} \tag{2.29}$$

For $q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ being a Gaussian distribution as in Equation (2.27), the function $g_\phi(\varepsilon_l, \boldsymbol{x})$ takes the simple form [KW14; RMW14]

$$\boldsymbol{z}_l = \boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\Sigma}(\boldsymbol{x})\,\varepsilon_l \tag{2.30}$$

For the prior $p_\theta(\boldsymbol{z})$ also being a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}_p, \Sigma_p)$, the KL divergence term in Equation (2.26) can be calculated in closed form and does not need to be approximated using MC samples.
The introduction to variational inference and variational autoencoders presented in this section is based on Blei, Kucukelbir, and McAuliffe [BKM17] and Zhang et al. [Zha+18].

2.3 MULTIMODAL AND WEAKLY-SUPERVISED MACHINE LEARNING

According to Merriam-Webster, a modality is one of the main avenues of sensation, such as vision [Mer]. A dataset is characterized as multimodal when it includes different sensory inputs. Primary examples of sensory inputs are visual and vocal signals and natural language. Still, they also include modalities from the medical domain, such as ultrasound [CP11] or magnetic resonance imaging [Mai+18]. Multimodal Machine Learning aims to build models to process and connect information from multiple modalities. Despite its potential, multimodal machine learning brings unique challenges rooted directly in the heterogeneity of the data and the limited knowledge of how different modalities relate to each other [BAM18]. On the other hand, *non-i.i.d.* data collection also appears in different settings. In Figure 1.2, we put time-series and multiview data next to multimodality. Hence, we see multimodal data as a particular case of non-i.i.d. data.

In this thesis, we are interested in learning from non-i.i.d. data without explicit supervision. Fully-supervised approaches [Fan+15; KFF15] perform well but labeling multiple data types, and group members is time-consuming and expensive. Hence, relying on fully-annotated datasets could hinder potential applications. In contrast, weakly-supervised learning offers the potential to leverage all the unlabelled data. Therefore, we want our methods to learn from non-i.i.d. data without needing expert labeling. We use the short notation weakly-supervised data when referring to the general case of non-i.i.d. data in the weakly-supervised learning setting.

In this section, we first discuss and define assumptions for multimodal data because it is the most general instance of non-i.i.d. data. Afterward, we discuss how other forms of non-i.i.d. data can be described and related to multimodal data.

2.3.1 *Desiderata for Multimodal Machine Learning*

We want multimodal methods to fulfill the following desiderata, which help overcome obstacles and leverage additional information (see Section 1.1).

HANDLING OF MISSING DATA If we can guarantee access to all data types at all times, there is no need to request the ability to handle missing data. In many real-world scenarios, though, there is a non-zero probability that some data might be missing. An example is a multi-sensor environment where some sensors might break. A second example is patient data, where an electronic health record (EHR) [Cas+16a] consists of different modalities like x-ray images, tabular data, lab values, and more. However, different patients require different medical tests, so developing methods that work well independently of the available multimodal subset is desirable.

SCALABILITY If we have a multimodal dataset consisting of M modalities, there are $2^M - 1$ subsets of modalities if we exclude the empty set $\emptyset$. Therefore, a straightforward implementation of a multimodal VAE, which handles missing data [SNM16; Ved+18], needs $2^M - 1$ different encoders: one encoder for every subset of modalities. This exponential growth in the number of encoders needed for multimodal VAEs is not feasible, and we need scalable multimodal VAEs.

MEANINGFUL REPRESENTATIONS We want our models to learn meaningful representations, i. e., representations that make it easier to extract useful information from the data for downstream tasks, e. g. regression or classification tasks [BCV13]. A multimodal method has access to multiple data types and, therefore, more information than an unimodal method. Hence, we are interested in methods that can benefit from this additional information regarding the learned representation and performance metrics in general.

GENERATION In the case of missing modalities, we want our method to be able to give an estimate of these modalities. Additionally, generated samples have to be coherent. A generated multimodal sample is coherent if it contains the same shared information as the conditioning subset. For example, we are interested in learning from a multimodal dataset consisting of X-rays from multiple views and the radiology report like in Johnson et al. [Mimic-Cxr, Joh+19]. We want to infer potential diseases and anomalies from the X-ray scans and generate a report based on the images. However, the generated report is only meaningful if coherent with the X-rays concerning diseases, anomalies, and findings. Next to being able to generate coherent samples, we want to generate high-quality samples.

2.3.2 *Definitions and Assumptions*

This work describes multimodal data using the following two Definitions 2.3.1 and 2.3.2.

Definition 2.3.1 (Multimodal dataset).
We define a multimodal dataset $\mathbb{X} = \left\{ \boldsymbol{X}^{(i)} \right\}_{i=1}^{N} = \{X_1, \ldots, X_M\}$ *as a set of* M *random vectors* X_m *where every random vector* $X_m = \left\{ \boldsymbol{x}_m^{(i)} \right\}_{i=1}^{N} \in \mathbb{R}^{N \times p_m}$ *consists of* N *unimodal samples.*

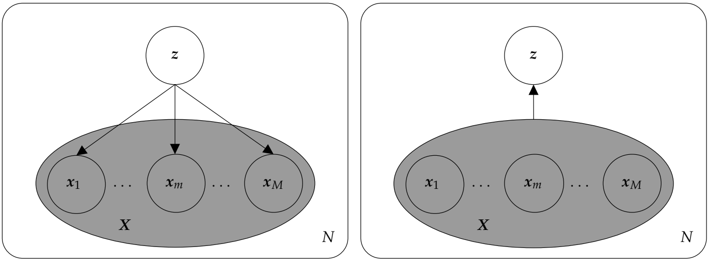
We denote a modality using its modality index $m \in \mathbb{M} = \{1, \ldots, M\}$. $\boldsymbol{X}^{(i)} = \{\boldsymbol{x}_1^{(i)}, \ldots, \boldsymbol{x}_M^{(i)}\}$ describes the random sample i from $\mathbb{X}$. To improve readability and reduce clutter in the notation, we write $\boldsymbol{X}$ and $\boldsymbol{x}_m$ instead of $\boldsymbol{X}^{(i)}$ and $\boldsymbol{x}_m^{(i)}$ if it is clear from the context that we refer to a single multimodal set or data point.

Definition 2.3.2 (Multimodal Subset).
We define a multimodal subset $\mathbb{X}_A = \left\{ \boldsymbol{X}_A^{(i)} \right\}_{i=1}^{N}$*, where* $\boldsymbol{X}_A^{(i)} = \{\boldsymbol{x}_m^{(i)} : m \in A, A \subseteq \mathbb{M}\}$.

Given the impossibility of recovering the disentangled set of attributes in the unsupervised setting [HP99; Loc+19], we use the modality attributes $\boldsymbol{y}$ as an illustrative concept and explanatory model for the group structure.

Definition 2.3.3 (Modality Attributes).
For every modality m*, there exists a set of attributes* $\left\{ \boldsymbol{y}_m^{(i)} \right\}_{i=1}^{N} \in \mathbb{R}^{N \times n_{y_m}}$ *such that* $\boldsymbol{x}_m = f(\boldsymbol{y}_m)$ *where* f *is some non-linear function.* n_{y_m} *is the number of attributes for modality* m*.*

We are interested in recovering the unknown generative factors of the data $\boldsymbol{y}_m$ of a unimodal dataset X_m based on the principles of VI using a VAE. The variational posterior $\boldsymbol{z} \sim q_\phi(\boldsymbol{z} \mid \boldsymbol{x}_m)$ ideally learns to map a data sample to its unknown set of attributes such that $\boldsymbol{z} \approx \boldsymbol{y}_m$ [BCV13].



(a) Generative Model (b) Non-Scalable Inference Model

FIGURE 2.1: Basic graphical models for multimodal VAEs. Figure 2.1a shows the generative model and Figure 2.1b the inference model. Note that the generative model reflects the conditional independence $p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$ of a data sample $\boldsymbol{x}_m$ given the latent vector $\boldsymbol{z}$. The inference model $q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X})$ does not specify how the different modalities $\boldsymbol{x}_m \in \boldsymbol{X}$ are aggregated into a single joint representation $\boldsymbol{z}$.

Definition 2.3.4 (Shared Attributes).
The shared attributes $\{\boldsymbol{y}_A^{(i)}\}_{i=1}^{N} \in \mathbb{R}^{N \times n_{y_A}}$ between a subset of modalities $\mathbb{X}_A$ are defined as the intersection of attributes $\boldsymbol{y}_m \in \mathbb{R}^{n_{y_m}}$ of all modalities $m \in A$, i. e.

$$\boldsymbol{y}_A = \bigcap_{m \in A} \boldsymbol{y}_m \tag{2.31}$$

n_{y_A} is the number of shared attributes of subset $A \subseteq \mathbb{M}$. The full set of attributes $\boldsymbol{y}$ for a multimodal set $\mathbb{X}$ is given as

$$\boldsymbol{y} = \bigcup_{A \subseteq \mathbb{M}} \boldsymbol{y}_A \tag{2.32}$$

From the set definition, it follows that multiple instances of an attribute $\boldsymbol{y}_j \in \boldsymbol{y}$ are not allowed [St079]. The concept of shared attributes is closely related to group structure. The complete set of shared attributes $\boldsymbol{y}$ describes how the modalities $\boldsymbol{x} \in \boldsymbol{X}$ are connected and, hence, the underlying group structure. If we have the complete set of attributes $\boldsymbol{y}$, we know the relationship and, hence, also the structure between a group of samples. In this thesis, we define the structure of a group as the set of shared attributes between a group of samples. Note again that there is no connection to group algebra or how the term structure is used in group algebra.

Similar to the assumptions for the unimodal setting (see Section 2.1)[1], we assume the data is generated by some random process involving a joint

[1] If not stated differently, the same assumptions hold as in the unimodal setting.

hidden random variable $\boldsymbol{z}$. See Figure 2.1 for the corresponding graphical model. We assume that the sets of shared attributes $\boldsymbol{y}_A, \forall A \in \mathcal{P}(\mathbb{M})$ as well as the modality attributes $\boldsymbol{y}_m, m \in \mathbb{M}$ are unknown. From the data collection process, we only assume that the set of modalities $\boldsymbol{X}$ is connected by some attributes.

In this setting, we want to learn the underlying attributes $\boldsymbol{y}$ of a multimodal sample $\boldsymbol{X}$. Although we can define a variational posterior $\boldsymbol{z} \sim q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$, compared to the unimodal setting, it is additionally unknown what connects the different modalities. Hence, we want to learn a variational posterior $\boldsymbol{z} \sim q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$ such that $\boldsymbol{z} \approx \boldsymbol{y}$, which reflects the complete set of shared and independent attributes.

2.3.3 *Multimodal ELBO*

Similar to the unimodal ELBO described in Section 2.2.1 and definition 2.2.1, we define the ELBO for multimodal datasets $\mathbb{X}$ as $\mathcal{L}(\Theta, \Phi; \mathbb{X})$. Following the amortized ELBO for unimodal data in Definition 2.2.2, we directly jump to the multimodal amortized ELBO $\mathcal{L}(\Theta, \Phi; \mathbb{X})$ where Θ and Φ are the amortization parameters of the generative and inference model. Having the same assumptions as in Definition 2.2.2, we can write the ELBO of the full dataset as a contribution of ELBOs of multimodal samples

$$\mathcal{L}(\Theta, \Phi; \mathbb{X}) = \sum_{\boldsymbol{X} \in \mathbb{X}} \mathcal{L}(\Theta, \Phi; \boldsymbol{X}) \tag{2.33}$$

Similar to Definition 2.2.2, we define the amortized ELBO $\mathcal{L}(\Theta, \Phi; \boldsymbol{X})$ for multimodal samples $\boldsymbol{X}$.

Definition 2.3.5 (Multimodal Amortized ELBO).
We write the multimodal evidence lower bound $\mathcal{L}(\Theta, \Phi; \boldsymbol{X})$ as

$$\mathcal{L}(\Theta, \Phi; \boldsymbol{X}) = \mathbb{E}_{q_\Phi(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_\Theta(\boldsymbol{X}, \boldsymbol{z}) - \log q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})\right] \tag{2.34}$$

where $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$ is the joint posterior approximation given the multimodal sample $\boldsymbol{X}$, Θ are the parameters of the generative model, and Φ of the inference model.

VAEs provide an efficient and elegant solution to optimize large-scale datasets [KW14]. Hence, we optimize the ELBO in Definition 2.3.5 using a multimodal VAE in this thesis.

We follow the assumptions in VAEs and multimodal VAEs [KW14; Shi+19; WG18] on the conditional independence of the generative model $p_\Theta(\boldsymbol{X} \mid \boldsymbol{z})$ given the latent variable $\boldsymbol{z}$, i.e. $p_\Theta(\boldsymbol{X} \mid \boldsymbol{z}) = \prod_m p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$. We have the amortization parameters $\Theta = [\theta_1, \ldots, \theta_M]$ where θ_m are the parameters for the generative model of modality $\boldsymbol{x}_m$. The objective in Definition 2.3.5 changes accordingly.

$$\begin{aligned}
\mathcal{L}(\Theta, \Phi; \boldsymbol{X}) &= \mathbb{E}_{q_\Phi(\boldsymbol{z}|\boldsymbol{X})}\left[\log\left(p_\Theta(\boldsymbol{z})\prod_{m=1}^{M} p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right)\right] \\
&\quad - \mathbb{E}_{q_\Phi(\boldsymbol{z}|\boldsymbol{X})}\left[\log q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})\right] \\
&= \mathbb{E}_{q_\Phi(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_\Theta(\boldsymbol{z}) + \sum_{m=1}^{M} \log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\
&\quad - \mathbb{E}_{q_\Phi(\boldsymbol{z}|\boldsymbol{X})}\left[\log q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})\right] \\
&= \sum_{m=1}^{M} \mathbb{E}_{q_\Phi(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\
&\quad - \mathbb{E}_{q_\Phi(\boldsymbol{z}|\boldsymbol{X})}\left[\log \frac{q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})}{p_\Theta(\boldsymbol{z})}\right] \\
&= \sum_{m=1}^{M} \mathbb{E}_{q_\Phi(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\
&\quad - D_{\mathrm{KL}}\left[q_\Phi(\boldsymbol{z} \mid \boldsymbol{X}) \mid\mid p_\Theta(\boldsymbol{z})\right] \qquad (2.35)
\end{aligned}$$

Optimizing the ELBO $\mathcal{L}(\Theta, \Phi; \boldsymbol{X})$, we maximize the expected conditional log-likelihoods $\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$ and regularize the joint posterior approximation $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$ with the KL divergence to the prior distribution $p_\Theta(\boldsymbol{z})$.

Following Section 2.2 and the desiderata for multimodal ML in Section 2.3.1, this thesis discusses VAE-based generative approaches to learning from multimodal data. VAEs offer to simultaneously learn meaningful representations and generation of data samples, which both are especially important when dealing with multimodal data as outlined above. Sections 2.3.4 and 2.3.5 describe approaches to learning multimodal representations and generation in more detail.

2.3.4 *Multimodal Representations*

Learning to represent and summarize multimodal data by exploiting the complementary and redundant properties of the different modalities is a fundamental challenge [BAM18]. The importance of good representations in machine learning has been shown in the unimodal setting [BCV13; Hin; KSH12; Mik+13]. Compared to the unimodal setting, additional difficulties surface in learning multimodal representations: how to combine the different data types? How to handle missing data? How to deal with the different noise levels of the individual modalities? While learning good representations is at the forefront of ML research and has even evolved into its subfield [BCV13], this has only recently changed for multimodal datasets [BAM18; LZM22; SM22].

We can write the joint latent representation $\boldsymbol{z}$ of a multimodal sample as

$$\boldsymbol{z} = f\left(\boldsymbol{X}\right) = f\left(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\right) \tag{2.36}$$

where $f(\cdot)$ is a function mapping all modalities $\boldsymbol{X}$ to a joint latent representation $\boldsymbol{z}$. Joint representations combine the unimodal signals into the same representation space. In this thesis, we want to learn multimodal representations that recover the shared and modality-specific attributes (see Section 2.3.2). Hence, the function $f(\cdot)$ should extract shared and modality-specific information from the different data types.

In the pre-DL era, there is only the difference between *early* and *late* fusion of modalities [Atr+10; MHA14]. In early fusion, concatenated modalities are used as input to the model, whereas in late fusion, the latent representations $\boldsymbol{z}_j \; \forall \, j$ are mapped to a joint representation $\boldsymbol{z}$ via, e.g., an aggregation function [Kol30].

With deep neural networks being the most popular method for learning representations $\boldsymbol{z}$ of unimodal datasets X, the ways to fuse modalities became more diverse. A standard scheme to construct a multimodal representation is that every modality $\boldsymbol{x}_j$ starts with several modality-specific neural layers $f_j(\cdot)$, followed by additional shared layers [Ant+15; BAM18; MMG15; OCW14; Wu+14]. One of the disadvantages of learning joint representations this way is the inability of the model to handle missing data naturally [BAM18; Ngi+11; Wan+15].

Following the definitions in Section 2.3.2, we are interested in learning joint representations of subsets $\boldsymbol{X}_A$ in a scalable way. Therefore, we restrict ourselves to aggregation methods following the *late* fusion principle.

2.3.5 *Multimodal Generation*

We are interested in deep generative models of multimodal data [Tom22]. Hence, we are not only interested in learning meaningful representations of multimodal samples (see Section 2.3.4) but also in the generation thereof.

We want to have methods that can generate random samples, i. e. samples that are conditioned on random input, and conditional samples, i. e. samples that are based on input from a subset $\boldsymbol{X}_A$. In probabilistic latent variable models, random generation is equal to generation based on a random input vector z where $z \sim p_\Theta(z)$ is drawn from the prior $p_\Theta(z)$. Conditional generation samples the full multimodal set $\boldsymbol{X}$ conditioned on some input set $\boldsymbol{X}_A$. In probabilistic latent variable models, we sample $z \sim q_{\Phi_A}(z \mid \boldsymbol{X}_A)$ from some posterior approximation $q_{\Phi_A}(z \mid \boldsymbol{X}_A)$ conditioned on the respective subset $\boldsymbol{X}_A$.

2.3.5.1 *Evaluation of Sample Quality*

A big challenge facing generative models is their evaluation. When generating an image based on text, translating a text from one language to another, or writing an image caption, often multiple correct solutions exist. Deciding which one is the best of these solutions is not straightforward. There does not even need to be one single best solution. The preference between different solutions might follow subjective reasons. Fortunately, there are approximate metrics for evaluating generative models [BAM18].

For most tasks, the gold standard in evaluating generated samples is to have a group of humans independently judge every sample. We can construct a single metric using a Likert-like scale [Lik32], which also helps scale responses in survey research [BAM18]. Another method is to perform preference studies where the participant sees at least two examples and decides on a favorite or ranking. However, despite being the gold standard because they result in an evaluation close to human judgment, user studies are time-consuming and expensive. Furthermore, special attention and care must be paid to avoid biases when designing and conducting studies [BAM18].

TEXT-DOMAIN METRICS For evaluating generated or translated text, multiple automatic metrics have been proposed, e.g., [Pap+02, BLEU], [LH03, ROUGE], [DL14, Meteor], and [VLZP15, CIDEr].

IMAGE-DOMAIN METRICS Various metrics have been proposed to evaluate the quality of generated images. The inception score [IS, Sal+16] calculates a score that uses a pre-trained Inception-Net [Sze+16]. The IS is based on the output of the pre-trained network. It is maximized under the following conditions:

1. The entropy of the output label distribution given an image $p(\cdot \mid \boldsymbol{x})$ should be small, i.e., the InceptionV3-Net should be able to predict the output class of the generated image confidently,

2. The entropy of the marginalized output label distribution $\int_{\boldsymbol{z}} p(y \mid \boldsymbol{x} = G(\boldsymbol{z})d\boldsymbol{z}$ should be large, i.e., the generated samples should generate as many different output classes as possible.

The higher the IS, the better the quality of generated samples.

Unlike IS, the Fréchet-Inception-distance [FID, Heu+17] calculates a metric based on the generated and the original images in the training set. The FID compares the mean and standard deviation of the last hidden layer of the Inception-Net [Sze+16] between original and generated samples using the Fréchet distance [Fré57].

Other metrics, which try to circumvent the one-dimensional nature of the FID, also exist. Given a dataset, we want the generated samples to be sharp and distinct but also reflect the total entropy of the training set, e.g., we do not want a model to produce a single perfect image but always the same one. Hence, precision-recall-like metrics to assess fidelity and diversity for evaluating generated samples have been proposed [Kyn+19; Nae+20; Saj+18; SWR19]. These metrics allow the evaluation of generated samples using two dimensions of performance metrics, which can highlight the trade-offs between precision and recall of generated samples.

The metrics for evaluating generated texts and images are modality-specific, and adapting them to other data types is not straightforward. They either rely on pre-trained networks as for the FID [Heu+17; Sze+16], or *a priori* knowledge about well-working architectures [Nae+20]. Additionally, not every type of modality $\boldsymbol{x}_m$ can be evaluated visually to get at least a qualitative impression of model performance, which poses additional challenges for learning from multimodal datasets if there is a lack of good evaluation metrics.

LIKELIHOOD-BASED EVALUATION Every VAE optimizes the marginal log-likelihood of the data $\log p(\boldsymbol{X})$ by maximizing the ELBO (see Section 2.2.5). Hence, given the assumptions of the model and the data distribution, we can evaluate the quality of the approximation to $\log p(\boldsymbol{X})$ using the achieved log-likelihood. Although Theis, Oord, and Bethge [TOB15] point out the difficulties of using likelihood-based evaluation of sample quality when comparing different classes of models, we find it adequate to compare the generated samples of different multimodal VAEs that are based on the same assumptions and use the same network architectures (see Section 3.2).

2.3.5.2 *Evaluation of Coherence*

For multimodal data, generated samples should be coherent. A coherent multimodal sample $\boldsymbol{X}$ is aligned by the same attributes $\boldsymbol{y}$. In other words, the attribute(s) shared by the modalities $m \in \boldsymbol{X}$ should be visible in a generated multimodal sample. We can illustrate the concept of coherence using an example dataset. We are given a dataset of images, text, and audio samples that shares the same digit information. This means that all modalities display the same digit in their respective modality, e. g. the image shows the number "8", the text writes "eight", and the audio sample is a recording of a person saying "eight". In this setting, we say that a conditionally generated sample of a missing modality is coherent if it displays the same digit information as the input modalities. A randomly generated multimodal sample is coherent if the samples of all modalities display the same digit in their respective modality.

Definition 2.3.6 (Coherence of randomly generated samples)**.**
A randomly generated multimodal sample $\boldsymbol{X}_A \sim p_{\Theta_A}(\boldsymbol{X}_A \mid \boldsymbol{z})$ where $\boldsymbol{z} \sim p_{\Theta_A}(\boldsymbol{z})$ is coherent if the shared attributes $\boldsymbol{y}_A \cap \boldsymbol{y}_m$ of all modalities $m \in A$ are equal, i. e.

$$(\boldsymbol{y}_A \cap \boldsymbol{y}_m) = (\boldsymbol{y}_A \cap \boldsymbol{y}_n) \;\; \forall\, m, n \in A \tag{2.37}$$

Definition 2.3.7 (Coherence of conditionally generated samples)**.**
A multimodal sample $\boldsymbol{X}_A \sim p_{\Theta_A}(\boldsymbol{X}_A \mid \boldsymbol{z}), A \subseteq \mathbb{M}$ that is generated conditioned on some multimodal subset $\boldsymbol{X}_B, B \subseteq \mathbb{M}$, i. e. $\boldsymbol{z} \sim q_{\Phi_B}(\boldsymbol{z} \mid \boldsymbol{X}_B)$, is coherent if it is invariant concerning all attributes that are shared between A and B, i. e., $\boldsymbol{y}_A \cap \boldsymbol{y}_B$. Additionally, the generated multimodal sample $\boldsymbol{X}_A$ needs to be coherent as defined in Definition 2.3.6.

For evaluating different methods based on their achieved coherence (see Definitions 2.3.6 and 2.3.7), we need to know the underlying attributes. Therefore, the coherence measure is more of a theoretical value and is important for the development of models. In real-world scenarios where the number of labeled samples is small for multimodal datasets, coherence as defined in Definitions 2.3.6 and 2.3.7 is challenging to calculate.

2.3.6 *Weakly-Supervised Data In General*

Given a multimodal dataset $\mathbb{X}$ that follows Definition 2.3.1, we assume that a sample $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ of M data types $\boldsymbol{x}_m$ is an unordered set. Generally, every data type $\boldsymbol{x}_m$ needs its particular architecture because the data dimensions and inductive biases differ. Hence, sharing network parameters [see CS96] to leverage the additional information more efficiently is impossible. In the multiview setting, on the other hand, we collect a set of images $\boldsymbol{X}_I = \{\boldsymbol{x}_{I,1}, \dots, \boldsymbol{x}_{I,M}\}$ where every image $\boldsymbol{x}_{I,m}$ has the same dimensionality. Because there is only one data type with the same dimensionality, we can use the same architecture for all images $\boldsymbol{x}_{I,m}$ with shared network parameters [e. g., Loc+20]. Like this, we leverage the additional information more directly than in the multimodal setting. Additionally, there is no ordering between the different data types of a multimodal sample, which differs from time series. In time-series data, on the other hand, we cannot only leverage the same network as in the multiview setting, but we also know from the data collection process that the time axis provides an ordering between measurement values. Hence, the time axis provides additional knowledge on the data structure that can and should be leveraged [e. g., For+20].

2.4 GRADIENT-BASED OPTIMIZATION OF DISCRETE STRUCTURES

VAEs (see Section 2.2.5) combine deep learning and the learning of probability distributions. Utilizing the reparameterization trick [KW14; RMW14; TLG15] and integrating continuous probability distributions in the computation graph as stochastic nodes enables the learning of the distributions' trainable parameters using gradient-based optimization [Hui+21]. However, discrete building blocks pose an additional challenge. Direct reparameterization of discrete distributions is infeasible due to their discontinuous nature [JGP16; MMT17; Pau+20]. On the other hand, discrete distributions enable us to describe structures and relations in a more interpretable way compared to continuous distributions.

DIFFERENTIABLE DISCRETE DISTRIBUTIONS In recent years, finding continuous relaxations for discrete distributions to integrate them into differentiable pipelines gained popularity following the Gumbel-Softmax trick [GST, JGP16; MMT17]. The GST enables reparameterized gradients with respect to the parameters of the categorical distribution and their use in differentiable models. Methods to select k elements - instead of only one - are subsequently introduced. Kool, Hoof, and Welling [KHW19; KHW20a] implemented sequential sampling without replacement using a stochastic beam search (including an extension [KHW20b]). Multiple works on differentiable sorting procedures and permutation matrices have been proposed, e.g., Linderman et al. [Lin+18], Petersen et al. [Pet+21], and Prillo and Eisenschlos [PE20]. Further, Grover et al. [Gro+19] described the distribution over permutation matrices $p(\pi)$ for a permutation matrix π using the Plackett-Luce (PL) distribution [Luc59; Pla75]. Prillo and Eisenschlos [PE20] proposed a computationally simpler variant of Grover et al. [Gro+19]. Based on [Gro+19], Xie and Ermon [XE19] proposed subset selection algorithm of a given number k out of n elements.

In the remaining part of this section, we first re-introduce the categorical distribution and the Gumbel-Max trick. We then describe the Gumbel-Softmax trick [GST, JGP16; MMT17], the continuous relaxation and reparameterization for the categorical distribution. Afterward, we discuss ranking models, how they relate to categorical distributions, and how they can be used to describe continuous relaxations for more complex structures and distributions. Toward the end of this chapter, we introduce random partition models and the hypergeometric distributions.

2.4.1 *Categorical Distribution*

A categorical distribution is a discrete probability distribution of K classes. We parameterize a categorical distribution using normalized probabilities $\boldsymbol{\alpha} \in [0,1]^K$ where $\sum_{k=1}^K \alpha_k = 1.0$, un-normalized scores $\boldsymbol{s} \in \mathbb{R}_+^K$ or un-normalized log-scores $\log \boldsymbol{s} \in \mathbb{R}^K$. The tempered categorical distribution $Cat(\boldsymbol{s}, \tau)$, also known as Gibbs [Gib02] or Boltzmann distribution [Bol68], introduces an additional temperature parameter $\tau \in \mathbb{R}_+$. We denote the tempered parameters with an additional subscript, e. g. $\boldsymbol{s}_\tau$. It follows

$$\alpha_{k,\tau} = \frac{\exp(\log s_k / \tau)}{\sum_{j=1}^K \exp(\log s_j / \tau)} = \frac{s_{k,\tau}}{\sum_{j=1}^K s_{j,\tau}} = \frac{s_{k,\tau}}{Z_\tau} \tag{2.38}$$

where $Z_\tau = \sum_{k=1}^K s_{k,\tau} \in \mathbb{R}_+$ is the normalizing constant of the distribution. For $\tau \rightarrow 0$, $Cat(\boldsymbol{s}, \tau)$ becomes a one-hot encoding equal to a deterministic function where the same class is always selected. For $\tau \rightarrow \infty$, the tempered categorical distribution equals a non-informative uniform distribution where every class is equally likely. Equation (2.38) is also known as the tempered softmax function with temperature parameter τ [Hui+21].

2.4.1.1 *Gumbel-Max Trick*

GUMBEL DISTRIBUTION The Gumbel-distribution [Gum35] is an extreme value distribution [Mis36], which models optima and rare events [Hui+21]. It is parameterized by location $\mu \in \mathbb{R}$ and scale $\beta \in \mathbb{R}_+$ parameters. The probability (PDF) and cumulative (CDF) density functions are defined as follows

$$f(x) = \frac{1}{\beta} \exp\left(-\frac{x-\mu}{\beta}\right) \exp\left(\exp\left(-\frac{x-\mu}{\beta}\right)\right) \tag{2.39}$$

$$F(x) = \exp\left(-\exp\left(-\frac{x-\mu}{\beta}\right)\right) \tag{2.40}$$

We use the short notation $G(\mu, \beta)$ for a Gumbel distribution with location parameter μ and scale parameter β. The standard distribution follows from $\mu = 0$ and $\beta = 1$ (i. e. $G(0, 1)$). μ and β are not the mean and variance of the distribution. Mean and variance are given as

$$\mathbb{E}_{G(\mu,\beta)}[x] = \mu + \gamma\beta, \tag{2.41}$$

$$\mathbb{E}_{G(\mu,\beta)}\left[(x - \mathbb{E}_{G(\mu,\beta)}[x])^2\right] = \frac{\pi^2}{6}\beta^2 \tag{2.42}$$

where $\gamma \approx 0.577$ is the Euler-Mascheroni constant [Bre37; Eul40], and $\pi \approx 3.14$ is the constant, which is the ratio of a circle's circumference and diameter. The inverse cumulative density function (ICDF) is given as

$$F^{-1}(u) = -\beta \log(-\log u) + \mu \tag{2.43}$$

From Equation (2.43), we see that the Gumbel distribution is closed under addition and scaling, i. e. any Gumbel variable can be generated by scaling a standard Gumbel variable $x \sim G(0, 1)$ with β and shifting it by μ [Hui+21]. We use Equation (2.43) for inverse transform sampling of Gumbel variables $x \sim G(\mu, \beta)$. In inverse transform sampling, we transform a sample $u \sim U(0, 1)$ where $U(0, 1)$ is a uniform distribution between 0 and 1 into a Gumbel sample with location parameter μ and scale parameter β via a double negative logarithm relation. Therefore, we refer to the Gumbel distribution as a double exponential distribution [Hui+21].

The Gumbel-Max trick [Gum54] draws a sample from a categorical distribution $I \sim Cat(\boldsymbol{\alpha})$ by adding i.i.d. Gumbel noise samples g_k to unnormalized log-scores (or log-probabilities). After perturbing the log-scores, the Gumbel-Max trick selects the index with the maximum value.

Theorem 2.4.1 (Gumbel-Max trick [Gum54]).
Given some scores $\boldsymbol{s} \in \mathbb{R}_{+}^{K}$, we can sample from a categorical distribution with weights $\boldsymbol{\alpha} = \boldsymbol{s}/Z$ where $Z = \sum_{k=1}^{K} s_k$ by adding i.i.d. *Gumbel noise $g_k \sim G(0,1)$, $\forall\, k \leq K$ to the un-normalized log-scores $\log s_k$, such that*

$$I = \underset{k \leq K}{\arg\max}\,(\log s_k + g_k) \sim Cat(\boldsymbol{\alpha}) \tag{2.44}$$

$$M = \max_{k \leq K}\,(\log s_k + g_k) \sim G(\log Z, 1) \tag{2.45}$$

We provide the proof to Theorem 2.4.1 in Appendix A.1.1. In other words, the Gumbel-Max trick describes an efficient sampling procedure from a categorical distribution $Cat(\boldsymbol{s}, \tau)$ with un-normalized parameters without the need of normalizing the values using softmax function, i. e.

$$\boldsymbol{z} = \text{one\_hot}(\underset{k}{\arg\max}(\log s_k + g_k)) \tag{2.46}$$

where g_k as in Theorem 2.4.1. However, the Gumbel-Max trick does not enable reparameterized gradients because of the jump discontinuities of the argmax function. The derivative of the $\arg\max$ function returns 0 everywhere except at the boundary of state changes, where it is undefined, which makes the Gumbel-Max trick not suitable for reparameterization [Pau+20]. Hence, we need to overcome this limitation to include categorical distributions as stochastic nodes in gradient-based optimization.

2.4.1.2 *Gumbel-Softmax Trick*

Gumbel-Softmax is a continuous distribution approximating the categorical distribution whose parameter gradients can be easily computed using the reparameterization trick [JGP16; MMT17]. Maddison, Mnih, and Teh [MMT17] name it *Concrete* distribution, *con*tinuous relaxations of dis*crete* random variables.

Although an efficient approach to sampling from a categorical distribution, the $\arg\max$ in Equation (2.46) still prohibits the use of a reparameterized gradient estimator, i. e.

$$\mathbb{E}\left[\frac{d\mathbb{L}(\boldsymbol{z})}{d\boldsymbol{s}}\right] = \frac{d}{d\boldsymbol{s}}\mathbb{E}\left[\mathbb{L}(\boldsymbol{z})\right] \tag{2.47}$$

is not possible for any loss function $\mathbb{L}(\cdot)$, random variable $\boldsymbol{z}$ and scores $\boldsymbol{s}$ as in Equation (2.46) due to the discontinuities of the $\arg\max$ function.

The Gumbel-Softmax trick [GST, JGP16; MMT17] overcomes this limitation by approximating the arg max using a tempered softmax function, i. e.

$$z_{k,\tau} = \frac{\exp((\log s_k + g_k)/\tau)}{\sum_{j=1}^{K} \exp((\log s_j + g_j)/\tau)} \quad \text{for} \quad k = 1, \dots, K \tag{2.48}$$

where τ is a temperature parameter and the distribution of $\boldsymbol{z}_{k,\tau}$ has a closed form [JGP16; MMT17].

Definition 2.4.1 (Concrete or Gumbel-Softmax Random Variables [JGP16; MMT17])**.**
Given scores $\boldsymbol{s} \in \mathbb{R}_+^K$ and temperature $\tau \in \mathbb{R}_+$. $\boldsymbol{z}_\tau$ has a concrete distribution, i. e. $\boldsymbol{z} \sim$ Concrete$(\boldsymbol{s}, \tau)$ with location parameter $\boldsymbol{s}$ and temperature parameter τ, if its density is given as

$$p_{\boldsymbol{s},\tau}(\boldsymbol{z}) = \Gamma(K)\tau^{K-1} \prod_{k=1}^{K} \left(\frac{s_k z_k^{-\tau-1}}{\sum_{j=1}^{K} s_j z_j^{-\tau}} \right) \tag{2.49}$$

Equation (2.48) enables the optimization of

$$\mathbb{E}\left[\frac{d\mathbb{L}(\boldsymbol{z}_\tau)}{d\boldsymbol{s}}\right] = \frac{d}{d\boldsymbol{s}}\mathbb{E}\left[\mathbb{L}(\boldsymbol{z}_\tau)\right] \quad \text{for} \quad \tau > 0 \tag{2.50}$$

using gradient-based optimization. Although Equation (2.48) leads to a biased estimation of Equation (2.47), it is an unbiased estimation of the right-hand side of Equation (2.50), and as τ goes to zero, the softmax function approximates the arg max function [Pau+20]. Hence, the Gumbel-Softmax distribution approximates the categorical distribution in the $\tau \to 0$ limit.

The straight-through Gumbel estimator [JGP16; MMT17] provides a way to use *hard* samples in the forward pass of a network but use its soft version in the backward pass to still allow the backpropagation of gradients. It is related to and based on the biased path derivative estimator proposed in Bengio, Léonard, and Courville [BLC13]. For the straight-through Gumbel estimator, we use the argmax to discretize the perturbed categorical weights in Equation (2.48), but still use the continuous approximation with the softmax function in the backward pass.

2.4.2 *Ranking Models*

Data consisting of rankings appear in Psychology, Animal Science, Educational Testing, Sociology, Economics, and Biology [Mar96], and problems involving ranked lists are widespread [GS09].

In statistics, it has been a field of study for some time [Gum54; Luc59; Pla68; Thu27] (or e. g. Marden [Mar96] for an overview). The field of ML approached ranking data with learning to rank applications [Joa+07; L009]. Explicitly modeling ranking data only recently gained attention in the ML community with the introduction of continuous relaxations for discrete distributions (see Section 2.4.1.2). This section gives a short overview of ranking models and their probability distributions. We introduce the underlying theory behind the Plackett-Luce (PL) ranking model [Luc59; Pla68], describe under which settings PL models are equal to Thurstonian models [Thu27; Yel77], and set the GST in context to ranking models.

We use the short notation $[n]$ to denote a set of n elements, which are named using the integers $1, \ldots, n$, i. e. $[n] = \{1, \ldots, n\}$. Additionally, we use $\pi \in \Pi_n$ to describe an ordering of $[n]$, where Π_n is the set of all orderings of n elements.

2.4.2.1 *Luce's Choice Axiom*

Luce's choice axiom [LCA, Luc59] envisions a setting in which an individual makes repeated choices from a set $[n]$ containing n alternatives [Yel01]. On each occasion, precisely one element i is chosen where choice is assumed to be probabilistic. $P(i; A)$ denotes the probability that i is chosen when the set of available alternatives is $A \subseteq [n]$. LCA allows us to describe relationships between choice probabilities for sets of available alternatives, where the particular case of 2-alternative choice (i. e. paired comparison) has its own notation $P(i, j)$ [Yel01].

Axiom 2.4.1.1 (Luce's choice axiom [Luc59]).
For $P(A; [n])$ being the probability that when the full set $[n]$ is available, the alternative chosen belongs to the subset A, i. e. $P(A; [n]) = \sum_{i \in A} P(i; [n])$, and $P(i, j)$ the probability of choosing i when the available set is $\{i, j\}$ (i. e. probability of the paired comparison), we have

1. if $P(i, j) \neq 0 \; \forall \; i, j \in A, i \neq j$, then $\forall \, A \subseteq [n]$ and every $j \in A$:

$$P(i; [n]) = P(i; A) P(A; [n]) \tag{2.51}$$

2. if $P(i, j) = 0$ for some pair $i, j \in [n]$, then $\forall \; A \subset [n]$:

$$P(A; [n]) = P(A \setminus \{i\}; [n] \setminus \{i\}) \tag{2.52}$$

Note that in the first part of Axiom 2.4.1.1, the assumption $P(i,j) > 0$ together with Equation (2.51), implies $P(i;A) > 0 \, \forall \; i$. As a result, $P(A;[n])$ is non-zero, and we can write Equation (2.51) as [Yel01]

$$P(i;A) = \frac{P(i;[n])}{P(A;[n])} \tag{2.53}$$

The right-hand side of Equation (2.53) is the conditional probability that we choose i from $[n]$, given that the choice of $[n]$ is some member of A [Yel01].

LCA implies that there is a function $v(\cdot)$ mapping the n alternatives in $[n]$ to n non-negative real numbers $v(1), \dots v(n)$ with the following property [Yel01]: $\forall \, i \in A \subseteq [n]$, we have

$$P(i;A) = \frac{v(i)}{\sum_{j \in A} v(j)} \tag{2.54}$$

From Equation (2.53), it is clear that $v(i) = P(i;[n])$ is such a scale function [Luc59]. Multiplying $P(i;A)$ by a constant factor $c > 0$ such that $\tilde{v}(i) = cv(i) \, \forall \, i \in [n]$ does not change Equation (2.54). Therefore, $v(\cdot)$ is a ratio scale [Yel01].

From Equation (2.54), it follows that

$$\frac{P(i;A)}{P(j;A)} = \frac{v(i)}{v(j)} = \frac{P(i;[n])}{P(j;[n])}, \tag{2.55}$$

which is called the constant-ratio rule. In other words, the ratio between the probability of selecting element i and the probability of selecting element j is independent of the remaining options [Luc59; Yel01].

Gumbel random variables also adhere to LCA [Hui+21]. For $s_i = v(i)$ being the score associated with element i, following Equation (2.54) we can write

$$P(i;A) = \frac{v(i)}{\sum_{j \in A} v(j)} = \frac{s_i}{\sum_{j \in A} s_j}, \tag{2.56}$$

which is equal to the right hand-side of Equation (2.38) for $\tau = 1$. Scaling the un-normalized probabilities of a categorical distribution $\boldsymbol{s}$ with a constant factor Z results in subtraction in log-space

$$\log \frac{s_i}{Z} = \log s_i - \log Z \tag{2.57}$$

Given the scaling-invariance of the arg max function, the Gumbel-Max trick also applies to normalized log-probabilities $\log s_i / Z$

$$I = \arg\max_{i \le n}(\log s_i + g_i) = \arg\max_{i \le n}(\log \frac{s_i}{Z} + g_i), \tag{2.58}$$

which is a direct result from Gumbel random variables following LCA [Hui+21]. As a result, we can apply the Gumbel-Max trick over sub-domains of categorical distributions using the un-normalized scores $\boldsymbol{s}$ [Luc59; Pla68].

2.4.2.2 *Plackett-Luce Model*

Given a set of n elements $[n] = \{1, \dots, n\}$, the Plackett-Luce model [PL, Luc59; Pla68] is based on some positive scores $\boldsymbol{s} = [s_1, \dots, s_n] \in \mathbb{R}^n_+$ where the score s_i is associated with element i. The PL model describes a ranking of elements using the probability of a random ordering $\pi \in \Pi_n$ where Π_n defines the set of all orderings of n elements. The underlying assumption of the PL model over rankings is that the larger the score s_i, the more preferred element i is. Hence, we write the probability that element i is selected first as $p(\pi_1 = i; \boldsymbol{s})$ with $\boldsymbol{s}$ being the parameters of the PL model over orderings and π_k the index of the element ranked at position k. The score s_i is proportional to the probability that i is selected first [Mar96], i. e.

$$p(\pi_1 = i; \boldsymbol{s}) = \frac{s_i}{\sum_{j=1}^n s_j} \tag{2.59}$$

One of the key assumptions of the PL model is that the probability $p(\pi_1 = i; \boldsymbol{s})$ to rank element i first is proportional to the identical scores $\boldsymbol{s}$ as the probability $p(\pi_k = j \mid \pi_1 = i_1, \dots, \pi_{k-1} = i_{k-1}; \boldsymbol{s})$ of ranking element j at the kth position given the elements ranked at positions 1 to $k - 1$ [Mar96] (see also Section 2.4.2.1). We write the probability $p(\pi_k = j \mid \pi_1 = i_1, \dots, \pi_{k-1} = i_{k-1}; \boldsymbol{s})$ as

$$p(\pi_k = j \mid \pi_1 = i_1, \dots, \pi_{k-1} = i_{k-1}; \boldsymbol{s}) = \frac{s_j}{\sum_{l \notin \{i_1, \dots, i_{k-1}\}} s_l} \tag{2.60}$$

We construct the probability $p(\pi; \boldsymbol{s})$ of the ordering of the set $[n]$ from the individual probabilities $p(\pi_k = j \mid \pi_1 = i_1, \dots, \pi_{k-1} = i_{k-1}; \boldsymbol{s})$

$$\begin{align} p(\pi; \boldsymbol{s}) &= p(\pi_1 = i_1) \cdots p(\pi_n = i_n \mid \pi_1 = i_1, \dots, \pi_{n-1} = i_{n-1}; \boldsymbol{s}) \tag{2.61} \\ &= p(\pi_1; \boldsymbol{s}) \cdots p(\pi_n \mid \pi_1, \dots, \pi_{n-1}; \boldsymbol{s}) \tag{2.62} \end{align}$$

where the second line is a short notation of the first one.

Definition 2.4.2 (Plackett-Luce Model [Luc59; Pla75]).
Given a set of elements $[n] = \{1, \dots, n\}$ and scores $\boldsymbol{s} = [s_1, \dots, s_n] \in \mathbb{R}^n_+$ where score s_i is associated with element i, the probability of an ordering $\pi \in \Pi_n$ of the set $[n]$ following the Plackett-Luce model is given as

$$p(\pi; \boldsymbol{s}) = \frac{s_1}{Z} \cdot \frac{s_2}{Z - s_1} \cdots \frac{s_{n-1}}{Z - \sum_{j=1}^{n-2} s_j} \cdot \frac{s_n}{Z - \sum_{j=1}^{n-1} s_j} \tag{2.63}$$

where $Z = \sum_{i=1}^{n} s_i$, and Π_n is the set of all orderings of n elements.

We can write every ordering as a permutation matrix, where the ith index of the first row indicates that the ith element is ranked at the first position. Multiplying the permutation matrix π with the scores $\boldsymbol{s}$ re-orders the scores according to the ordering induced by π, i. e.

$$p(\pi; \boldsymbol{s}) = \frac{(\pi \boldsymbol{s})_1}{Z} \cdots \frac{(\pi \boldsymbol{s})_n}{Z - \sum_{j=1}^{n-1} (\pi \boldsymbol{s})_j} \tag{2.64}$$

where $(\pi \boldsymbol{s})_k$ is a short notation for $(\pi \boldsymbol{s})_k = \sum_{i=1}^{n} (\pi \boldsymbol{s})_k$, which denotes the score of the element j ranked at position k.

2.4.2.3 *Thurstonian Model*

A Thurstonian model [Thu27] assumes an unobserved (and typically independent) random score variable s_i for every item $i \in [n]$. Drawing from the score distribution and sorting according to the sampled scores provides a sample ranking π. In consequence, the score distribution results in a distribution over orderings [GS09].

Definition 2.4.3 (Thurstonian Model [Thu27])**.**
Given some unobserved and independent scores $\boldsymbol{s} \in \mathbb{R}_+^n$ where score s_i is associated with element i, we can generate a ranking $\pi \in \Pi_n$, which is derived from $\boldsymbol{s}$ using a deterministic function f, i. e. $\pi = f(\boldsymbol{s})$, such that $(\pi \boldsymbol{s})_1 > \cdots > (\pi \boldsymbol{s})_n$.

The parameters of a Thurstonian model are the parameters of the distribution of the underlying scores $\boldsymbol{s}$. Definition 2.4.3 shows that the probability of an ordering is described using the parameters of the underlying score distribution. Yellott [Yel77] provides the result of the relation between PL- and Thurstonian models.

Theorem 2.4.2 (Equality between Placket-Luce [Luc59; Pla75] and Thurstonian Models [Thu27; Yel77])**.**
Suppose the score variables are independent, and the score distributions are identical up to their mean. In that case, the score distributions give rise to the PL model if and only if the scores $\boldsymbol{s}$ are distributed according to a Gumbel distribution.

We provide the proof to Theorem 2.4.2 [Yel77] in Appendix A.1.2. Following Definitions 2.4.2 and 2.4.3 and Theorem 2.4.2, we can describe the probability of a random ordering $\pi = f(\tilde{\boldsymbol{s}})$ using the parameters $\boldsymbol{s}$ of the underlying score distribution, and sample from the distribution.

Corollary 2.4.1 (Sampling from Plackett-Luce Models [Gro+19; Yel77]). *Given scores $\boldsymbol{s} \in \mathbb{R}^n_+$ where the score s_i corresponds to element $i \in [n]$, we sample $g_i \sim \text{Gumbel}(0, \beta)$ independently with zero mean and fixed scale β. Let $\tilde{\boldsymbol{s}}$ denote the perturbed log-scores such that $\tilde{s}_i = \beta \log s_i + g_i$ and $\pi = f(\tilde{\boldsymbol{s}})$, where $f(\cdot)$ is a deterministic function, then*

$$p((\pi\tilde{\boldsymbol{s}})_1 \geq \cdots \geq (\pi\tilde{\boldsymbol{s}})_n) = \frac{(\pi\boldsymbol{s})_1}{Z} \cdots \frac{(\pi\boldsymbol{s})_n}{Z - \sum_{j=1}^{n-1}(\pi\boldsymbol{s})_j} \tag{2.65}$$

where $Z = \sum_{i=1}^{n} s_i$

In other words, the probability of a random ordering π that follows the probability distribution $p(\pi; \boldsymbol{s})$ as defined in Corollary 2.4.1 has a closed form solution based on the random ordering π itself and the underlying and non-perturbed scores $\boldsymbol{s}$ of the distribution.

2.4.2.4 *Differentiable Ranking Models*

To integrate probability distributions over orderings $p(\pi; \boldsymbol{s})$ (see Sections 2.4.2.2 and 2.4.2.3) into gradient-based optimization pipelines, the deterministic function $f(\cdot)$ generating the random ordering π needs to be differentiable.

Permutation matrices belong to the class of doubly-stochastic matrices, i. e. every column and every row sum to one,

$$\sum_{k=1}^{n} \pi[k, i] = 1, \quad \text{and} \quad \sum_{i=1}^{n} \pi[k, i] = 1 \tag{2.66}$$

Hence, a permutation matrix π ranks element i at position k, if $\pi[k, i] = 1$.

Following Definition 2.4.3, the deterministic function $f(\cdot)$ generating the permutation matrix π has to be the sort operator, i. e.

$$\pi = f(\boldsymbol{s}) = \text{sort}(\boldsymbol{s}) \tag{2.67}$$

However, the vanilla sort operator is not differentiable. In the remaining part of this section, we describe the work of Grover et al. [Gro+19], which introduced a differentiable sorting method based on PL models, where the scores $\boldsymbol{s}$ follow a Gumbel distribution. In combination with Corollary 2.4.1, their work, `NeuralSort`, describes a continuous relaxation for $p(\pi; \boldsymbol{s})$. At the beginning of this section, we described other differentiable sorting mechanisms, some of which could be used too.

Corollary 2.4.2 (From Grover et al. [Gro+19]).
Given a vector of scores $\boldsymbol{s} = [s_1, \dots, s_n]^T \in \mathbb{R}^n_+$, and the matrix $A_{\boldsymbol{s}}$ of pairwise absolute differences of the elements of $\boldsymbol{s}$, i. e. $A_{\boldsymbol{s}}[i,j] = |s_i - s_j|$. The permutation matrix $\pi = sort(\boldsymbol{s})$ is given by

$$\pi[i,j] = \begin{cases} 1 & \textit{if} \quad j = \arg\max((n+1-2i)\boldsymbol{s} - A_{\boldsymbol{s}}\mathbb{1}) \\ 0 & \textit{otherwise} \end{cases} \tag{2.68}$$

where $\mathbb{1} = [1, \dots, 1]^T$.

Corollary 2.4.2 (we provide the proof in Appendix A.1.4) formulates a sorting operator based on the pairwise differences between elements $A_{\boldsymbol{s}}$. The arg max operator is not differentiable (see Section 2.4.1.2), which prohibits taking derivatives of π with respect to $\boldsymbol{s}$. Similar to Section 2.4.1.2, we use the tempered softmax function such that

$$\pi_\tau[i,:] = \text{softmax}\left(\frac{(n+1-2i)\boldsymbol{s} - A_{\boldsymbol{s}}\mathbb{1}}{\tau}\right) \tag{2.69}$$

where τ is again the temperature parameter for the softmax function. The resulting relaxation is continuous everywhere and differentiable almost everywhere with respect to $\boldsymbol{s}$ [Gro+19].

Combining Corollaries 2.4.1 and 2.4.2 enables a continuous relaxation for the distribution $p(\pi; \boldsymbol{s})$, which is differentiable and reparameterizable. `NeuralSort` scales $\mathcal{O}(n^2)$ in computation steps and memory requirements, whereas the fastest non-differentiable sorting methods only need $\mathcal{O}(n \log n)$ computation steps [Gro+19; Knu97].

In the remaining part of this work, we remove the sub-script τ for all continuous relaxations. However, we always use their tempered versions.

2.5 SET PARTITIONS AND RANDOM PARTITION MODELS

Partitioning a set of elements into subsets is a classical mathematical problem that has attracted much interest over the last few decades. A partition over a given set is a collection of non-overlapping subsets such that their union results in the original set. While there are many well-studied combinatorial partitioning problems [GKP89; Rot64], recent advances in machine learning give rise to new challenges revolving around set-partitioning.

A random partition model [RPM, Har90; MQR11] defines a probability distribution on the space of set partitions. Previous works on RPMs include product partition models [Har90], species sampling models [Pit96], and model-based clustering approaches [BS04]. Further, Lee and Sang [LS22] investigate the balancedness of subset sizes of RPMs. They all require tedious manual adjustment, are non-differentiable, and are unsuitable for modern ML pipelines.

In ML, partitioning a set of elements into different subsets is essential for many applications, such as classification or clustering, e. g. [Dil+16; Jia+16; Man+21] implicitly define RPMs to perform clustering. They compute partitions using VAEs by making *i.i.d.* assumptions about the samples in the dataset and imposing soft assignments of the clusters to data points during training.

In the following, we introduce the notation for random partition models, some key properties, and popular non-differentiable random partition models. Let $\mathcal{S}$ be a set of natural numbers. In case of $\mathcal{S}$ being the first n natural numbers, we use the same notation as in Section 2.4.2, i. e. $[n] = \{1, \ldots, n\}$.

Definition 2.5.1 (Set partition ρ [MS16]).
A set partition $\rho = [\mathcal{S}_1, \ldots, \mathcal{S}_K]$ of a set $[n] = \{1, \ldots, n\}$ is a collection $\mathcal{S}_1, \ldots, \mathcal{S}_K$ of non-empty disjoint subsets of $[n]$ such that $\cup_{k=1}^{K} \mathcal{S}_k = [n]$. The size of a subset is given by $n_k = |\mathcal{S}|$.

Hence, a set partition assigns every element i to precisely one of K subsets where K is *a priori* unknown. We use the terms set partition and partition model interchangeably in this thesis.

We denote the set of all set partitions of $[n]$ by P_n, and the number of all set partitions of $[n]$ by $|P_n|$, with $|P_0| = 1$. $|P_0| = 1$ comes from the fact that there is only a single partition for the empty set $\emptyset$. The numbers $|P_n|$ are known as Bell numbers.

Definition 2.5.2 (Stirling Number of the second kind [GKP89; MS16]).
The set of all set partitions of $[n]$ with exactly K subsets is denoted by $P_{n,K}$. The number of $|P_{n,K}|$ of set partitions of $[n]$ into K blocks is denoted by $S(n, K)$ and is called the Stirling number of the second kind.

The Stirling number of the second kind $S(n, K)$ for $n \geq K \geq 1$ is given as

$$S(n, K) = \frac{1}{K!} \sum_{j=1}^{K} \binom{K}{j} (-1)^{K-j} j^n \tag{2.70}$$

The Bell number $|P_n|$ follows by definition

$$|P_n| = \sum_{k=1}^{n} S(n, k) \tag{2.71}$$

Famous examples of RPMS include discrete random probability measures induced RPMS, such as Dirichlet process [Fer73], Pitman-Yor process [PPY92; PY97], and the mixture of finite mixtures [MH18].

2.5.1 *Dirichlect Prior Partition Model*

Sampling from a Dirichlet process prior [DP, Fer73] is one of the most popular RPMs [Mül+15]. Consider a model of the type

$$X_1, \dots, X_n \mid F \sim F, \qquad F \sim DP(\lambda F_0) \tag{2.72}$$

where DP is a Dirichlet process with base measure F_0 and weight parameter $\lambda > 0$ [Qui06]. Since F is discrete, there can be ties among the different X_i. The Polya urn representation [BM73] offers a second view of this property. A partition of $[n]$ can be formed by defining clusters over equivalence classes under the relation $i \sim j$ if and only if $X_i = X_j$. Let c_i denote the cluster membership of element i, then, let $c_1 = 1, X_1^* = X_1$, and for $j > 1$, let $c_j = c_i$ if $X_j = X_i$ for some $1 \leq j - 1$ and $c_j = \max\{c_1, \dots, c_{j-1}\}$ if $X_j \notin \{X_1, \dots, X_{j-1}\}$, in which case $X_{c_j}^* = X_j$. See Arratia, Barbour, and Tavaré [ABT92] for the *Chinese restaurant process*, a more colorful description of the induced partition structure using a DP prior.

2.5.2 *Product Partition Models*

Product partition models [PPM, Har90] define an alternative probability distribution over partition models [Mül+15; Qui06]. For any partition $\rho = \{\mathcal{S}_1, \dots, \mathcal{S}_K\}$ and data samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$, we assume that

$$p(\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \mid \rho) = \prod_{k=1}^{K} p_{\mathcal{S}_k}(X_{\mathcal{S}_k}), \tag{2.73}$$

where $X_{\mathcal{S}_k} = \{\boldsymbol{x}_i : i \in \mathcal{S}_k\}$ and $p_{\mathcal{S}_k}(X_{\mathcal{S}_k})$ depends only on $\mathcal{S}_k$ and not on any of the other subsets $\mathcal{S}_j, j \neq k$. In turn, the partition ρ is assigned a prior distribution as follows

$$p(\rho = \{\mathcal{S}_1, \dots, \mathcal{S}_K\}) = \frac{1}{P_0} \prod_{k=1}^{K} coh(\mathcal{S}_k), \tag{2.74}$$

where $coh(A)$ is the cohesion function of a subset $A \subseteq [n]$ such that $coh(A) \geq 0$, and P_0 is the normalizing constant such that $\sum_\rho p(\rho) = 1$.

Additional applications of PPMs can be found in Barry and Hartigan [BH92], Barry and Hartigan [BH93], Loschi and Cruz [LC02], and Loschi et al. [Los+03].

2.5.3 *Model-Based Clustering*

RPMs based on mixture distributions are known as model-based clustering [MBC, BR93; DR98; MR84]. The data points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are modelled independently as a mixture distribution

$$p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \mid K, \boldsymbol{\theta}) = \prod_{i=1}^{n} \sum_{k=1}^{K} \alpha_k p_k(\boldsymbol{x}_i \mid \theta_k), \tag{2.75}$$

where K is the number of components in the mixture distribution, α_k is the weight of the kth component such that it holds $\alpha_k \geq 0, \forall\, k$ and $\sum_{k=1}^{K} \alpha_k = 1$. $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_K]$ are the distribution parameters with potentially component-specific parameters [Qui06]. Mixture distributions are also used for regression tasks, such as in Jordan and Jacobs [JJ94] or Bishop and Svensén [BS04].

2.6 MULTIVARIATE HYPERGEOMETRIC DISTRIBUTION

The hypergeometric distribution is applied in various areas of science, such as social and computer science and biology [Sut+23a], e. g., modeling gene mutations, recommender systems, and analyzing social networks [Bec+11; Cas+16b; Las+07; Lod+18; Lod+15; OMT03; PCT06]. It can also be used as a modeling assumption in more theoretical work, e. g. in submodular maximization [FHK17; Har+19], k-means clustering variants [CPM18], or random permutation graphs [BM17].

The hypergeometric distribution is a discrete probability distribution that describes the probability of n_1 successes in n draws without replacement from a finite population of size $(m_1 + m_2)$ with m_1 elements that are part of the success class. Unlike the binomial distribution, which describes the probability distribution of n_1 successes in n draws with replacement.

Definition 2.6.1 (Hypergeometric Distribution [Gon36][2]).
A random variable N_1 follows the hypergeometric distribution if its probability mass function is given by

$$P(N_1 = n_1) = p_{N_1}(n_1) = \frac{\binom{m_1}{n_1}\binom{m_2}{n-n_1}}{\binom{m_1+m_2}{n}} \tag{2.76}$$

Urn models are typical examples of hypergeometric probability distributions. Suppose we think of an urn with marbles in two different colors, e. g., green and purple. We can label the drawing of a green marble as a success. Then $(m_1 + m_2)$ defines the total number of marbles and m_1 the number of green marbles in the urn. n_1 is the number of green marbles, and $n - n_1$ is the number of drawn purple marbles.

The multivariate hypergeometric distribution describes an urn with more than two colors, e. g., green, purple, and yellow, in the simplest case, with three colors. As described by Johnson [Joh87], the definition is given by:

Definition 2.6.2 (Multivariate Hypergeometric Distribution).
A random vector $\boldsymbol{N}$ follows the multivariate hypergeometric distribution, if its joint probability mass function is given by

$$P(\boldsymbol{N} = \boldsymbol{n}) = p_{\boldsymbol{N}}(\boldsymbol{n}) = \frac{\prod_{k=1}^{K}\binom{m_k}{n_k}}{\binom{\sum_k m_k}{n}} \tag{2.77}$$

where $K \in \mathbb{N}$ is the number of different classes (e. g. marble colors in the urn), $\boldsymbol{m} = [m_1, \ldots, m_K] \in \mathbb{N}^K$ describes the number of elements per class (e.g. marbles per color), $\sum_{k=1}^{K} m_k$ is the total number of elements (e. g. all marbles in the urn) and $n \in \{0, \ldots, \sum_k m_k\}$ is the number of elements (e. g. marbles) to draw.

The support S of the probability mass function is given by

$$S = \left\{ \boldsymbol{n} \in \mathbb{N}_0^K : \forall k \quad n_k \leq m_i, \sum_{k=1}^{K} n_k = n \right\} \tag{2.78}$$

Every marble is picked with equal probability under the *central* hypergeometric distribution. The number of selected elements per class is then proportional to the ratio between the number of elements per class and the total number of elements in the urn. This assumption is often too restrictive, and we want an additional modeling parameter for the importance of a class. We call generalizations, which make certain classes more likely to

[2] Although the distribution itself is older, Gonin [Gon36] were the first to name it hypergeometric distribution

be picked, *noncentral* hypergeometric distributions. In the remaining part of this dissertation, we refer to the noncentral hypergeometric distribution even if we write hypergeometric distribution.

In the literature, Fisher's [Fis35] and Wallenius' [Che76; Wal63] distribution are two different versions of the noncentral hypergeometric distribution. This thesis refers to Fisher's version of the noncentral hypergeometric distribution.

Definition 2.6.3 (Multivariate Fisher's Noncentral Hypergeometric Distribution [Fis35]).
A random vector $\boldsymbol{N}$ follows Fisher's noncentral multivariate distribution, if its joint probability mass function is given by

$$P(\boldsymbol{N} = \boldsymbol{n}; \boldsymbol{\omega}) = p_{\boldsymbol{N}}(\boldsymbol{n}; \boldsymbol{\omega}) = \frac{1}{P_0} \prod_{k=1}^{K} \binom{m_k}{n_k} \omega_k^{n_k} \tag{2.79}$$

$$\textit{where} \quad P_0 = \sum_{\boldsymbol{y} \in S} \prod_{k=1}^{K} \binom{m_k}{y_k} \omega_k^{y_k} \tag{2.80}$$

The support S of the noncentral hypergeometric distribution is independent of the group importance $\boldsymbol{\omega}$. The total number of samples per class $\boldsymbol{m}$, the number of samples to draw n, and the class importance $\boldsymbol{\omega}$ parameterize the multivariate distribution. Throughout this thesis, we assume $\boldsymbol{m}$ and n constant per experiment. In our applications of the hypergeometric distributions, the group importance $\boldsymbol{\omega}$ is the unknown parameter of interest. Consequently, we use $\boldsymbol{\omega}$ as the distribution parameter in Equation (2.79) and this thesis.

3

MODELING GROUP STRUCTURE USING A JOINT LATENT SPACE

In this chapter, we describe different implementations of scalable multimodal VAEs. We start by explaining the need for methods that are scalable in the number of modalities and why the definition of the joint posterior approximation is an essential building block for these models. We evaluate three different methods according to the desiderata of multimodal methods introduced in Section 2.3. Toward the end of this chapter, we discuss and explain why none of the introduced models can fulfill the complete set of desiderata.

We extend and generalize existing work in scalable multimodal generative models. We use the definition of scalability as we defined it in Section 2.3.1, i. e. scalable in the sense that a single model approximates the joint distribution over all modalities (including all marginal and conditional distributions) instead of requiring individual models for every subset of modalities [e.g., HG18; Hua+18; TE19]. The latter approach requires a prohibitive number of models, exponential in the number of modalities.

multimodal vaes Among scalable multimodal generative models, multimodal VAEs [KGS19; Shi+20; Shi+19; SDV20; SNM16; Tsa+19; Ved+18; WG18] have recently been the dominant approach for learning a joint distribution of multiple modalities. They are suitable for learning a joint distribution over multiple modalities and enable joint inference given a subset of modalities. However, to efficiently approximate the joint posterior for all subsets of modalities, it is required to introduce additional assumptions on the form of the joint posterior. Previous work relies on either the Product of Experts [KGS19; WG18] or the Mixture of Experts [Shi+20; Shi+19] distribution to combine the unimodal posteriors and overcome the scalability issue. This chapter describes a method to unite these approaches in a generalized formulation – a mixture of products joint posterior – that encapsulates both approaches and combines their benefits [SDV21].

multimodal posteriors The MVAE [WG18] assumes that the joint posterior is a product of unimodal posteriors - a product of experts [PoE, Hin99; Hin02]. The PoE has the benefit of aggregating information across any subset of unimodal posteriors. It, therefore, provides an efficient way of dealing with missing modalities for specific types of unimodal posteriors (e.g., Gaussians). However, to handle missing modalities, the MVAE relies on an additional sub-sampling of unimodal log-likelihoods, which no longer guarantees a valid lower bound on the joint log-likelihood [WG19]. Previous work provides empirical results that exhibit the shortcomings of the MVAE, attributing them to a precision miscalibration of experts [Shi+19] or the averaging over inseparable individual beliefs [KGS19]. In Sutter, Daunhawer, and Vogt [SDV21], we can show that the PoE works well in practice if it is applied to all subsets of modalities, which naturally leads to the proposed mixture of products of experts (MoPoE) generalization, which yields a valid lower bound on the joint log-likelihood. In this thesis, we define PoE-VAE as a multimodal VAE, which uses a product of experts aggregation function for the joint multimodal posterior approximation distribution but no additional loss terms.

On the other hand, the MMVAE [Shi+19] assumes that the joint posterior approximation is a mixture of unimodal posteriors – a mixture of experts (MoE). The MMVAE is suitable for the approximation of unimodal posteriors and translation between pairs of modalities. However, it cannot take advantage of multiple modalities because it only considers the unimodal posteriors during training. In contrast, the proposed MoPoE-VAE computes the joint posterior for all subsets of modalities, enabling efficient many-to-many translations.

While multiple extensions of multimodal VAEs [Dau+20; KGS19; Shi+20; SDV20] have introduced additional loss terms, these are mostly independent of the used aggregation function. This chapter compares the three probabilistic aggregation functions, PoE, MoE, and MoPoE, and their application to multimodal VAEs. Therefore, we do not consider any additional objectives in our experimental setup.

3.1 SCALABLE MULTIMODAL LEARNING

Following Definition 2.3.5 of the multimodal ELBO, the particular case of scalable multimodal learning is not apparent. It is only apparent in having access to multiple data types and the desiderata, which follow from there as described in Section 2.3.1.

Definition 3.1.1 (Scalable Multimodal VAEs)**.**
A multimodal VAE is scalable if the number of encoder and decoder networks scales linearly with the number of modalities M.

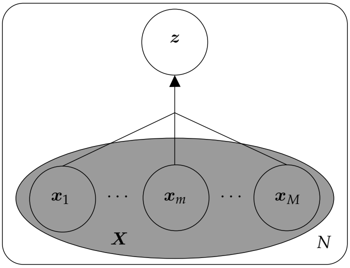


FIGURE 3.1: Graphical model for scalable inference in the multimodal setting. Compared to the graphical model in Figure 2.1b, the unimodal samples $\boldsymbol{x}_m \in \boldsymbol{X}$ are aggregated outside of the grey area representing the multimodal set $\boldsymbol{X}$. The aggregation represents the late fusion step of the unimodal experts $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$ into the joint posterior $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$ approximation using the scalable function f_{agg} (see Definitions 3.1.1 and 3.1.2).

Following Definitions 2.3.5 and 3.1.1, the focus in this section is on achieving scalability in multimodal VAEs. The methods that we discuss in this chapter differ in their definition of the joint posterior approximation distribution $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$. They rely on defining the joint posterior approximation $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$ as a non-parametric function f_{agg} of the unimodal approximations $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$ [Shi+19; SDV21; WG18].

Definition 3.1.2 (Scalable Posterior Approximation).
A joint posterior approximation distribution $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$ is scalable if it is a function f_{agg} of unimodal posterior approximations $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$

$$q_\Phi(\boldsymbol{z} \mid \boldsymbol{X}) = f_{agg}(q_{\phi_1}(\boldsymbol{z} \mid \boldsymbol{x}_1), \ldots, q_{\phi_M}(\boldsymbol{z} \mid \boldsymbol{x}_M)) \tag{3.1}$$

where the amortization parameters of the inference networks are given as $\Phi = [\phi_1, \ldots, \phi_M]$.

In Definition 3.1.2, there are no learnable parameters for the function f_{agg} and, therefore, the amortization parameters of the inference function are just a concatenation of the unimodal posterior amortization parameters, i.e. $\Phi = [\phi_1, \ldots, \phi_M]$. Definition 3.1.2 implies that every modality $\boldsymbol{x}_m$ is encoded individually using its variational posterior $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$. The joint posterior approximation $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$ follows from a late fusion [e.g., BAM18] of the unimodal posterior approximations $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$ using the function f_{agg}. Scalable aggregation functions f_{agg} follow the concept of abstract mean functions [NP05]. Abstract mean functions unify a family of mean function $\mathcal{F}_{agg}$

$$\mathcal{F}_{agg}(\boldsymbol{p}) = f{-1}\left(\frac{1}{M}\sum_{m=1}^{M} f(p_m)\right), \tag{3.2}$$

where M is the number of elements to be aggregated. The function f needs to be injective for f^{-1} to exist. For $f(p_m) = ap_m + b$ we receive the arithmetic mean, i. e. MoE, and for $f(p_m) = \log p_m$ the geometric mean, i. e. PoE [Nie19].

Definition 3.1.3 (Valid Joint Posterior approximation).
A joint posterior distribution $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$ following Definition 3.1.2 is a valid variational distribution if the function f_{agg} satisfies

$$\int_{\boldsymbol{z}} f_{agg}\left(q_{\phi_1}(\boldsymbol{z} \mid \boldsymbol{x}_1), \ldots, q_{\phi_M}(\boldsymbol{z} \mid \boldsymbol{x}_M)\right) d\boldsymbol{z} = 1 \tag{3.3}$$

3.2 PROBABILISTIC AGGREGATION FUNCTIONS FOR MULTIMODAL VAES

In Section 3.1, we introduced the concept of scalable multimodal VAEs. We introduced a general class of scalable multimodal VAEs, which use a function f_{agg} to combine unimodal posterior approximations $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$ into a joint posterior approximation $q_\Phi(\boldsymbol{z} \mid \boldsymbol{X})$. This section introduces different implementations of the function f_{agg}.

3.2.1 *Product of Experts*

Wu and Goodman [WG18] define the joint posterior approximation as a product of experts [PoE, Hin99; Hin02]. A PoE joint posterior approximation is given by

$$q_\Phi(\boldsymbol{z} \mid \boldsymbol{X}) = \frac{1}{Z}\prod_{m=1}^{M} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) \tag{3.4}$$

where Z is normalization constant such that Definition 3.1.3 is fulfilled and Equation (3.4) defines a probability distribution, i. e. $\int q(\boldsymbol{z} \mid \boldsymbol{X}) d\boldsymbol{z} = 1$.

No general closed-form solution exists for Equation (3.4). There is an analytical solution for Equation (3.4) for all unimodal experts being Gaussian distributions, i.e. $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}(\boldsymbol{x}_m), \Sigma(\boldsymbol{x}_m))$. The product distribution is itself proportional to a Gaussian distribution $\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_{\text{PoE}}, \Sigma_{\text{PoE}})$, where $\boldsymbol{\mu}_{\text{PoE}}$ and Σ_{PoE} are given as [PP008]

$$\Sigma_{\text{PoE}} = \sum_i \Sigma_i^{-1} \tag{3.5}$$

$$\boldsymbol{\mu}_{\text{PoE}} = \Sigma_{\text{PoE}}^{-1} \sum_i \Sigma_i^{-1} \boldsymbol{\mu}_i \tag{3.6}$$

for $\boldsymbol{\mu}_i$ and Σ_i being the distribution parameters of the individual experts. Murphy [Mur07] showed that the product of Gaussians equals a Gaussian distribution after it is normalized to integrate to one. Hence, using

$$\int_{\boldsymbol{z}} \prod_{m=1}^{M} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) = Z \tag{3.7}$$

leads to a properly normalized Gaussian distribution, i.e.

$$\frac{1}{Z} \prod_{m=1}^{M} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) = \mathcal{N}_{\Phi}(\boldsymbol{z}; \boldsymbol{\mu}_{\text{PoE}}(\boldsymbol{X}), \Sigma_{\text{PoE}}(\boldsymbol{X})) \tag{3.8}$$

Accordingly, we write the joint posterior approximation $q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X})$ as

$$q_{\Phi,\text{PoE}}(\boldsymbol{z} \mid \boldsymbol{X}) = \mathcal{N}_{\Phi}(\boldsymbol{z}; \boldsymbol{\mu}_{\text{PoE}}(\boldsymbol{X}), \Sigma_{\text{PoE}}(\boldsymbol{X})) \tag{3.9}$$

3.2.1.1 *Multimodal PoE ELBO*

Defining the joint posterior approximation distribution using a PoE leads to the following ELBO formulation $\mathcal{L}_{\text{PoE}}(\Theta, \Phi; \boldsymbol{X})$

$$\begin{aligned}
\mathcal{L}_{\text{PoE}}(\Theta, \Phi; \boldsymbol{X}) &= \mathbb{E}_{q_{\Phi,\text{PoE}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\Theta}(\boldsymbol{X} \mid \boldsymbol{z}) - \log \frac{q_{\Phi,\text{PoE}}(\boldsymbol{z} \mid \boldsymbol{X})}{p_{\Theta}(\boldsymbol{z})}\right] \\
&= \sum_{m=1}^{M} \mathbb{E}_{q_{\Phi,\text{PoE}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\
&\quad - D_{\text{KL}}\left(q_{\Phi,\text{PoE}}(\boldsymbol{z} \mid \boldsymbol{X}) \,||\, p_{\Theta}(\boldsymbol{z})\right) &(3.10)\\
&= \sum_{m=1}^{M} \mathbb{E}_{q_{\Phi,\text{PoE}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\
&\quad - D_{\text{KL}}\left[\mathcal{N}_{\Phi}(\boldsymbol{z}; \boldsymbol{\mu}_{\text{PoE}}(\boldsymbol{X}), \Sigma_{\text{PoE}}(\boldsymbol{X})) \,||\, p_{\Theta}(\boldsymbol{z})\right] &(3.11)
\end{aligned}$$

One of the advantages of the PoE ELBO $\mathcal{L}_{\text{PoE}}$ formulation is the calculation of the KL divergence term. The KL divergence between two Gaussian distributions can be calculated in closed form, which makes $\mathcal{L}_{\text{PoE}}$ an efficient objective.

3.2.2 *Mixture of Experts*

Shi et al. [Shi+19] define the joint posterior approximation as a mixture of experts distribution [MoE, Lin95] where the mixture components are the unimodal posterior approximations

$$q_{\Phi,\text{MoE}}(\boldsymbol{z} \mid \boldsymbol{X}) = \sum_{m \in \mathbb{M}} \alpha_m q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) \tag{3.12}$$

$\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_M] \in \mathbb{R}^M$ are the mixture weights such that $0 \leq \alpha_m \leq 1 \; \forall \; m$ and $\sum_{m=1}^{M} \alpha_m = 1$. If not stated differently, we assume equal weights α_m for all experts in the mixture distribution, i. e. $\alpha_m = \frac{1}{M}, \; \forall \; m$. It follows

$$q_{\Phi,\text{MoE}}(\boldsymbol{z} \mid \boldsymbol{X}) = \frac{1}{M} \sum_{m \in \mathbb{M}} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) \tag{3.13}$$

3.2.2.1 *Multimodal MoE ELBO*

Defining the joint posterior approximation distribution as a MoE leads to the following ELBO

$$\begin{aligned} \mathcal{L}_{\text{MoE}}(\Theta, \Phi; \boldsymbol{X}) &= \mathbb{E}_{q_{\Phi,\text{MoE}}(\boldsymbol{z}|\boldsymbol{X})} \left[\log p_\Theta(\boldsymbol{X} \mid \boldsymbol{z}) - \log \frac{q_{\Phi,\text{MoE}}(\boldsymbol{z} \mid \boldsymbol{X})}{p_\Theta(\boldsymbol{z})} \right] \\ &= \sum_{m=1}^{M} \mathbb{E}_{q_{\Phi,\text{MoE}}(\boldsymbol{z}|\boldsymbol{X})} \left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}) \right] \\ &\quad - D_{\text{KL}} \left[q_{\Phi,\text{MoE}}(\boldsymbol{z} \mid \boldsymbol{X}) \mid\mid p_\Theta(\boldsymbol{z}) \right] \end{aligned} \tag{3.14}$$

Unlike the PoE distribution, the KL divergence between the MoE posterior approximation and the prior distribution cannot be calculated exactly in closed form. This is not only impossible for the general case but also for the particular case of both unimodal posterior approximations and prior distribution being Gaussian distributions.

Using Jensen's inequality [Jen06] and the fact that the KL divergence is a convex function [see e. g. CT06], we can upper bound the KL divergence of a convex sum by a sum of KL divergences

$$D_{\text{KL}} \left[\frac{1}{M} \sum_{m=1}^{M} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) \mid\mid p_\Theta(\boldsymbol{z}) \right] \leq \frac{1}{M} \sum_{m=1}^{M} D_{\text{KL}} \left[q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) \mid\mid p_\Theta(\boldsymbol{z}) \right] \tag{3.15}$$

In the case of Gaussian distributions for unimodal posterior approximations $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$ and prior distribution $p_\Theta(\boldsymbol{z})$, the sum of KL divergences offers again a closed form solution. We write the ELBO $\mathcal{L}_{\text{MoE}}(\Theta, \Phi; \boldsymbol{X})$ accordingly

$$\begin{aligned}\mathcal{L}_{\text{MoE}}(\Theta, \Phi; \boldsymbol{X}) = & \sum_{m=1}^{M} \mathbb{E}_{q_{\Phi,\text{MoE}}(\boldsymbol{z}|\boldsymbol{X})} \left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\ & - D_{\text{KL}} \left(q_{\Phi,\text{MoE}}(\boldsymbol{z} \mid \boldsymbol{X}) \mid\mid p_{\Theta}(\boldsymbol{z})\right) \\ \geq & \sum_{m=1}^{M} \mathbb{E}_{q_{\Phi,\text{MoE}}(\boldsymbol{z}|\boldsymbol{X})} \left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\ & - \frac{1}{M} \sum_{m=1}^{M} D_{\text{KL}} \left(q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) \mid\mid p_{\Theta}(\boldsymbol{z})\right) \end{aligned} \tag{3.16}$$

The prize for being able to optimize a closed form divergence measure is the lower bound that is optimized compared to the original ELBO $\mathcal{L}_{\text{MoE}}(\Theta, \Phi; \boldsymbol{X})$.

3.2.3 *Mixture of Products of Experts*

In Sutter, Daunhawer, and Vogt [SDV21], we introduced the multimodal mixture of products of experts (MoPoE) distribution. It is also a mixture distribution, but different to Shi et al. [Shi+19], the set of experts does not only contain the unimodal posterior approximations. MoPoE instead includes all subsets of modalities in the set of experts for the mixture distribution, i. e. the set of experts is the powerset $\mathcal{P}(\mathbb{M})$ without the empty set $\emptyset$. For ease of notation, we, by default, exclude the empty set $\emptyset$ when we talk about the powerset $\mathcal{P}(\mathbb{M})$ of a set of modalities $\mathbb{M}$. Therefore,

$$\mathcal{P}(\mathbb{M}) = \{A : A \subseteq \mathbb{M} \setminus \emptyset\} \tag{3.17}$$

The MoPoE distribution is given by

$$q_{\Phi,\text{MoPoE}}(\boldsymbol{z} \mid \boldsymbol{X}) = \frac{1}{|\mathcal{P}(\mathbb{M})|} \sum_{A \subseteq \mathbb{M}} q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A) \tag{3.18}$$

As the name suggests, the posterior approximation given any input subset $\boldsymbol{X}_A$ follows a PoE [Hin99; Hin02]

$$q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A) \propto \prod_{m \in A} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) \tag{3.19}$$

The variational amortization parameters for every subset Φ_A are given by the unimodal parameters ϕ_m of all modalities $m \in A$:

$$\Phi_A = \{\phi_m : m \in A\} \tag{3.20}$$

Despite iterating over all subsets $A \subseteq \mathbb{M}$, it follows from Equation (3.20) and definition 3.1.2 that the total number of variational parameters $|\Phi|$ is equivalent to the sum of the unimodal variational parameters ϕ_m, i. e.

$$|\Phi_{\text{MoPoE}}| = \sum_{m \in \mathbb{M}} |\phi_m| \tag{3.21}$$

It is clear from Equations (3.20) and (3.21) that all probabilistic aggregation functions discussed so far, i. e. MoE, PoE, and MoPoE, are non-parametric functions. They are all a function of the unimodal variational posteriors and their learnable parameters. Hence, the functions are instances of *late fusion* probabilistic aggregation methods.

3.2.3.1 *Multimodal MoPoE ELBO*

Defining the joint posterior distribution as MoPoE distribution leads to the following ELBO

$$\begin{aligned}\mathcal{L}_{\text{MoPoE}}(\Theta, \Phi; \boldsymbol{X}) =& \mathbb{E}_{q_{\Phi,\text{MoPoE}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\Theta}(\boldsymbol{X} \mid \boldsymbol{z}) - \log \frac{q_{\Phi,\text{MoPoE}}(\boldsymbol{z} \mid \boldsymbol{X})}{p_{\Theta}(\boldsymbol{z})}\right] \\ =& \sum_{m=1}^{M} \mathbb{E}_{q_{\Phi,\text{MoPoE}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\ & - D_{\text{KL}}\left[q_{\Phi,\text{MoPoE}}(\boldsymbol{z} \mid \boldsymbol{X}) \mid\mid p_{\Theta}(\boldsymbol{x}_m)\right]\end{aligned} \tag{3.22}$$

We can again use the upper bound to the KL divergence of a mixture distribution (see Equation (3.15)), leading to a looser ELBO.

$$\begin{aligned}\mathcal{L}_{\text{MoPoE}}(\Theta, \Phi; \boldsymbol{X}) \geq & \sum_{m=1}^{M} \mathbb{E}_{q_{\Phi,\text{MoPoE}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\ & - \frac{1}{|\mathcal{P}(\mathbb{M})|} \sum_{A \subseteq \mathbb{M}} D_{\text{KL}}\left[q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}_A) \mid\mid p_{\Theta}(\boldsymbol{z})\right]\end{aligned} \tag{3.23}$$

3.2.4 *A Generalized Formulation for Probabilistic Aggregation Functions*

The MoPoE formulation can be further generalized. In Section 3.2.3, we define the mixture distribution over the powerset of possible subsets of modalities $\mathcal{P}(\mathbb{M})$, i. e.

$$q_{\Phi,\text{MoPoE}}(\boldsymbol{z} \mid \boldsymbol{X}) = \frac{1}{|\mathcal{P}(\mathbb{M})|} \sum_{A \subseteq \mathbb{M}} q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A) \tag{3.24}$$

Instead of the powerset $\mathcal{P}(\mathbb{M})$, any set of subsets $\mathbb{A} \subseteq \mathcal{P}(\mathbb{M})$ is eligible to define a mixture distribution.

Definition 3.2.1 (Generalized Multimodal Posterior Approximation [SDV21]). *Given a set of subsets $\mathbb{A}$ where $\mathbb{A} \subseteq \mathcal{P}(\mathbb{M})$, the generalized mixture of products of experts joint posterior approximation is given as*

$$q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathbb{A}) = \frac{1}{|\mathbb{A}|} \sum_{A \in \mathbb{A}} q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A) \tag{3.25}$$

where

$$q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A) \propto \prod_{m \in A} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) \tag{3.26}$$

Using Definition 3.2.1, we derive the following generalized ELBO

$$\begin{aligned} \mathcal{L}(\Theta, \Phi; \boldsymbol{X}, \mathbb{A}) \geq & \sum_{m=1}^{M} \mathbb{E}_{q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathbb{A})} \left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\ & - D_{\mathrm{KL}} \left(q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathbb{A}) \mid\mid p_{\Theta}(\boldsymbol{x}_m \mid \boldsymbol{z})\right) \end{aligned} \tag{3.27}$$

The slightly modified $\mathcal{L}(\Theta, \Phi; \boldsymbol{X}, \mathbb{A})$ and $q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathbb{A})$ highlight their dependency on the set of subsets $\mathbb{A}$. We are now able to rewrite the three methods based on PoE (Section 3.2.1), MoE (Section 3.2.2), and MoPoE (Section 3.2.3) as special cases of the general scalable multimodal formulation in Definition 3.2.1. We use the set of subsets $\mathbb{A}$ in the notation of the ELBO $\mathcal{L}(\Theta, \Phi; \boldsymbol{X}, \mathbb{A})$ to distinguish between the three aggregation methods discussed. In the following, we formalize this finding in three corollaries. We start with the PoE (Corollary 3.2.1), then the MoE (Corollary 3.2.2), and last the MoPoE (Corollary 3.2.3).

Corollary 3.2.1. *For $\mathbb{A} = \{\mathbb{M}\}$ in Definition 3.2.1, the resulting joint posterior approximation is equal to the PoE formulation introduced in Section 3.2.1 [WG18].*

Proof.

$$\begin{aligned} q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathbb{A}) &= q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \{\mathbb{M}\}) && (3.28) \\ &= \frac{1}{|\{\mathbb{M}\}|} \sum_{A \in \{\mathbb{M}\}} q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A) && (3.29) \\ &= q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}) && (3.30) \\ &\propto \prod_{m \in \mathbb{M}} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) && (3.31) \end{aligned}$$

$$= q_{\Phi,\text{PoE}}(\boldsymbol{z} \mid \boldsymbol{X}) \tag{3.32}$$

The ELBO $\mathcal{L}(\Theta, \Phi; \boldsymbol{X}, \{\mathbb{M}\})$ is already derived in Section 3.2.1.1. □

Corollary 3.2.2. *For $\mathbb{A} = \mathbb{M}$ in Definition 3.2.1, the resulting joint posterior approximation is equal to the MoE formulation introduced in Section 3.2.2 [Shi+19].*

Proof.

$$\begin{aligned} q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathbb{A}) &= q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathbb{M}) && (3.33)\\ &= \frac{1}{|\mathbb{M}|} \sum_{A \in \mathbb{M}} q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A) && (3.34)\\ &= \frac{1}{M} \sum_{m \in \mathbb{M}} q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m) && (3.35)\\ &= q_{\Phi,\text{MoE}}(\boldsymbol{z} \mid \boldsymbol{X}) && (3.36) \end{aligned}$$

The ELBO $\mathcal{L}(\Theta, \Phi; \boldsymbol{X}, \mathbb{M})$ is already derived in Section 3.2.2.1. □

Corollary 3.2.3. *For $\mathbb{A} = \mathcal{P}(\mathbb{M})$ in Definition 3.2.1, the resulting joint posterior approximation is equal to the MoPoE formulation introduced in Section 3.2.3 [SV21].*

Proof.

$$\begin{aligned} q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathbb{A}) &= q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}; \mathcal{P}(\mathbb{M})) && (3.37)\\ &= \frac{1}{|\mathcal{P}(\mathbb{M})|} \sum_{A \in \mathcal{P}(\mathbb{M})} q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A) && (3.38)\\ &= q_{\Phi,\text{MoPoE}}(\boldsymbol{z} \mid \boldsymbol{X}) && (3.39) \end{aligned}$$

The ELBO $\mathcal{L}(\Theta, \Phi; \boldsymbol{X}, \mathcal{P}(\mathbb{M}))$ is already derived in Section 3.2.3.1. □

3.3 EXPERIMENTS & RESULTS

The following section describes the experiments, datasets, and results we used to evaluate multimodal methods. We designed the experiments such that we can assess the methods' ability to fulfill the desiderata for multimodal learning described in Section 2.3. Figure 3.2 depicts the basic architecture for the methods used in these experiments. As already described in Section 3.2, they only differ in how they fuse the unimodal posterior approximations $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$ to a joint posterior approximation $q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X})$, which is depicted abstractly in the $f_{agg}(\cdot)$ building block in Figure 3.2

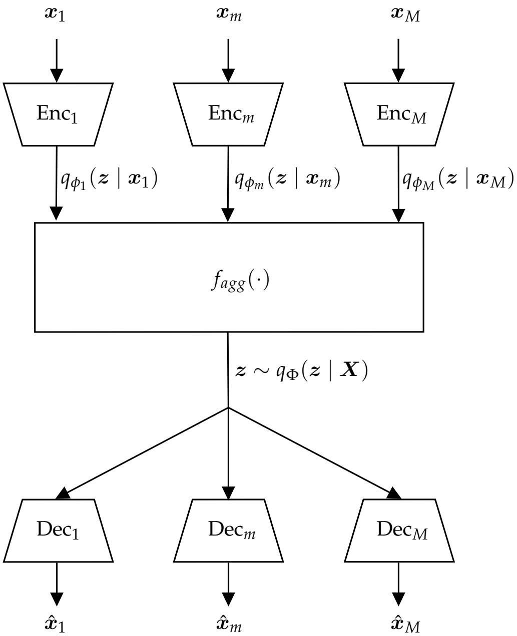


FIGURE 3.2: The basic architecture of all methods is described in Section 3.2. They only differ in the aggregation function building block f_{agg}. Every modality separately infers $q_{\phi_m}(z \mid x_m)$. A scalable f_{agg} (see Section 3.2) generates $q_\Phi(z \mid X)$ using the unimodal posterior approximations $q_{\phi_m}(z \mid x_m)$.

3.3.1 *Datasets*

To evaluate the methods proposed in Section 3.2, we perform experiments on two different datasets, *MNIST-SVHN-Text* and *PolyMNIST* [SV21].

3.3.1.1 *MNIST-SVHN-Text*

MNIST-SVHN-Text (MST) is a multimodal combination of the MNIST digit dataset [LeC+98], the SVHN digit dataset [Net+11], and a text modality. We use the MST dataset to highlight the ability of the different methods to adapt to modalities of different difficulties. See Figure 3.3 for MST samples of the digits 0 to 9.

The MNIST modality consists of white handwritten digits on a black background of size 28×28. It consists of $60'000$ training images and $10'000$

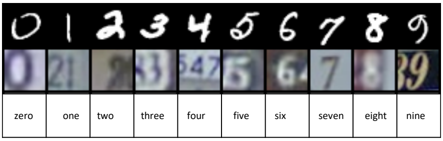


Figure 3.3: Samples from the *MNIST-SVHN-Text* dataset. Every column is an example of $\boldsymbol{X} = \{\boldsymbol{x}_M, \boldsymbol{x}_S, \boldsymbol{x}_T\}$, where the weak supervision comes from the alignment of the multimodal sets concerning their digit information. $\boldsymbol{x}_M$ denotes the MNIST modality, $\boldsymbol{x}_S$ SVHN, and $\boldsymbol{x}_T$ the text modality. We see the increased difficulty of the SVHN digit dataset in the exemplary multi-digit samples of digits 1 and 4.

test images. Every image displays a digit between 0 and 9. It contains writing samples from approximately 250 different writers [LeC+98].

The *Street View House Numbers* modality [SVHN, Net+11] is a real-world image modality displaying digits in natural scenes. It consists of 73′257 training samples and 26′032 test samples. For our purpose, we use the dataset version with cropped images, where a square of size 32 × 32 is cut around the digit to be approximately in the center of the image. Note that this does not necessarily mean only a single digit is visible in the image. The image's label, though, corresponds to the center digit (see Figure 3.3). Given the real-world scenario of these images and the background, which stems from a natural scene, the SVHN dataset is more challenging than MNIST. Shi et al. [Shi+19] already designed a bimodal dataset consisting of MNIST and SVHN samples. We add a text modality to this dataset to have a non-image modality and to show how different multimodal methods perform in settings with more than two modalities. The text modality is the digit written as a word in English, e. g. "five" for the label 5. To have an additional difficulty level, we randomly shift the start point of the digit string. The length of the string per sample is set to 8. Following Shi et al. [Shi+19], we also pair every MNIST image with 20 SVHN images which increases the dataset size by a factor of 20 to a total of 1′200′000 training tuples and 200′000 test tuples.

3.3.1.2 *PolyMNIST*

PolyMNIST is also based on the MNIST digit dataset [LeC+98]. Different from the MST dataset, it only consists of image-based modalities. This dataset compares the different methods in Section 3.2 according to their ability to adapt to more modalities. In PolyMNIST, the modalities differ at

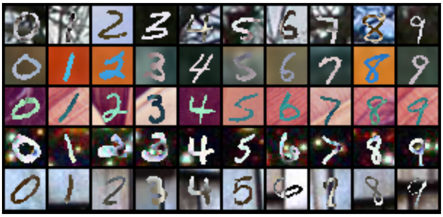


FIGURE 3.4: Sample from the *PolyMNIST* dataset.

first by their background. Every MNIST image $\boldsymbol{x}_m \in [0,1]^{28\times 28}$ of modality m is fused with a background image $\boldsymbol{BG}_m \in [0,255]^{h\times w\times 3}$ where $h \gg 28$ and $w \gg 28$. We take a random 28×28 patch from $\boldsymbol{BG}_m$ as a background for modality m (see Appendix A.2.1 for the source information on the background images). Additionally, we binarize the MNIST image and invert the binarized version of the digit concerning the background color. Equal to the experiments in this section, Figure 3.4 shows an example with $M = 5$ modalities.

To compare the three methods introduced in Section 3.2, we evaluate their ability to infer meaningful latent representations and generate expressive samples that reflect the distribution of the dataset $\mathbb{X}$. We ablate all measures over the different possible input modalities, i. e. for $M = 5$, and evaluate the methods based on their averaged performance for $|A| \in \{1,2,3,4,5\}$. We want the methods to benefit from having access to multiple modalities. Hence, their performance measures should benefit from an increasing number of input modalities.

3.3.2 *Implementation & Training*

In this subsection, we describe the network architectures for the encoders and decoders of the multimodal VAEs and the training procedure. We first report the hyperparameter settings, which are invariant for both the MST and the PolyMNIST datasets. Afterwards, we detail the dataset-specific settings in Sections 3.3.2.1 and 3.3.2.2.

All unimodal posterior approximations are assumed to be Gaussian distributed $\mathcal{N}(\boldsymbol{\mu}_m(\boldsymbol{x}_m), diag(\boldsymbol{\sigma}^2_m(\boldsymbol{x}_m)))$, as well as the prior distribution $p_\theta(\boldsymbol{z})$, which we define as $\mathcal{N}(\boldsymbol{0}, \mathbb{I})$. For all encoders, the last layers map to $\boldsymbol{\mu}_m(\boldsymbol{x}_m)$ and $diag(\sigma^2_m(\boldsymbol{x}_m))$ of the approximate posterior distribution of the respective modalities $\mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_m(\boldsymbol{x}_m), diag(\boldsymbol{\sigma}^2_m(\boldsymbol{x}_m)))$.

The image modalities are modeled with a Laplace likelihood, and the text modality is modeled with a categorical likelihood. The scaling between log-likelihood values $\log p_{\Theta}(\boldsymbol{X} \mid \boldsymbol{z})$ and KL-divergence $D_{\mathrm{KL}}[q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}) \mid\mid p_{\Phi}(\boldsymbol{z})]$ is crucial for making any VAE-based method work. The focus is on the β hyperparameter, which scales the KL divergence [Hig+16]. The same holds for multimodal VAEs. With multimodal datasets, the log-likelihoods $\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$ may have different ranges that make it difficult to optimize them equally well for all modalities $\boldsymbol{x}_m \in \boldsymbol{X}$. Therefore, we introduce additional hyperparameters α_m to weight the log-likelihoods $p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$.

$$\begin{aligned}\mathcal{L}(\Theta, \Phi; \boldsymbol{X}) \propto & \sum_{m=1}^{M} \alpha_m \mathbb{E}_{q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X})} \left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})\right] \\ & - \beta D_{\mathrm{KL}}[q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}) \mid\mid p_{\Theta}(\boldsymbol{z})] \end{aligned} \tag{3.40}$$

We choose the parameters α_m according to the data dimension of every modality $\boldsymbol{x}_m$ such that

$$\alpha_m = \frac{\max_{\mu \in \mathbb{M}} |\boldsymbol{x}_{\mu}|}{|\boldsymbol{x}_m|} \tag{3.41}$$

where $|\boldsymbol{x}_m|$ denotes the dimensionsality of modality $m \in \mathbb{M}$. In mixture distributions, we weight every mixture component with $\frac{1}{|\mathbb{A}|}$ as it is already the case in Section 3.2.

We use the Adam optimizer [KB14] with a starting learning rate of 0.001 for all datasets.

3.3.2.1 *MNIST-SVHN-Text*

Because the MST dataset is based on the MNIST-SVHN dataset [Shi+19], we stick to the network architectures used in Shi et al. [Shi+19] for the MNIST and SVHN modality (see Tables A.1 and A.3). The network architecture used for the text modality is described in Table A.5. All network architectures are described in detail in Appendix A.2.2. Following Equation (3.41), we choose the parameters α_m according to the data dimensionality of every modality $\boldsymbol{x}_m$ where

$$|\boldsymbol{x}_M| = w \times h = 28 \cdot 28 = 784 \tag{3.42}$$

$$|\boldsymbol{x}_S| = w \times h \times c = 32 \cdot 32 \cdot 3 = 3072 \tag{3.43}$$

$$|\boldsymbol{x}_T| = \text{len(string)} = 8 \tag{3.44}$$

We choose the modality $\boldsymbol{x}_m$ with the biggest data dimension, i. e. SVHN, as a reference and set $\alpha_m = 1.0$. From there, we set the following weights

$$\alpha_M = \frac{|\boldsymbol{x}_S|}{|\boldsymbol{x}_M|} = 3.92 \tag{3.45}$$

$$\alpha_S = \frac{|\boldsymbol{x}_S|}{|\boldsymbol{x}_S|} = 1.0 \tag{3.46}$$

$$\alpha_T = \frac{|\boldsymbol{x}_S|}{|\boldsymbol{x}_T|} = 384.0 \tag{3.47}$$

We set $\beta = 5.0$ for all MST experiments and use a latent space of size 20. We use a batch size of 256 for training and train all methods for 50 epochs.

3.3.2.2 *PolyMNIST*

The latent space dimension is 512 for all modalities, models, and runs. All results are based on $\beta = 2.5$. We use the same architectures for all methods and train all models for 300 epochs. Unlike the MST dataset, all modalities are image-based in the PolyMNIST dataset. Hence, we use different initializations of the same network structure for all modalities in the PolyMNIST dataset instead of 5 different architectures. However, no network parameters are shared. The architecture is based on feedforward convolutional neural networks. Table A.7 in Appendix A.2.2 explains the network architecture used for all modalities $\boldsymbol{x}_m$.

3.3.3 *Evaluation & Results*

Tables 3.1, 3.3 and 3.6 show the results of the methods introduced in Section 3.2 for the MST dataset. We see the improved performance of the proposed MoPoE-VAE model [SDV21] compared to previous works, i. e. MoE-VAE [Shi+19] and PoE-VAE [WG18]. Figure 3.5 shows the results for the PolyMNIST dataset. MoPoE-VAE performs well when applied to a heterogeneous dataset such as MST and scales well to many modalities, as in PolyMNIST. It improves the performance with an increasing number of modalities and still performs well if only a single modality is given as input to the model. We describe the results in detail in the following sections. We assess the methods' performance using the latent representation classification and the coherence and quality of generated samples.

	$q_{\phi_M}(\boldsymbol{z} \mid \boldsymbol{x}_M)$	$q_{\phi_S}(\boldsymbol{z} \mid \boldsymbol{x}_S)$	$q_{\phi_T}(\boldsymbol{z} \mid \boldsymbol{x}_T)$
PoE	0.71 ± 0.05	0.10 ± 0.00	1.00 ± 0.01
MoE	0.96 ± 0.00	0.80 ± 0.02	0.97 ± 0.02
MoPoE	0.96 ± 0.00	0.82 ± 0.01	1.00 ± 0.00

(a) Single Modality: $\forall\, A : |A| = 1$

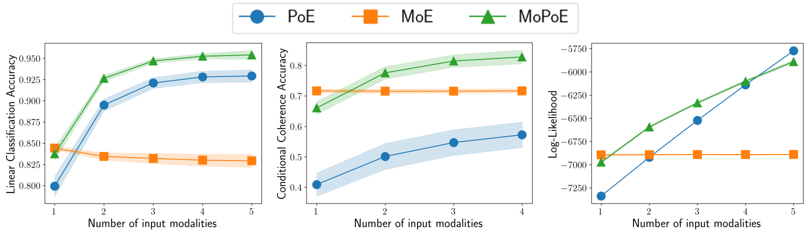
	$q_{\Phi_{M,S}}(\boldsymbol{z} \mid \boldsymbol{X}_{M,S})$	$q_{\Phi_{M,T}}(\boldsymbol{z} \mid \boldsymbol{X}_{M,T})$	$q_{\Phi_{S,T}}(\boldsymbol{z} \mid \boldsymbol{X}_{S,T})$
PoE	0.70 ± 0.05	0.99 ± 0.01	1.00 ± 0.00
MoE	0.88 ± 0.01	0.93 ± 0.04	0.83 ± 0.04
MoPoE	0.97 ± 0.00	0.99 ± 0.01	0.98 ± 0.01

(b) Subsets of two modalities: $\forall\, A : |A| = 2$

	$q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X})$
PoE	0.99 ± 0.01
MoE	0.86 ± 0.03
MoPoE	0.98 ± 0.01

(c) Full set of modalities: $|A| = 3$

Table 3.1: Linear classification accuracy of latent representations on the MNIST-SVHN-Text dataset for models using a single joint latent space. We evaluate the joint representations $\boldsymbol{z} \sim q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A)$ of all subsets $\boldsymbol{X}_A$. The abbreviations of the modalities are $\boldsymbol{x}_M$ for MNIST, $\boldsymbol{x}_S$ for SVHN, and $\boldsymbol{x}_T$ for text. The reported results are the mean and standard deviation of five runs.



(a) Latent Representation (b) Coherence (c) Log-Likelihood

FIGURE 3.5: Results for the PolyMNIST dataset. We compare three methods, PoE-VAE, MoE-VAE, and MoPoE-VAE, regarding their ability to infer meaningful latent representations (fig. 3.5a), generate coherent multimodal samples (fig. 3.5b), and quality of generated samples (fig. 3.5c). For all metrics, we see a similar behavior of the methods: MoE-VAE performs well in case only a single modality is given as input to the model, PoE-VAE benefits from an increasing number of input modalities, and MoPoE-VAE performs well independent of the input subset.

3.3.3.1 *Latent Representation Classification*

Given the goal of self-supervised multimodal learning to infer meaningful latent representations, we are interested in evaluating the learned representations of the three methods proposed in Section 3.2 (see also Section 2.3.4).

We infer the latent representations of N_{train} samples from the original training set using the encoder of a trained VAE model. Using these N_{train} representations, we train a linear classifier to predict the shared information of the multimodal samples. The prediction performance of the classifier is evaluated on the N_{test} encodings of the original test set. We use the performance of the linear classifier as a measure for the quality of the learned latent representations [BCV13; Loc+19]. For all experiments in this section, we set $N_{\text{train}} = 500$ and N_{test} to the size of the test set.

In Table 3.1, we report the performance of the three methods, PoE-VAE, MoE-VAE, and MoPoE-VAE, concerning their learned latent representation on the MST dataset. Given the goal of robustness regarding missing data, we assess the performance of a model by evaluating the latent representations of all possible subsets $\boldsymbol{X}_A, A \in \mathcal{P}(\mathbb{M})$. We see that the PoE-VAE benefits from having access to the complete set of modalities $\boldsymbol{X} = \{\boldsymbol{x}_M, \boldsymbol{x}_S, \boldsymbol{x}_T\}$. Put differently, there is a performance drop if we evaluate the latent representation inferred by a subset of modalities $\boldsymbol{X}_A$. On the other hand, MoE-VAE can achieve high classification accuracy if only a single modality $\boldsymbol{x}_m$ is given as input. Still, it cannot benefit from subsets $\boldsymbol{X}_A$ of more than one modality as input to the model, i. e. $|A| > 1$.

We see the same behavior in Figure 3.5a. Different to the results from Table 3.1, we report the average classification accuracy of all subsets $\boldsymbol{X}_A, A \in \mathcal{P}(\mathbb{M})$ of a given size $|A|$ where $|A| \in \{1, 2, 3, 4, 5\}$. Similar to the MST results, the PolyMNIST results show how PoE-VAE can utilize an increasing number of modalities in inferring its latent representation and how MoE-VAE reaches a stable classification accuracy independent of the input subset size.

MoPoE-VAE benefits from being the generalized version of MoE-VAE and PoE-VAE. It can infer meaningful latent representations independent of the input subset $\boldsymbol{X}_A$. It benefits as much from an increasing number of modalities as PoE-VAE but achieves the same stable performance for a single modality $\boldsymbol{x}_m$ as MoE-VAE (see Table 3.1 and fig. 3.5a).

3.3.3.2 *Generation Coherence*

Unlike unimodal generative models, especially VAEs, generated samples of multimodal VAEs must approximate the empirical data distribution and be coherent (see Section 2.3.5.2). We train a deep classifier for every modality m to evaluate the coherence of generated multimodal samples. The deep classifier has the same architecture as the corresponding encoder of the multimodal VAE. It is trained on the original training set to predict the shared information based on a sample from the original dataset $\boldsymbol{x}_m$. For all modalities $m \in \mathbb{M}$, the trained classifier predicts the shared information for every generated sample of modality m. Coherence is the amount of agreement between the classifiers' predictions. We call a generated sample coherent if the classifiers of all modalities $m \in \mathbb{M}$ predict the same shared information. In the case of randomly generated samples, they only must agree amongst themselves. In the case of conditionally generated samples, they must agree amongst themselves and with the shared information from the input. Coherence measures how well the generated samples follow the multimodal structure. See Section Section 2.3.5.2 for a detailed discussion of coherence in generative multimodal models.

Table 3.3 shows the coherence results for the MST dataset. The table structure is designed such that every subtable reflects the conditional generation of one modality $\boldsymbol{x}_m$ given the remaining subsets $\boldsymbol{X}_A, m \notin A$. The classification accuracy of the generated samples follows the dynamics already seen in the evaluation of the classification accuracy of the latent

	$\boldsymbol{x}_M \sim p_{\theta_M}(\boldsymbol{x}_M \mid \boldsymbol{z})$		
Model	$q_{\phi_S}(\boldsymbol{z} \mid \boldsymbol{x}_S)$	$q_{\phi_T}(\boldsymbol{z} \mid \boldsymbol{x}_T)$	$q_{\Phi_{S,T}}(\boldsymbol{z} \mid \boldsymbol{X}_{S,T})$
PoE	0.10 ± 0.00	0.27 ± 0.10	0.27 ± 0.10
MoE	0.79 ± 0.02	0.99 ± 0.01	0.89 ± 0.01
MoPoE	0.79 ± 0.01	0.99 ± 0.01	0.97 ± 0.01

(a) Conditional Generation: $\boldsymbol{x}_M$

	$\boldsymbol{x}_S \sim p_{\theta_S}(\boldsymbol{x}_S \mid \boldsymbol{z})$		
Model	$q_{\phi_M}(\boldsymbol{z} \mid \boldsymbol{x}_M)$	$q_{\phi_T}(\boldsymbol{z} \mid \boldsymbol{x}_T)$	$q_{\Phi_{M,T}}(\boldsymbol{z} \mid \boldsymbol{X}_{M,T})$
PoE	0.20 ± 0.04	0.36 ± 0.27	0.42 ± 0.31
MoE	0.38 ± 0.04	0.32 ± 0.03	0.35 ± 0.03
MoPoE	0.42 ± 0.05	0.36 ± 0.02	0.40 ± 0.04

(b) Conditional Generation: $\boldsymbol{x}_S$

	$\boldsymbol{x}_T \sim p_{\theta_T}(\boldsymbol{x}_T \mid \boldsymbol{z})$		
Model	$q_{\phi_M}(\boldsymbol{z} \mid \boldsymbol{x}_M)$	$q_{\phi_S}(\boldsymbol{z} \mid \boldsymbol{x}_S)$	$q_{\Phi_{M,S}}(\boldsymbol{z} \mid \boldsymbol{X}_{M,S})$
PoE	0.16 ± 0.01	0.10 ± 0.00	0.19 ± 0.04
MoE	0.96 ± 0.00	0.80 ± 0.02	0.88 ± 0.01
MoPoE	0.96 ± 0.00	0.81 ± 0.01	0.97 ± 0.00

(c) Conditional Generation: $\boldsymbol{x}_T$

Table 3.3: Generation Coherence on the MNIST-SVHN-Text dataset for methods using a single joint latent space. The modality above the second horizontal line is generated based on the subsets below the same line in every subtable. The first row on the right side of every table shows the generative model $p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$ whereas the second line shows the different variational approximations $q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A), m \notin A$. The abbreviations of the different modalities are $\boldsymbol{x}_M$ for MNIST, $\boldsymbol{x}_S$ for SVHN, and $\boldsymbol{x}_T$ for text. The reported results are the mean and standard deviation of five runs.

Model	$\boldsymbol{X} \sim p_\Theta(\boldsymbol{X} \mid \boldsymbol{z})$
PoE	0.24 ± 0.00
MoE	0.33 ± 0.00
MoPoE	0.31 ± 0.00

Table 3.5: Random Generation Coherence on the MNIST-SVHN-Text dataset for methods using a single joint latent space. We condition the generation of multimodal samples $\boldsymbol{X}$ on random latent vectors $\boldsymbol{z} \sim p_\Theta(\boldsymbol{z})$. The reported results are the mean and standard deviation of five runs.

representations. MoE-VAE achieves good performance for the generation of samples conditioned on a single modality $\boldsymbol{x}_m$ but is not able to benefit from having access to more than one modality (most right column in Tables 3.4a to 3.4c). Similar to the evaluation of the latent representation, we also report the average performance over subsets $\boldsymbol{X}_A$ of the same size $|A|$ for assessing the coherence for the PolyMNIST dataset in Figure 3.5b. The evaluation of the PolyMNIST dataset confirms the results from the MST dataset for MoE-VAE.

PoE-VAE shows difficulties generating coherent samples if not conditioned on the complete set of modalities $\boldsymbol{X}$. Similar to the latent representation classification accuracy, the coherence of generated samples improves when the generation is conditioned on two modalities instead of a single modality (see the first row in Tables 3.4a to 3.4c). Again, we confirm the MST dataset results with the results on the PolyMNIST dataset (see Figure 3.5b). The coherence of generated samples by PoE-VAE improves with an increasing number of input modalities but cannot compete with MoE-VAE and MoPoE-VAE.

MoPoE-VAE outperforms the two previous works, PoE-VAE and MoE-VAE, regarding the coherence of generated samples. We see good results in Table 3.3 and Figure 3.5b.

3.3.3.3 *Test Set Log-Likelihood*

To measure the quality of the approximation to the empirical data distribution, we estimate the test set log-likelihoods of modalities $m \in \mathbb{M}$. The test set log-likelihood measures the probability of the samples in the test set given the learned model. Multimodal VAEs optimize the likelihood of a

	$q_{\phi_M}(\boldsymbol{z} \mid \boldsymbol{x}_M)$	$q_{\phi_S}(\boldsymbol{z} \mid \boldsymbol{x}_S)$	$q_{\phi_T}(\boldsymbol{z} \mid \boldsymbol{x}_T)$
PoE	-2127.1 ± 31.6	-1939.5 ± 8.6	-2533.6 ± 483.5
MoE	-1985.7 ± 1.5	-1856.2 ± 10.8	-2018.3 ± 1.7
MoPoE	-1988.4 ± 1.7	-1856.3 ± 3.6	-2021.5 ± 1.8

(a) $\forall A : |A| = 1$ (Single Modality)

	$q_{\Phi_{M,S}}(\boldsymbol{z} \mid \boldsymbol{X}_{M,S})$	$q_{\Phi_{M,T}}(\boldsymbol{z} \mid \boldsymbol{X}_{M,T})$	$q_{\Phi_{S,T}}(\boldsymbol{z} \mid \boldsymbol{X}_{S,T})$
PoE	-1882.4 ± 9.6	-2436.6 ± 472.7	-1884.1 ± 14.5
MoE	-1911.8 ± 6.6	-2001.7 ± 1.4	-1924.5 ± 7.4
MoPoE	-1819.9 ± 2.9	-1983.7 ± 1.8	-1848.2 ± 3.3

(b) $\forall A : |A| = 2$

	$q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X})$
PoE	-1801.4 ± 3.0
MoE	-1940.7 ± 5.7
MoPoE	-1815.5 ± 2.8

(c) $\forall A : |A| = 3$ (Full set of modalities)

Table 3.6: Test set log-likelihoods on MNIST-SVHN-Text. We report the test set log-likelihoods of the joint generative model $p_\Theta(\boldsymbol{X} \mid \boldsymbol{z})$ conditioned on the variational posterior of subsets of modalities $q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A)$. The abbreviations for the different modalities are $\boldsymbol{x}_M$ for MNIST, $\boldsymbol{x}_S$ for SVHN, and $\boldsymbol{x}_T$ for text. We denote the full set of modalities as $\boldsymbol{X} = \{\boldsymbol{x}_M, \boldsymbol{x}_S, \boldsymbol{x}_T\}$.

data sample $\boldsymbol{X}$ given some latent vector $\boldsymbol{z}$. Hence, it is possible to evaluate a trained model based on the achieved test-set log-likelihood.

Table 3.6 reports the test set log-likelihoods for the three methods PoE-VAE, MoE-VAE, and MoPoE-VAE. We report the log-likelihoods based on the posterior approximations of different subsets $\boldsymbol{X}_A$. Similar to the reported results for the latent representation classification and the coherence of generated samples, we also want a high quality of generated samples independent of the available modalities. The behavior of the three methods is again similar for both datasets, MST and PolyMNIST (see Sections 3.3.3.1 and 3.3.3.2). In Table 3.6, MoE-VAE achieves the best log-likelihoods if only a single modality is available. For more than a single input modality, PoE-VAE and MoPoE-VAE outperform MoE-VAE.

MoPoE-VAE improves its log-likelihood numbers with increasing modalities, but not as much as PoE-VAE, achieving the best log-likelihood results given the complete set $\boldsymbol{X}$ as input. Figure 3.5c confirms the results from the MST dataset. We aggregate the achieved log-likelihood numbers from different subsets $\boldsymbol{X}_A$ of the same size $|A|$. We see that also for the PolyMNIST dataset, MoE-VAE achieves the highest log-likelihood if only a single modality is given as input. PoE-VAE and MoPoE-VAE improve their performance for multiple modalities, outperforming MoE-VAE.

The performance of PoE-VAE for some input subsets seems unstable over multiple seeds, which we see in the high standard deviations.

3.3.3.4 *Fidelity Metrics for Image Modalities*

We additionally evaluate the sample quality of image modalities based on fidelity metrics, as discussed in Section 2.3.5.1. Fidelity metrics offer an alternative view of the quality of generated samples compared to test set log-likelihoods. Table 3.8 reports the average precision of the precision-recall metric for generative models [Saj+18] (see also Section 2.3.5.1) for the MST dataset. In Table 3.9a, we report the results for the generation of MNIST samples, and in Table 3.9b for the generation of SVHN samples. In both cases, we evaluate the conditional generation performance of $p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$ based on the posterior approximations $q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A),\ \forall\ A \in \{A : m \notin A\}$. Interestingly, the impact on the performance of PoE-VAE of having access to only a single modality instead of the complete set $\boldsymbol{X}$ is smaller than the impact on the reported test set log-likelihoods. Besides that, PoE-VaE can outperform the other two methods, MoE- and MoPoE-VAE, concerning the visual quality of generated image modalities. Among the latter two, MoPoE-VAE outperforms MoE-VAE, which follows the trend we already

	$\boldsymbol{x}_M \sim p_{\theta_M}(\boldsymbol{x}_M \mid \boldsymbol{z})$			
Model	$q_{\phi_S}(\boldsymbol{z} \mid \boldsymbol{x}_S)$	$q_{\phi_T}(\boldsymbol{z} \mid \boldsymbol{x}_T)$	$q_{\Phi_{S,T}}(\boldsymbol{z} \mid \boldsymbol{X}_{S,T})$	$p_\Theta(\boldsymbol{z})$
PoE	0.58 ± 0.02	0.40 ± 0.04	0.40 ± 0.05	0.58 ± 0.02
MoE	0.17 ± 0.02	0.10 ± 0.02	0.15 ± 0.02	0.32 ± 0.01
MoPoE	0.28 ± 0.02	0.18 ± 0.02	0.13 ± 0.02	0.42 ± 0.02

(a) MNIST

	$\boldsymbol{x}_S \sim p_{\theta_S}(\boldsymbol{x}_S \mid \boldsymbol{z})$			
Model	$q_{\phi_M}(\boldsymbol{z} \mid \boldsymbol{x}_M)$	$q_{\phi_T}(\boldsymbol{z} \mid \boldsymbol{x}_T)$	$q_{\Phi_{M,T}}(\boldsymbol{z} \mid \boldsymbol{X}_{M,T})$	$p_\Theta(\boldsymbol{z})$
PoE	0.37 ± 0.06	0.19 ± 0.10	0.19 ± 0.10	0.37 ± 0.07
MoE	0.01 ± 0.00	0.01 ± 0.00	0.01 ± 0.00	0.23 ± 0.02
MoPoE	0.07 ± 0.03	0.05 ± 0.03	0.04 ± 0.03	0.25 ± 0.03

(b) SVHN

Table 3.8: Quality of generated samples on MNIST-SVHN-Text. We report the average precision based on the precision-recall metric for generative models (higher is better) for conditionally and randomly generated image data. We denote the type of generation (random or conditional) using the variational approximation distribution $q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A)$ or the prior distribution $p_\Theta(\boldsymbol{z})$ in the second row of every table.

saw in the log-likelihood results (see Table 3.6). Also, we see that the average reported numbers for generating SVHN samples are lower than the MNIST ones, reflecting the two datasets' relative difficulty.

3.3.4 *Discussion of Results*

The evaluation of methods in Section 3.3.3 shows the improved performance of MoPoE-VAE over PoE-VAE and MoE-VAE. Its generalized formulation, which takes all subsets of modalities $\boldsymbol{X}_A, A \in \mathcal{P}(\mathbb{M})$ into account, can overcome most of the limitations of previous works. We can show that MoPoE-VAE infers meaningful latent representations and generates coherent samples for all input subsets $\boldsymbol{X}_A, A \in \mathcal{P}(\mathbb{M})$. In addition, MoPoE-VAE also produces high-quality samples according to the achieved test set log-likelihoods.

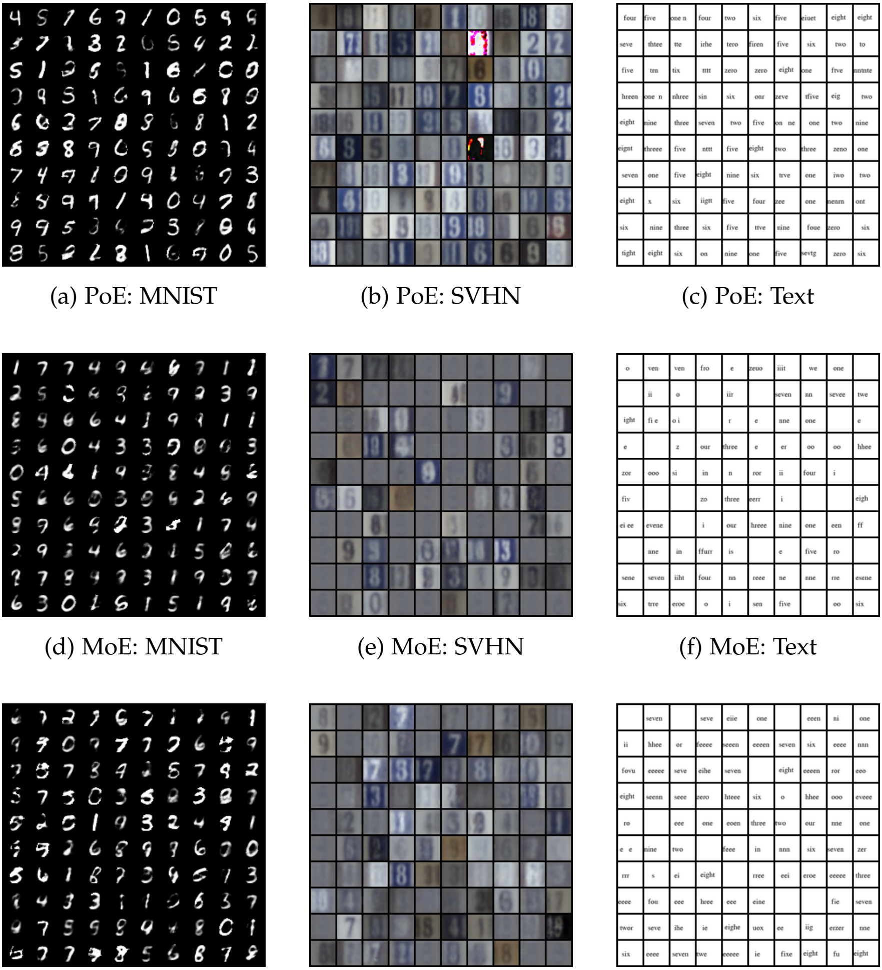
Figure 3.6 shows a qualitative comparison of random samples generated by the three multimodal VAEs, complementing the evaluation based on log-likelihoods and fidelity metrics. Qualitatively, the samples in Figure 3.6 confirm the evaluation and results described in Section 3.3: PoE-VAE provides better generative quality than MoE-VAE, leading to more coherent samples, and MoPoE-VAE provides the best trade-off between coherence and quality of all three methods.

3.4 LIMITATIONS OF SCALABLE MULTIMODAL VAES

Multimodal VAEs are a promising method for efficient generative learning. However, there seems to be a trade-off between the generative quality of samples and their coherence [Dau+22]. In Section 3.3, we see that MoPoE-VAE [SDV21] is not able to surpass the PoE-VAE [WG18] regarding generative quality if the complete set $\boldsymbol{X}$ is present.

In Daunhawer et al. [Dau+22], we investigate the drawbacks of *mixture*-based multimodal VAEs concerning the quality of their generated samples. We can show a gap in the quality of generated samples between mixture-based models and their unimodal counterparts on the PolyMNIST dataset (see Section 3.3.1.2). In Section 3.2.4, we showed how the MoPoE-based joint posterior approximation generalizes the formulation of previous works [Shi+19; SDV21; WG18]. Therefore, the discovered limitations apply to all methods that follow Definition 3.2.1.

Figure 3.8 shows the results for the three multimodal VAEs we introduced in Section 3.2. We evaluate the methods for different β values.

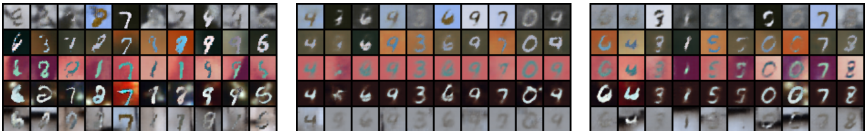


(a) PoE: MNIST (b) PoE: SVHN (c) PoE: Text

(d) MoE: MNIST (e) MoE: SVHN (f) MoE: Text

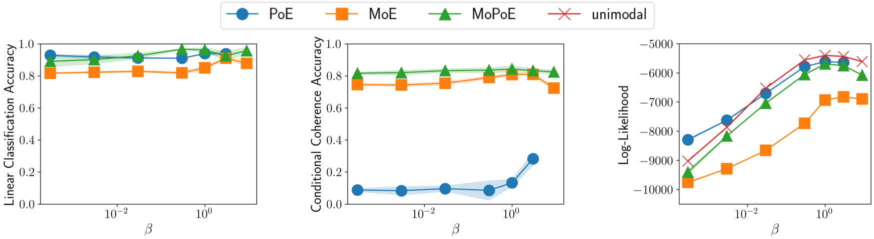
(g) MoPoE: MNIST (h) MoPoE: SVHN (i) MoPoE: Text

FIGURE 3.6: We qualitatively compare the random generation of the three methods: PoE-VAE, MoE-VAE, and MoPoE-VAE. From the prior distribution, we generate samples by first sampling $\boldsymbol{z} \sim p_\Theta(\boldsymbol{z})$, which are then input to the respective decoder $p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$. We use the same $\boldsymbol{z}$ for corresponding cells of the image matrices of every row such that we have coherence in the generated samples. The samples reflect the log-likelihoods reported in Table 3.7c and coherence results in Table 3.3 with PoE-VAE producing the highest quality of samples but lacking coherence.



(a) PoE (b) MoE (c) MoPoE

Figure 3.7: Ten unconditionally generated multimodal PolyMNIST samples. Column-wise, we use the same latent codes sampled from the prior distribution. Note that, row-wise, the digits should not be ordered.



(a) Latent Representation (b) Coherence (c) Log-Likelihood

Figure 3.8: Sensitivity analysis over different β values for three multimodal VAEs on the PolyMNIST dataset. We show the results for PoE-VAE, MoE-VAE, and MoPoE-VAE. All methods are trained and evaluated for $\beta \in \{3e^{-4}, 3e^{-3}, 3e^{-2}, 3e^{-1}, 1, 3, 9\}$.

We assess the quality of the learned joint latent representations by classifying them according to the shared information of the multimodal set (see Section 3.3.3). We evaluate the quality of the learned latent representation in the same way as described in Section 3.3.3. We train a linear classifier on 500 samples of the joint representations of the training set to predict the digit information, which is shared among all modalities. This linear classifier is then used to predict the digit information from joint representations of the test set. In Figure 3.8a, all methods perform well for different β values. The results reflect those reported in Section 3.3.3.1.

We also evaluate the coherence of generated samples. In Figure 3.8b, we see the results of the methods' conditional generation coherence. We use the same pre-trained classifiers as in Section 3.3.3.2 to predict the digit information from generated samples. We give 4 out of 5 modalities as input and generate the missing modality conditioned on the input modalities. Similar to the quality of the learned latent representations, we see that the results for the coherence evaluation of generated samples are also robust for a big part of the β range. The coherence of the PoE-VAE increases only for high β values but cannot reach the performance of both MoE-VAE and MoPoE-VAE. Additionally, using a high β value reduces the stability of the PoE-VAE to the extent that we could not report results for $\beta = 9.0$. Besides that, MoPoE-VAE achieves the highest coherence for all β values, reflecting the results from Section 3.3.3.2.

To evaluate the quality of samples for different β values, we look again at the test set log-likelihood as in Section 3.3.3.3. In Figure 3.8c, we report the joint log-likelihood for the three different multimodal VAEs introduced in Section 3.2. Additionally, we train unimodal VAEs on all 5 PolyMNIST modalities and report the sum of their test set log-likelihoods. No multimodal VAE can reach the same quality of samples as independent unimodal VAEs [Dau+22].

3.4.1 *Discussion of Limitations*

In this section, we show the limitations of multimodal VAEs that use a MoPoE-based distribution for their joint variational posterior approximation in combination with a single joint latent space according to Section 3.2.3. For the complete set of input modalities, MoPoE-based multimodal VAEs lack generative quality compared to the unimodal VAE despite their advantage of having access to more information (see Figure 3.8c). From the methods following Definition 3.2.1, PoE-VAE shows the best generative quality but is still worse than the unimodal VAE. On the other hand, PoE-VAE shows inferior performance regarding the coherence measure in case of missing modalities (see Figure 3.8b). Interestingly, the relative performance of methods seems to be consistent for different β-values (see Figure 3.8) [Hig+16].

3.5 CONCLUSION

This chapter discusses three VAE-based methods to learn from multimodal data. We introduce a new probabilistic aggregation function [MoPoE, SDV21], which outperforms and generalizes previous works. MoPoE is a mixture of products of experts, which defines a mixture distribution over all multimodal subsets, which it combines using the product of experts. We show how the idea behind MoPoE-VAE [SDV21] generalizes previous works [Shi+19; WG18] and is critical to a better trade-off between the quality and coherence of generated samples. While MoE-VAE [Shi+19] performs best if only a single modality is given as input, PoE-VAE [WG18] reaches the best performance if the full set of modalities is given as input. Only MoPoE-VAE achieves a stable and good performance across all possible input subsets.

Nevertheless, MoPoE-based multimodal VAEs show a gap in generative quality compared to unimodal VAEs despite their theoretical advantage of belonging to the class of weakly-supervised learning problems and, therefore, having access to more data and information than their unimodal counterpart [Dau+22]. Multimodal data is a complex form of weakly-supervised data, as the relationship between modalities is unknown. Therefore, PoE-VAE [WG18], MoE-VAE [Shi+19], and MoPoE-VAE [SDV21] implicitly make restrictive assumptions regarding the underlying group structure of a multimodal sample. The assumptions focus on extracting the shared information between modalities while neglecting the information only present in the individual modalities, which seems to hurt the generative quality of multimodal VAEs. In Chapter 4, we introduce an extension to learning from multimodal data, which is targeted toward providing more flexible assumptions regarding the underlying group structure.

4

MORE FLEXIBILITY BY INTRODUCING MODALITY-SPECIFIC LATENT SUBSPACES

In Chapter 3, we describe and evaluate different approaches to implement scalable multimodal VAEs. In the limitations section (Section 3.4) we explain and discuss why none of the proposed methods, PoE (section 3.2.1), MoE (section 3.2.2) or MoPoE (section 3.2.3), can fulfill all desiderata for multimodal models (see Section 2.3). In this chapter, we introduce modality-specific latent subspaces for multimodal VAEs. Instead of only having a single joint latent space as in the previous chapter, we assume a shared multimodal latent space and additional modality-specific latent subspaces. The shared latent factors represent the information shared between all modalities, and the modality-specific latent factors encode the information specific to individual modalities.

Hsu and Glass [HG18] and Tsai et al. [Tsa+19] proposed models with modality-specific and shared latent distributions. The former [HG18] relies on supervision by labels to extract shared generative factors, while the latter approach [Tsa+19] is non-scalable. Bouchacourt, Tomioka, and Nowozin [BTN18] and Hosoya [Hos18] introduce latent subspaces for datasets, where they assume knowledge of the grouping information. Recently, Palumbo, Daunhawer, and Vogt [PDV22] proposed a multimodal VAE using shared and modality-specific latent subspaces, which uses auxiliary prior distributions.

Learning disentangled representations holds the promise of being more interpretable and intervenable, more suitable for downstream tasks and transfer learning approaches [BCV13; Lak+17; Loc+19; PJS17; Sch92; TBL18]. Locatello et al. [Loc+20] and Shu et al. [Shu+19] discuss the disentanglement of groups of shared and independent latent factors in the weakly-supervised setting. Methods using additional loss terms or contrastive approaches as regularizers to achieve a better disentanglement of modality-specific and shared latent factors are available [Che+18; Hwa+21; Joy+21; Shi+20], and probably would improve the performance and results presented in this section further. Adding more regularizers and loss terms makes analyzing and interpreting results and methods more difficult. We base our findings on clean loss functions without additional objectives; the shared encoding is the output of either an MoE, PoE, or MoPoE aggregation function. On purpose, we relinquish any additional loss terms or objectives, improving disentanglement between the latent spaces, as we are interested in the effect of different aggregation functions and assumptions on multimodal metrics.

4.1 SHARED AND MODALITY-SPECIFIC LATENT SPACES

In Chapter 3, we aggregate the unimodal latent experts $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$ based on some probabilistic aggregation function f resulting in the joint posterior distribution $q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X})$, where the aggregation is independent of the information stored in the latent dimension $z_j \in \boldsymbol{z}$. I. e., the aggregation function f applies the same operation to all latent factors.

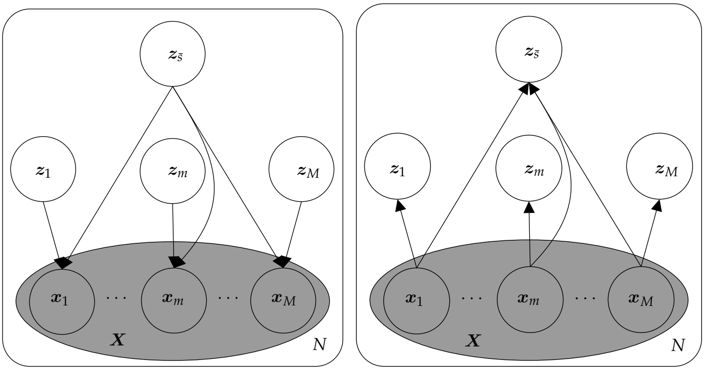
Bouchacourt, Tomioka, and Nowozin [BTN18] and Hosoya [Hos18] propose extensions to the vanilla VAE incorporating group information between samples. Group information is a form of weak supervision as it provides additional information compared to the *i.i.d.* setting [Loc+20]. Grouped observations are assumed to be invariant regarding specific generative factors that define the group. In Section 2.3, we describe how multiple modalities naturally form a group depending on the data collection process. We assume that a multimodal sample $\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M\}$ is a group of M samples, which are connected by some shared latent factors $\boldsymbol{y}$. Please note that, in general, the shared factors $\boldsymbol{y}$ do not need to be shared between all modalities $\boldsymbol{X}$, as there can be a more complicated structure of shared attributes between subsets $\boldsymbol{y}_A$ (see Section 2.3.2). In this section, the underlying modeling assumption is that all modalities $\boldsymbol{x}_m, m \in \mathbb{M}$ have the same shared attributes $\boldsymbol{y} \subset \boldsymbol{y}_m$ such that it holds

$$\boldsymbol{y} = \bigcap_{m \in A} \boldsymbol{y}_m \quad \forall \; A \subseteq \mathbb{M}, \tag{4.1}$$

where $\boldsymbol{y}_m$ are the generative factors of modality $m \in \mathbb{M}$.

Hence, instead of aggregating all dimensions of the unimodal posterior approximations $q_{\phi_m}(\boldsymbol{z} \mid \boldsymbol{x}_m)$, we introduce latent subspaces that enable us to aggregate only shared information between the modalities $m \in \mathbb{M}$ [Dau+20; SDV20]. And instead of a single latent variable $\boldsymbol{z} \in \mathbb{R}^d$, we now have $\boldsymbol{z}_{\bar{s}} \in \mathbb{R}^{d_{\bar{s}}}$ for the shared information between modalities and $\boldsymbol{z}_m \in \mathbb{R}^{d_m}, \forall \; m \in \mathbb{M}$ for the modality-specific information, i. e.

$$\boldsymbol{z} = [\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_M] \tag{4.2}$$



(a) Generative Model (b) Inference Model

FIGURE 4.1: Graphical models for multimodal VAEs utilizing additional modality-specific latent subspaces $\boldsymbol{z}_m$. We assume that every modality $\boldsymbol{x}_m$ is generated by some underlying latent factors $[\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m]$. The latent factors $\boldsymbol{z}_{\bar{s}}$ are shared between all modalities $\boldsymbol{x}_m \in \boldsymbol{X}$ whereas the factors $\boldsymbol{z}_m$ are modality-specific, i. e. they are only present in modality $\boldsymbol{x}_m$.

Lemma 4.1.1 (Multimodal ELBO using latent subspaces [Dau+20; SDV20]).

For $d_{\bar{s}} \in \mathbb{N}$ and $d_m \in \mathbb{N}, m \in \mathbb{M}$, the multimodal ELBO $\mathcal{L}(\Theta, \Phi_{\mathbb{M},\bar{s}}; \boldsymbol{X})$ using a shared latent subspace $\boldsymbol{z}_{\bar{s}} \in \mathbb{R}^{d_{\bar{s}}}$ and M modality-specific subspaces $\boldsymbol{z}_m \in \mathbb{R}^{d_m}$, $\forall\ m \in \mathbb{M}$ is given as

$$\begin{aligned}\mathcal{L}(\Theta, \Phi_{\mathbb{M},\bar{s}}; \boldsymbol{X}) = \mathbb{E}_{q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}}|\boldsymbol{X})} &\left[\sum_{m=1}^{M} \mathbb{E}_{q_{\phi_m}(\boldsymbol{z}_m|\boldsymbol{x}_m)} \left[\log p_{\theta_m}\left(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m\right)\right] \right] \\ &- D_{\mathrm{KL}}\left[q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}) \mid\mid p_{\Theta}(\boldsymbol{z}_{\bar{s}})\right] \qquad (4.3) \\ &- \sum_{m=1}^{M} D_{\mathrm{KL}}\left[q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m) \mid\mid p_{\theta_m}(\boldsymbol{z}_m)\right]\end{aligned}$$

where $q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m)$ are the modality-specific posterior approximations and $q_{\Phi}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X})$ the shared posterior approximation. The variational parameters $\Phi_{\mathbb{M},\bar{s}} = [\Phi_{\mathbb{M}}, \Phi_{\bar{s}}]$ are split into modality-specific parameters $\Phi_{\mathbb{M}} = [\phi_1, \ldots, \phi_M]$ and shared parameters $\Phi_{\bar{s}}$.

Proof. By assumption, the variational posterior $q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m)$ inferring the modality-specific factors $\boldsymbol{z}_m$ of modality m is only conditioned on

$\boldsymbol{x}_m$. The variational posterior $q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X})$ inferring the shared factors $\boldsymbol{z}_{\bar{s}}$ is instead conditioned on the full set of modalities $\boldsymbol{X}$. Additionally, we assume conditional independence of the latent vectors $\boldsymbol{z}_{\bar{s}}$ and $\boldsymbol{z}_m$, $\forall\ m \in \mathbb{M}$.

$$\begin{aligned} q_{\Phi}(\boldsymbol{z} \mid \boldsymbol{X}) =& q_{\Phi_{\mathbb{M},\bar{s}}}(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \ldots, \boldsymbol{z}_M \mid \boldsymbol{X}) && (4.4)\\ =& q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}) q_{\Phi_{\mathbb{M}}}(\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M \mid \boldsymbol{X}) && (4.5)\\ =& q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}) \prod_{m \in \mathbb{M}} q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m) && (4.6) \end{aligned}$$

Also by assumption, the probability of a data sample of modality $p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z})$ is reformulated to be conditioned on the shared latent $\bar{z}_s$ and the modality-specific information $\boldsymbol{z}_m$ only, $\forall\ m \in \mathbb{M}$

$$\begin{aligned} p_{\Theta}(\boldsymbol{X} \mid \boldsymbol{z}) =& p_{\Theta}(\boldsymbol{X} \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \ldots \boldsymbol{z}_M) && (4.7)\\ =& \prod_{m \in \mathbb{M}} p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \ldots \boldsymbol{z}_M) && (4.8)\\ =& \prod_{m \in \mathbb{M}} p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m) && (4.9) \end{aligned}$$

Hence, we formulate the multimodal ELBO $\mathcal{L}(\Theta, \Phi_{\mathbb{M},\bar{s}}; \boldsymbol{X})$ using $\boldsymbol{z}_{\bar{s}}$ and $\boldsymbol{z}_m$ as follows

$$\begin{aligned} \mathcal{L}_{LS} =& \mathbb{E}_{q_{\Phi_{\mathbb{M},\bar{s}}}(\boldsymbol{z}|\boldsymbol{X})}\left[\sum_{m=1}^{M} \log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m)\right] && (4.10)\\ &- \mathbb{E}_{q_{\Phi_{\mathbb{M},\bar{s}}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log \frac{q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}) \prod_{m=1}^{M} q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m)}{p_{\Theta}(\boldsymbol{z}_{\bar{s}}) \prod_{m=1}^{M} p_{\theta_m}(\boldsymbol{z}_m)}\right] \\ =& \mathbb{E}_{q_{\Phi_{\mathbb{M},\bar{s}}}(\boldsymbol{z}|\boldsymbol{X})}\left[\sum_{m=1}^{M} \log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m)\right] && (4.11)\\ &- \mathbb{E}_{q_{\Phi_{\bar{s}}}(\boldsymbol{z}|\boldsymbol{X})}\left[\log \frac{q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X})}{p_{\Theta}(\boldsymbol{z}_{\bar{s}})}\right] \\ &- \sum_{m=1}^{M} \mathbb{E}_{q_{\phi_m}(\boldsymbol{z}|\boldsymbol{x}_m)}\left[\log \frac{q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m)}{p_{\theta_m}(\boldsymbol{z}_m)}\right] \\ =& \mathbb{E}_{q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}}|\boldsymbol{X})}\left[\sum_{m=1}^{M} \mathbb{E}_{q_{\phi_m}(\boldsymbol{z}_m|\boldsymbol{x}_m)}\left[\log p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m)\right]\right] && (4.12)\\ &- D_{\mathrm{KL}}\left[q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}) \mid\mid p_{\Theta}(\boldsymbol{z}_{\bar{s}})\right] - \sum_{m=1}^{M} D_{\mathrm{KL}}\left[q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m) \mid\mid p_{\theta_m}(\boldsymbol{z}_m)\right] \end{aligned}$$

Please note that we use the short notation $\mathcal{L}_{LS}$ to denote the multimodal ELBO $\mathcal{L}(\Theta, \Phi_{\mathbb{M},\bar{s}}; \boldsymbol{X})$. □

To infer the variational posterior $q_\Phi(\bar{z}_s \mid \boldsymbol{X})$, we leverage the aggregation functions introduced in Section 3.2. Following Lemma 4.1.1, the latent space capacity of the split model is equal to single joint space models described in Section 3.2 if

$$d = d_{\bar{s}} + d_m \quad \forall\ m \in \mathbb{M} \tag{4.13}$$

In addition, introducing latent subspaces for the modality-specific information would enable the latent spaces of different sizes for different modalities $i, j \in \mathbb{M}$. If $d_i \neq d_j$, we can adapt the size of the modality-specific subspace to the difficulty of the respective modalities i and j. In this thesis, we set the size of the modality-specific latent spaces to be equal for all $m \in \mathbb{M}$.

4.2 EXPERIMENTS & RESULTS

This section describes the experiments, datasets, and results used to evaluate the proposed multimodal VAEs using modality-specific latent subspaces. We use the same evaluation metrics as in Chapter 3 such that the results from different modeling assumptions are comparable. We also perform experiments on the MNIST-SVHN-Text dataset. In addition, we introduce a bimodal version of the CelebA dataset to highlight the improved generative performance from the more flexible modeling assumptions. At the end of this chapter, we again discuss the results and limitations of the introduced method.

4.2.1 *Datasets*

We evaluate the different methods on two different datasets. First, the MNIST-SVHN-Text (MST) dataset, which we already introduced in Section 3.3.1. Second, we introduce a bimodal version of the CelebA dataset [Liu+15], which consists of images and text.

4.2.1.1 *Bimodal CelebA*

CelebA [Liu+15] is an image dataset showing the faces of celebrities. It consists of $N = 202'559$ samples from $10'177$ individuals, and every image is labeled according to 40 different attributes. We generate a bimodal dataset by converting each of the 40 labels to text. The text modality is a concatenation of the textual description of the available attributes into a comma-separated text. Underline characters are replaced by a blank space.

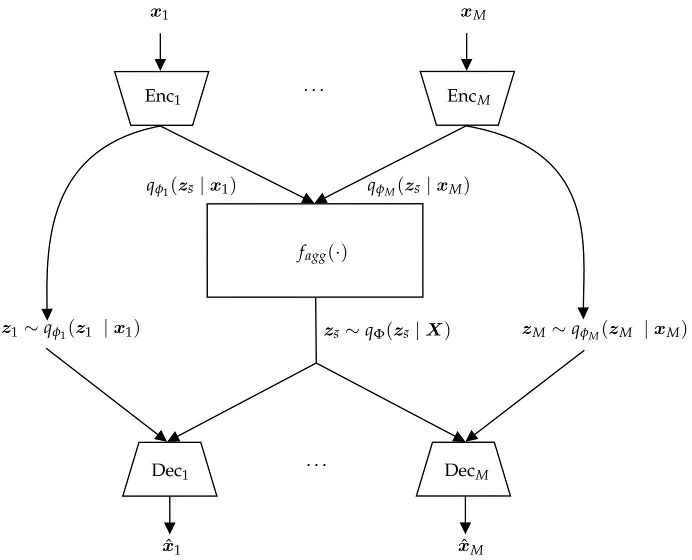
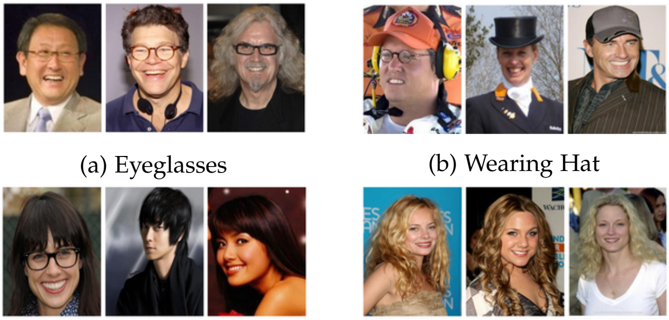

$\boldsymbol{x}_1$
$\boldsymbol{x}_M$
Enc$_1$
...
Enc$_M$
$q_{\phi_1}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_1)$
$q_{\phi_M}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_M)$
$f_{agg}(\cdot)$
$\boldsymbol{z}_1 \sim q_{\phi_1}(\boldsymbol{z}_1 \mid \boldsymbol{x}_1)$
$\boldsymbol{z}_{\bar{s}} \sim q_{\Phi}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X})$
$\boldsymbol{z}_M \sim q_{\phi_M}(\boldsymbol{z}_M \mid \boldsymbol{x}_M)$
Dec$_1$
...
Dec$_M$
$\hat{\boldsymbol{x}}_1$
$\hat{\boldsymbol{x}}_M$

FIGURE 4.2: The basic architecture of all methods is described in Section 4.1. They only differ in the f_{agg} building block. Different to the basic architecture in Figure 3.2, the latent representation of every encoder Enc$_m$ is the input to fully-connected layers mapping to the shared latent subspace $q_{\phi_m}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_m)$ and the modality-specific latent subspace $q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m)$. To reduce clutter, we illustrate it with two modalities only.



(a) Eyeglasses

(b) Wearing Hat

(c) Bangs

(d) Wavy Hair

Figure 4.3: Examples of Celeba images. Every subfigure shows three samples that are labeled positively with the corresponding attribute, eyeglasses (Figure 4.3a), wearing hat (Figure 4.3b), bangs (Figure 4.3c), and wavy hair (Figure 4.3d). The images are taken from the project website `http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html` [Liu+15].

We create strings of 256 characters (the maximum string length possible following the described rules). If a given face has only a small number of attributes, which would result in a short string, we fill the remaining space with the asterisk character $*$. Figure 4.3 shows image examples, and Table 4.1 shows examples of strings.

4.2.2 *Implementation & Training*

We generally follow the same assumptions as in Section 3.3.2. The significant difference is how to implement modality-specific and shared latent subspaces. We use a single encoder per modality where the split between modality-specific and shared latent subspace, $q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m)$ and $q_{\phi_m}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_m)$, only happens in the last linear layer. E. g. for the MST dataset we have 4 linear layers for every modality m instead of only 2 as in Tables A.2a, A.4a and A.6a. Two linear layers encode $\boldsymbol{\mu}_m(\boldsymbol{x}_m)$ and $diag(\boldsymbol{\sigma}_m(\boldsymbol{x}_m))$, and the remaining two $\boldsymbol{\mu}_{\bar{s}}(\boldsymbol{x}_m)$ and $diag(\boldsymbol{\sigma}_{\bar{s}}(\boldsymbol{x}_m))$. In a simplification of notation, we remove the additional subscripts $\mathbb{M}$ and $\bar{s}$ in the description of the variational parameters in this section, as it is clear from context whether we infer the shared $\boldsymbol{z}_{\bar{s}}$ or modality-specific $\boldsymbol{z}_m$ latent variable.

bags under eyes, chubby, eyeglasses, gray hair, male, mouth slightly open, oval face, sideburns, smiling, straight hair
big nose, male, no beard, young
attractive, big nose, black hair, bushy eyebrows, high cheekbones, male, mouth slightly open, oval face, smiling, young
5 o clock shadow, bushy eyebrows, chubby, double chin, gray hair, high cheekbones, male, smiling, straight hair
arched eyebrows, attractive, bangs, black hair, heavy makeup, smiling, straight hair, wearing lipstick, young
attractive, brown hair, bushy eyebrows, high cheekbones, male, oval face, smiling, young
attractive, high cheekbones, oval face, smiling, wearing lipstick, young
attractive, blond hair, heavy makeup, high cheekbones, oval face, smiling, wearing lipstick, young
attractive, brown hair, heavy makeup, oval face, pointy nose, straight hair, wearing lipstick, young
5 o clock shadow, bags under eyes, big nose, brown hair, male, mouth slightly open, smiling, young
attractive, brown hair, heavy makeup, high cheekbones, smiling, wavy hair, wearing earrings, wearing lipstick, young
attractive, bangs, blond hair, heavy makeup, high cheekbones, smiling, wavy hair, wearing earrings, wearing lipstick, young

TABLE 4.1: Examples of strings from the text modality of CelebA. We can generate pairs of images and texts using the text we generated from the label descriptions, resulting in a bimodal dataset. For illustrative reasons, we dropped the asterisk characters.

For methods using a single joint latent space, the latent spaces $\boldsymbol{z}$ of all modalities $\boldsymbol{x}_m$ have the same number of dimensions due to the aggregation step. By introducing modality-specific subspaces, we get rid of this additional restriction. Only the shared subspaces of all data types must have the same number of latent dimensions. The number of latent space dimensions could account for different difficulty levels between modalities.

We use a batch size of 256 and the Adam optimizer [KB14] with a starting learning rate of 0.001 for both experiments. We train all models for 100 epochs.

4.2.2.1 *MNIST-SVHN-Text*

We use the same network architectures as in the experiments with methods using a single joint latent space (see Tables A.1, A.3 and A.5). To have an equal number of parameters, we set the modality-specific and shared latent subspace size to 4 and 16 for all modalities. This allows for a fair comparison between the two variants regarding the capacity of the latent space. We model the modalities with the same likelihoods as in Section 3.3.2.

We set β to 5.0 for all MST experiments. Unlike the methods with a single joint latent space, we introduce an additional hyperparameter β_{MS} for regularizing the sum of the KL divergences of the modality-specific subspaces. For all experiments, the β_{MS} for the modality-specific subspaces is set equal to the number of modalities, i. e.. $\beta_{MS} = 3.0$. Additionally, we add β_m for regularizing the KL-divergences of the modality-specific subspaces individually. β_T for the text modality is set to 5.0, and $\beta_m = 1.0$ for the other 2 modalities.

4.2.2.2 *Bimodal CelebA*

For the CelebA experiments, we switched to a ResNet architecture [He+16] for encoders and decoders of image and text modality due to the difficulty of the dataset. We describe the specifications of the individual layers for the image and text networks in Tables A.9 and A.11 in Appendix A.3.1. The image modality $p_{\theta_I}(\boldsymbol{x}_I \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_I)$ is modelled with a Laplace likelihood and the text modality $p_{\theta_T}(\boldsymbol{x}_T \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_T)$ with a categorical likelihood. The likelihoods are scaled following Equation (3.41) where the data dimensions are

$$|\boldsymbol{x}_I| = 64 \times 64 \times 3 = 12288 \tag{4.14}$$

$$|\boldsymbol{x}_T| = 256 \tag{4.15}$$

Therefore, we have the following $\boldsymbol{\alpha}$

$$\alpha_I = \frac{|\boldsymbol{x}_I|}{|\boldsymbol{x}_I|} = 1.0 \tag{4.16}$$

$$\alpha_T = \frac{|\boldsymbol{x}_I|}{|\boldsymbol{x}_T|} = 48.0 \tag{4.17}$$

The global β is set to 2.5, and the β_{MS} of the modality-specific subspaces again to the number of modalities, i. e., 2. The shared as well as the modality-specific latent spaces consist of 32 dimensions.

4.2.3 *Evaluation & Results*

To evaluate the additional modality-specific latent subspaces, we compare the same three methods, PoE-VAE, MoE-VAE, and MoPoE-VAE, as in Chapter 3. This time, every method combines shared and modality-specific latent subspaces as described in Section 4.1. Hence, we not only compare the new modeling assumptions to the single joint latent space models in Chapter 3, we also compare the effect of the new assumption on the three probabilistic aggregation methods, PoE, MoE, and MoPoE. We evaluate the methods using the same performance metrics and experiments as in Chapter 3: latent representation classification and generative coherence and quality.

4.2.3.1 *Latent Representation Classification*

To evaluate the learned representations, we train linear classifiers on the inferred latent representations of the training set. The procedure is the same as for methods with a single joint latent space (see Section 3.3.3.1 for details) except we train and evaluate the linear classifiers on the shared subspace $\boldsymbol{z}_{\bar{s}} \sim q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A)$, $A \in \mathcal{P}(\mathbb{M})$. Otherwise, we keep the setting from Section 3.3.3.1.

Compared to the models using only a single joint latent space, the performance remains stable with respect to the classification accuracy. We see that reducing the number of latent dimensions of the shared subspace to 16 dimensions compared to the 20 dimensions of the joint space does not hurt the quality of the learned latent representation. For the PoE-VAE, we see that adding modality-specific latent subspaces does not help in overcoming the inferior performance for subsets with $|A| < 3$ (see Tables 4.3a and 4.3b).

4.2.3.2 *Generation Coherence*

We rely on the same procedure as described in Section 3.3.3 to evaluate the coherence of the generated samples. We also use the same pre-trained classifiers when evaluating the joint latent space models.

Conditionally generating samples is different, though. For methods using a single joint latent space, we generate samples $\boldsymbol{x}_m$ conditioned on the shared latent representation $\boldsymbol{z}$, where $\boldsymbol{z} \sim q_{\Phi_A}(\boldsymbol{z} \mid \boldsymbol{X}_A)$ follows the variational posterior of the input subset A. In the case of conditional generation, we only choose subsets A where $m \notin A$. And for random generation, the subset A is the empty set, i. e. $A = \emptyset$ and $\boldsymbol{z} \sim p_{\Theta}(\boldsymbol{z})$.

For methods using shared and modality-specific subspaces, we generate samples $\boldsymbol{x}_m$ conditioned on the shared $\boldsymbol{z}_{\bar{s}}$ and the modality-specific $\boldsymbol{z}_m$ latent representations, where $\boldsymbol{z}_{\bar{s}} \sim q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A)$ follows the variational posterior for the shared subspace of the input subset A, and $\boldsymbol{z}_m \sim q_{\phi_m}(\boldsymbol{z}_m \mid \boldsymbol{x}_m)$ the variational approximation for the modality-specific subspace of modality m. For the conditional generation of modality m, we again only choose subsets A where $m \notin A$. Hence, we sample $\boldsymbol{z}_{\bar{s}}$ from the variational approximation, i. e. $\boldsymbol{z}_{\bar{s}} \sim q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A)$, and $\boldsymbol{z}_m$ from its prior distribution, i. e. $\boldsymbol{z}_m \sim p_{\theta_m}(\boldsymbol{z}_m)$. For random generations, we sample both $\boldsymbol{z}_{\bar{s}}$ and $\boldsymbol{z}_m$ from their prior distributions, i. e. $\boldsymbol{z}_{\bar{s}} \sim p_{\Theta}(\boldsymbol{z}_{\bar{s}})$ and $\boldsymbol{z}_m \sim p_{\theta_m}(\boldsymbol{z}_m)$ respectively.

Table 4.4 shows the coherence results of PoE-VAE, MoE-VAE, and MoPoE-VAE using modality-specific latent subspaces on the MST dataset. The performance is approximately equal to the joint space models when evalu-

	$q_{\phi_M}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_M)$	$q_{\phi_S}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_S)$	$q_{\phi_T}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_T)$
PoE	0.72 ± 0.07	0.10 ± 0.01	1.00 ± 0.00
MoE	0.96 ± 0.01	0.81 ± 0.03	1.00 ± 0.00
MoPoE	0.96 ± 0.00	0.81 ± 0.02	1.00 ± 0.00

(a) Single Modality: $\forall\, A : |A| = 1$

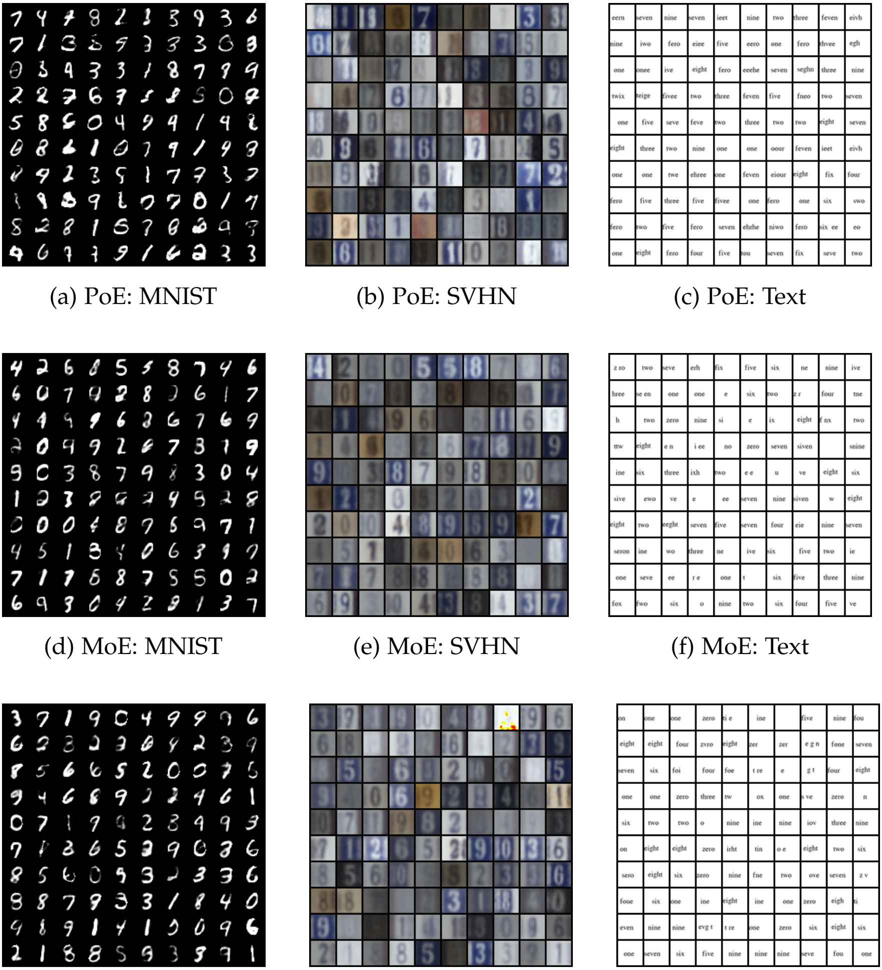
	$q_{\Phi_{M,S}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{M,S})$	$q_{\Phi_{M,T}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{M,T})$	$q_{\Phi_{S,T}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{S,T})$
PoE	0.71 ± 0.07	0.99 ± 0.01	1.00 ± 0.01
MoE	0.88 ± 0.02	0.98 ± 0.00	0.90 ± 0.01
MoPoE	0.97 ± 0.00	0.99 ± 0.00	0.99 ± 0.00

(b) Subsets of two modalities: $\forall\, A : |A| = 2$

	$q_{\Phi}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X})$
PoE	0.98 ± 0.01
MoE	0.92 ± 0.01
MoPoE	0.99 ± 0.00

(c) Full set of modalities: $|A| = 3$

TABLE 4.2: Linear classification accuracy of latent representations on the MNIST-SVHN-Text dataset for models using shared and modality-specific latent subspaces. We evaluate the shared representations $\boldsymbol{z}_{\bar{s}} \sim q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A)$ of all subsets $\boldsymbol{X}_A$. The abbreviations of the modalities are $\boldsymbol{x}_M$ for MNIST, $\boldsymbol{x}_S$ for SVHN, and $\boldsymbol{x}_T$ for text. The reported results are the mean and standard deviation of five runs.



(a) PoE: MNIST (b) PoE: SVHN (c) PoE: Text

(d) MoE: MNIST (e) MoE: SVHN (f) MoE: Text

(g) MoPoE: MNIST (h) MoPoE: SVHN (i) MoPoE: Text

FIGURE 4.4: Using shared and modality-specific subspaces, we qualitatively compare the random generations of the three methods, PoE-VAE, MoE-VAE, and MoPoE-VAE. We first sample $z_{\bar{s}} \sim p_{\Theta}(z_{\bar{s}})$ and $z_m \sim p_{\theta_m}(z_m)$ from their prior distributions, which are then input to the respective decoder $p_{\theta_m}(\boldsymbol{x}_m \mid z_{\bar{s}}, z_m)$. We use the same $z_{\bar{s}}$ for corresponding cells of the image matrices of every row such that we have coherence in the generated samples.

	$\boldsymbol{x}_M \sim p_{\theta_M}(\boldsymbol{x}_M \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_M)$		
Model	$q_{\phi_S}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_S)$	$q_{\phi_T}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_T)$	$q_{\Phi_{S,T}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{S,T})$
PoE	0.10 ± 0.00	0.42 ± 0.12	0.42 ± 0.11
MoE	0.75 ± 0.05	0.95 ± 0.03	0.85 ± 0.04
MoPoE	0.76 ± 0.02	0.98 ± 0.00	0.96 ± 0.00

(a) Conditional Generation: $\boldsymbol{x}_M$

	$\boldsymbol{x}_S \sim p_{\theta_S}(\boldsymbol{x}_S \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_S)$		
Model	$q_{\phi_M}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_M)$	$q_{\phi_T}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_T)$	$q_{\Phi_{M,T}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{M,T})$
PoE	0.17 ± 0.02	0.38 ± 0.19	0.44 ± 0.22
MoE	0.88 ± 0.00	0.93 ± 0.00	0.91 ± 0.00
MoPoE	0.89 ± 0.00	0.93 ± 0.01	0.93 ± 0.00

(b) Conditional Generation: $\boldsymbol{x}_S$

	$\boldsymbol{x}_T \sim p_{\theta_T}(\boldsymbol{x}_T \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_S)$		
Model	$q_{\phi_M}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_M)$	$q_{\phi_S}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_S)$	$q_{\Phi_{M,S}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{M,S})$
PoE	0.18 ± 0.03	0.10 ± 0.00	0.18 ± 0.03
MoE	0.76 ± 0.11	0.65 ± 0.10	0.70 ± 0.10
MoPoE	0.91 ± 0.01	0.74 ± 0.03	0.91 ± 0.02

(c) Conditional Generation: $\boldsymbol{x}_T$

TABLE 4.4: Generation Coherence on the MNIST-SVHN-Text dataset for methods using shared and modality-specific latent subspaces. The modality above the second horizontal line is generated based on the subsets below the same line for every subtable. The first row on the right side of every table shows the generative model $p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m)$ whereas the second line shows the different variational approximations $q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A), m \notin A$. The abbreviations of the different modalities are $\boldsymbol{x}_M$ for MNIST, $\boldsymbol{x}_S$ for SVHN, and $\boldsymbol{x}_T$ for text. The reported results are the mean and standard deviation of five runs.

Model	$\boldsymbol{X} \sim p_\Theta(\boldsymbol{X} \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_M, \boldsymbol{z}_S, \boldsymbol{z}_T)$
PoE	0.23 ± 0.00
MoE	0.30 ± 0.00
MoPoE	0.41 ± 0.00

TABLE 4.6: Random Generation Coherence on the MNIST-SVHN-Text dataset for methods including shared and modality-specific latent subspaces. We draw random latent vectors $\boldsymbol{z}_{\bar{s}} \sim p_\Theta(\boldsymbol{z}_{\bar{s}})$ and $\boldsymbol{z}_m \sim p_{\theta_m}(\boldsymbol{z}_m)$ from the respective prior distributions. The reported results are the mean and standard deviation of five runs.

ating the coherence of conditionally generated MNIST samples. MoE-VAE and MoPoE-VAE perform slightly inferior, but PoE-VAE can increase its coherence numbers for the MNIST modality. Assessing the coherence of generated SVHN samples, we see that PoE-VAE achieves similar results to its joint latent space version. Moe-VAE and MoPoE-VAE can improve their coherence significantly compared to their joint latent space versions. For the text modality, the coherence of all methods is slightly worse compared to the joint space models. Interestingly, only MoPoE-VAE can increase its joint random coherence compared to the joint space models. PoE-VAE and MoE-VAE show the same performance.

4.2.3.3 *Generative Quality of Samples*

To evaluate the generative quality of samples, we again look at test set log-likelihoods and fidelity metrics for image modalities. The generation of samples follows the scheme described in Section 4.2.3.2. Different from the coherence results, we always approximate the joint log-likelihoods $\log p_\Theta(\boldsymbol{X})$ and not the likelihoods of single modalities $\log p_{\theta_m}(\boldsymbol{x}_m)$. In Table 4.7, we report the test set log-likelihoods, and in Table 4.9, the average precision of the precision-recall metric for generative models [Saj+18]. We report the log-likelihood results in three subtables for the generation conditioned on subsets of the same size $|A|$. Table 4.8a shows the result for $|A| = 1$, Table 4.8b for $|A| = 2$, and Table 4.8c for the full input set.

Compared to the test set log-likelihoods from models with a single joint latent space, the numbers for models with shared and modality-specific latent subspaces are slightly inferior. Surprisingly, this result is independent of the input subset's size $|A|$. When generating the complete

	$q_{\phi_M}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_M)$	$q_{\phi_S}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_S)$	$q_{\phi_T}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_T)$
PoE	-2139.0 ± 13.6	-1940.7 ± 7.9	-2525.5 ± 614.7
MoE	-2068.8 ± 2.3	-1916.6 ± 2.6	-2082.7 ± 4.0
MoPoE	-2065.8 ± 4.7	-1915.2 ± 4.7	-2084.0 ± 2.7

(a) Subsets consisting of a single modality: $\forall A : |A| = 1$

	$q_{\Phi_{M,S}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{M,S})$	$q_{\Phi_{M,T}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{M,T})$	$q_{\Phi_{S,T}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{S,T})$
PoE	-1907.9 ± 19.7	-2443.3 ± 614.5	-1895.1 ± 12.0
MoE	-1914.1 ± 2.2	-2083.4 ± 4.0	-2086.2 ± 3.9
MoPoE	-1907.1 ± 5.6	-2079.7 ± 5.4	-2082.6 ± 6.9

(b) Subsets consisting of two modalities: $\forall A : |A| = 2$

	$q_{\Phi}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X})$
PoE	-1821.1 ± 14.5
MoE	-2085.6 ± 3.8
MoPoE	-2079.4 ± 9.5

(c) Full set of modalities: $\forall A : |A| = 3$

Table 4.7: Test set log-likelihoods on the MNIST-SVHN-Text dataset for methods including shared and modality-specific latent subspaces. We report the test set log-likelihoods of the joint generative model $\sum_{m \in \mathbb{M}} p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m)$ conditioned on the variational posterior of subsets of modalities $q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A)$. The modalities are given as $\boldsymbol{x}_M$ for MNIST, $\boldsymbol{x}_S$ for SVHN, $\boldsymbol{x}_T$ for text. We denote the full set of modalities as $\boldsymbol{X} = \{\boldsymbol{x}_M, \boldsymbol{x}_S, \boldsymbol{x}_T\}$. The reported results are the mean and standard deviation of five runs.

set of modalities $\boldsymbol{X}$ from a subset of modalities $\boldsymbol{X}_A$, the modality-specific information $\boldsymbol{z}_m$ for the missing modalities $\boldsymbol{x}_m, m \notin A$ has to be sampled from the prior $p_{\theta_m}(\boldsymbol{z}_m)$. Hence, compared to the joint space models, a lower test-set log-likelihood is reasonable because joint space models do not have to sample from the prior distribution $p_\Theta(\boldsymbol{z})$ when conditionally generating samples. More surprisingly, we see the same results pattern even for the complete set of input modalities $\boldsymbol{X}$, where we do not have to sample from the prior distributions of the modality-specific subspaces $p_{\theta_m}(\boldsymbol{z}_m)$ either. Again, we see the unstable performance of PoE-VAE (see Section 3.3.3.3) over different seeds, which we see in the high standard deviations.

In Table 4.9, we report the results for the precision-recall metric for generative models for the image modalities, MNIST (Table 4.10a) and SVHN (Table 4.10b). Compared to Table 3.8, where we report the performance of models using a single joint latent space, we see that all models improve their performance when using shared and modality-specific latent subspaces for both modalities. MoE-VAE and MoPoE-VAE benefit a lot and surpass PoE-VAE in most cases regarding the generative quality of samples, although PoE-VAE also reports better numbers.

Visually comparing randomly generated samples, i. e., samples, which are sampled from the prior distribution $p_\Theta(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_M, \boldsymbol{z}_S, \boldsymbol{z}_T)$ or $p_\Theta(\boldsymbol{z})$ respectively, confirms the findings on a qualitative level (see Figure 4.4 to the ones in Figure 3.6): extending multimodal VAEs with modality-specific latent subspaces improves the quality of their generated samples. The two sets of generated samples were selected randomly in Figure 3.6 and Figure 4.4.

4.2.3.4 *Results for Bimodal CelebA*

We evaluate the learned latent representations and the coherence and quality of generated samples. Unlike the experiments on the MST and PolyMNIST datasets, we measure the performance on the bimodal CelebA dataset using the average precision metric. There are 40 different attributes for every multimodal sample. The class imbalance for the different attributes varies, so average precision is better suited than accuracy to report the results. To simplify the reporting of classification metrics, we report the mean average precision of the 40 attributes in Tables 4.11 and 4.12. Similar to the experiments on the MST and PolyMNIST datasets, we assess the coherence and latent representations of the different methods according to all the possible input subsets. The sample quality is evaluated in Figure 4.5 on a qualitative level only.

	$\boldsymbol{x}_M \sim p_{\theta_M}(\boldsymbol{x}_M \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_M)$			
Model	$q_{\phi_S}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_S)$	$q_{\phi_T}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_T)$	$q_{\Phi_{S,T}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{S,T})$	$p_\Theta(\boldsymbol{z}_{\bar{s}})$
PoE	0.55 ± 0.01	0.33 ± 0.03	0.33 ± 0.03	0.56 ± 0.02
MoE	0.62 ± 0.01	0.64 ± 0.01	0.64 ± 0.01	0.51 ± 0.02
MoPoE	0.62 ± 0.01	0.64 ± 0.01	0.64 ± 0.01	0.51 ± 0.01

(a) MNIST

	$\boldsymbol{x}_S \sim p_{\theta_S}(\boldsymbol{x}_S \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_S)$			
Model	$q_{\phi_M}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_M)$	$q_{\phi_T}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_T)$	$q_{\Phi_{M,T}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_{M,T})$	$p_\Theta(\boldsymbol{z}_{\bar{s}})$
PoE	0.38 ± 0.02	0.18 ± 0.05	0.20 ± 0.05	0.38 ± 0.02
MoE	0.23 ± 0.02	0.23 ± 0.02	0.22 ± 0.02	0.19 ± 0.02
MoPoE	0.23 ± 0.02	0.22 ± 0.02	0.23 ± 0.02	0.20 ± 0.02

(b) SVHN

TABLE 4.9: Quality of generated samples on MNIST-SVHN-Text. We report the average precision based on the precision-recall metric for generative models (higher is better) for conditionally and randomly generated image data. We denote the type of generation (random or conditional) using the variational approximation distribution $q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A)$ or the prior distribution $p_\Theta(\boldsymbol{z})$ in the second row of every table. The generation follows the process described in Section 4.2.3.2. The modalities are given as $\boldsymbol{x}_M$ for MNIST, $\boldsymbol{x}_S$ for SVHN, $\boldsymbol{x}_T$ for text.

Table 4.11 shows the performance of the three methods, PoE-VAE, MoE-VAE, and MoPoE-VAE, regarding their learned latent representations. Again, MoE-VAE and MoPoE-VAE outperform PoE-VAE regarding the quality of learned representation according to linear classification accuracy. Given the bimodal nature of the dataset, the performances of MoE-VAE and MoPoE-VAE are approximately equal, as the aggregation of different modalities is less critical in the two-modality setting.

Interestingly, the coherence results in Table 4.12 for the conditional generation of text based on an input image $\boldsymbol{x}_T \sim p_{\theta_T}(\boldsymbol{x}_T \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_T)$ do not reflect the results seen on the MST and PolyMNIST datasets. PoE-VAE achieves the best coherence among the three methods. MoE-VAE and MoPoE-VAE seem to have difficulties learning the generation of text conditioned on input images. For the text-to-image generation, $\boldsymbol{x}_I \sim p_{\theta_I}(\boldsymbol{x}_I \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_I)$, MoPoE-VAE and MoE-VAE again achieve better numbers compared to PoE-VAE.

Model	$q_{\phi_{I,\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_I)$	$q_{\phi_{T,\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_I)$	$q_{\Phi_{\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X})$
PoE	0.30	0.31	0.32
MoE	0.35	0.38	0.35
MoPoE	0.40	0.39	0.39

Table 4.11: Linear classification results of latent representations on the bimodal CelebA dataset for models using shared and modality-specific subspaces. We report the mean average precision across all 40 attributes to assess the quality of the learned representations. We evaluate the shared representation $\boldsymbol{z}_{\bar{s}} \sim q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A), A \subseteq \{I, T\}$. $\boldsymbol{x}_I$ denotes the image modality, and $\boldsymbol{x}_T$ the text modality such that $\boldsymbol{X} = \{\boldsymbol{x}_I, \boldsymbol{x}_T\}$.

4.3 DISCUSSION AND LIMITATIONS

In Section 4.2.3, methods using shared and modality-specific latent subspaces improve on most metrics over models using a single joint latent space only (see Section 3.3.3). More flexible modeling assumptions improve the quality of generated samples. Latent representation classification and coherence of generated samples overall benefit as well, which is more surprising as these metrics are based on the shared latent factors $\boldsymbol{z}_{\bar{s}}$. As described in Section 4.2.2, we reduce the shared dimensions by the size of the modality-specific subspaces $\boldsymbol{z}_m$. Hence, we can achieve a better perfor-

	$\boldsymbol{x}_T \sim p_{\theta_T}(\boldsymbol{x}_T \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_T)$	$\boldsymbol{x}_T \sim p_{\theta_I}(\boldsymbol{x}_I \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_I)$
Model	$q_{\phi_{I,\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_I)$	$q_{\phi_{T,\bar{s}}}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{x}_T)$
PoE	0.26	0.33
MoE	0.14	0.41
MoPoE	0.15	0.43

TABLE 4.12: Coherence results on the bimodal CelebA dataset for methods using shared and modality-specific latent subspaces. The first row on the right side of every table shows the generative model $p_{\theta_m}(\boldsymbol{x}_m \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_m)$ whereas the second line shows the different variational approximations $q_{\Phi_A}(\boldsymbol{z}_{\bar{s}} \mid \boldsymbol{X}_A), m \notin A$. We report the mean average precision across all 40 attributes to assess the coherence of generated samples. $\boldsymbol{x}_I$ denotes the image modality, and $\boldsymbol{x}_T$ the text modality.

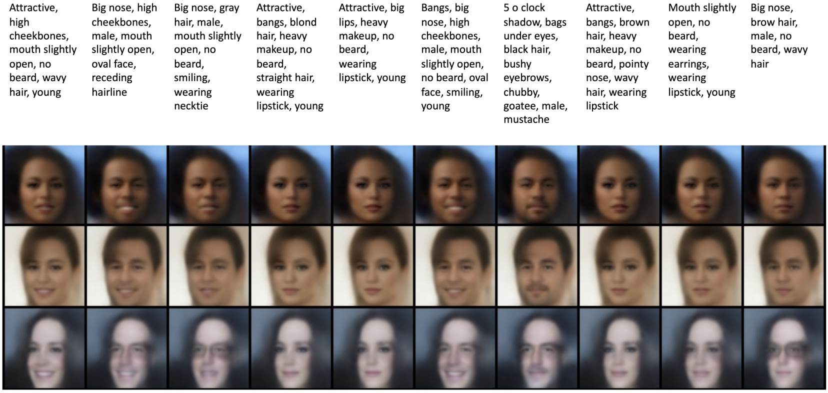
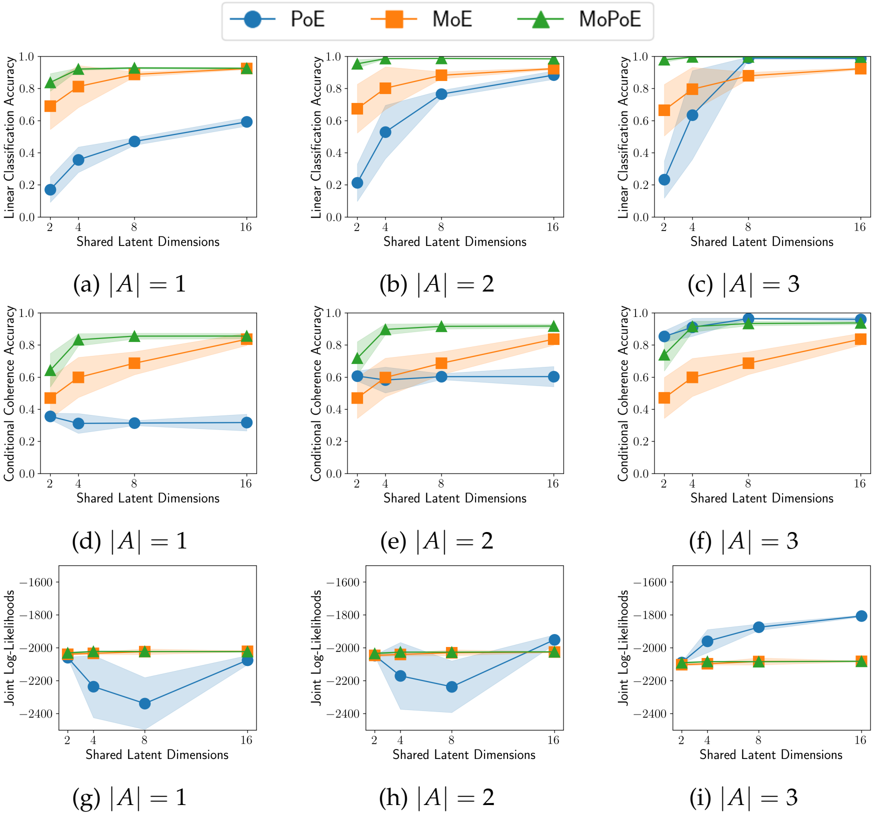

Attractive, high cheekbones, mouth slightly open, no beard, wavy hair, young
Big nose, high cheekbones, male, mouth slightly open, oval face, receding hairline
Big nose, gray hair, male, mouth slightly open, no beard, smiling, wearing necktie
Attractive, bangs, blond hair, heavy makeup, no beard, straight hair, wearing lipstick, young
Attractive, big lips, heavy makeup, no beard, wearing lipstick, young
Bangs, big nose, high cheekbones, male, mouth slightly open, no beard, oval face, smiling, young
5 o clock shadow, bags under eyes, black hair, bushy eyebrows, chubby, goatee, male, mustache
Attractive, bangs, brown hair, heavy makeup, no beard, pointy nose, wavy hair, wearing lipstick
Mouth slightly open, no beard, wearing earrings, wearing lipstick, young
Big nose, brow hair, male, no beard, wavy hair

FIGURE 4.5: Qualitative results for bimodal CelebA. The images are conditionally generated by MoPoE-VAE using the text on top of each column.



(a) $|A| = 1$

(b) $|A| = 2$

(c) $|A| = 3$

(d) $|A| = 1$

(e) $|A| = 2$

(f) $|A| = 3$

(g) $|A| = 1$

(h) $|A| = 2$

(i) $|A| = 3$

FIGURE 4.6: Results for different latent subspace sizes on the MNIST-SVHN-Text dataset for methods using shared and modality-specific latent subspaces. For all plots, we use the same subspace size d for all modalities, i. e. $d = d_{\bar{s}} + d_m = 20, \ \forall \ m \in \{M, S, T\}$. The x-axis of every plot shows the size of the shared space $d_{\bar{s}}$, where the y-axis shows the performance metric. The first row shows the results for evaluating the latent representations, the second for the coherence, and the last for the joint log-likelihoods. The columns sort the plots according to the size of the input subset $|A|$. We report the mean and standard deviations over 5 runs.

mance although the shared latent space in this chapter $d_{\bar{s}}$ is smaller than the joint latent space d in Chapter 3, i. e., $d_{\bar{s}} < d$.

Modeling multimodal data as having shared and modality-specific generative factors improves the quality of generated samples compared to multimodal VAEs using the more restrictive assumption of having only a single joint latent space.

However, adding latent subspaces leads to additional hyperparameters that must be carefully selected. Hyperparameters such as the size of the modality-specific and shared latent subspaces are crucial for achieving good results. In Figure 4.6, we show the performance of PoE-VAE, MoE-VAE, and MoPoE-VAE for different dimensions $d_{\bar{s}}$ of the shared latent subspace. For all values of $d_{\bar{s}}$, the combined capacity of shared and modality-specific latent subspaces equals the one in Section 4.2. Figure 4.6 shows a non-negligible influence of the subspace size on the performance concerning the latent representation classification and coherence and quality of generated samples. Hence, an increasing number of modalities in a dataset simultaneously increases the effort needed to find good hyperparameter settings. Although not part of the initial definition of scalability for multimodal ML (see Definition 3.1.1), the need for excessive hyperparameter tuning slows the development of new models. To some extent, it contradicts our goal of scalable multimodal ML.

It will depend on practical constraints whether the performance improvements of methods using shared and modality-specific latent subspaces compared to methods using a single joint latent space justify the additional overhead in finding a good set of hyperparameters.

Nevertheless, Section 4.2 shows that using more flexible modeling assumptions is beneficial for a method's performance, as we can see for MST (e. g., Figure 4.4) and bimodal CelebA (Section 4.2.1.1).

In these two datasets, the shared generative factors are shared between all modalities, reflecting a simplified relationship between multimodal samples and might originate in their generation. Both datasets are originally unimodal, which we combined with other datasets for MST or extended using additional information for Bimodal CelebA. However, more complicated group structures are possible, such as only pairwise shared generative factors. We must expect a more complex group structure for real-world multimodal or weakly-supervised datasets. In such a setting, the modeling assumption of shared and modality-specific latent subspaces would not hold anymore. In a real-world setting, it is unlikely that we will have precise knowledge about the underlying group structure. Hence, our assumptions will either be too restrictive or too flexible, making an adaptive approach inevitable. Given the performance improvements across all tasks between having a single joint latent space to shared and modality-specific subspaces, we need to be able to infer the relationships between group members in the weakly-supervised setting.

5 LEARNING THE RELATIONSHIP BETWEEN GROUPS OF SAMPLES

The proposed multimodal VAEs in Chapters 3 and 4 leveraged assumptions regarding the multimodal group structure. While they enabled the first promising results, restrictive assumptions are a limiting factor in learning from multimodal data. In this chapter, we want to overcome these assumptions and learn the relationship between groups of modalities or views. In contrast to previous chapters, we do not impose any assumption on the relationship between generative factors of weakly-supervised data but integrate a random partition model into the data-generating process. The random partition model selects groups of shared and independent generative factors between samples.

We present a differentiable random partition model. Our approach follows a two-stage procedure: first, we model the number of elements per subset, and second, we learn an ordering of the elements to assign the correct number of elements to every subset.

We introduce a differentiable approximation to the multivariate noncentral hypergeometric distribution [MVHG, Sut+23a] for modeling the number of elements per subset. Despite being non-differentiable, current sampling schemes for the MVHG are a trade-off between numerical stability and computational efficiency [Fog08a; Fog08b; LR01]. In Sutter et al. [Sut+23a], we overcome this trade-off.

For assigning the already inferred number of elements to subsets, we propose to use a Plackett-Luce [Luc59; Pla68] model for sorting the elements. Based on this ordering, we assign the first n_1 elements to the first subset, the second n_2 elements to the second subset, and continue until we reach the last subset. n_k is the number of elements for the kth subset according to the MVHG. Using differentiable distributions over sorting procedures [Gro+19; PE20] allows us to learn their parameters using gradient-based optimization.

We propose the differentiable random partition model [DRPM, Sut+23b], a fully-differentiable relaxation for RPMs that allows reparameterizable sampling. DRPM enables their integration into modern machine learning frameworks and their learning from data using stochastic optimization.

Jiang et al. [Jia+16] approached differentiable RPMs by tackling non-continuous loss functions and zero gradients almost everywhere with over-restrictive *i.i.d* assumptions. Every element is modeled with a categorical distribution over the number of subsets, where the assignment of elements to a subset is independent of other elements. Most deep probabilistic clustering approaches belong to this class of random partition models [see e. g. Dil+16; Man+21].

A problem related to set partitioning is the earth mover's distance problem [EMD, Mon81; RTG00]. However, EMD aims to assign a set's elements to different subsets based on a cost function and given subset sizes. Efficient solutions to the problem exist [Sin64], and various methods have recently been proposed, e.g., for document ranking [AZ11] or permutation learning [Men+18; SC+17].

Differentiable reparameterizations of complex distributions with learnable parameters enable new applications, as shown in Section 5.3, where we highlight the versatility of the new approach in three different experiments: clustering, multitask learning, and weakly-supervised learning.

In the remaining part of this chapter, we introduce our differentiable formulation for the MVHG, followed by our two-stage procedure for our DRPM method and experiments to highlight the versatility and general applicability of differentiable random partition models in ML.

5.1 THE DIFFERENTIABLE MULTIVARIATE HYPERGEOMETRIC DISTRIBUTION

We first describe the sampling procedure used for the differentiable hypergeometric distribution [Sut+23a]. Doing so, we connect it to the Gumbel-Softmax trick [GST, JGP16; MMT17] (or see Section 2.4.1.2), and how we relax the formulation of the hypergeometric distribution to make it differentiable. We show the correctness of the approach by comparing random samples of the proposed approach to samples from a reference distribution and evaluating their distance using the Kolmogorov-Smirnov test [KS, Kol30; Smi39].

We base the reparameterizable sample for the differentiable hypergeometric distribution, which is introduced in Sutter et al. [Sut+23a], on three steps:

1. Reformulate the multivariate distribution as a sequence of interdependent and conditional univariate hypergeometric distributions (Algorithms 1 and 2).

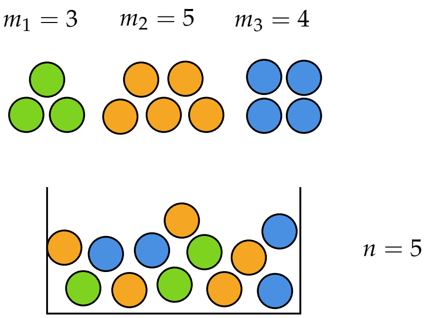


FIGURE 5.1: Illustration of the basic setting of the multivariate hypergeometric distribution. We have $K = 3$ classes of elements (green, orange, and blue) with different and unknown class importance $\omega_k, k \in \{1, \ldots, K\}$. The total number of elements in our urn is given by the sum of elements of all classes, i. e., $\sum_k m_k$. From this urn, we draw a group of n samples. In this example, $n = 5$. The group importance ω_k is often unknown and difficult to estimate. Our formulation helps to learn ω_k using gradient-based optimization when simulating how given samples are drawn from the urn.

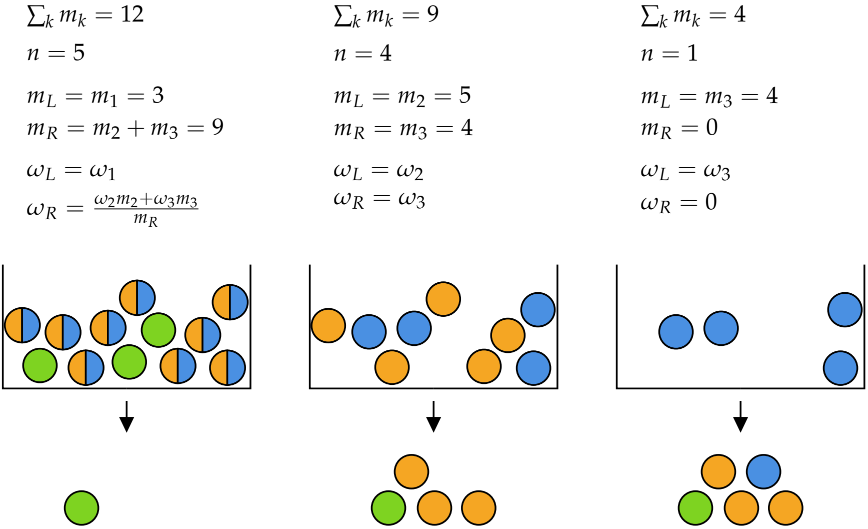
2. Calculate the probability mass function of the respective univariate distributions (Algorithm 3).

3. Sample from the conditional distributions utilizing the Gumbel-Softmax trick (Algorithm 4).

We explain all steps in the following Sections 5.1.1 to 5.1.3. Additionally, Algorithm 1 and Algorithms 2 to 4 describe the full reparameterizable sampling method using pseudo-code and Figures 5.1 and 5.2 illustrate it graphically.

5.1.1 *Sequential Sampling Using Conditional Distributions*

We use the conditional sampling algorithm [Fog08b; LR01] because it scales linearly with the number of classes and not with the size of the support S of the hypergeometric distribution (see Definition 2.6.3). Using the chain rule of probability, we reformulate the joint probability $p_{\boldsymbol{N}}(\boldsymbol{n}; \boldsymbol{\omega})$ into a sequence



(a) Step 1: $N_1 = 1$ (b) Step 2: $N_2 = 3$ (c) Step 3: $N_3 = 1$

FIGURE 5.2: Illustration of the proposed conditional sampling from the multivariate noncentral hypergeometric distribution. We use the same urn as in Figure 5.1 with $\boldsymbol{m} = [3, 5, 4]$ and $n = 5$. As described, we sequentially sample random variates of the individual classes. Hence, we start by sampling class 1 (fig. 5.2a). For that, we merge classes 2 and 3 (illustrated by the half-blue and half-orange balls), creating the necessary parameters $m_L, m_R, \omega_L, \omega_R$ for $p_{N_L}(\cdot)$ (described in the left column). This is also described in Algorithms 1 to 4. Using the univariate distribution $p_{N_L}(\cdot)$, we sample the random variable N_1, which is equal to 1 in our example (symbolized by the single green ball). We continue with the sampling of class 2 (fig. 5.2b). The merge operation simplifies to assigning $m_L = m_2$ and $m_R = m_3$, and n is the original n minus N_1. We draw $N_2 = 3$ in our example (again illustrated by the three orange balls below the urn). Because the number of drawn balls must sum to n, the last class N_3 is determined already (fig. 5.2c).

Algorithm 1 Sampling from the differentiable hypergeometric distribution. The individual building blocks are explained in more detail in Sections 5.1.1 to 5.1.3 and Algorithms 2 to 4.

```
**Input:** $\boldsymbol{m} \in \mathbb{N}^K$, $\boldsymbol{\omega} \in \mathbb{R}^K_{0+}$, $n \in \mathbb{N}$, $\tau \in \mathbb{R}_{0+}$
**Output:** $\boldsymbol{n} \in \mathbb{N}^K_0$, $\{\boldsymbol{\alpha}_i \in \mathbb{R}^{m_i}\}_{i=1}^K$, $\{\hat{\boldsymbol{r}}_i \in \mathbb{R}^{m_i}\}_{i=1}^K$
# Formulate the multivariate as a univariate distribution (section 5.1.1)
for $i \in \{1, \dots, K\}$ do
    $L \leftarrow i, R \leftarrow \{\bigcup_{j=i+1}^K j\}$
    $\boldsymbol{m} \rightarrow m_L, m_R \in \mathbb{N}_0$, $\boldsymbol{\omega} \rightarrow \omega_L, \omega_R \in \mathbb{R}_{0+}$
    $n_L, \boldsymbol{\alpha}_L, \hat{\boldsymbol{r}}_L \leftarrow$sampleUNCHG$(m_L, m_R, \omega_L, \omega_R, n, \tau)$
    # Re-assign classes for next step
    $n \leftarrow n - n_L$, $\boldsymbol{m} \leftarrow \boldsymbol{m} \setminus \boldsymbol{m}_L$, $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} \setminus \boldsymbol{\omega}_L$
    # Assign values for class $i$
    $n_i \leftarrow n_L, \boldsymbol{\alpha}_i \leftarrow \boldsymbol{\alpha}_L, \hat{\boldsymbol{r}}_i \leftarrow \hat{\boldsymbol{r}}_L$
end for
return $\boldsymbol{n}$, $\{\boldsymbol{\alpha}_i\}_{i=1}^K$, $\{\hat{\boldsymbol{r}}_i\}_{i=1}^K$
```

of conditional probabilities, which allows us to sample the different classes in the urn sequentially.

$$p_{\boldsymbol{N}}(\boldsymbol{n}; \boldsymbol{\omega}) = p_{N_1}(n_1; \boldsymbol{\omega}) \prod_{k=2}^{K} p_{N_k}\left(n_k \mid \left\{\bigcup_{l<k} n_l\right\}; \boldsymbol{\omega}\right) \tag{5.1}$$

Following Equation (5.1), every $p_{N_k}(\cdot)$ describes the probability of the number of samples of a single class k given the already sampled classes $l < k$. In the conditional sampling method, we model every conditional distribution $p_{N_k}(\cdot)$ as a univariate hypergeometric distribution with two classes L and R: for $k \in \{1, \dots, K\}$, we define a new class $L := \{k\}$ as the left class and a second new class $R := \{l : l > k \wedge l \leq K\}$ as the right class [Sut+23a]. We can now sequentially sample from the new univariate hypergeometric distributions with classes L and R, approximating sampling from the original MVHG. The parameters for each of the univariate distributions of the new classes L and R can be computed as [Fog08b]

$$m_L = \sum_{l \in L} m_l \tag{5.2}$$

$$m_R = \sum_{r \in R} m_r \tag{5.3}$$

$$\omega_L = \frac{\sum_{l \in L} \omega_l \cdot m_l}{m_L} \tag{5.4}$$

Algorithm 2 Sampling from the differentiable hypergeometric distribution. We describe in pseudo-code the sampling procedure for every univariate hypergeometric distribution. We explain the sub-routines in Sections 5.1.2 and 5.1.3 and Algorithms 3 and 4.

```
Input: $m_i, m_j \in \mathbb{N}_0; \omega_i, \omega_j \in \mathbb{R}_{0+}; n \in \mathbb{N}_0; \tau \in \mathbb{R}_{0+}$
Output: $n_i \in \mathbb{N}_0, \boldsymbol{\alpha}_i \in \mathbb{R}^{m_i}, \hat{\boldsymbol{r}}_i \in \mathbb{R}^{m_i}$
function sampleUNCHG($m_i, m_j, \omega_i, \omega_j, n, \tau$)
    $\boldsymbol{\alpha}_i \leftarrow$ calcLogPMF($m_i, m_j, \omega_i, \omega_j, n$)          # section 5.1.2
    $n_i, \hat{\boldsymbol{r}}_i \leftarrow$ contRelaxSample($\boldsymbol{\alpha}_i, \tau$))          # section 5.1.3
    return $n_i, \boldsymbol{\alpha}_i, \hat{\boldsymbol{r}}_i$
end function
```

$$\omega_R = \frac{\sum_{r \in R} \omega_r \cdot m_r}{m_R} \tag{5.5}$$

Sampling strategies based on univariate conditional distributions are only approximately equal to samples from the joint noncentral MVHG with equal $\tilde{\boldsymbol{\omega}}$. The merging operation defined by Equations (5.2) to (5.5) introduces a bias, which is only equal to zero for the central MVHG. Different grouping strategies or class subset selection algorithms can be a strategy to reduce this approximation error [Fog08b]. However, it is essential to note that the approximation error relates to a non-differentiable reference implementation with the same $\boldsymbol{\omega}$ but not the underlying and desired true class importance. Comparing the average number of drawn samples per class, we can still recover the true class importance if $\boldsymbol{\omega}$ is not a hyperparameter but learned as in our applications. Hence, we found the error irrelevant to our applications and left the exploration of different grouping strategies for future work.

5.1.2 *Calculate Probability Mass Function*

The exponent of $\boldsymbol{\omega}$ and the combinatorial terms can lead to numerical instabilities, making the direct calculation of the PMF in Equation (5.17) at least non-desirable. Hence, we focus on the log-probability distribution instead.

Algorithm 3 Subroutine for calculating the un-normalized logits of the probability mass function (PMF) of the hypergeometric distribution using pseudo-code (see Section 5.1.2).

```
Input: $m_l, m_r \in \mathbb{N}_0; \omega_l, \omega_r \in \mathbb{R}_{0+}; n \in \mathbb{N}_0$
Output: $\boldsymbol{\alpha}_l \in \mathbb{R}^{m_l}$
function CALCLOGPMF($m_l, m_r, \omega_l, \omega_r, n$)
    for $k \in \{0, \dots, m_l\}$ do
        $\boldsymbol{n}_{l,k} \leftarrow (k+1)$
        $\boldsymbol{n}_{r,k} \leftarrow (\mathrm{ReLU}(n-k)+1)$
    end for
    $\boldsymbol{l} \leftarrow \log\Gamma(\boldsymbol{n}_l + 1) + \log\Gamma(m_l - \boldsymbol{n}_{\boldsymbol{l}} + 1)$
    $\boldsymbol{r} \leftarrow \log\Gamma(\boldsymbol{n}_r + 1) + \log\Gamma(m_r - \boldsymbol{n}_{\boldsymbol{r}} + 1)$
    $\boldsymbol{\alpha}_l \leftarrow \boldsymbol{n}_l \log\omega_l + \boldsymbol{n}_r \log\omega_r - (\boldsymbol{l} + \boldsymbol{r})$
    return $\boldsymbol{\alpha}_l$
end function
```

DERIVING $\log p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})$ Calculations in log-domain increase numerical stability for such large domains while keeping the relative ordering. We start with the PMF of the MVHG

$$p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega}) = \frac{1}{P_0}\prod_{k=1}^{K}\binom{m_k}{n_k}\omega_k^{n_k} \tag{5.6}$$

where P_0 is defined as in Equation (2.79). From there, it follows

$$\begin{aligned}
\log p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega}) &= \log\left(\frac{1}{P_0}\prod_{k=1}^{K}\binom{m_k}{n_k}\omega_k^{n_k}\right) && (5.7)\\
&= \log\left(\frac{1}{P_0}\right) + \log\left(\prod_{k=1}^{K}\binom{m_k}{n_k}\omega_k^{n_k}\right) && (5.8)\\
&= \log\left(\frac{1}{P_0}\right) + \sum_{k=1}^{K}\log\left(\binom{m_k}{n_k}\omega_k^{n_k}\right) && (5.9)\\
&= \log\left(\frac{1}{P_0}\right) + \sum_{k=1}^{K}\left(\log\binom{m_k}{n_k} + \log\left(\omega_k^{n_k}\right)\right) && (5.10)\\
&= \log\left(\frac{1}{P_0}\right) + \sum_{k=1}^{K}\left(\log\binom{m_k}{n_k} + n_k\log\left(\omega_k\right)\right) && (5.11)
\end{aligned}$$

Because the ordering of categories is not influenced by scaling with a constant factor (due to the arg max, see Section 2.4.1), we can remove any normalization term of the PMF and still end up with the unnormalized log-probabilities of the noncentral MVHG. It follows

$$\begin{aligned}\log p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega}) &= \sum_{k=1}^{K}\left(\log\binom{m_k}{n_k} + n_k\log\left(\omega_k\right)\right) + C && (5.12)\\ &= \sum_{k=1}^{K}\left(\log\frac{1}{n_k!(m_k-n_k)!} + n_k\log\left(\omega_k\right)\right) + \tilde{C} && (5.13)\\ &= \sum_{k=1}^{K}\left(-\log\left(\Gamma(n_k+1)\Gamma(m_k-n_k+1)\right) + n_k\log\left(\omega_k\right)\right)\\ &\quad + \tilde{C} && (5.14)\end{aligned}$$

We used the relation $\Gamma(k+1) = k!$ in the last line. Setting $C = \tilde{C}$, it directly follows

$$\log p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega}) = \sum_{k=1}^{K} n_k\log\omega_k + \psi_F(\boldsymbol{n}) + C \qquad (5.15)$$

where $\psi_F(\boldsymbol{n}) = -\sum_{k=1}^{K}\log\left(\Gamma(n_k+1)\Gamma(m_k-n_k+1)\right)$ and the Gamma function is defined as [WW96]

$$\Gamma(z) = \int_0^\infty x^{z-1}e^{-x}dx \qquad (5.16)$$

We first derived the log-probability distribution for the multivariate case but defined Lemma 5.1.1 for the univariate case as in Sutter et al. [Sut+23a]. Following Section 2.6 and the previous paragraph, we write the probability mass function (PMF) $p_{N_L}(n_L;\boldsymbol{\omega})$ for the univariate hypergeometric distribution of two classes L and R as

$$p_{N_L}(n_L;\boldsymbol{\omega}) = \frac{1}{P_0}\binom{m_L}{n_L}\omega_L^{n_L}\binom{m_R}{n-x_L}\omega_R^{n-n_L} \qquad (5.17)$$

P_0 is defined as in Equation (2.79), ω_L, ω_R and their derivation from $\boldsymbol{\omega}$, and m_L, m_R as in Equations (5.2) to (5.5).

Lemma 5.1.1 (Sutter et al. [Sut+23a]).
The unnormalized log-probabilities

$$\log p_{N_L}(n_L;\boldsymbol{\omega}) = n_L\log\omega_L + (n-n_L)\log\omega_R + \psi_F(n_L) + C \qquad (5.18)$$

define the unnormalized weights of a categorical distribution that follows Fisher's noncentral hypergeometric distribution. C is a constant and $\psi_F(n_L)$ is defined as

$$\begin{aligned} \psi_F(n_L) = &- \log\left(\Gamma(n_L+1)\Gamma(n-n_L+1)\right) \\ &- \log\left(\Gamma(m_L-n_L+1)\Gamma(m_R-n+n_L+1)\right) \end{aligned} \tag{5.19}$$

Proof. Factors that are constant for all n do not change the relative ordering between different values of n. Hence, removing them preserves the ordering of values n [Bar17].

$$\log p_{N_L}(n_L;\boldsymbol{\omega}) = \log\left(\frac{1}{P_0}\binom{m_L}{n}\omega_L^{n_L}\binom{m_R}{n-n_L}\omega_R^{n-n_L}\right) \tag{5.20}$$

$$\begin{aligned} &= \log\binom{m_L}{n_L} + \log\binom{m_R}{n-n_L} \\ &\quad + \log\left(\omega_L^{n_L}\right) + \log\left(\omega_R^{n-n_L}\right) + C \end{aligned} \tag{5.21}$$

Using the definition of the binomial coefficient (see Section 2.6) and the Gamma function $\Gamma(k+1) = k!$, it follows

$$\begin{aligned} \log p_{N_L}(n_L;\boldsymbol{\omega}) = &n \cdot \log \omega_L + (n-n_L)\cdot \log\omega_R \\ &- \log\left(\Gamma(n_L+1)\Gamma(n-n_L+1)\right) \\ &- \log\left(\Gamma(m_L-n_L+1)\Gamma(m_R-n+n_L+1)\right) + C \end{aligned} \tag{5.22}$$

With $\psi_F(n)$ defined as in Equation (5.19), Equation (5.18) follows directly. □

Automatic differentiation frameworks[1] have numerically stable implementations of $\log\Gamma(\cdot)$, which allows to compute Equations (5.18) and (5.19) efficiently and reliably. Lemma 5.1.1 relates to the `calcLogPMF` function in Algorithm 2, and Algorithm 3 describes `calcLogPMF` in more detail.

Using the multivariate form of Lemma 5.1.1 (see Section 2.6), it is possible to directly calculate the categorical weights for the MVHG, which would result in a computational speed-up for large K compared to the proposed formulation. However, the size of the support S of the MVHG is $\prod_{k=1}^{K} m_k$, which results in unfeasible memory constraints for most interesting applications.

Algorithm 4 The `contRelaxSample` sub-routine of the differentiable sampling procedure for the multivariate hypergeometric distribution (see Section 5.1.3).

```
Input: $\boldsymbol{\alpha}_l \in \mathbb{R}^{m_l}, \tau \in {}_{0+}$
Output: $n_l \in \mathbb{N}_0, \hat{\boldsymbol{r}}_l \in \mathbb{R}^{m_l}$
function CONTRELAXSAMPLE($\boldsymbol{\alpha}_l, \tau$)
    $\boldsymbol{u} \leftarrow \boldsymbol{U}(\boldsymbol{0}, \boldsymbol{1})$
    $\boldsymbol{g} \leftarrow -\log(-\log \boldsymbol{u})$
    $\hat{\boldsymbol{r}}_l \leftarrow \boldsymbol{\alpha}_l + \boldsymbol{g}$
    $\boldsymbol{p}_l \leftarrow \text{Softmax}(\hat{\boldsymbol{r}}_l / \tau)$
    $n_l \leftarrow \text{Count-Index}(\text{Straight-Through}(\boldsymbol{p}_l))$
    return $n_l, \hat{\boldsymbol{r}}_l$
end function
```

5.1.3 *Continuous Relaxation for the Conditional Distribution*

Following Lemma 5.1.1, we use the GST to reparameterize the MVHG based on its conditional distributions $p_{N_L}(\cdot)$.

Lemma 5.1.2 (Sutter et al. [Sut+23a]**).**
The Gumbel-Softmax trick can be applied to the conditional distribution $p_{N_k}(n_k \mid \{n_l\}_{l<k}; \boldsymbol{\omega})$ of class k given the already sampled classes $l < k$.

Proof. When sampling class k, we draw n_k samples from class k such that $n_k \leq m_k$. The conditional distribution $p_{N_k}(n_k \mid \{n_l\}_{l<k}; \boldsymbol{\omega})$ for class k given the already sampled classes $l < k$ simultaneously defines the weights of a categorical distribution. Sampling n_k elements from class k can be seen as selecting the $(n_k + 1)$th category from the distribution defined by the weights $p_{N_k}(n_k \mid \{n_l\}_{l<k}; \boldsymbol{\omega})$. Therefore,

$$\sum_{0 \leq n_k \leq m_k} p_{N_k}\left(n_k \mid \{n_l\}_{l<k}; \boldsymbol{\omega}\right) = 1, \tag{5.23}$$

which allows us to apply the Gumbel-Max trick and, respectively, the GST trick. □

In Lemma 5.1.2, we connect the GST trick to the MVHG. Hence, reparameterizing enables gradients with respect to the parameter $\boldsymbol{\omega}$ of the MVHG:

$$\boldsymbol{u} \sim \boldsymbol{U}(0, 1), \qquad \boldsymbol{g}_k = -\log(-\log(\boldsymbol{u})), \qquad \hat{\boldsymbol{r}}_k = \boldsymbol{\alpha}_k(\boldsymbol{\omega}) + \boldsymbol{g}_k \tag{5.24}$$

[1] E. g. Tensorflow [Aba+16] or PyTorch [Pas+19]

where $\boldsymbol{u} \in [0,1]^{m_k+1}$ is a random vector of m_k *i.i.d.* uniform distributions $\boldsymbol{U}$ and $\boldsymbol{g}_k$ is therefore *i.i.d.* gumbel noise (see Sections 2.4.1.1 and 2.4.1.2). We perturb the unnormalized weights of the conditional univariate distributions $\boldsymbol{\alpha}_k(\boldsymbol{\omega})$, which are given as

$$\begin{aligned}\boldsymbol{\alpha}_k(\boldsymbol{\omega}) &= \log \boldsymbol{p}_{N_k}(\boldsymbol{n}_k; \boldsymbol{\omega}) - C \\ &= [\log p_{N_k}(0; \boldsymbol{\omega}), \dots, \log p_{N_k}(m_k; \boldsymbol{\omega})] - C, \end{aligned} \tag{5.25}$$

Perturbing $\boldsymbol{\alpha}_k(\boldsymbol{\omega})$ results in $\hat{\boldsymbol{r}}_k$. Equal to the original GST, we use the tempered softmax function to generate (m_k+1)-dimensional sample vectors from the perturbed unnormalized weights $\hat{\boldsymbol{r}}_k/\tau$, where τ is the temperature parameter (see Section 2.4.1.2 for more details on the GST). Due to Lemma 5.1.2, we do not need to calculate the constant C in Equations (5.18) and (5.25). Algorithm 4 describes Lemma 5.1.2 using pseudo code. Using the straight-through operator [BLC13] in Algorithm 4 (see Section 2.4.1.2), we can use hard samples in the forward path, but the relaxed vector in the backward path, which allows the computation of the derivatives. The Count-Index in Algorithm 4 maps the one-hot vector to an index, which is equal to the number of selected class elements in our case.

Although we use the GST, there is a difference compared to the vanilla GST. The log-weights $\boldsymbol{\alpha}_k$ of class k are a function of the class importance $\boldsymbol{\omega}$ and the pre-defined $\boldsymbol{n}_k = [0, \dots, m_k]$. A sequence of categorical distributions would result in $\sum_{k=1}^{K} m_k$ learnable parameters, whereas there are only K learnable parameters for the differentiable MVHG.

5.1.4 *Kolmogorov-Smirnov Test*

We evaluate the accuracy of the proposed method against a reference implementation using the Kolmogorov-Smirnov test [KS, Kol33; Smi39]. It is a nonparametric test to estimate the equality of two distributions by quantifying the distance between the empirical distributions of their samples. The null distribution of this test is calculated under the null hypothesis that the two groups of samples are drawn from the same distribution. If the test fails to reject the null hypothesis, the same distribution has generated the two groups of samples, i.e., the two underlying distributions are equal. This experiment is from Sutter et al. [Sut+23a].

For this experiment, we use an MVHG of three classes, where we compare samples from our differentiable formulation to samples from a non-differentiable reference implementation [SciPy, Vir+20]. We perform a sensitivity analysis concerning the class weights $\boldsymbol{\omega}$. We keep ω_1 and ω_3 fixed at

1.0, and ω_2 is increased from 1.0 to 10.0 in steps of 1.0. For every value of ω_2, we sample 50′000 *i.i.d.* random vectors. We use the Benjamini-Hochberg correction [BH95] to adjust the p-values for the false discovery rate of multiple comparisons as we perform $K = 3$ tests per joint distribution. Given a significance threshold of $t = 0.05$, $p > 0.05$ implies that we cannot reject the null hypothesis, which is desirable for our application as the proposed and the reference distribution are similar.

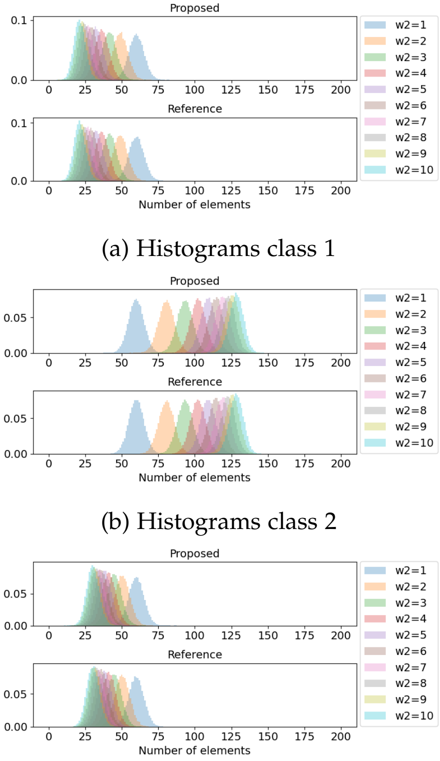
In Figure 5.3, we compare samples from the proposed distribution to the reference distribution. We show the histograms of samples generated by the reference and the proposed distribution for a visual comparison. The histograms per subfigure relate to samples generated by distributions with different $\boldsymbol{\omega}$. Figure 5.4 shows the results of the KS test for all classes k for the different values of ω_2. We see that the calculated distances of the KS test are small, and the corrected p-values are well above the threshold. Many are even close to 1.0. Hence, the test fails to reject the null hypothesis in 30 out of 30 cases. Additionally, the proposed and the reference implementation histograms are visually similar. The results of the KS test strongly imply that the proposed differentiable formulation effectively follows a noncentral hypergeometric distribution.

5.1.5 *Minimal Example*

We present the minimal example of Sutter et al. [Sut+23a] using a step-by-step procedure to provide intuition and further illustrate the proposed method. Here, we learn the generative model of an urn model using stochastic gradient descent when given samples from an urn model with a priori unknown weights $\boldsymbol{\omega}$. We use a generative approach to demonstrate how our method allows backpropagation when modeling the generative process of the samples by reparameterizing the MVHG. Additionally, we illustrate the minimal example with two figures (Figures 5.1 and 5.2), which explain the sampling procedure visually.

We are given a dataset of ν *i.i.d* samples $\mathbb{D} \in \mathbb{N}_0^{\nu \times K}$ from an MVHG distribution with unknown $\boldsymbol{\omega}_{gt} \in \mathbb{R}_+^K$. ν denotes the number of samples in the dataset, K the number of classes, and $\boldsymbol{N}_D \in \mathbb{D}$ a random sample from the dataset. For every $\boldsymbol{N}_D \in \mathbb{D}$, it holds that $\sum_k \boldsymbol{N}_{D,k} = n$. Additionally, we assume that we know the total number of elements in the urn, e.g. $\boldsymbol{m} = [m_1, m_2, ..., m_K]$.

We want to learn the unknown group importance $\boldsymbol{\omega}$ with a generative model using stochastic gradient descent (SGD). Hence, we assume a data



(a) Histograms class 1

(b) Histograms class 2

(c) Histograms class 3

FIGURE 5.3: Comparing random variables from the proposed differentiable formulation to a non-differentiable reference implementation. We draw samples from a multivariate noncentral hypergeometric distribution consisting of three classes: $m_k = 200 \; \forall \, k$ and $n = 180$. For all classes, we show the histograms over the number of elements sampled per class for different values of ω_2. The class weights ω_1 and ω_3 for classes 1 and 3 are set to 1.0, ω_2 is increased from 1.0 to 10.0 with a step size of 1.0 (w2 in the figure).

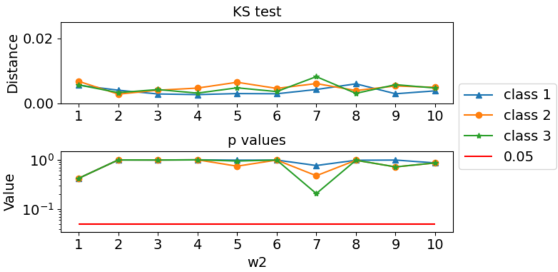


Figure 5.4: Comparing random variables from the proposed differentiable formulation to a non-differentiable reference implementation. For the same configurations as in Figure 5.3, we show the calculated distance values of the KS test between the reference and proposed implementation (upper plot) and their respective p-values (lower plot). The distance values of the KS test are small, while the p-values are high, indicating that the proposed and the reference distributions are similar.

generating distribution $p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})$ such that $\boldsymbol{N} \sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})$. The loss function $\mathbb{L}$ is given as

$$\mathbb{L} = \sum_{\boldsymbol{N}_D \in \mathbb{D}} \mathbb{E}_{\boldsymbol{N} \sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})} \left[(\boldsymbol{N}_D - \boldsymbol{N})^2 \right] \tag{5.26}$$

$$= \sum_{\boldsymbol{N}_D \in \mathbb{D}} \mathbb{E}_{\boldsymbol{N} \sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})} \left[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N}) \right] \tag{5.27}$$

where $\mathbb{L}$ is the loss per sample. $p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})$ is a noncentral multivariate hypergeometric distribution as defined in Definition 2.6.3 where the class importance $\boldsymbol{\omega}$ is unknown.

To minimize $\mathbb{E}[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N})]$, we want to optimize $\boldsymbol{\omega}$. Using SGD, we optimize the parameters $\boldsymbol{\omega}$ in an iterative manner:

$$\boldsymbol{\omega}_{t+1} := \boldsymbol{\omega}_t - \eta \frac{d}{d\boldsymbol{\omega}} \mathbb{E}_{\boldsymbol{N} \sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})} \left[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N}) \right] \tag{5.28}$$

where η is the learning rate, and t is the step in the optimization process. Unfortunately, we do not have a reparameterization estimator $\frac{d}{d\boldsymbol{\omega}}\mathbb{E}[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N})]$ because of the jump discontinuities of the arg max function in the categorical distributions.

As described in sections 5.1.1 and 5.1.3, we can rewrite $p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})$ as a sequence of conditional distributions.

In more detail, we rewrite the joint probability distribution $p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})$ as

$$p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega}) = p_{N_1}(n_1;\boldsymbol{\omega}) \prod_{k=2}^{n_k} p_{N_c}(n_k \mid n_1, ..., n_{k-1};\boldsymbol{\omega}) \tag{5.29}$$

where every distribution $p_{N_k}(\cdot;\boldsymbol{\omega})$ is a categorical distribution. We sample every N_k using Equation (5.18), i.e.

$$p_{N_k}(n_{L_k};\boldsymbol{\omega}) = n_{L_k} \log \omega_{L_k} + (n_k - n_L) \log \omega_{R_k} + \psi_F(n_{L_k}) + C \tag{5.30}$$

$\omega_{L_k}, \omega_{R_k}, m_{L_k}, m_{R_k}$, and $n_k = \sum_{l<k} N_l$ are calculated according to eqs. (5.2) to (5.5) and sequentially for every class.

The expected element-wise loss $\mathbb{E}_{\boldsymbol{N}\sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})}\left[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N})\right]$ changes to

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{N}\sim p_{\boldsymbol{N}}(\boldsymbol{x};\boldsymbol{\omega})}\left[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N})\right] &= \mathbb{E}_{\boldsymbol{N}\sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})}\left[\sum_{k=1}^{K} (N_{D,k} - N_k)^2\right] && (5.31)\\
&= \mathbb{E}_{\boldsymbol{N}\sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})}\left[\sum_{k=1}^{K} \mathbb{L}(N_{D,k}, N_k)\right] && (5.32)\\
&= \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{N}\sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})}\left[\mathbb{L}(N_{D,k}, N_k)\right] && (5.33)
\end{aligned}$$

Hence,

$$\frac{d}{d\boldsymbol{\omega}}\mathbb{E}[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N})] = \sum_{k=1}^{K} \frac{d}{d\boldsymbol{\omega}} \mathbb{E}_{\boldsymbol{N}\sim p_{\boldsymbol{N}}(\boldsymbol{n};\boldsymbol{\omega})}\left[\mathbb{L}(N_{D,k}, N_k)\right] \tag{5.34}$$

Unfortunately, for every $\frac{d}{d\boldsymbol{\omega}}\mathbb{E}\left[\mathbb{L}(N_{D,c}, N_c)\right]$, we face the problem of not having a reparameterizable gradient estimator. We cannot calculate the gradients of the loss directly, but $p_{N_c}(\cdot)$ being categorical distributions allows us to use the GST [JGP16; MTM14; Pau+20].

It follows [JGP16]

$$\begin{aligned}
\boldsymbol{y} &= \text{softmax}((\boldsymbol{\alpha} + \boldsymbol{g})/\tau) && (5.35)\\
&= \text{softmax}_\tau(\boldsymbol{\alpha} + \boldsymbol{g}) && (5.36)
\end{aligned}$$

where $g_1, \ldots, g_k$ are *i.i.d.* samples drawn from Gumbel$(0, 1)$, and τ is a temperature parameter. $\boldsymbol{y}$ is a continuous approximation to a one-hot vector, i.e. $0 \leq y_i \leq 1$ such that $\sum_i y_i = 1$.

Different to the standard GST, we infer the log-scores $\boldsymbol{\alpha}$ from the probability density function $\log p_{\boldsymbol{N}}(\cdot)$ (see eqs. (5.24) and (5.25)), which results in $\boldsymbol{\alpha}_k(\boldsymbol{\omega})$ in Section 5.1.3. We write for a single conditional class n_{L_k} as the procedure is the same for all classes. It follows

$$N_{\tau,k}(\boldsymbol{\omega}, \boldsymbol{g}) = \text{softmax}_\tau(\boldsymbol{\alpha}_k(\boldsymbol{\omega}) + \boldsymbol{g}) \tag{5.37}$$

where $n_{L_k} = [0, m_{L_k}]$. See Section 5.1.3 for the computation of $\boldsymbol{\alpha}_k(\boldsymbol{\omega})$. The Gumbel-Softmax approximation is smooth for $\tau > 0$, and therefore $\mathbb{E}[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N}_\tau)]$ has well-defined gradients $\frac{d}{d\boldsymbol{\omega}}$.

We write the loss function to optimize its gradients as

$$\mathbb{E}_{\boldsymbol{g}}\left[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N}_\tau(\boldsymbol{\omega}, \boldsymbol{g}))\right] = \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{g}}\left[\mathbb{L}(N_{D,c}, N_{\tau,k}(\boldsymbol{\omega}, \boldsymbol{g}))\right] \tag{5.38}$$

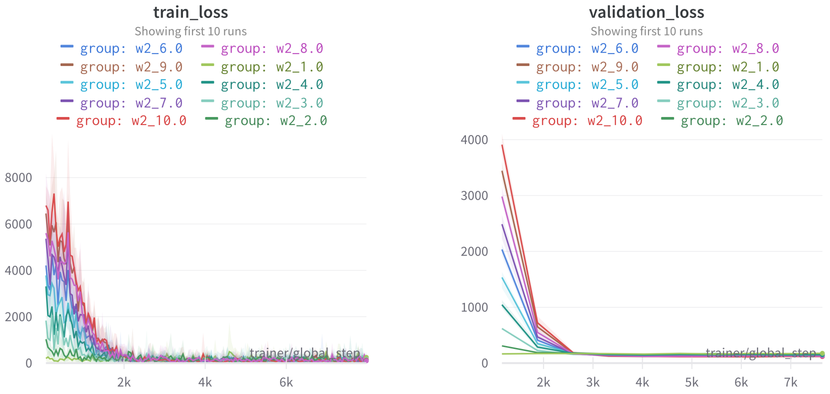
$$\frac{d}{d\boldsymbol{\omega}}\mathbb{E}_{\boldsymbol{g}}\left[\mathbb{L}(\boldsymbol{N}_D, \boldsymbol{N}_\tau(\boldsymbol{\omega}, \boldsymbol{g}))\right] = \sum_{k=1}^{K} \mathbb{E}_{\boldsymbol{g}}\left[\frac{d}{d\boldsymbol{\omega}}\mathbb{L}(N_{D,k}, N_{\tau,k}(\boldsymbol{\omega}, \boldsymbol{g}))\right] \tag{5.39}$$

By replacing the categorical distribution in eq. (5.18) with the Gumbel-Softmax distribution (see lemma 5.1.2), we can thus use backpropagation and automatic differentiation frameworks to compute gradients and optimize the parameters $\boldsymbol{\omega}$ [JGP16].

We implemented our minimal example for $K = 3$ classes. We set $\boldsymbol{m} = [m_1, m_2, m_3] = [200, 200, 200]$ and $n = 180$. We create 10 datasets $\mathbb{D} \in \mathbb{N}^{1000 \times 3}$ generated from different $\boldsymbol{\omega}_{gt}$ and show the performance of the proposed method. From these 1000 samples, we use 800 for training and 200 for validation. Similar to the setting we use for the KS test (see section 5.1.4), we choose 10 values for $\omega_{gt,2}$, i. e., $\omega_{gt,2} = [1.0, 2.0, \ldots, 10.0]$. The values for $\omega_{gt,1}$ and $\omega_{gt,3}$ are set to 1.0 for all datasets versions.

As described above, the model cannot access the data generating $\boldsymbol{\omega}_{gt}$. So, for every dataset $\mathbb{D}$, we optimize the unknown $\boldsymbol{\omega}$ based on the loss $\mathbb{L}$ defined in eq. (5.33).

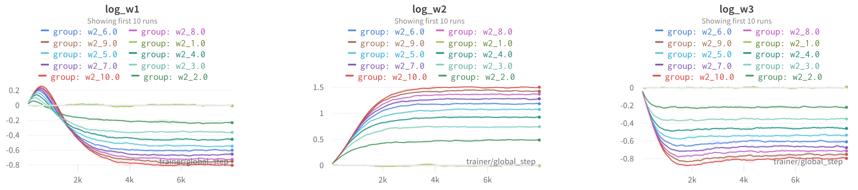
Figure 5.5 shows the training and validation losses over the training steps. We train the model for 10 epochs, but we see that the model converges earlier. The losses only differ at the beginning of the training procedure, which is probably an initialization effect, but quickly converge to similar values independent of the $\omega_{gt,2}$ value that generated the dataset. Figure 5.6 shows the estimation of $\log \boldsymbol{\omega}$. The x-axis shows the training step, and the y-axis shows the estimated value. Figures 5.6a to 5.6b demonstrate that the hypergeometric distribution is invariant to the scale of $\boldsymbol{\omega}$. With increasing value of $\omega_{gt,2}$, the values of ω_1 and ω_3 decrease, although their ground truth values $\omega_{gt,1}$ and $\omega_{gt,3}$ do not change. Neither the training nor the validation loss increases though (figs. 5.5a and 5.5b), which demonstrates the scale-invariance of $\boldsymbol{\omega}$.



(a) Training Loss

(b) Validation Loss

Figure 5.5: Training and validation losses for different values of $\boldsymbol{\omega}_{gt}$ of our minimal example described in section 5.1.5.



(a) Estimation of $\log \omega_1$

(b) Estimation of $\log \omega_2$

(c) Estimation of $\log \omega_3$

Figure 5.6: The estimated $\log \boldsymbol{\omega}$ values over the training procedure for different ground truth $\boldsymbol{\omega}_{gt}$ values of our minimal example (see section 5.1.5). These plots illustrate the scale invariance of the $\boldsymbol{\omega}$ parameter. With the value of ω_2 increasing, the estimated values for ω_1 and ω_3 change as well, but the training and validation loss remain low (see fig. 5.5).

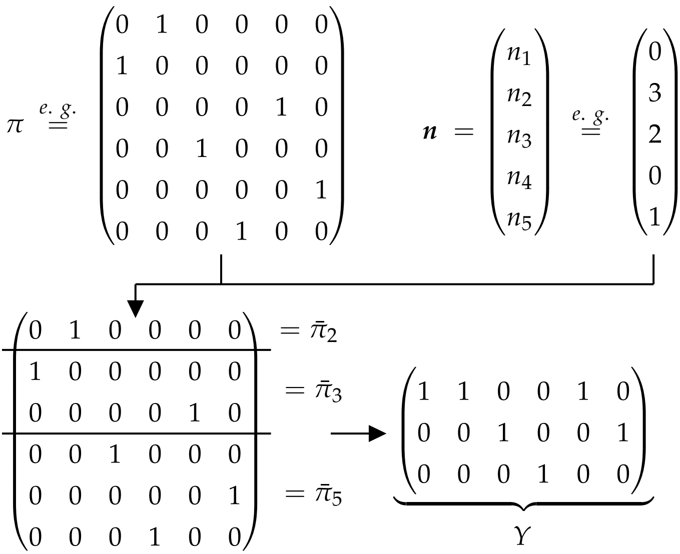

$\pi \overset{e.g.}{=}$
$\boldsymbol{n} = (n_1, n_2, n_3, n_4, n_5)^T \overset{e.g.}{=} (0, 3, 2, 0, 1)^T$
$= \bar{\pi}_2$
$= \bar{\pi}_3$
$= \bar{\pi}_5$
Y

FIGURE 5.7: Illustration of the proposed DRPM method. We first sample a permutation matrix π, and a set of subset sizes $\boldsymbol{n}$ separately in two stages. We then use $\boldsymbol{n}$ and π to generate the assignment matrix Y, the matrix representation of a partition ρ.

5.2 TOWARDS DIFFERENTIABLE RANDOM PARTITION MODELS

We can infer the number of elements per subset using the MVHG (Section 5.1). To have a random partition model, we need to formulate a probability distribution for assigning a given number of elements to the respective subset in combination with having a distribution over the number of elements per subset. We derive the differentiable random partition model using the newly proposed MVHG in this section.

We want to partition n elements $[n] = \{1, \ldots, n\}$ into K subsets $(\mathcal{S}_1, \ldots, \mathcal{S}_K)$ where K is *a priori* unknown. For a partition $\rho = (\mathcal{S}_1, \ldots, \mathcal{S}_K)$ to be valid, it must hold that

$$\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_K = [n] \quad \text{and} \quad \forall\, i \neq j : \ \mathcal{S}_i \cap \mathcal{S}_j = \emptyset \tag{5.40}$$

Put differently, every element i has to be assigned to precisely one subset $\mathcal{S}_k$. Alternatively to ρ, we describe a partition ρ as an assignment matrix $Y = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K]^T \in \{0,1\}^{K \times n}$. Every row $\boldsymbol{y}_k \in \{0,1\}^{1 \times n}$ is a multi-hot vector, where $y_{ki} = 1$ assigns element i to subset $\mathcal{S}_k$. We denote the size $|\mathcal{S}_k|$ of the k-th subset $\mathcal{S}_k$ as n_k.

In this work, we propose a new two-stage procedure for partition models. The proposed formulation separately infers the number of elements per subset n_k and the assignment of elements to subsets $\mathcal{S}_k$ by inducing an order on the n elements and filling $\mathcal{S}_1, ..., \mathcal{S}_K$ sequentially in this order. Figure 5.7 illustrates the proposed sampling method.

Definition 5.2.1 (Two-stage partition model [Sut+23b]).
Let $\boldsymbol{n} = [n_1, \ldots, n_K] \in \mathbb{N}_0^K$ be the subset sizes in ρ, with $\mathbb{N}_0$ the set of natural numbers including 0 and $\sum_{k=1}^K n_k = n$, where n is the total number of elements. Let $\pi \in \{0,1\}^{n \times n}$ be a permutation matrix that defines an order over the n elements. We define the two-stage partition model of n elements into K subsets as an assignment matrix $Y = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K]^T \in \{0,1\}^{K \times n}$ with

$$\boldsymbol{y}_k = \sum_{i=\nu_k+1}^{\nu_k+n_k} \boldsymbol{\pi}_i, \quad \textit{where} \;\; \nu_k = \sum_{\iota=1}^{k-1} n_\iota \tag{5.41}$$

such that $Y = [\{\boldsymbol{y}_k \mid n_k\}_{k=1}^K]^T$.

In contrast to previous works on partition models [MS16], we allow $\mathcal{S}_k$ to be the empty set $\emptyset$. Hence, K defines the maximum number of possible subsets, not the actual number of non-empty subsets.

To model the order of the elements, we use a permutation matrix $\pi = [\boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_n]^T \in \{0,1\}^{n \times n}$ which is a square matrix where every row and column sums to 1. This doubly stochastic property of all permutation matrices π [Mar60] thus ensures that the columns of Y remain one-hot vectors. At the same time, its rows correspond to n_k-hot vectors $\boldsymbol{y}_k$ in Definition 5.2.1 and therefore serve as subset assignment vectors.

Corollary 5.2.1 ([Sut+23b]). *A two-stage partition model Y, which follows Definition 5.2.1, is a valid partition model satisfying Equation* (5.40).

Proof. By definition, every row $\boldsymbol{\pi}_i$ and column $\boldsymbol{\pi}_j$ of π is a one-hot vector, hence $\sum_{n_k} \boldsymbol{\pi}_i$ results in a n_k-hot encoding. Therefore, $\sum_{i=1}^{n_k} \sum_{j=1}^{n} \pi_{ij} = n_k$ follows directly from π being a permutation matrix. Hence, if $\sum_k n_k = n$, every element i is assigned to one and only one $\boldsymbol{y}_k$. Thus, Definition 5.2.1 fulfills $\mathcal{S}_1 \cup \cdots \cup \mathcal{S}_K = [n]$, and $\mathcal{S}_i \cap \mathcal{S}_j = \emptyset \;\; \forall \;\; i, j$ and $i \neq j$. □

5.2.1 *Differentiable Random Partition Models*

An RPM $p(Y)$ defines a probability distribution over partitions Y. In this section, we derive how to extend the two-stage procedure from Definition 5.2.1 to the probabilistic setting, creating a two-stage RPM. To derive the two-stage RPM's probability distribution $p(Y)$, we must model distributions over $\boldsymbol{n}$ and π. We choose the MVHG distribution $p(\boldsymbol{n}; \boldsymbol{\omega})$ (see Sections 2.6 and 5.1) and the PL distribution $p(\pi; \boldsymbol{s})$ (see Section 2.4.2). For the remainder of this thesis, we denote $p(Y)$ as $p(Y; \boldsymbol{\omega}, \boldsymbol{s})$ to indicate dependence on the MVHG parameter $\boldsymbol{\omega}$ and PL parameter $\boldsymbol{s}$.

We calculate the probability $p(Y; \boldsymbol{\omega}, \boldsymbol{s})$ sequentially over the probabilities of subsets $p_{\boldsymbol{y}_k} := p(\boldsymbol{y}_k \mid \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s})$. $p_{\boldsymbol{y}_k}$ itself depends on the probability over subset permutations $p_{\bar{\pi}_k} := p(\bar{\pi} \mid n_k, \boldsymbol{y}_{<k}; \boldsymbol{s})$, where a subset permutation matrix $\bar{\pi}$ represents an ordering over n_k out of n elements.

Definition 5.2.2 (Subset permutation matrix $\bar{\pi}$ [Sut+23b]).
A subset permutation matrix $\bar{\pi} \in \{0,1\}^{n_k \times n}$, where $n_k \leq n$, must fulfill

$$\forall\, i \leq n_k : \sum_{j=1}^{n} \bar{\pi}_{ij} = 1 \quad \textit{and} \quad \forall\, j \leq n : \sum_{i=1}^{n_k} \bar{\pi}_{ij} \leq 1.$$

We describe the probability distribution over subset permutation matrices $p_{\bar{\pi}_k}$ using Definition 5.2.2 and corollary 2.4.1.

Lemma 5.2.1 (Probability over subset permutations $p_{\bar{\pi}_k}$ [Sut+23b]).
The probability $p_{\bar{\pi}_k}$ of any subset permutation matrix

$$\bar{\pi}_k = [\bar{\boldsymbol{\pi}}_1, \ldots, \bar{\boldsymbol{\pi}}_{n_k}]^T \in \{0,1\}^{n_k \times n} \tag{5.42}$$

is given by

$$p_{\bar{\pi}_k} := p(\bar{\pi} \mid n_k, \boldsymbol{y}_{<k}; \boldsymbol{s}) \tag{5.43}$$

$$= \prod_{i=1}^{n_k} \frac{(\bar{\pi}\boldsymbol{s})_i}{Z_k - \sum_{j=1}^{i-1} (\bar{\pi}\boldsymbol{s})_j} \tag{5.44}$$

where $\boldsymbol{y}_{<k} = \{\boldsymbol{y}_1, ..., \boldsymbol{y}_{k-1}\}$, $Z_k = Z - \sum_{j \in \mathcal{S}_{<k}} s_j$ and $\mathcal{S}_{<k} = \bigcup_{j=1}^{k-1} \mathcal{S}_j$.

Proof. We provide the proof for $p_{\bar{\pi}_1}$, but it is equivalent for all other subsets. Without loss of generality, we assume that there are n_1 elements in $\mathcal{S}_1$. Following Corollary 2.4.1, the probability of a permutation matrix $p(\pi; \boldsymbol{s})$ is given by

$$p(\pi; \boldsymbol{s}) = \frac{(\pi \boldsymbol{s})_1}{Z} \frac{(\pi \boldsymbol{s})_2}{Z - (\pi \boldsymbol{s})_1} \cdots \frac{(\pi \boldsymbol{s})_n}{Z - \sum_{j=1}^{n-1} (\pi \boldsymbol{s})_j} \tag{5.45}$$

At the moment, we are only interested in the ordering of the first n_1 elements. The probability of the first n_1 is given by marginalizing over the remaining $n - n_1$ elements:

$$p(\bar{\pi} \mid n_1; \boldsymbol{\omega}) = \sum_{\pi \in \Pi_1} p(\pi \mid \boldsymbol{s}) \tag{5.46}$$

where Π_1 is the set of permutation matrices such that the top n_1 rows select the elements in a specific ordering $\bar{\pi} \in \{0,1\}^{n_1 \times n}$, i.e. $\Pi_1 = \{\pi : [\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_{n_1}]^T = \bar{\pi}\}$. It follows

$$\begin{aligned}
p(\bar{\pi} \mid n_1; \boldsymbol{\omega}) &= \sum_{\pi \in \Pi_1} p(\pi \mid \boldsymbol{s}) \\
&= \sum_{\pi \in \Pi_1} \prod_{i=1}^{n} \frac{(\pi \boldsymbol{s})_i}{Z - \sum_{j=1}^{i-1} (\pi \boldsymbol{s})_j} \\
&= \prod_{i=1}^{n_1} \frac{(\bar{\pi} \boldsymbol{s})_i}{Z - \sum_{j=1}^{i-1} (\bar{\pi} \boldsymbol{s})_j} \sum_{\pi \in \Pi_1} \prod_{i=1}^{n-n_1} \frac{(\pi \boldsymbol{s})_{n_1+i}}{Z - \sum_{j=1}^{n_1} (\bar{\pi} \boldsymbol{s})_j - \sum_{j=1}^{i-1} (\bar{\pi} \boldsymbol{s})_j} \\
&= \prod_{i=1}^{n_1} \frac{(\bar{\pi} \boldsymbol{s})_i}{Z - \sum_{j=1}^{i-1} (\bar{\pi} \boldsymbol{s})_j} \sum_{\pi \in \Pi_1} \prod_{i=1}^{n-n_1} \frac{(\pi \boldsymbol{s})_{n_1+i}}{Z_1 - \sum_{j=1}^{i-1} (\bar{\pi} \boldsymbol{s})_j}
\end{aligned}$$

where $Z_1 = Z - \sum_{j=1}^{n_1} (\bar{\pi} \boldsymbol{s})_j$. It follows

$$p(\bar{\pi} \mid n_1; \boldsymbol{\omega}) = \prod_{i=1}^{n_1} \frac{(\bar{\pi} \boldsymbol{s})_i}{Z - \sum_{j=1}^{i-1} (\bar{\pi} \boldsymbol{s})_j}$$

□

Lemma 5.2.1 describes the probability of drawing the elements $i \in \mathcal{S}_k$ in the order described by the subset permutation matrix $\bar{\pi}$ given that the elements in $\mathcal{S}_{<k}$ are already determined. Note that in a slight abuse of notation, we use $p(\bar{\pi} \mid n_k, \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s})$ as the probability of a subset permutation $\bar{\pi}$ given that there are n_k elements in $\mathcal{S}_k$ and thus $\bar{\pi} \in \{0,1\}^{n_k \times n}$. Additionally, we condition on the subsets $\boldsymbol{y}_{<k}$ and n_k, the size of subset $\mathcal{S}_k$. In contrast to the distribution over permutations matrices $p(\pi; \boldsymbol{s})$ in Corollary 2.4.1, we take the product over n_k terms and have a different normalization constant Z_k. Although we induce an ordering over all elements i in Definition 5.2.1, the probability $p_{\boldsymbol{y}_k}$ is invariant to intra-subset orderings of elements $i \in \mathcal{S}_k$.

Lemma 5.2.2 (Probability distribution $p_{\boldsymbol{y}_k}$ [Sut+23b]).
The probability distribution over subset assignments $p_{\boldsymbol{y}_k}$ is given by

$$p_{\boldsymbol{y}_k} := p(\boldsymbol{y}_k \mid \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s})$$

$$= p(n_k \mid n_{<k}; \boldsymbol{\omega}) \sum_{\bar{\pi} \in \Pi_{\boldsymbol{y}_k}} p(\bar{\pi} \mid n_k, \boldsymbol{y}_{<k}; \boldsymbol{s})$$

where $\Pi_{\boldsymbol{y}_k} = \{\bar{\pi} \in \{0,1\}^{n_k \times n} : \boldsymbol{y}_k = \sum_{i=1}^{n_k} \bar{\boldsymbol{\pi}}_i\}$ *and* $p(\bar{\pi} \mid n_k, \boldsymbol{y}_{<k}; \boldsymbol{s})$ *as in Lemma 5.2.1.*

Proof. We can proof the statement of Lemma 5.2.2 as follows:

$$\begin{aligned} p_{\boldsymbol{y}_k} &= p(\boldsymbol{y}_k \mid \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s}) \\ &= \sum_{n'_k} p(\boldsymbol{y}_k, n'_k \mid \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s}) && (5.47) \\ &= \sum_{n'_k} p(n'_k \mid \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s}) p(\boldsymbol{y}_k \mid n'_k, \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s}) && (5.48) \\ &= \sum_{n'_k} p(n'_k \mid n_{<k}; \boldsymbol{\omega}, \boldsymbol{s}) p(\boldsymbol{y}_k \mid n'_k, \boldsymbol{y}_{<k}; \boldsymbol{s}) && (5.49) \\ &= p(n_k \mid n_{<k}; \boldsymbol{\omega}, \boldsymbol{s}) p(\boldsymbol{y}_k \mid n_k, \boldsymbol{y}_{<k}; \boldsymbol{s}) && (5.50) \\ &= p(n_k \mid n_{<k}; \boldsymbol{\omega}) \sum_{\bar{\pi} \in \Pi_{\boldsymbol{y}_k}} p(\bar{\pi} \mid n_k, \boldsymbol{y}_{<k}; \boldsymbol{s}) && (5.51) \end{aligned}$$

Equation (5.47) holds by marginalization, where n'_k denotes the random variable that stands for the size of the subset $\mathcal{S}_k$. By Bayes' rule, we can then derive Equation (5.48). The next derivations stem from the fact that we can compute $n_{<k}$ if $\boldsymbol{y}_{<k}$ is given, as the assignments $\boldsymbol{y}_{<k}$ hold information on the size of subsets $\mathcal{S}_{<k}$. More explicitly, $n_k = \sum_{j=1}^{n} y_{kj}$. Further, $\boldsymbol{y}_k$ is independent of $\boldsymbol{\omega}$ if the size n'_k of subset $\mathcal{S}_k$ is given, leading to Equation (5.49). We further observe that $p(\boldsymbol{y}_k \mid n'_k, \boldsymbol{y}_{<k}; \boldsymbol{s})$ is only non-zero, if $n'_k = \sum_{i=1}^{n} y_{ki} = n_k$. Dropping all zero terms from the sum in Equation (5.49) thus results in Equation (5.50). Finally, by Definition 5.2.1, we know that $\boldsymbol{y}_k = \sum_{i=\nu_k+1}^{\nu_k+n_k} \boldsymbol{\pi}_i$, where $\nu_k = \sum_{\iota=1}^{k-1} n_\iota$ and $\pi \in \{0,1\}^{n \times n}$ a permutation matrix. Hence, to get $\boldsymbol{y}_k$ given $\boldsymbol{y}_{<k}$, we need to marginalize over all permutations of the elements of $\boldsymbol{y}_k$ given that the elements in $\boldsymbol{y}_{<k}$ are already ordered. This corresponds exactly to marginalizing over all subset permutation matrices $\bar{\pi}$, such that $\boldsymbol{y}_k = \sum_{i=1}^{n_k} \bar{\boldsymbol{\pi}}_i$, resulting in Equation (5.51). □

Here, we describe the set of all subset permutations $\bar{\pi}$ of elements $i \in \mathcal{S}_k$ by $\Pi_{\boldsymbol{y}_k}$. Put differently, we make $p(\boldsymbol{y}_k \mid \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s})$ invariant to the ordering of elements $i \in \mathcal{S}_k$ by marginalizing over the probabilities of subset permutations $p_{\bar{\pi}_k}$ [XE19].

We propose the DRPM $p(Y; \boldsymbol{\omega}, \boldsymbol{s})$, a differentiable and reparameterizable two-stage RPM. Since $Y = [\boldsymbol{y}_1, \dots, \boldsymbol{y}_K]^T$, we calculate $p(Y; \boldsymbol{\omega}, \boldsymbol{s})$, the distribution of the differentiable RPM, sequentially using Lemmas 5.2.1 and 5.2.2, where we leverage the PL distribution for permutation matrices $p(\pi; \boldsymbol{s})$ to describe the probability distribution over subsets $p(\boldsymbol{y}_k \mid \boldsymbol{y}_{<k}; \boldsymbol{\omega}, \boldsymbol{s})$.

Proposition 5.2.1 (Two-stage Random Partition Model [Sut+23b]).
Given Lemmas 5.2.1 and 5.2.2, the probability mass function $p(Y; \boldsymbol{\omega}, \boldsymbol{n})$ of the two-stage RPM is given by

$$p(Y; \boldsymbol{\omega}, \boldsymbol{s}) = p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_K; \boldsymbol{\omega}, \boldsymbol{s}) \tag{5.52}$$

$$= p(\boldsymbol{n}; \boldsymbol{\omega}) \sum_{\pi \in \Pi_Y} p(\pi; \boldsymbol{s}) \tag{5.53}$$

where $\Pi_Y = \{\pi : \boldsymbol{y}_k = \sum_{i=\nu_k+1}^{\nu_k+n_k} \boldsymbol{\pi}_i, k = 1, \dots, K\}$, $p(\boldsymbol{n}; \boldsymbol{\omega})$ and $p(\pi; \boldsymbol{s})$ as in Section 5.1 and corollary 2.4.1, and ν_k as in Definition 5.2.1.

Proof. From Lemmas 5.2.1 and 5.2.2, we write

$$\begin{aligned}
p(Y) &= p(\boldsymbol{y}_1, \dots, \boldsymbol{y}_K; \boldsymbol{\omega}, \boldsymbol{s}) = p(\boldsymbol{y}_1; \boldsymbol{\omega}, \boldsymbol{s}) \cdots p(\boldsymbol{y}_K \mid \{\boldsymbol{y}_j\}_{j<K}; \boldsymbol{\omega}, \boldsymbol{s}) \\
&= \left(p(n_1; \boldsymbol{\omega}) \sum_{\bar{\pi}_1 \in \Pi_{\boldsymbol{y}_1}} p(\bar{\pi}_1 \mid n_1; \boldsymbol{s}) \right) \\
&\quad \cdots \left(p(n_K \mid \{n_j\}_{j<K}; \boldsymbol{\omega}) \sum_{\bar{\pi}_K \in \Pi_{\boldsymbol{y}_K}} p(\bar{\pi}_K \mid \{n_j\}_{j \leq K}; \boldsymbol{s}) \right) \\
&= p(n_1; \boldsymbol{\omega}) \cdots p(n_K \mid \{n_K\}_{j<K}; \boldsymbol{\omega}) \\
&\quad \cdot \left(\sum_{\bar{\pi}_1 \in \Pi_{\boldsymbol{y}_1}} p(\bar{\pi}_1 \mid n_1; \boldsymbol{s}) \cdots \sum_{\bar{\pi}_K \in \Pi_{\boldsymbol{y}_K}} p(\bar{\pi}_K \mid \{n_j\}_{j \leq K}; \boldsymbol{s}) \right) \\
&= p(\boldsymbol{n}; \boldsymbol{\omega}) \left(\sum_{\bar{\pi}_1 \in \Pi_{\boldsymbol{y}_1}} \cdots \sum_{\bar{\pi}_K \in \Pi_{\boldsymbol{y}_K}} p(\bar{\pi}_1 \mid n_1; \boldsymbol{s}) \cdots p(\bar{\pi}_K \mid \{n_j\}_{j \leq K}; \boldsymbol{s}) \right) \\
&= p(\boldsymbol{n}; \boldsymbol{\omega}) \sum_{\pi \in \Pi_Y} p(\pi \mid \boldsymbol{n}; \boldsymbol{s}) \\
&= p(\boldsymbol{n}; \boldsymbol{\omega}) \sum_{\pi \in \Pi_Y} p(\pi; \boldsymbol{s})
\end{aligned}$$

□

5.2.1.1 *Approximating the distribution over RPMs*

The number of permutations per subset $|\Pi_{\boldsymbol{y}_k}|$ scales factorially with the subset size n_k, i.e. $|\Pi_{\boldsymbol{y}_k}| = n_k!$. Consequently, the number of valid permutation matrices $|\Pi_Y|$ is given as a function of $\boldsymbol{n}$, i.e.

$$|\Pi_Y| = \prod_{k=1}^{K} |\Pi_{\boldsymbol{y}_k}| = \prod_{k=1}^{K} n_k! \tag{5.54}$$

Although Proposition 5.2.1 describes a well-defined distribution for $p(Y;\boldsymbol{\omega},\boldsymbol{s})$, it is in general computationally intractable due to Equation (5.54). In practice, we thus approximate $p(Y;\boldsymbol{\omega},\boldsymbol{s})$ using the following Lemma.

Lemma 5.2.3 (Upper and lower bounds for $p(Y;\boldsymbol{\omega},\boldsymbol{s})$ [Sut+23b])**.** *$p(Y;\boldsymbol{\omega},\boldsymbol{s})$ can be upper and lower bounded as follows*

$$\forall\, \pi \in \Pi_Y : \; p(Y;\boldsymbol{\omega},\boldsymbol{s}) \geq p(\boldsymbol{n};\boldsymbol{\omega}) p(\pi;\boldsymbol{s}) \tag{5.55}$$

$$p(Y;\boldsymbol{\omega},\boldsymbol{s}) \leq |\Pi_Y| p(\boldsymbol{n};\boldsymbol{\omega}) \max_{\pi} p(\pi;\boldsymbol{s}) \tag{5.56}$$

Proof. Since $p(\pi;\boldsymbol{s})$ is a probability distribution we know that

$$\forall\, \pi \in \{0,1\}^{n\times n} : \quad p(\pi;\boldsymbol{s}) \geq 0 \tag{5.57}$$

Thus, it follows directly that:

$$\forall\, \pi \in \Pi_Y : \quad p(Y;\boldsymbol{\omega},\boldsymbol{s}) = p(\boldsymbol{n};\boldsymbol{\omega}) \sum_{\pi' \in \Pi_Y} p(\pi';\boldsymbol{s}) \geq p(\boldsymbol{n};\boldsymbol{\omega}) p(\pi;\boldsymbol{s}),$$

proving Equation (5.55).
On the other hand, we can prove Equation (5.56) by:

$$\begin{aligned} p(Y;\boldsymbol{\omega},\boldsymbol{s}) &= p(\boldsymbol{n};\boldsymbol{\omega}) \sum_{\pi' \in \Pi_Y} p(\pi';\boldsymbol{s}) \\ &\leq p(\boldsymbol{n};\boldsymbol{\omega}) \sum_{\pi' \in \Pi_Y} \max_{\pi \in \Pi_Y} p(\pi;\boldsymbol{s}) \\ &= p(\boldsymbol{n};\boldsymbol{\omega}) \max_{\pi \in \Pi_Y} p(\pi;\boldsymbol{s}) \sum_{\pi' \in \Pi_Y} 1 \\ &= |\Pi_Y| \cdot p(\boldsymbol{n};\boldsymbol{\omega}) \max_{\pi \in \Pi_Y} p(\pi;\boldsymbol{s}) \\ &\leq |\Pi_Y| \cdot p(\boldsymbol{n};\boldsymbol{\omega}) \max_{\pi} p(\pi;\boldsymbol{s}) \end{aligned}$$

We can compute the maximum probability $\max_\pi p(\pi;\boldsymbol{s})$ with the probability of the permutation matrix $f_\pi(\boldsymbol{s})$, which sorts the unperturbed scores in decreasing order. □

Note that from Corollary 2.4.1 we see that $\max_\pi p(\pi; \boldsymbol{s}) = p(f_\pi(\boldsymbol{s}); \boldsymbol{s})$, since $p(\pi; \boldsymbol{s})$ is maximal if $\pi = f_\pi(\boldsymbol{s})$, i.e. when sorting the unperturbed scores $\boldsymbol{s}$.

5.2.1.2 *Differentiable two-stage RPM*

The following Lemma guarantees differentiability, allowing us to integrate the proposed DRPM into gradient-based optimization methods:

Lemma 5.2.4 (DRPM [Sut+23b]). *A two-stage RPM as proposed in Proposition 5.2.1 is differentiable and reparameterizable if the distribution over subset sizes $p(\boldsymbol{n}; \boldsymbol{\omega})$ and the distribution over orderings $p(\pi; \boldsymbol{s})$ are differentiable and reparameterizable.*

Proof. To prove that our two-stage RPM is differentiable, we need to prove that we can compute gradients for the bounds in Lemma 5.2.3 and to provide a reparameterization scheme for the two-stage approach in Definition 5.2.1.

Gradients for the bounds: Since we assume that $p(\boldsymbol{n}; \boldsymbol{\omega})$ and $p(\pi; \boldsymbol{s})$ are differentiable and reparameterizable, we only need to show that we can compute $|\Pi_Y|$ and $\max_{\tilde{\pi}} p(\tilde{\pi}; \boldsymbol{s})$ in a differentiable manner to prove that the bounds in Lemma 5.2.3 are differentiable. By definition (see Section 5.2.1.1),

$$|\Pi_Y| = \prod_{k=1}^{K} |\Pi_{\boldsymbol{y}_k}| = \prod_{k=1}^{K} n_k!.$$

Hence, $|\Pi_Y|$ can be computed given a reparametrized n_k, which is provided by the reparametrization trick for the MVHG $p(\boldsymbol{n}; \boldsymbol{\omega})$. Further, from Equation (2.63), we immediately see that the most probable permutation is given by the order induced by sorting the original, unperturbed scores $\boldsymbol{s}$ from highest to lowest. This implies that $\max_{\tilde{\pi}} p(\tilde{\pi}; \boldsymbol{s}) = p(\pi_{\boldsymbol{s}}; \boldsymbol{s})$, which we can compute due to $p(\pi_{\boldsymbol{s}}; \boldsymbol{s})$ being differentiable according to our assumptions.

Reparametrization of the two-stage approach: Given reparametrized versions of $\boldsymbol{n}$ and π, we compute a partition as follows:

$$\boldsymbol{y}_k = \sum_{i=\nu_k+1}^{\nu_k+n_k} \boldsymbol{\pi}_i, \qquad \text{where} \;\; \nu_k = \sum_{\iota=1}^{k-1} n_\iota \tag{5.58}$$

The challenge here is that we need to be able to backpropagate through n_k, which appears as an index in the sum. Let $\boldsymbol{\alpha}_k = \{0,1\}^n$, such that

$$(\boldsymbol{\alpha}_k)_i = \begin{cases} 1 & \text{if } \nu_k < i \leq \nu_{k+1} \\ 0 & \text{otherwise} \end{cases}$$

Given such $\boldsymbol{\alpha}_k$, we can rewrite Equation (5.58) with

$$\boldsymbol{y}_k = \sum_{i=1}^{n} (\boldsymbol{\alpha}_k)_i \boldsymbol{\pi}_i. \tag{5.59}$$

While this solves the problem of propagating through sum indices, it is not clear how to compute $\boldsymbol{\alpha}_k$ in a differentiable manner. Similar to other works on continuous relaxations [JGP16; MMT17], we can compute a relaxation of $\boldsymbol{\alpha}_k$ by introducing a temperature τ. Let us introduce auxiliary function $f : \mathbb{N} \to [0,1]^n$, that maps an integer x to a vector with entries

$$f_i(x;\tau) = \sigma\left(\frac{x - i + \epsilon}{\tau}\right),$$

such that $f_i(x;\tau) \approx 0$ if $\frac{x-i}{\tau} < 0$ and $f_i(x;\tau) \approx 1$ if $\frac{x-i}{\tau} \geq 0$. Note that $\sigma(\cdot)$ is the standard sigmoid function, and $\epsilon << 1$ is a small positive constant to break the tie at $\sigma(0)$. We then compute an approximation of $\boldsymbol{\alpha}_k$ with

$$\tilde{\boldsymbol{\alpha}}_k(\tau) = f(\nu_k;\tau) - f(\nu_{k-1};\tau),$$

$\tilde{\boldsymbol{\alpha}}_k(\tau) \in [0,1]^n$. Then, for $\tau \to 0$ we have $\tilde{\boldsymbol{\alpha}}_k(\tau) \to \boldsymbol{\alpha}_k$. In practice, we cannot set $\tau = 0$ since this would amount to a division by 0. Instead, we can apply the straight-through estimator [BLC13] to the auxiliary function $f(x;\tau)$ in order to get $\tilde{\boldsymbol{\alpha}}_k \in \{0,1\}^n$ and use it to compute Equation (5.59). □

5.2.2 *Sampling partitions from the DRPM*

To sample a partition Y from our DRPM, i.e. $Y \sim p(Y;\boldsymbol{\omega},\boldsymbol{s})$, we use the two methods from Sections 2.4.2.4 and 5.1, which introduced differentiable and reparameterizable distributions for $p(\pi;\boldsymbol{s})$ and $p(\boldsymbol{n};\boldsymbol{\omega})$ respectively [Gro+19; Sut+23a]. We thus propose the following sampling procedure:

1. sample $\pi \sim p(\pi;\boldsymbol{s})$
2. sample $\boldsymbol{n} \sim p(\boldsymbol{n};\boldsymbol{\omega})$
3. calculate $Y = f(\pi,\boldsymbol{n})$ according to Definition 5.2.1 by summing the rows of π according to $\boldsymbol{n}$. Hence, $\forall\ k = 1,\ldots,K$ we have:

$$\boldsymbol{y}_k = \sum_{i=\nu_k+1}^{\nu_k+n_k} \boldsymbol{\pi}_i \quad \text{where} \quad \nu_k = \sum_{j<k} n_j \tag{5.60}$$

Using this two-stage procedure, we can infer and resample partitions in a differentiable and reparameterizable way. Additionally, due to Proposition 5.2.1 and lemma 5.2.3, we are also able to efficiently estimate $p(Y; \boldsymbol{\omega}, \boldsymbol{s})$ for a given DRPM sample Y.

In summary, we introduce an efficient model to deterministically and probabilistically learn partitions end-to-end. In contrast to previous RPMs, which often need exponentially many distribution parameters [Pla75], the proposed DRPM needs only $(n + K)$ parameters to create an RPM for n elements: the score parameters $\boldsymbol{s} \in \mathbb{R}^n_+$ and the group importance parameters $\boldsymbol{\omega} \in \mathbb{R}^K_+$. Our DRPM enables us to integrate partition models in any gradient-based optimization pipeline. In the following experiments, we present how to use the DRPM with both deterministic and probabilistic models.

5.3 EXPERIMENTS

We demonstrate the versatility and effectiveness of the proposed DRPM in three different experiments. First, we propose a novel generative clustering method based on the DRPM and validate it against state-of-the-art differentiable clustering methods. Then, we apply the DRPM to multitask learning (MTL), where the DRPM enables an adaptive neural network architecture that partitions layers based on task difficulty. Finally, we demonstrate how the DRPM can infer shared and independent generative factors under weak supervision.

In Chapters 1, 3 and 4, we discuss the limitations of implying restrictive assumptions on the group structure in a weakly-supervised setting. The proposed DRPM shows the benefits of inferring shared and independent generative factors during optimization, equivalent to learning the group structure.

5.3.1 *Variational Clustering with Random Partition Models*

Our first experiment introduces a new clustering method based on the DRPM.

5.3.1.1 *Method*

Because of the intractable generative process of the dataset, we perform variational clustering. We assume that each sample $\boldsymbol{x}^{(i)}$ of a dataset $X =$

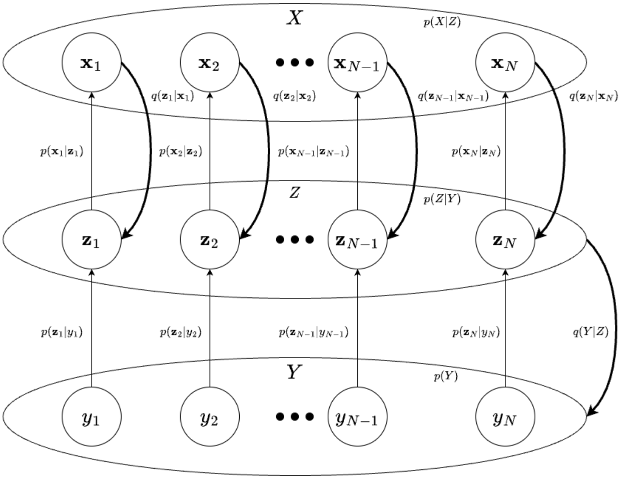

X
$p(X|Z)$
$\mathbf{x}_1$
$\mathbf{x}_2$
$\mathbf{x}_{N-1}$
$\mathbf{x}_N$
$q(\mathbf{z}_1|\mathbf{x}_1)$
$q(\mathbf{z}_2|\mathbf{x}_2)$
$q(\mathbf{z}_{N-1}|\mathbf{x}_{N-1})$
$q(\mathbf{z}_N|\mathbf{x}_N)$
$p(\mathbf{x}_1|\mathbf{z}_1)$
$p(\mathbf{x}_2|\mathbf{z}_2)$
$p(\mathbf{x}_{N-1}|\mathbf{z}_{N-1})$
$p(\mathbf{x}_N|\mathbf{z}_N)$
Z
$p(Z|Y)$
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_{N-1}$
$\mathbf{z}_N$
$p(\mathbf{z}_1|y_1)$
$p(\mathbf{z}_2|y_2)$
$p(\mathbf{z}_{N-1}|y_{N-1})$
$p(\mathbf{z}_N|y_N)$
$q(Y|Z)$
Y
$p(Y)$
y_1
y_2
y_{N-1}
y_N

Figure 5.8: Graphical model of the DRPM clustering model. Generative paths are marked with thin arrows, whereas inference is in bold.

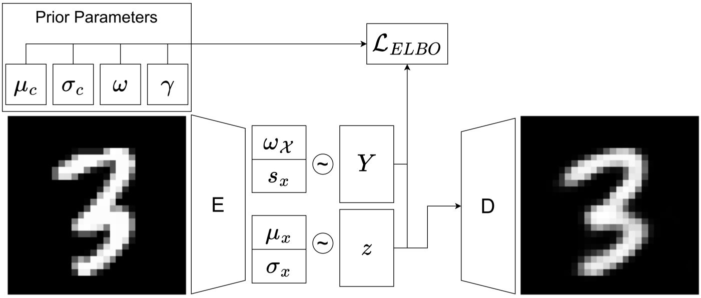


Figure 5.9: Architecture of the DRPM clustering model.

	NMI	ARI	ACC
GMM	0.32±0.01	0.22±0.02	0.41±0.01
LATENT GMM	0.86±0.02	0.83±0.06	0.88±0.07
VADE	0.84±0.01	0.76±0.05	0.82±0.04
DRPM-VC	**0.89**±0.01	**0.88**±0.03	**0.94**±0.02

(a) MNIST

	NMI	ARI	ACC
GMM	0.49±0.01	0.33±0.00	0.44±0.01
LATENT GMM	0.60±0.00	0.47±0.01	0.62±0.01
VADE	0.56±0.02	0.40±0.04	0.56±0.03
DRPM-VC	**0.64**±0.00	**0.51**±0.01	**0.65**±0.00

(b) FMNIST

TABLE 5.1: We compare the clustering performance of the DRPM-VC on test sets of MNIST and FMNIST between Gaussian Mixture Models (GMM), GMM in latent space (Latent GMM), and Variational Deep Embedding (VADE). We measure performance in terms of the Normalized Mutual Information (NMI), Adjusted Rand Index (ARI), and cluster accuracy (ACC) over five seeds and put the best model in bold.

$\{\boldsymbol{x}^{(i)}\}_{i=1}^{N} \in \mathbb{R}^{N \times p}$ is generated by a latent vector $\boldsymbol{z}^{(i)}$. Instead of assuming a single Gaussian prior $\mathcal{N}(\boldsymbol{\mu}, diag(\boldsymbol{\sigma}))$ for the latent vectors like in vanilla variational autoencoders [VAE, KW14], we assume every $\boldsymbol{z}^{(i)}$ to be sampled from one of K different latent Gaussian distributions $\mathcal{N}(\boldsymbol{\mu}_k, diag(\boldsymbol{\sigma}_k)), k \in \{1, \dots, K\}$. The assignments $\boldsymbol{y}^{(i)}$ of every $\boldsymbol{z}^{(i)}$ to their respective clusters are then distributed according to an RPM, potentially resulting in dependencies between latent vectors $\boldsymbol{z}^{(i)}$. In the following, let $Z = \{\boldsymbol{z}^{(1)}, ..., \boldsymbol{z}^{(N)}\} \in \mathbb{R}^{d \times N}$, and $Y = \{\boldsymbol{y}^{(1)}, ..., \boldsymbol{y}^{(N)}\} \in \{0, 1\}^{K \times N}$ contain the respective latent vectors and cluster assignments for each sample of a given dataset $X = \{\boldsymbol{x}^{(1)}, ..., \boldsymbol{x}^{(N)}\}$ with N samples. Every $\boldsymbol{y}^{(i)}$ is a one-hot vector where the index set to one defines the cluster assignment.

We illustrate the generative process in the graphical model in Figure 5.8 and derive it as follows: First, we sample the cluster assignments Y from an RPM, i.e., $Y \sim p(Y; \boldsymbol{\omega}, \boldsymbol{s})$. Given Y, we can sample the latent variables Z, where $\boldsymbol{z}^{(i)} \sim \mathcal{N}(\boldsymbol{\mu}_{k^{(i)}}, diag(\boldsymbol{\sigma}_{k^{(i)}}))$ where $k^{(i)} = \arg\max_k \boldsymbol{y}^{(i)}$ in a slight abuse of notation. Finally, we sample X, where $\boldsymbol{x}^{(i)} \sim p_\theta(\boldsymbol{x}^{(i)} \mid \boldsymbol{z}^{(i)})$ is sampled from the data distribution $p_\theta(\cdot)$ given the respective latent vector $\boldsymbol{z}^{(i)}$. In the derivation of the ELBO $\mathcal{L}(\theta, \phi; X)$, we remove the super scripts (i) to remove clutter.

Assuming this generative process and using Bayes' rule and Jensen's inequality, we derive the following ELBO $\mathcal{L}(\theta, \phi; X)$ for $p(X)$ as follows:

$$\begin{aligned}
\log p(X) = {} & \log \left(\int \sum_{Y} p_\theta(X, Y, Z) dZ \right) && (5.61) \\
& \geq \mathbb{E}_{q_\phi(Z,Y|X)} \left[\log \left(\frac{p(X \mid Z) p_\theta(Z \mid Y) p_\theta(Y)}{q_\phi(Z, Y \mid X)} \right) \right] && (5.62) \\
& =: \mathcal{L}(\theta, \phi; X) && (5.63)
\end{aligned}$$

where θ are the amortization parameters of the generative distribution. We then assume that we can factorize the approximate posterior as follows:

$$q_\phi(Z, Y \mid X) = q_\phi(Y \mid X) \prod_{\boldsymbol{x} \in X} q_\phi(\boldsymbol{z} | \boldsymbol{x}), \tag{5.64}$$

where ϕ are the amortization parameters of the variational distribution. Note that while we do assume conditional independence between $\boldsymbol{z}$ given its corresponding $\boldsymbol{x}$, we model $q_\phi(Y \mid X)$ with the DRPM and do not have to assume conditional independence between different cluster assignments. This has the advantage that we directly leverage dependencies between samples from the dataset. Hence, we can rewrite the ELBO as follows:

$$\begin{aligned}
\mathcal{L}(\theta, \phi; X) = {} & \mathbb{E}_{q_\phi(Z|X)} \left[\log(p_\theta(X \mid Z)) \right] \\
& - \mathbb{E}_{q_\phi(Y|X)} \left[D_{\mathrm{KL}}[q_\phi(Z \mid X) \mid\mid p_\theta(Z \mid Y)] \right] \\
& - D_{\mathrm{KL}}[q_\phi(Y \mid X) \mid\mid p_\theta(Y)] && (5.65) \\
= {} & \sum_{\boldsymbol{x} \in X} \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}) \right] \\
& - \sum_{\boldsymbol{x} \in X} \mathbb{E}_{q_\phi(Y|X)} \left[D_{\mathrm{KL}}[q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p_\theta(\boldsymbol{z} \mid Y)] \right] \\
& - D_{\mathrm{KL}}[q_\phi(Y \mid X) \mid\mid p_\theta(Y)] && (5.66)
\end{aligned}$$

Note that computing $D_{\mathrm{KL}}[q(Y \mid X) \mid\mid p(Y)]$ directly is computationally intractable, and we need to upper bound it according to Lemma 5.2.3, i.e.

$$D_{\mathrm{KL}}[q_\phi(Y \mid X) \mid\mid p_\theta(Y)] \leq \mathbb{E}_{q_\phi(Y|X)}\left[\log \frac{|\Pi_Y| \cdot q_\phi(\boldsymbol{n}; \boldsymbol{\omega}(X))}{p_\theta(\boldsymbol{n}; \boldsymbol{\omega}) p(\pi_Y; \boldsymbol{s})}\right] + \log\left(\max_\pi q_\phi(\pi; \boldsymbol{s}(X))\right), \tag{5.67}$$

where π_Y is any $\pi \in \Pi_Y$.

5.3.1.2 *Dataset & Implementation*

We train and compare our method to three baselines on two different datasets, the MNIST [LeC+98] and Fashion-MNIST [FMNIST, XRV17] dataset. We use a simple, fully connected autoencoder architecture for our clustering experiments. Figure 5.9 shows a high-level overview of its structure. We have a fully connected encoder E with four layers mapping the input to 500, 500, 500, and 2000 neurons, respectively. We then compute each parameter by passing the encoder output through a linear layer and mapping to the respective parameter dimension in the last layer. In our experiments, we use a latent dimension size of 10, hence $\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i \in \mathbb{R}^{10}$. To learn dependencies between samples, we map the last layer of the $\boldsymbol{\omega}_X$ block in Figure 5.9 to K, where K is the number of dimensions. We then apply a softmax activation for each sample, average the resulting vector over the batch, and take the logarithm since we want to model $\log \boldsymbol{\omega}_X$. To compute the score s_i of a sample $\boldsymbol{x}$, we map the last layer to dimension K and apply a softmax activation to that representation to compute the intermediate representation $\boldsymbol{r} \in [0, 1]^K$. Since we know that the scores for samples in the same cluster should be approximately equal, we compute s_i by $\log(s_i) = \lambda \sum_{k=1}^{K} k \cdot r_k$, where λ is a learnable parameter that is responsible for scaling the scores to an appropriate magnitude. Note that we thus compute s_i per sample independently of the other samples in the batch. Finally, once we resample $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{\mu}_i, diag(\boldsymbol{\sigma}_i))$, we pass it through a fully connected decoder D with four layers mapping $\boldsymbol{z}$ to 2000, 500, and 500 neurons in the first three layers and then finally back to the input dimension in the last layer to end up with the reconstructed sample $\hat{\boldsymbol{x}}$. Figure 5.9 shows a simplified architecture model used for the DRPM clustering method.

Based on the derivations in Section 5.3.1.1, we use the following loss to train the clustering experiment:

$$\mathcal{L}(\theta, \phi; X) = \sum_{\boldsymbol{x} \in X} \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})] \tag{5.68}$$

$$
\begin{aligned}
&- \sum_{\boldsymbol{x} \in X} \mathbb{E}_{q_\phi(Y|X)} \left[D_{\mathrm{KL}}[q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p_\theta(\boldsymbol{z} \mid Y)] \right] && (5.69)\\
&- \mathbb{E}_{q_\phi(Y|X)} \left[\log \frac{|\Pi_Y| \cdot q_\phi(\boldsymbol{n}; \boldsymbol{\omega}(X))}{p_\theta(\boldsymbol{n}; \boldsymbol{\omega}) p_\theta(\pi_Y; \boldsymbol{s})} \right] && (5.70)\\
&- \log \left(\max_\pi q_\phi(\pi; \boldsymbol{s}(X)) \right) && (5.71)
\end{aligned}
$$

For GST-based experiments, the temperature is – especially at the beginning of training – a sensitive parameter [Gro+19; JGP16]. In order to resample $\boldsymbol{n}$ and π, we need to apply temperature annealing [Gro+19; JGP16; Sut+23a]. To do this, we apply the exponential schedule that was originally proposed together with the GST in Jang, Gu, and Poole [JGP16], i.e.

$$\tau = \max(\tau_{final}, \exp(-rt)), \tag{5.72}$$

where t is the current training step and r is the annealing rate. For our experiments, we choose the following annealing configuration

$$
\begin{aligned}
r &= \frac{\log(\tau_{final}) - \log(\tau_{init})}{\#\text{steps}} && (5.73)\\
\tau_{init} &= 1.0 && (5.74)\\
\tau_{final} &= 0.5 && (5.75)\\
\#\text{steps} &= 100000 && (5.76)
\end{aligned}
$$

in order to anneal the temperature τ from 1.0 to 0.5 over 100000 training steps.

Similar to Jiang et al. [Jia+16], we quickly realized that proper initialization of the cluster parameters and network weights is crucial for variational clustering. In our experiments, we pre-trained the autoencoder structure by adapting the contrastive loss of [Li+22], as they demonstrated that their representations manage to retain clusters in low-dimensional space. Further, we also added a reconstruction loss to initialize the decoder properly. To initialize the prior parameters, we fit a GMM to the pre-trained embeddings of the training set and took the resulting Gaussian parameters to initialize our priors. Note that we used the same initialization across all baselines.

To optimize the DRPM-VC and VADE in our experiments, we used the AdamW [LH17] optimizer with a learning rate of 0.0001 with a batch size of 256 for 1024 epochs. Appendix A.4.1 provides more details on the training specifications. During initial experiments with the DRPM-VC, we realized that the pre-trained weights of the encoder would often lose the learned structure in the first couple of training epochs. We suspect this to be an

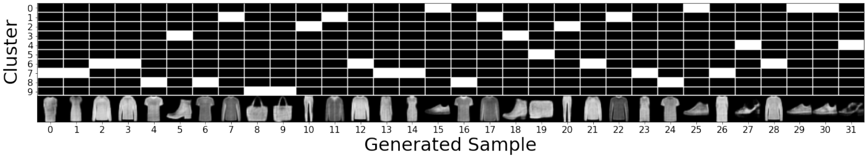


FIGURE 5.10: A sample drawn from a DRPM-VC model trained on FMNIST. On top is the sampled partition with the cluster assignments, and on the bottom are generated images corresponding to the sampled assignment matrix. The DRPM-VC learns consistent clusters for different pieces of clothing and can generate new samples of each cluster with great variability.

artifact of instabilities induced by temperature annealing. To deal with these problems, we decided to freeze the first three layers of the encoder when training the DRPM-VC, giving us much better results.

5.3.1.3 *Experiments & Results*

Two of the baselines are based on a Gaussian Mixture Model, where one is directly trained on the original data space (GMM), whereas the other takes the embeddings from a pre-trained encoder as input (Latent GMM). The third baseline is variational deep embedding [VADE, Jia+16], which is similar to the DRPM-VC but assumes *i.i.d.* categorical cluster assignments. For all methods except GMM, we use the weights of a pre-trained encoder to initialize the models and priors at the start of training. All baseline methods assume *i.i.d.* sampling of the latent clusters through a categorical distribution with categorical weights γ, i.e., $\boldsymbol{y}^{(i)} \sim Cat(\gamma)$.
In Table 5.1, we compare the normalized mutual information [NMI, Dan+05; LFK09; MGH11] and adjusted rand index [ARI, HA85; Ran71; VEB09] scores of the different baselines to our DRPM model. As can be seen, we outperform all baselines, indicating that modeling the inherent dependencies implied by finite datasets benefits the performance of variational clustering. While achieving decent clustering performance, another benefit of variational clustering methods is that their reconstruction-based nature intrinsically allows unsupervised conditional generation. In Figure 5.10, we present the result of sampling a partition and the corresponding generations from the respective clusters after training the DRPM-VC on FMNIST. The model produces coherent generations despite not having access to labels, allowing us to investigate the structures learned by the model more closely.

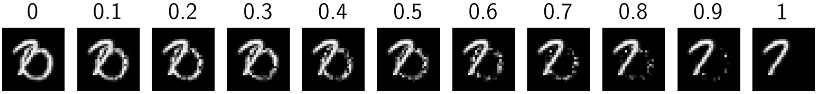


FIGURE 5.11: Samples from the noisyMultiMNIST dataset with increasing noise ratio in the right task.

5.3.2 *Multitask Learning*

Many machine learning applications aim to solve specific tasks, optimizing for a single objective while ignoring potentially helpful information from related tasks. In multitask learning [MTL, Car93; CS96], we share representations between related tasks to improve the generalization on all tasks [Car93]. Hard parameter sharing [CS96] is helpful in many scenarios as the tasks share the same base network with only task-specific sub-networks. For MTL methods utilizing hard parameter sharing, the loss is simply the sum of the losses of all tasks. Recent works [Kur+22; Xin+22] show that using a convex combination of task losses is difficult to outperform if the task losses are scaled according to the task difficulty. Although challenging to outperform, finding the optimal weights is a tedious and inefficient approach to MTL. A more automated way of weighting multiple tasks would thus be vastly appreciated.

In this experiment, we demonstrate how the DRPM, in its deterministic setting, achieves precisely that by automatically learning task importance and assigning specific neurons to different output layers, therefore specializing them to particular tasks.

5.3.2.1 *Dataset, Implementation & Training*

We use the MTL pipeline from Sener and Koltun [SK18] and perform experiments on MultiMNIST [SFH17]. In MultiMNIST, there are two overlapping MNIST digits on every image. Hence, the two tasks, classification of the digit on the left- and the right-hand side (see Figure 5.11 for an example), are of approximately equal difficulty by default (left-most image in Figure 5.11). In Sutter et al. [Sut+23b], we introduce the *noisyMultiMNIST* dataset to increase the difficulty of one of the two tasks. There, we control the task difficulty by adding salt and pepper noise to one of the two digits, subsequently increasing the difficulty of that task with increasing noise ratios (from left to right in Figure 5.11). Varying the noise, we evaluate

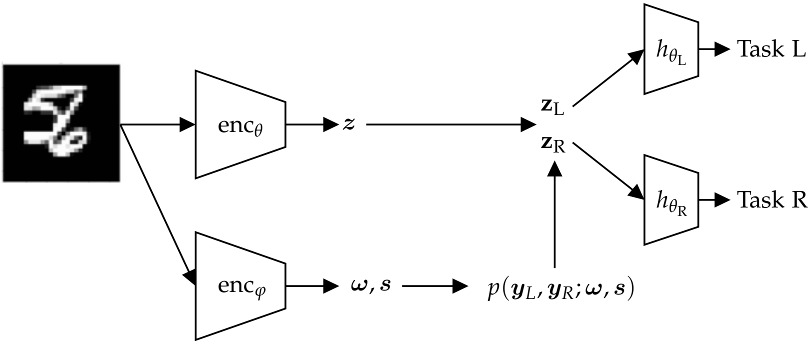


FIGURE 5.12: Overview of the multitask learning pipeline of the DRPM-MTL method.

how our DRPM-MTL method adapts to imbalanced difficulties, where one usually has to tediously search for optimal loss weights to reach a good performance. We compare DRPM-MTL to a unitary loss scaling method (ULS), which weights the two tasks equally. ULS is a standard baseline for MTL methods [see e. g., Kur+22; SK18; Xin+22].

The multitask loss function for the *MultiMNIST* dataset is

$$\mathbb{L} = w_L \mathbb{L}_L + w_R \mathbb{L}_R \tag{5.77}$$

where w_L and w_L are the loss weights, and $\mathbb{L}_L$ and $\mathbb{L}_R$ are the individual loss terms for the respective tasks L and R. We are interested in showing the effects of equal task weights if the tasks increasingly diverge in difficulty. Hence, in our experiments, we set the same task weights for DRPM-MTL and ULS and for all dataset versions, i. e., $w_L = w_R = 0.5$. On the other hand, the DRPM-MTL method does not need additional weighting of loss terms as it should adapt the selection of task-specific neurons to the task difficulty.

The task losses are defined as cross-entropy losses

$$\mathbb{L}_t = -\sum_{c=1}^{C_t} \boldsymbol{gt}_c \log \boldsymbol{p}_c = -\boldsymbol{gt}^T \log \boldsymbol{p} \tag{5.78}$$

where $C_L = C_R = 10$ for MultiMNIST, $\boldsymbol{gt}$ is a one-hot encoded label vector and $\boldsymbol{p}$ is a categorical vector of estimated class assignments probabilities, i. e. $\sum_c \boldsymbol{p}_c = 1$.

The predictions for the individual tasks $\boldsymbol{p}_t$ are given as

$$\boldsymbol{p}_t = h_{\theta_t}(\boldsymbol{z}), \quad \text{where} \tag{5.79}$$

$$z = \text{enc}_\theta(\boldsymbol{x}) \tag{5.80}$$

for a sample $\boldsymbol{x} \in X$ (see also Figure 5.12). We use an adaptation of the LeNet-5 architecture [LeC+98] to the multitask learning problem [SK18]. Both DRPM-MTL and ULS use the same network $\text{enc}_\theta(\cdot)$, where the neurons are shared between the two tasks up to some layer, after which the network branches into two task-specific sub-networks that perform the classifications (see Figures 5.12 and A.1). Figure A.1 shows the MTL pipeline used for the ULS method.

Unlike the ULS method, the task-specific networks of DRPM-MTL predict the digit using only a subset of $\boldsymbol{z}$. DRPM-MTL uses the following prediction scheme (see Figure 5.12)

$$\boldsymbol{p}_t = h_{\theta_t}(\boldsymbol{z}_t), \quad \text{where} \tag{5.81}$$

$$\boldsymbol{z}_t = \boldsymbol{z} \odot \boldsymbol{y}_t \tag{5.82}$$

$$\boldsymbol{y}_t = p(Y; \boldsymbol{\omega}(\boldsymbol{x}), \boldsymbol{s}(\boldsymbol{x}))_t = p(Y; \text{enc}_\varphi(\boldsymbol{x}))_t \tag{5.83}$$

The DRPM-MTL encoder first predicts a latent representation $\mathbf{z} \leftarrow \text{enc}_\theta(\boldsymbol{x})$, where $\boldsymbol{x}$ is the input image. Using the same encoder architecture but different parameters φ, we predict a partitioning encoding $\boldsymbol{z}' \leftarrow \text{enc}_\varphi(\boldsymbol{x})$. With a single linear layer per DRPM log-parameter $\log \boldsymbol{\omega}$ and $\log \boldsymbol{s}$ are computed. Next we infer the partition masks $\boldsymbol{y}_\text{L}, \boldsymbol{y}_\text{R} \sim p(\boldsymbol{y}_\text{L}, \boldsymbol{y}_\text{R}; \boldsymbol{\omega}, \boldsymbol{s})$. We then feed the masked latent representations $\boldsymbol{z}_\text{L} \leftarrow \boldsymbol{z} \odot \boldsymbol{y}_\text{L}$ and $\boldsymbol{z}_\text{R} \leftarrow \boldsymbol{z} \odot \boldsymbol{y}_\text{R}$ into the task specific classification networks $h_{\theta_\text{L}}(\boldsymbol{z}_\text{L})$ and $h_{\theta_\text{R}}(\boldsymbol{z}_\text{R})$ respectively to obtain the task specific predictions. Since the two tasks in the MultiMNIST dataset are similar, the task-specific networks h_{θ_L} and h_{θ_R} share the same architecture, but have different parameters.

5.3.2.2 *Experiments & Results*

Recent works [Kur+22; Xin+22] show that maximum performance can be reached by perfectly scaling the task losses. I.e., in case of equal difficulty of the two tasks, a classifier with equal weighting of the two classification losses serves as an upper bound in terms of performance. On the other hand, weighting losses equally is not ideal for increasing noise levels and would require a reweighting of the loss terms to adapt their weights to the task difficulty. Instead of weighting the task losses, we assume that the representation size per task implicitly serves as a weighting of task losses depending on their difficulty or importance. DRPM-MTL allows us to partition the weights of the last shared layer using gradient-based

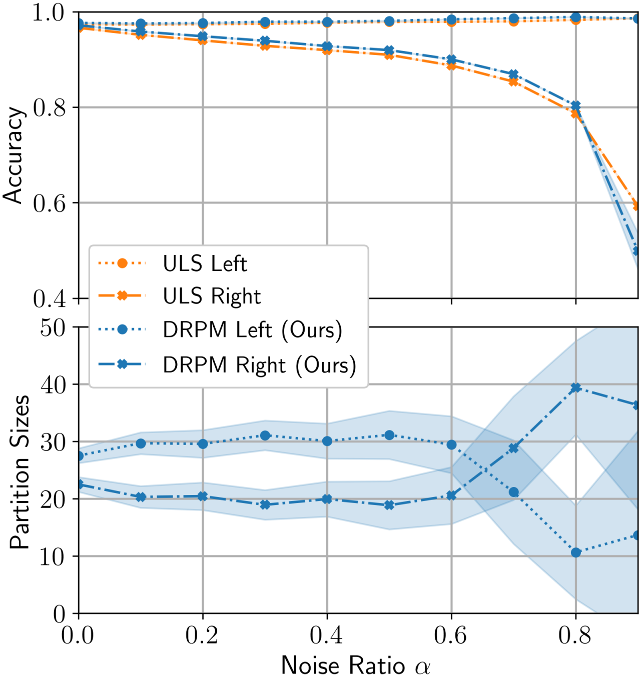
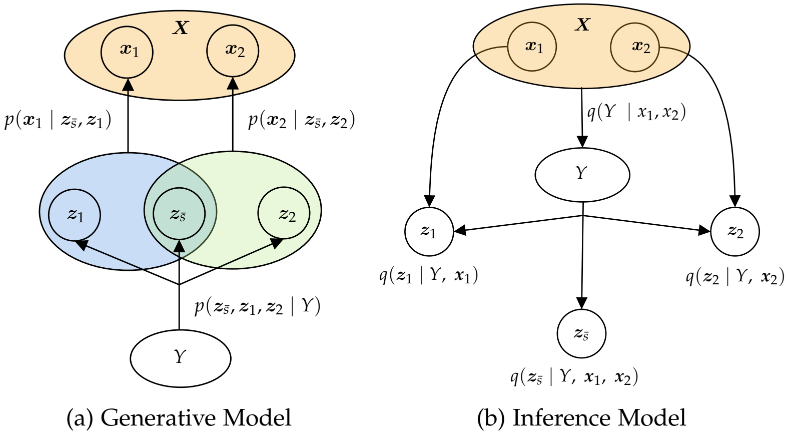


FIGURE 5.13: Results for the noisyMultiMNIST experiment. We compare the task accuracy of the two methods, ULS and the proposed DRPM-MTL, for five seeds and different noise ratios α (upper plot). DRPM-MTL can reach higher accuracy for most noise levels α. DRPM-MTL intuitively assigns the number of dimensions per task according to their difficulty (lower plot).


$\boldsymbol{X}$
$\boldsymbol{x}_1$
$\boldsymbol{x}_2$
$p(\boldsymbol{x}_1 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1)$
$p(\boldsymbol{x}_2 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_2)$
$\boldsymbol{z}_1$
$\boldsymbol{z}_{\bar{s}}$
$\boldsymbol{z}_2$
$p(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2 \mid Y)$
Y
$q(Y \mid x_1, x_2)$
$q(\boldsymbol{z}_1 \mid Y, \boldsymbol{x}_1)$
$q(\boldsymbol{z}_2 \mid Y, \boldsymbol{x}_2)$
$q(\boldsymbol{z}_{\bar{s}} \mid Y, \boldsymbol{x}_1, \boldsymbol{x}_2)$

(a) Generative Model

(b) Inference Model

FIGURE 5.14: Graphical Model for the weakly-supervised experiment. Following the assumption of weakly-supervised data, we use a random partition model for inference and generation of shared $\boldsymbol{z}_{\bar{s}}$ and independent $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ factors.

optimization such that only a subset of the neurons are used for every task. Note the difference to the ULS method, where the task-specific branches access all neurons of the last shared layer.

In contrast to the other experiments (Sections 5.3.1 and 5.3.3), we use DRPM-MTL as a deterministic model by inferring Y in the two-step procedure of Definition 5.2.1. We learn $\boldsymbol{n}$ and π by adapting the deterministic versions of the PL and MVHG distributions [Gro+19; Sut+23a]. We evaluate the DRPM-MTL method concerning its classification accuracy on the two tasks and the inferred subset sizes per task for different noise ratios $\alpha \in \{0.0, \ldots, 0.9\}$ of the noisyMultiMNIST dataset (see Figure 5.13). The DRPM-MTL method achieves the same or better accuracy on both tasks for most noise levels (upper part of Figure 5.13). It is interesting to see that DRPM-MTL tries to overcome the increasing difficulty of the task on the right-hand side by assigning more dimensions to it (lower part of Figure 5.13, noise ratio α 0.6-0.8). For the maximum noise ratio $\alpha = 0.9$, DRPM-MTL can no longer estimate an RPM that reaches state-of-the-art performance.

5.3.3 *Weakly-Supervised Learning*

Data modalities that are not collected as *i.i.d.* samples, such as consecutive frames in a video, provide a weak-supervision signal for generative models and representation learning [Sut+23a] (see also Chapter 1 and section 2.3) Working with coupled datasets can provide both advantages and additional challenges compared to the *i.i.d.* setting (Chapters 3 and 4). Here, on top of learning meaningful representations of the data samples, we are also interested in discovering the relation between the coupled samples. In such a setting, our DRPM enables us to infer the number of shared and independent generative factors and assign latent factors to be shared or independent. Shared and independent factors relate to the underlying group structure as explained in Chapter 1 and section 2.3.2. Next, we introduce DRPM-VAE for learning from weakly-supervised datasets.

5.3.3.1 *Method*

For DRPM-VAE, we model the distribution of shared and independent latent factors as RPM using the proposed DRPM $p(Y; \boldsymbol{\omega}, \boldsymbol{s})$ (see Proposition 5.2.1). We add a posterior approximation of the form $q(Y; \boldsymbol{\omega}(\boldsymbol{X}), \boldsymbol{s}(\boldsymbol{X}))$ where the notation $\boldsymbol{\omega}(\boldsymbol{X})$ and $\boldsymbol{s}(\boldsymbol{X})$ implies that the distribution parameters are inferred from data $\boldsymbol{X}$, and additionally a prior distribution of the form $p(Y; \boldsymbol{\omega}_p, \boldsymbol{s}_p)$.

Without loss of generality, we assume a weakly-supervised dataset $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$ of two views. We assume the following generative model for DRPM-VAE

$$p(\boldsymbol{X}) = \int_{\boldsymbol{z}} p(\boldsymbol{X}, \boldsymbol{z}) d\boldsymbol{z} = \int_{\boldsymbol{z}} p(\boldsymbol{X} \mid \boldsymbol{z}) p(\boldsymbol{z}) d\boldsymbol{z} \tag{5.84}$$

where $\boldsymbol{z} = \{\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2\}$. The two frames share an unknown number $n_{\bar{s}}$ of generative latent factors $\boldsymbol{z}_{\bar{s}}$, and an unknown number, n_1 and n_2, of independent factors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. Given that we use the same encoding function for both views (see Figure 5.15), we have $n_1 = n_2$. The RPM samples n_k and $\boldsymbol{z}_k$ using Y. Hence, the generative model extends to

$$
\begin{aligned}
p(\boldsymbol{X}) &= \int_{\boldsymbol{z}} p(\boldsymbol{X} \mid z) \sum_{Y} p(\boldsymbol{z} \mid Y) p(Y) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}} p(\boldsymbol{x}_1, \boldsymbol{x}_2 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2) \sum_{Y} p(\boldsymbol{z} \mid Y) p(Y) d\boldsymbol{z} \\
&= \int_{\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2} p(\boldsymbol{x}_1 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1) p(\boldsymbol{x}_2 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_2) \\
&\quad \sum_{Y} p(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2 \mid Y) p(Y) d\boldsymbol{z}_{\bar{s}} d\boldsymbol{z}_1 d\boldsymbol{z}_2
\end{aligned}
\tag{5.85}
$$

Figure 5.14 shows the generative and inference model assumptions. Following Lemma 5.2.3 and Equation (5.85), we are able to optimize DRPM-VAE using the following ELBO $\mathcal{L}(\theta, \phi; \boldsymbol{X})$:

$$
\begin{aligned}
\mathcal{L}(\theta, \phi; \boldsymbol{X}) &= \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log p_\theta(\boldsymbol{X} \mid \boldsymbol{z}, Y) - \log \frac{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})}{p_\theta(\boldsymbol{z}, Y)} \right] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \Big[\log p_\theta(\boldsymbol{x}_1, \boldsymbol{x}_2 \mid \boldsymbol{z}) \\
&\quad - \log \frac{q_\phi(\boldsymbol{z} \mid Y, \boldsymbol{X})}{p_\theta(\boldsymbol{z})} - \log \frac{q_\phi(Y \mid \boldsymbol{X})}{p_\theta(Y)} \Big] \\
&= \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log p_\theta(\boldsymbol{x}_1 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1) \right] + \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log p_\theta(\boldsymbol{x}_2 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_2) \right] \\
&\quad - \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log \frac{q_\phi(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2 \mid Y, \boldsymbol{X})}{p_\theta(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2)} \right] \\
&\quad - \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log \frac{q_\phi(Y \mid \boldsymbol{X})}{p_\theta(Y)} \right] \\
&\geq \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log p_\theta(\boldsymbol{x}_1 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1) \right] + \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log p_\theta(\boldsymbol{x}_2 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_2) \right] \\
&\quad - \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log \frac{q_\phi(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2 \mid Y, \boldsymbol{X})}{p_\theta(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2)} \right] \\
&\quad - \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log \frac{q_\phi(\boldsymbol{n} \mid \boldsymbol{X}; \boldsymbol{\omega}) \cdot |\Pi_Y|}{p_\theta(\boldsymbol{n}; \boldsymbol{\omega}_p) p_\theta(\pi \mid \boldsymbol{X}; \boldsymbol{s}_p)} \right] \\
&\quad - \log \max_{\pi \in \Pi_Y} q_\phi(\pi \mid \boldsymbol{X}; \boldsymbol{s})
\end{aligned}
\tag{5.86}
$$

θ are the amortization parameters for the generative model, and ϕ for the inference model. Please note the similarity to the ELBO used in the clustering experiment in Section 5.3.1, which also shows the versatility of the proposed DRPM formulation. Similar to the ELBO in Section 5.3.1 or Higgins et al. [Hig+16], the terms need some manual adjustment for optimal results (see Appendix A.4.3).

5.3.3.2 *Dataset, Implementation & Training*

In this experiment, we use paired frames $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2]$ from the *mpi3d* toy dataset [Gon+19]. In Chapter 1, Figure 1.2 shows an example pair of images of a robot arm. Every pair of frames shares a subset of its seven generative factors. The model maximizes the ELBO on the marginal log-likelihood of the images through a VAE [KW14]. In this experiment, we use the `disentanglement_lib` from Locatello et al. [Loc+20]. We use the same architectures proposed in the original paper for all comparison methods. Figure 5.15 shows the pipeline shared between all methods.

We compare the proposed DRPM-VAE to three methods, which only differ in how they infer shared and latent dimensions. While the Label-VAE [BTN18; Hos18] implicitly assumes that the number of independent factors is known, the Ada-VAE [Loc+20] relies on a heuristics-based approach to infer shared and independent latent factors. Like in Locatello et al. [Loc+20], we assume a single known independent generative factor for Label-VAE in all experiments[2]. Additionally, we compare DRPM-VAE to HG-VAE [Sut+23a]. HG-VAE only models the number of shared and independent factors using the MVHG (see Section 5.1), but not the assignment step needed for a RPM. HG-VAE relies on a heuristic for the assignment step, which is not part of the ELBO formulation.

The baseline algorithms, Label-VAE [BTN18; Hos18] and Ada-VAE [Loc+20] are already implemented in `disentanglement_lib`[3]. We did not change any hyperparameters or network settings.

The different methods only differ in the `View Aggregation` module (see Figure 5.15), which contains the procedure to select shared and independent latent factors. Given a subset $\mathcal{S}_{\bar{s}}$ of shared latent factors, it follows

$$q_\phi(z_i \mid \boldsymbol{x}_j) = f_{agg}(q_\phi(z_i \mid \boldsymbol{x}_1), q_\phi(z_i \mid \boldsymbol{x}_2)) \qquad \forall\ i \in \mathcal{S}_{\bar{s}} \tag{5.87}$$

$$q_\phi(z_i \mid \boldsymbol{x}_j) = q_\phi(z_i \mid \boldsymbol{x}_j) \qquad \text{else} \tag{5.88}$$

where f_{agg} is the aggregation function and $j \in \{1,2\}$. The aggregation function $f_{agg}(\cdot)$ is the same for all methods. They aggregate shared latent factors using the GroupVAE method proposed in Hosoya [Hos18], aggregating shared latent factors using an arithmetic mean. We stick to GroupVAE

[2] Please note the similarity of Label-VAE to the multimodal VAE in Chapter 4 using shared and modality-specific latent subspaces. Both methods share the same assumptions regarding the underlying group structure.

[3] For additional details on the implementation of these methods we refer to the original paper from Locatello et al. [Loc+20].

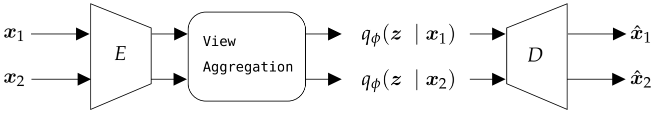


Figure 5.15: The basic architecture for all methods used for the weakly-supervised experiments. They only differ in determining the number of shared $n_{\bar{s}}$ and independent n_1 and n_2 generative factors.

because of its simplicity. Any other aggregation method (see Section 3.2) would be a feasible option too.

Figure 5.15 shows the building blocks used. With a single encoder, all methods independently encode both images to some latent representation, jointly used to infer shared latent dimensions. A single decoder, which represents $p_\theta(\boldsymbol{x}_j \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_j)$, sequentially reconstructs the two views from an aggregated latent vector consisting of a combination of shared and independent factors.

All models are trained on five random seeds, and the reported results are averaged over the five seeds. We report mean performance with standard deviations. We train all models for $300'000$ steps using the Adam optimizer [KB14] with an initial learning rate of 10^{-5} and a batch size of 64. The temperature annealing of the Gumbel-Softmax modules in DRPM follows the same schedule as in our clustering experiments (see Section 5.3.1.2). Also, it anneals the temperature from $\tau_{init} = 1.0$ to $\tau_{final} = 0.5$ over the $300'000$ training steps. Appendix A.4.3 provides more details on the precise loss function and its hyperparameters.

5.3.3.3 *Experiments & Results*

EVALUATION To evaluate the methods, we compare their performance on two different tasks, which challenge the methods regarding their estimation of the relationship between images. Because we have access to the data-generating process, we can control the number of shared $n_{\bar{s}}$ and independent n_i factors. We compare the methods on four different weakly-supervised datasets with $n_{\bar{s}} \in \{0, 1, 3, 5\}$. On purpose, we also evaluate the edge case of $n_{\bar{s}} = 0$, which is equal to the two views not sharing any generative factors. We assess the methods according to their ability to estimate the number of shared generative factors (Figure 5.16) and how well they encode the latent representations into shared and independent factors (Table 5.3). We measure the mean squared error $\text{MSE}(n_{\bar{s}}, \hat{n}_{\bar{s}})$ between the actual number of shared latent factors $n_{\bar{s}}$ and the estimated number $\hat{n}_{\bar{s}}$ (Figure 5.16) and

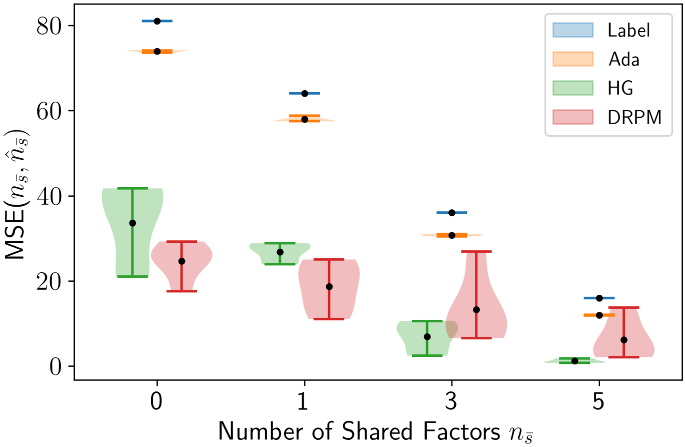


FIGURE 5.16: The mean squared error between the estimated number of shared factors $\hat{n}_{\bar{s}}$ and the true number of shared factors $n_{\bar{s}}$ across five seeds for the Label-VAE, Ada-VAE, HG-VAE, and DRPM-VAE.

the classification accuracy of predicting the generative factors on the shared and independent subsets of the learned representations (Table 5.3).

For the downstream task, we randomly sample 10′000 samples from the training set and 5′000 samples from the test set. We extract both views' predicted shared and independent parts for each sample. Then, for every generative factor of the dataset, three individual classifiers are trained on the respective latent representations of the 10′000 training samples. Afterward, every classifier evaluates its predictive performance on the latent representations of the 5′000 test samples. To arrive at the final scores, we extract the prediction of the shared factors on the shared representation and compute the balanced accuracy. Similarly, we calculate the balanced accuracy of the independent factors on the respective independent representation classifiers and average their balanced accuracy. Because the number of classes differs between generative factors, we report the adjusted balanced accuracy[4].

For all shared generative factors, we average the accuracies of the individual classifiers into a single average balanced accuracy. We do the same for the independent factors. This allows us to report the amount of shared

[4] We use the `scikit-learn` [Ped+11] implementation. For details, see `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html`.

Model		Label	Ada	HG	DRPM
$n_s = 0$	I	0.14 ± 0.01	0.12 ± 0.01	0.18 ± 0.01	$\mathbf{0.26} \pm 0.02$
$n_s = 1$	S	0.19 ± 0.03	0.19 ± 0.01	0.22 ± 0.05	$\mathbf{0.39} \pm 0.07$
	I	0.16 ± 0.01	0.15 ± 0.01	0.19 ± 0.01	$\mathbf{0.20} \pm 0.01$
$n_s = 3$	S	0.10 ± 0.00	0.10 ± 0.03	0.08 ± 0.02	$\mathbf{0.15} \pm 0.01$
	I	0.23 ± 0.01	0.22 ± 0.02	0.28 ± 0.01	$\mathbf{0.29} \pm 0.02$
$n_s = 5$	S	0.34 ± 0.00	0.33 ± 0.03	0.28 ± 0.01	$\mathbf{0.42} \pm 0.03$
	I	0.00 ± 0.00	0.00 ± 0.00	0.01 ± 0.00	0.01 ± 0.00

Table 5.3: We evaluate the learned latent representations of the four methods (Label-VAE, Ada-VAE, HG-VAE, DRPM-VAE) in the weakly-supervised experiment with respect to the shared (S) and independent (I) generative factors. We do this by fitting linear classifiers on the shared and independent dimensions of the representation, predicting the respective generative factors. We report the results in adjusted balanced accuracy [Sut+23a] across five random seeds.

and independent information in the learned latent representation and the respective subspaces.

Similar to the latent representation in Sections 3.3.3.1 and 4.2.3.1, we train linear classifiers to evaluate the latent representation. Specifically, we use logistic regression classifiers [Cox58] from scikit-learn [Ped+11]. To train the model, we increased the *max_iter* parameter so that all models converged and left everything else on default settings.

RESULTS In Figure 5.16, we see that DRPM-VAE can accurately estimate the true number of shared generative factors. It matches the performance of HG-VAE and outperforms the other two baselines, which consistently overestimate the true number of shared factors. In Table 5.3, we see a considerable performance improvement for DRPM-VAE compared to previous work when assessing the learned latent representations. We assume this to be due to its ability to estimate the subset sizes of latent and shared factors like HG-VAE and to learn to assign latent dimensions to corresponding shared or independent representations. Thus, DRPM-VAE can dynamically learn more meaningful representations of shared and independent subspaces for all dataset versions.

DRPM-VAE provides empirical evidence of how RPMs can help with specific weakly supervised learning tasks, in which we are interested in maximizing the data likelihood while also learning representations that capture the relation between coupled data samples. Additionally, we can explicitly model the data-generating process in a theoretically grounded fashion instead of relying on heuristics.

5.4 DISCUSSION

This chapter presents the first steps toward learning the relationship between group members in a weakly-supervised setting. Similar to the multimodal setting, we assume an unknown set of shared generative factors. Using the hypergeometric distribution, we present a novel two-step RPM. Given its differentiable and reparameterizable formulation, we can integrate the proposed DRPM into any modern ML framework and learn the parameters of its distribution using gradient-based optimization.

We proposed two novel continuous relaxations for discrete distributions. Our new formulations enable us to learn the distribution parameters using gradient-based optimization.

In a weakly supervised multiview setting, we show that DRPM-VAE, a VAE integrating DRPM into its variational distribution, learns the number of shared factors and encodes shared and independent information accordingly into the shared and independent latent subspaces. Compared to the baseline methods, which must rely on restrictive assumptions and heuristics, DRPM-VAE achieves better performance. Like the methods presented in Chapters 3 and 4, over-restrictive assumptions and heuristics hurt the learning process of the baseline methods.

In addition, we show that DRPM is a versatile building block and use it in a clustering application (Section 5.3.1), where we partly overcome the *i.i.d.* assumption and an MTL setting (Section 5.3.2), where we partition the output layer of a neural network according to different tasks.

On the other hand, learning the needed scores $\boldsymbol{s}$ and group importance $\boldsymbol{\omega}$ is a complex optimization problem. Although we reduce the number of hyperparameters and restrictive assumptions compared to the methods using shared and modality-specific latent subspaces in Chapter 4, we must choose the remaining hyperparameters carefully.

In future steps, we want to apply DRPM to other weakly-supervised settings. We want to use it in a multiview setting with more than two views, where more complicated relationships between images appear, and multimodal data, where not the same encoder generates the latent representations, and, as such, presents a more complex problem.

6 SUMMARY AND CONCLUSION

In this chapter, we summarize the results and discuss the limitations of this thesis, which is based on findings and results of the research publications during my PhD [Dau+22; Dau+20; SDV20; SDV21; Sut+23a; Sut+23b; SV21]. We describe our contributions in three chapters, where every chapter discusses and tackles a different but related aspect of multimodal and weakly supervised learning.

First (Chapter 3), we analyzed the implications of different probabilistic aggregation methods for multimodal scalable VAEs in combination with the assumption of a single joint latent space. Hence, we infer a single joint latent variable, which we then use to condition the generative distributions of all modalities. Given potentially missing data types, we require multimodal VAEs to have a joint posterior approximation, which is robust to missing modalities. Hence, early-fusion multimodal models, where the aggregation of different modalities is part of the network architecture, are not a valid option because they always require the same number of input modalities and cannot handle missing modalities.

Unlike early fusion, late fusion aggregation functions efficiently handle any number of inputs. Therefore, we introduced a new probabilistic aggregation function, a mixture of products of experts distribution (MoPoE). The MoPoE joint posterior approximation combines all multimodal subsets using their mixture distribution and aggregates the modalities in every subset using their product distribution. We showed that the proposed MoPoE distribution improves performance and generalizes previous works [Shi+19; WG18]. Previous works on scalable multimodal VAEs only considered specific subsets in their probabilistic aggregation functions and are only special cases of the MoPoE formulation. In our experiments, we showed that the performance of every method is directly related to the choice of subsets considered in the aggregation function. While there is a trade-off between generative quality and coherence, we showed that the choice of aggregation function strongly impacts the performance of scalable multimodal VAEs. Hence, only our MoPoE-based formulation fulfills the requirements and desiderata of multimodal learning across all subsets and is robust to missing input modalities.

However, we uncovered that no method under our generalized formulation generates samples of the same quality as if a comparable unimodal VAE generated them. We attributed this limitation to the over-restrictive assumption of having a single joint latent space and the drawbacks of mixture distributions as probabilistic aggregation functions in the multimodal setting.

Second (Chapter 4), we showed how a more flexible assumption of the underlying multimodal group structure improves the generative performance of multimodal VAEs. We introduced additional modality-specific latent subspaces to overcome the over-restrictive assumption of a single joint latent space. Hence, we model every data type as having shared and modality-specific generative factors. In the experiments, the more flexible assumptions led to the better generative quality of samples than more restrictive assumptions. More surprisingly, the learned latent representations and the coherence of generated samples improved, too.

Our experiments showed that all multimodal VAEs benefit from additional modality-specific latent subspaces — independent of the probabilistic aggregation function used for the joint variational posterior. Mixture-based multimodal VAEs such as MoPoE – in addition to their high coherence in generated samples when using a single joint latent space – show the high quality of generated samples when using modality-specific latent subspaces. The downside of additional modality-specific subspaces is the increased number of hyperparameters. During the experiments, we saw a high sensitivity of the model performance to the choice of hyperparameters, which is not surprising for a VAE-based approach [Hig+16; Loc+19].

However, additional latent subspaces requiring additional hyperparameters intensify an already existing issue. Therefore, we pay for the increased quality with a more intense hyperparameter selection procedure, arguably contradictory to our original goal of designing scalable multimodal VAEs.

In the last chapter (Chapter 5), we worked toward integrating the modeling of group structures into the learning process. The step from restrictive (Chapter 3) to more flexible modeling assumptions (Chapter 4) led to performance improvements regarding the coherence and quality of samples. However, the assumptions still reflect a simplistic group structure of a weakly-supervised dataset. In the chapters on multimodal learning, a particular case of the general weakly-supervised setting, we showed the importance of having assumptions that match the relationship between group members. Following the difficulty of getting precise knowledge about the underlying group structure, there was a need to incorporate the inference of this relationship into the learning process. We needed probabilistic formulations that do not rely on *a priori* knowledge about the underlying group structure of weakly supervised data.

Chapter 5 showed how partition models enable learning the group structure instead of relying on assumptions or pre-defined heuristics. We introduced a differentiable and reparameterizable random partition model, which learns to select shared and independent latent factors while maximizing the ELBO in a weakly supervised setting. The proposed differentiable RPM is based on a two-stage approach. First, we sample the number of elements per subset using the multivariate noncentral hypergeometric distribution. Second, we assign the corresponding number of elements to the respective subsets based on a Plackett-Luce model. For the first step, we introduced a new differentiable and reparameterizable formulation of the hypergeometric distribution. Next to the weakly supervised experiment, we show the versatility of the proposed RPM formulation in two additional ML experiments, clustering and multitask learning. The selection of experiments in the last chapter highlights the importance of differentiable discrete distributions for ML.

Overall, we additionally motivate comparing methods with respect to multiple performance metrics – and not only a single one. We assess the performance of multimodal and weakly-supervised VAEs using several performance measures. In the multimodal setting, which can be seen as a challenging weakly-supervised setting, we see that different methods perform well on single tasks such as latent representations or generative quality. Hence, it is essential to not only compare different tasks but also different inputs. We can only see the limitations and strengths of different methods and assumptions by evaluating all metrics with respect to different multimodal subsets.

We only looked at amortized inference schemes in our experiments. Cremer, Li, and Duvenaud [CLD18] show that the inference of VAEs is suboptimal compared to methods that infer the latent representation using gradient-based optimization. This "amortization gap" is likely caused by the amortization of posterior approximations using inference networks. Combining gradient-based optimization of inference and multimodal VAEs could help improve the performance of multimodal models - independent of their aggregation function - and presents an interesting line of research for future research. However, the results of the experiments of this thesis showed that learning a model that fulfills all desiderata of multimodality goes beyond approximating the training distribution (Sections 3.3 and 4.2). For e. g., the Poe-VAE achieves the best log-likelihood numbers if all modalities are given. However, it shows reduced performance if only a subset of modalities is given as input, making it a sub-optimal multimodal model.

This thesis aims to uncover and overcome limiting assumptions in learning from data under weak supervision. During this process, we evaluate our hypotheses mainly on synthetic datasets. The more realistic an application gets, the more complicated the involved networks and architectures of the models are. Hence, the disentanglement of contributions of the individual building blocks and methods is difficult. On the other hand, synthetic data allows the controllable generation of datasets and their evaluation with respect to generative factors.

Nevertheless, the missing experiments on real-world data are a limitation of this thesis. Foreseeing all the difficulties and caveats of real-world data is unlikely. Hence, the assumptions and relationships built into a synthetic dataset will differ from the complete set of problems surfacing in a real-world experiment. Therefore, future work must include an evaluation of the proposed methods and formulations on real-world data and applications. Transferring the gained knowledge and improved methods to more realistic settings would give us more insights and test the ability of the proposed methods to learn in real-world scenarios.

This thesis's proposed methods and formulations rely on the weak supervision of the collected data. However, datasets of different data types describing a specific phenomenon are often unaligned, which makes it difficult to leverage this rich source of information. For example, the information from x-ray images and computer tomography scans of different patient populations but the same part of the human body cannot be integrated by methods described in this work. However, leveraging unaligned datasets would improve multimodal methods. Using this source information is an interesting next step for multimodal learning requiring different approaches [e. g., GNX22; Mos+22].

Current state-of-the-art generative models train conditional text-to-image models [Ram+22; Ram+21]. Hence, they can only generate images based on text, but not text based on images, raising the question of how much *multimodality* in conditional methods is. While for a bimodal dataset of images and text, it is feasible to train models for both conditional distributions, i. e., image to text and text to image, the number of conditional distributions increases exponentially with the number of modalities. Given the computational cost of training a single state-of-the-art text-to-image generative model [Bro+20, section 6.3 and appendix B], it is questionable whether training an exponential number of models is a good development strategy. This thesis focuses on learning the joint distribution of weakly-supervised datasets, where multimodal data is a particular case. Given the conceptual problems with conditional models, we must tackle the more difficult problem of learning the joint data distribution. The proposed approaches to learning from weakly supervised data outlined in this thesis enable models that scale linearly in the number of group members and naturally provide many-to-many mappings between group members or modalities. However, learning the joint distribution of weakly supervised data poses additional challenges. Coherent and high-quality generation of multimodal samples, in combination with scalability, is a non-trivial challenge in the weakly-supervised setting.

Currently, contrastive methods [Che+20] are the state-of-the-art self-supervised representation learning approach, diffusion models [Rom+22] for image generation and transformer-based architectures for natural language processing and generation [Bro+20; Vas+17]. While extensions to the vanilla VAE, such as hierarchical VAEs[Chi20; VK20], show impressive generative quality, the prime spot in conditional generation belongs to the more specialized models or combinations thereof. E. g. conditional text-to-image models such as Dall-E [Ram+22; Ram+21] rely on a combination of transformers, contrastive methods, and diffusion models. The possibility of learning meaningful representations and generating samples while optimizing only a single objective is both a strength and weakness for VAEs [see also Mil21]. Nevertheless, the strength of multimodal and weakly-supervised VAEs presented in this work is their formulation based on the principles of variational inference. This principled approach allows us to simultaneously learn meaningful representations and high-quality generations for any data type and propose new probabilistic formulations.

In multimodal learning, we want to learn from any data type, not just the ubiquitous text and image modality. Therefore, providing methods to learn from and generate any data type is essential. Hence, the ELBO optimization makes multimodal VAEs a suitable method for any data type we can define and evaluate the log-likelihood. However, multimodal VAEs adopt not only the strengths but also the weaknesses of a VAE-based approach. The log-likelihood definitions to evaluate a generated sample of a data type with unknown data distribution, e. g., images, rely on heuristics. Likelihood-free methods, e. g., GANs [Goo+14], provide a better objective leading to a better quality of generated samples. Bridging the gap between the likelihood-based objective applicable to any data type and state-of-the-art specialized models by reducing over-restrictive assumptions of the defined data log-likelihoods is an essential step for unimodal and multimodal VAEs.

Throughout this thesis, we are concerned with learning from weakly-supervised data and how different assumptions on the group structure and its generative factors affect the learning process. Using the principles of variational approximation, we can uncover the hidden relation between samples in the form of shared and independent latent factors. In summary, this thesis shows the importance of integrating the learning of group structure into the optimization process and the value of overcoming simplistic assumptions.

BIBLIOGRAPHY

[Aba+16] Martin Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Gregory S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian J Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Józefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Gordon Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul A Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda B Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". In: *CoRR* abs/1603.0 (2016).

[Aco+22] Julián N Acosta, Guido J Falcone, Pranav Rajpurkar, and Eric J Topol. "Multimodal biomedical AI". In: *Nature Medicine* 28.9 (2022), 1773.

[AZ11] Ryan P Adams and Richard S Zemel. "Ranking via sinkhorn propagation". In: *arXiv preprint arXiv:1106.1925* (2011).

[Ant+15] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. "Vqa: Visual question answering". In: *Proceedings of the IEEE International Conference on Computer vision*. 2015, 2425.

[AF17] Jeffrey K Aronson and Robin E Ferner. "Biomarkers—a general review". In: *Current protocols in pharmacology* 76.1 (2017), 9.

[ABT92] Richard Arratia, Andrew D Barbour, and Simon Tavaré. "Poisson process approximations for the Ewens sampling formula". In: *The Annals of Applied Probability* (1992), 519.

[Atr+10] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. "Multimodal fusion for multimedia analysis: a survey". In: *Multimedia systems* 16 (2010), 345.

[Azi+21] Shekoofeh Azizi, Basil Mustafa, Fiona Ryan, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, and others. "Big self-supervised models advance medical image classification". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, 3478.

[Bae+20] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. "wav2vec 2.0: A framework for self-supervised learning of speech representations". In: *Advances in Neural Information Processing Systems* (2020), 12449.

[Bal+17] Matej Balog, Nilesh Tripuraneni, Zoubin Ghahramani, and Adrian Weller. "Lost relatives of the Gumbel trick". In: *Proceedings of the International Conference on Machine Learning*. 2017, 371.

[BAM18] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. "Multimodal machine learning: A survey and taxonomy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41.2 (2018), 423.

[BR93] Jeffrey D Banfield and Adrian E Raftery. "Model-based Gaussian and non-Gaussian clustering". In: *Biometrics* (1993), 803.

[Bar17] Bruce Barrett. "A note on exact calculation of the non central hypergeometric distribution". In: *Communications in Statistics - Theory and Methods* 46.13 (2017), 6737.

[BH92] Daniel Barry and J A Hartigan. "Product Partition Models for Change Point Problems". In: *The Annals of Statistics* 20.1 (1992), 260.

[BH93] Daniel Barry and John A Hartigan. "A Bayesian analysis for change point problems". In: *Journal of the American Statistical Association* 88.421 (1993), 309.

[Bec+11] Luca Becchetti, Ugo M Colesanti, Alberto Marchetti-Spaccamela, and Andrea Vitaletti. "Recommending items in pervasive scenarios: models and experimental analysis". In: *Knowledge and information systems* 28.3 (2011), 555.

[BCV13] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35.8 (2013).

[BLC13] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. “Estimating or propagating gradients through stochastic neurons for conditional computation”. In: *arXiv preprint arXiv:1308.3432* (2013).

[BH95] Yoav Benjamini and Yosef Hochberg. “Controlling the false discovery rate: a practical and powerful approach to multiple testing”. In: *Journal of the royal statistical society series b-methodological* 57 (1995), 289.

[BM17] Bhaswar B Bhattacharya and Sumit Mukherjee. “Degree sequence of random permutation graphs”. In: *The Annals of Applied Probability* 27.1 (2017), 439.

[Bio+01] Biomarkers Definitions Working Group, Arthur J Atkinson Jr, Wayne A Colburn, Victor G DeGruttola, David L DeMets, Gregory J Downing, Daniel F Hoth, John A Oates, Carl C Peck, Robert T Schooley, and others. “Biomarkers and surrogate endpoints: preferred definitions and conceptual framework”. In: *Clinical pharmacology & therapeutics* 69.3 (2001), 89.

[BN06] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Vol. 4. 4. Springer, 2006.

[BS04] Christopher M Bishop and Markus Svensén. “Robust Bayesian Mixture Modelling.” In: *ESANN*. 2004, 69.

[Bla47] David Blackwell. “Conditional expectation and unbiased sequential estimation”. In: *The Annals of Mathematical Statistics* (1947), 105.

[BM73] David Blackwell and James B MacQueen. “Ferguson distributions via Pólya urn schemes”. In: *The annals of statistics* 1.2 (1973), 353.

[BKM17] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. “Variational Inference: A Review for Statisticians”. In: *Journal of the American Statistical Association* 112.518 (2017), 859.

[Bol68] Ludwig Boltzmann. “Studien über das Gleichgewicht der lebenden Kraft”. In: *Wissenschafiliche Abhandlungen* 1 (1868), 49.

[BMVH03] Denny Borsboom, Gideon J Mellenbergh, and Jaap Van Heerden. “The theoretical status of latent variables.” In: *Psychological review* 110.2 (2003), 203.

[Bot10] Léon Bottou. "Large-Scale Machine Learning with Stochastic Gradient Descent". In: *COMPSTAT*. 2010.

[BTN18] Diane Bouchacourt, Ryota Tomioka, and Sebastian Nowozin. "Multi-level variational autoencoder: Learning disentangled representations from grouped observations". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[Bre37] Car A Bretschneider. "Theoriae logarithmi integralis lineamenta nova." In: (1837).

[Bro+20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and others. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems* 33 (2020), 1877.

[Bub+23] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, and others. "Sparks of artificial general intelligence: Early experiments with gpt-4". In: *arXiv preprint arXiv:2303.12712* (2023).

[Car93] Rich Caruana. "Multitask learning: A knowledge-based source of inductive bias". In: *Proceedings of the International Conference on Machine Learning*. 1993, 41.

[CS96] Rich Caruana and Virginia R de Sa. "Promoting poor features to supervisors: Some inputs work better as outputs". In: *Advances in Neural Information Processing Systems* 9 (1996).

[Cas+16a] Joan A Casey, Brian S Schwartz, Walter F Stewart, and Nancy E Adler. "Using electronic health records for population health research: a review of methods and applications". In: *Annual review of public health* 37 (2016), 61.

[Cas+16b] Giona Casiraghi, Vahan Nanumyan, Ingo Scholtes, and Frank Schweitzer. "Generalized Hypergeometric Ensembles: Statistical Hypothesis Testing in Complex Networks". In: (2016).

[Cha+20] Krishna Chaitanya, Ertunc Erdil, Neerav Karani, and Ender Konukoglu. "Contrastive learning of global and local features for medical image segmentation with limited annotations". In: *Advances in Neural Information Processing Systems* 33 (2020), 12546.

[CP11] Vincent Chan and Anahi Perlas. “Basics of Ultrasound Imaging”. In: *Atlas of Ultrasound-Guided Procedures in Interventional Pain Management*. New York, NY: Springer New York, 2011, 13.

[Che+18] Ricky T Q Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. “Isolating sources of disentanglement in variational autoencoders”. In: *Advances in Neural Information Processing Systems* 31 (2018).

[Che+20] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. “A simple framework for contrastive learning of visual representations”. In: *Proceedings of the International Conference on Machine Learning*. 2020, 1597.

[Che76] Jean Chesson. “A non-central multivariate hypergeometric distribution arising from biased sampling with application to selective predation”. In: *Journal of Applied Probability* 13.4 (1976), 795–797.

[CPM18] I Chien, Chao Pan, and Olgica Milenkovic. “Query k-means clustering and the double dixie cup problem”. In: *arXiv preprint arXiv:1806.05938* (2018).

[Chi20] Rewon Child. “Very deep vaes generalize autoregressive models and can outperform them on images”. In: *arXiv preprint arXiv:2011.10650* (2020).

[CJ18] Hyunsun Choi and Eric Jang. “Generative ensembles for robust anomaly detection”. In: (2018).

[CT06] Thomas M Cover and Joy A Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. USA: Wiley-Interscience, 2006.

[Cox58] David R Cox. “The regression analysis of binary sequences”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.2 (1958), 215.

[CHCV19] Alexander Craik, Yongtian He, and Jose L Contreras-Vidal. “Deep learning for electroencephalogram (EEG) classification tasks: a review”. In: *Journal of neural engineering* 16.3 (2019), 31001.

[CLD18] Chris Cremer, Xuechen Li, and David Duvenaud. “Inference Suboptimality in Variational Autoencoders”. In: *Proceedings of the International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. 2018, 1086.

[Dan+05] Leon Danon, Albert Diaz-Guilera, Jordi Duch, and Alex Arenas. “Comparing community structure identification”. In: *Journal of statistical mechanics: Theory and experiment* 2005.09 (2005), P09008.

[DR98] Abhijit Dasgupta and Adrian E Raftery. “Detecting features in spatial point processes with clutter via model-based clustering”. In: *Journal of the American statistical Association* 93.441 (1998), 294.

[Dau+23] Imant Daunhawer, Alice Bizeul, Emanuele Palumbo, Alexander Marx, and Julia E Vogt. “Identifiability results for multimodal contrastive learning”. In: *arXiv preprint arXiv:2303.09166* (2023).

[Dau+22] Imant Daunhawer, Thomas M. Sutter, Kieran Chin-Cheong, Emanuele Palumbo, and Julia E Vogt. “On the Limitations of Multimodal VAEs”. In: *International Conference on Learning Representations* (2022).

[Dau+20] Imant Daunhawer, Thomas Marco Sutter, Ricards Marcinkevics, and Julia E Vogt. “Self-supervised Disentanglement of Modality-specific and Shared Factors Improves Multimodal Generative Models”. In: *German Conference on Pattern Recognition* (2020).

[Day+95] Peter Dayan, Geoffrey E Hinton, Radford M Neal, and Richard S Zemel. “The Helmholtz Machine”. In: *Neural Computation* 7.5 (1995), 889.

[Hin] “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups”. In: *IEEE Signal Process. Mag.* 29.6 (2012), 82.

[Den+09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. “Imagenet: A large-scale hierarchical image database”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2009, 248.

[DL14] Michael Denkowski and Alon Lavie. “Meteor universal: Language specific translation evaluation for any target language”. In: *Proceedings of the ninth workshop on statistical machine translation*. 2014, 376.

[DZR12] James J DiCarlo, Davide Zoccolan, and Nicole C Rust. “How does the brain solve visual object recognition?” In: *Neuron* 73.3 (2012), 415.

[Dil+16] Nat Dilokthanakul, Pedro A M Mediano, Marta Garnelo, Matthew C H Lee, Hugh Salimbeni, Kai Arulkumaran, and Murray Shanahan. “Deep unsupervised clustering with gaussian mixture variational autoencoders”. In: *arXiv preprint arXiv:1611.02648* (2016).

[DCS18] Jiarui Ding, Anne Condon, and Sohrab P Shah. “Interpretable dimensionality reduction of single cell transcriptome data with deep generative models”. In: *Nature communications* 9.1 (2018), 2002.

[Dwi+19] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. “Temporal cycle-consistency learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, 1801.

[Eul40] Leonhard Euler. “De progressionibus harmonicis observationes”. In: *Commentarii academiae scientiarum Petropolitanae* (1740), 150.

[Eve84] Brian S Everitt. “An Introduction to Latent Variable Models”. In: 1984.

[Fan+15] Hao Fang, Saurabh Gupta, Forrest Iandola, Rupesh K Srivastava, Li Deng, Piotr Dollár, Jianfeng Gao, Xiaodong He, Margaret Mitchell, John C Platt, and others. “From captions to visual concepts and back”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, 1473.

[FHK17] Moran Feldman, Christopher Harshaw, and Amin Karbasi. “Greed is good: Near-optimal submodular maximization via greedy optimization”. In: *Conference on Learning Theory*. PMLR. 2017, 758.

[Fer73] Thomas S Ferguson. “A Bayesian analysis of some nonparametric problems”. In: *The annals of statistics* (1973), 209.

[Fis35] Ronald A Fisher. “The logic of inductive inference”. In: *Journal of the royal statistical society* 98.1 (1935), 39.

[Fog08a] Agner Fog. “Calculation methods for Wallenius’ noncentral hypergeometric distribution”. In: *Communications in Statistics—Simulation and Computation* 37.2 (2008), 258.

[Fog08b] Agner Fog. “Sampling methods for Wallenius’ and Fisher’s noncentral hypergeometric distributions”. In: *Communications in Statistics—Simulation and Computation* 37.2 (2008), 241.

[For+20] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. “GP-VAE: Deep probabilistic time series imputation”. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 2020, 1651.

[Fré57] Maurice Fréchet. “Sur la distance de deux lois de probabilité”. In: *Comptes Rendus Hebdomadaires des Seances de L Academie des Sciences* 244.6 (1957), 689.

[Gal16] Yarin Gal. “Uncertainty in Deep Learning”. PhD thesis. University of Cambridge, 2016.

[GS90] Alan E Gelfand and Adrian F M Smith. “Sampling-Based Approaches to Calculating Marginal Densities”. In: *Journal of the American Statistical Association* 85.410 (1990), 398.

[GG84] Stuart Geman and Donal Geman. “Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-6.6 (1984), 721.

[GG14] Samuel J Gershman and Noah D Goodman. “Amortized Inference in Probabilistic Reasoning”. In: *Cognitive Science* 36 (2014).

[Gib02] Josiah W Gibbs. *Elementary principles in statistical mechanics: developed with especial reference to the rational foundations of thermodynamics*. C. Scribner’s sons, 1902.

[Gon+19] Muhammad W Gondal, Manuel Wuthrich, Djordje Miladinovic, Francesco Locatello, Martin Breidt, Valentin Volchkov, Joel Akpo, Olivier Bachem, Bernhard Schölkopf, and Stefan Bauer. “On the Transfer of Inductive Bias from Simulation to the Real World: a New Disentanglement Dataset”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019.

[GNX22] Fengjiao Gong, Yuzhou Nie, and Hongteng Xu. “Gromov-Wasserstein multi-modal alignment and clustering”. In: *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2022, 603.

[Gon36] H T Gonin. “The use of factorial moments in the treatment of the hypergeometric distribution and in tests for regression”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 21.139 (1936), 215.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[Goo+14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. “Generative Adversarial Nets”. In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014.

[GHL17] Jonathan Gordon and José M Hernández-Lobato. “Bayesian semisupervised learning with deep generative models”. In: *arXiv preprint arXiv:1706.09751* (2017).

[GKP89] Ronald L Graham, Donald E Knuth, and Oren Patashnik. *Concrete mathematics: a foundation for computer science*. 5. Addison-Wesley, 1989.

[Gre+20] Luigi Gresele, Paul K Rubenstein, Arash Mehrjou, Francesco Locatello, and Bernhard Schölkopf. “The Incomplete Rosetta Stone problem: Identifiability results for Multi-view Nonlinear ICA”. In: *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*. Vol. 115. Proceedings of Machine Learning Research. 2020, 217.

[Gro+19] Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. “Stochastic Optimization of Sorting Networks via Continuous Relaxations”. In: *International Conference on Learning Representations*. 2019.

[GS09] John Guiver and Edward L Snelson. “Bayesian inference for Plackett-Luce ranking models”. In: *Proceedings of the 26th International Conference on Machine Learning*. Vol. 382. ACM International Conference Proceeding Series. 2009, 377.

[Gum35] Emil J Gumbel. “Les valeurs extrêmes des distributions statistiques”. In: *Annales de l’institut Henri Poincaré*. Vol. 5. 2. 1935, 115.

[Gum54] Emil J Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*. Vol. 33. US Government Printing Office, 1954.

[GH10] Michael Gutmann and Aapo Hyvärinen. "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models". In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 2010, 297.

[Har+19] Chris Harshaw, Moran Feldman, Justin Ward, and Amin Karbasi. "Submodular maximization beyond non-negativity: Guarantees, fast algorithms, and applications". In: *Proceedings of the International Conference on Machine Learning*. PMLR, 2019, 2634.

[Har90] John A Hartigan. "Partition models". In: *Communications in Statistics - Theory and Methods* 19.8 (1990), 2745.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[Has70] W Keith Hastings. "Monte Carlo Sampling Methods Using Markov Chains and Their Applications". In: *Biometrika* 57.1 (1970), 97.

[He+16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, 770.

[Heu+17] 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". In: *Advances in Neural Information Processing Systems*. 2017, 6626.

[Hig+16] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. "beta-vae: Learning basic visual concepts with a constrained variational framework". In: *International Conference on Learning Representations*. 2016.

[Hin99] Geoffrey E Hinton. "Products of experts". In: (1999).

[Hin02] Geoffrey E Hinton. "Training products of experts by minimizing contrastive divergence". In: *Neural computation* 14.8 (2002), 1771.

[HW20] Michelle S Hirsch and Jaclyn Watkins. “A comprehensive review of biomarker use in the gynecologic tract including differential diagnoses and diagnostic pitfalls”. In: *Advances in anatomic pathology* 27.3 (2020), 164.

[Hof+13] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. “Stochastic Variational Inference”. In: *Journal of Machine Learning Research* 14.40 (2013), 1303.

[HV03] Antti Honkela and Harri Valpola. “On-line variational Bayesian learning”. In: *4th International Symposium on Independent Component Analysis and Blind Signal Separation*. Vol. 76. 93. 2003, 102.

[Hos18] Haruo Hosoya. “A simple probabilistic deep generative model for learning generalizable disentangled representations from grouped data”. In: *CoRR* abs/1809.0 (2018).

[HG18] Wei-Ning Hsu and James Glass. “Disentangling by Partitioning: A Representation Learning Framework for Multimodal Sensory Data”. In: (2018).

[Hua+20] Shih-Cheng Huang, Anuj Pareek, Saeed Seyyedi, Imon Banerjee, and Matthew P Lungren. “Fusion of medical imaging and electronic health records using deep learning: a systematic review and implementation guidelines”. In: *NPJ digital medicine* 3.1 (2020), 136.

[Hua+18] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. “Multimodal unsupervised image-to-image translation”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, 172.

[HA85] Lawrence Hubert and Phipps Arabie. “Comparing partitions”. In: *Journal of classification* 2 (1985), 193.

[Hui+21] Iris A M Huijben, Wouter Kool, Max B Paulus, and Ruud J G van Sloun. “A Review of the Gumbel-max Trick and its Extensions for Discrete Stochasticity in Machine Learning”. In: *arXiv preprint arXiv:2110.01515* (2021).

[Hwa+21] HyeongJoo Hwang, Geon-Hyeong Kim, Seunghoon Hong, and Kee-Eung Kim. “Multi-view representation learning via total correlation objective”. In: *Advances in Neural Information Processing Systems* 34 (2021), 12194.

[HP99] Aapo Hyvärinen and Petteri Pajunen. “Nonlinear independent component analysis: Existence and uniqueness results”. In: *Neural Networks* 12.3 (1999), 429.

[Irv+19] Jeremy Irvin, Pranav Rajpurkar, Michael Ko, Yifan Yu, Silviana Ciurea-Ilcus, Chris Chute, Henrik Marklund, Behzad Haghgoo, Robyn Ball, Katie Shpanskaya, and others. “Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, 590.

[JJ13] Tommi S Jaakkola and Michael I Jordan. “Computing upper and lower bounds on likelihoods in intractable networks”. In: *arXiv preprint arXiv:1302.3586* (2013).

[JGP16] Eric Jang, Shixiang Gu, and Ben Poole. “Categorical reparameterization with gumbel-softmax”. In: *arXiv preprint arXiv:1611.01144* (2016).

[Jen06] Johan L W V Jensen. “Sur les fonctions convexes et les inégalités entre les valeurs moyennes”. In: *Acta mathematica* 30.1 (1906), 175.

[Jia+16] Zhuxi Jiang, Yin Zheng, Huachun Tan, Bangsheng Tang, and Hanning Zhou. “Variational Deep Embedding: An Unsupervised and Generative Approach to Clustering”. In: *International Joint Conference on Artificial Intelligence* (2016), 1965.

[Joa+07] Thorsten Joachims, Hang Li, Tie-Yan Liu, and ChengXiang Zhai. “Learning to rank for information retrieval”. In: *Acm Sigir Forum*. Vol. 41. 2. 2007, 58.

[Joh+19] Alistair E W Johnson, Tom J Pollard, Seth J Berkowitz, Nathaniel R Greenbaum, Matthew P Lungren, Chih-ying Deng, Roger G Mark, and Steven Horng. “MIMIC-CXR, a de-identified publicly available database of chest radiographs with free-text reports”. In: *Scientific Data* 6 (2019).

[Joh87] Mark E Johnson. *Multivariate statistical simulation: A guide to selecting and generating continuous multivariate distributions*. Vol. 192. John Wiley & Sons, 1987.

[Jor+99] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. “An Introduction to Variational Methods for Graphical Models”. In: *Machine Learning* 37.2 (1999), 183.

[JJ94] Michael I Jordan and Robert A Jacobs. "Hierarchical mixtures of experts and the EM algorithm". In: *Neural computation* 6.2 (1994), 181.

[Joy+21] Tom Joy, Yuge Shi, Philip H S Torr, Tom Rainforth, Sebastian M Schmon, and N Siddharth. "Learning multimodal VAEs through mutual supervision". In: *arXiv preprint arXiv:2106.12570* (2021).

[Jum+21] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, and others. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), 583.

[KFF15] Andrej Karpathy and Li Fei-Fei. "Deep visual-semantic alignments for generating image descriptions". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, 3128.

[KB14] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[KW14] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *International Conference on Learning Representations*. 2014.

[KL21] Nahum Kiryati and Yuval Landau. "Dataset growth in medical image analysis research". In: *Journal of imaging* 7.8 (2021), 155.

[Knu97] Donald E Knuth. *The art of computer programming*. Vol. 3. Pearson Education, 1997.

[Kol30] Andrey N Kolmogorov. "On the Notion of Mean". In: *Mathematics and Mechanics* 199.1 (1930), 144.

[Kol33] Andrey N Kolmogorov. "Sulla determinazione empirica di una legge di distribuzione". In: *Inst. Ital. Attuari, Giorn.* 4 (1933), 83.

[Kol50] Andrey N Kolmogorov. "Unbiased estimates". In: *Izvestiya Rossiiskoi Akademii Nauk. Seriya Matematicheskaya* 14.4 (1950), 303.

[KHW19] Wouter Kool, Herke van Hoof, and Max Welling. "Stochastic Beams and Where To Find Them: The Gumbel-Top-k Trick for Sampling Sequences Without Replacement". In: *Proceedings of the International Conference on Machine Learning*. 2019, 3499.

[KHW20a] Wouter Kool, Herke van Hoof, and Max Welling. "Ancestral Gumbel-Top-k Sampling for Sampling Without Replacement". In: *Journal of Machine Learning Research* 21.47 (2020), 1.

[KHW20b] Wouter Kool, Herke van Hoof, and Max Welling. "Estimating Gradients for Discrete Random Variables by Sampling without Replacement". In: *International Conference on Learning Representations*. 2020.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. 2012, 1106.

[KL51] Solomon Kullback and Richard A Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), 79.

[Kur+22] Vitaly Kurin, Alessandro De Palma, Ilya Kostrikov, Shimon Whiteson, and Pawan K Mudigonda. "In defense of the unitary scalarization for deep multi-task learning". In: *Advances in Neural Information Processing Systems* 35 (2022), 12169.

[KGS19] Richard Kurle, Stephan Günnemann, and Patrick van der Smagt. "Multi-Source Neural Variational Inference". In: *The Thirty-Third Conference on the Advancement of Artificial Intelligence (AAAI)*. 2019, 4114.

[Kyn+19] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. "Improved precision and recall metric for assessing generative models". In: *Advances in Neural Information Processing Systems* 32 (2019).

[Lak+17] Brenden M Lake, Tomer D Ullman, Joshua B Tenenbaum, and Samuel J Gershman. "Building machines that learn and think like people". In: *Behavioral and brain sciences* 40 (2017), e253.

[LFK09] Andrea Lancichinetti, Santo Fortunato, and János Kertész. "Detecting the overlapping and hierarchical community structure in complex networks". In: *New journal of physics* 11.3 (2009), 33015.

[Las+07] Sergey A Lashin, Valentin V Suslov, Nikolay A Kolchanov, and Yury G Matushkin. “Simulation of coevolution in community by using the" Evolutionary Constructor" program”. In: *In silico biology* 7.3 (2007), 261.

[LeC+98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), 2278.

[LS22] Changwoo J Lee and Huiyan Sang. “Why the rich get richer? On the balancedness of random partition models”. In: *Proceedings of the International Conference on Machine Learning*. 2022, 12521.

[Lem14] Christiane Lemieux. “Control variates”. In: *Wiley StatsRef: Statistics Reference Online* (2014), 1.

[Li+22] Yunfan Li, Mouxing Yang, Dezhong Peng, Taihao Li, Jiantao Huang, and Xi Peng. “Twin contrastive learning for online clustering”. In: *International Journal of Computer Vision* 130.9 (2022), 2205.

[LZM22] Paul Pu Liang, Amir Zadeh, and Louis-Philippe Morency. “Foundations and Recent Trends in Multimodal Machine Learning: Principles, Challenges, and Open Questions”. In: *arXiv preprint arXiv:2209.03430* (2022).

[LR01] Jason G Liao and Ori Rosen. “Fast and stable algorithms for computing and sampling from the noncentral hypergeometric distribution”. In: *The American Statistician* 55.4 (2001), 366.

[Lik32] Rensis Likert. “A technique for the measurement of attitudes.” In: *Archives of psychology* (1932).

[LH03] Chin-Yew Lin and Eduard Hovy. “Automatic evaluation of summaries using n-gram co-occurrence statistics”. In: *Proceedings of the 2003 human language technology conference of the North American chapter of the association for computational linguistics*. 2003, 150.

[Lin+18] Scott W Linderman, Gonzalo E Mena, Hal J Cooper, Liam Paninski, and John P Cunningham. “Reparameterizing the Birkhoff Polytope for Variational Permutation Inference”. In: *International Conference on Artificial Intelligence and Statistics*. Vol. 84. Proceedings of Machine Learning Research. 2018, 1618.

[Lin95] Bruce G Lindsay. *Mixture models: theory, geometry, and applications*. Ims, 1995.

[Lo09] Tie-Yan Liu and others. "Learning to rank for information retrieval". In: *Foundations and Trends in Information Retrieval* 3.3 (2009), 225.

[Liu+15] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. "Deep Learning Face Attributes in the Wild". In: *The IEEE International Conference on Computer Vision (ICCV)*. 2015.

[Loc+19] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Raetsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. "Challenging common assumptions in the unsupervised learning of disentangled representations". In: *Proceedings of the International Conference on Machine Learning*. PMLR. 2019, 4114.

[Loc+20] Francesco Locatello, Ben Poole, Gunnar Rätsch, Bernhard Schölkopf, Olivier Bachem, and Michael Tschannen. "Weakly-supervised disentanglement without compromises". In: *Proceddings of the International Conference on Machine Learning*. PMLR. 2020, 6348.

[Lod+18] Michael A. Lodato, Rachel E. Rodin, Craig L. Bohrson, Michael E. Coulter, Alison R. Barton, Minseok Kwon, Maxwell A. Sherman, Carl M. Vitzthum, Lovelace J. Luquette, Chandri N. Yandava, Pengwei Yang, Thomas W. Chittenden, Nicole E. Hatem, Steven C. Ryu, Mollie B. Woodworth, Peter J. Park, and Christopher A. Walsh. "Aging and neurodegeneration are associated with increased mutations in single human neurons". In: *Science* 359.6375 (2018), 555.

[Lod+15] Michael A. Lodato, Mollie B. Woodworth, Semin Lee, Gilad D. Evrony, Bhaven K. Mehta, Amir Karger, Soohyun Lee, Thomas W. Chittenden, Alissa M. D'Gama, Xuyu Cai, Lovelace J. Luquette, Eunjung Lee, Peter J. Park, and Christopher A. Walsh. "Somatic mutation in single human neurons tracks developmental and transcriptional history". In: *Science* 350.6256 (2015), 94.

[LC02] Rosangela H Loschi and Frederico R B Cruz. "An analysis of the influence of some prior specifications in the identification of change points via product partition model". In: *Computational Statistics & Data Analysis* 39.4 (2002), 477.

[Los+03] Rosangela H Loschi, Frederico R B Cruz, Pilar Loreto Iglesias, and Reinaldo B Arellano-Valle. "A Gibbs sampling scheme to the product partition model: an application to change-point problems". In: *Computers & Operations Research* 30.3 (2003), 463.

[LH17] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[Luc59] Robert D Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 1959.

[LHM19] Arno Lukas, Andreas Heinzel, and Bernd Mayer. "Biomarkers for capturing disease pathology as molecular process hyperstructure". In: *bioRxiv* (2019), 573402.

[Luo+15] Yong Luo, Dacheng Tao, Kotagiri Ramamohanarao, Chao Xu, and Yonggang Wen. "Tensor canonical correlation analysis for multi-view dimension reduction". In: *IEEE transactions on Knowledge and Data Engineering* 27.11 (2015), 3111.

[Mac67] James MacQueen. "Classification and analysis of multivariate observations". In: *5th Berkeley Symp. Math. Statist. Probability*. 1967, 281.

[MMT17] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. "The concrete distribution: A continuous relaxation of discrete random variables". In: *International Conference on Learning Representations*. OpenReview.net, 2017.

[MTM14] Chris J Maddison, Daniel Tarlow, and Tom Minka. "A* sampling". In: *Advances in neural information processing systems* 27 (2014).

[Mai+18] Andreas K Maier, Stefan Steidl, Vincent Christlein, and Joachim Hornegger. "Medical Imaging Systems: An Introductory Guide". In: Springer, 2018.

[Man+21] Laura Manduchi, Kieran Chin-Cheong, Holger Michel, Sven Wellmann, and Julia Vogt. "Deep conditional gaussian mixture model for constrained clustering". In: *Advances in Neural Information Processing Systems* 34 (2021), 11303.

[MS16] Toufik Mansour and Matthias Schork. *Commutation relations, normal ordering, and Stirling numbers*. CRC Press Boca Raton, 2016.

[Mar60] Marvin Marcus. "Some Properties and Applications of Doubly Stochastic Matrices". In: *The American Mathematical Monthly* 67.3 (1960), 215.

[Mar96] John I Marden. *Analyzing and modeling rank data*. CRC Press, 1996.

[MGH11] Aaron F McDaid, Derek Greene, and Neil Hurley. "Normalized mutual information to evaluate overlapping community finding algorithms". In: *arXiv preprint arXiv:1110.2515* (2011).

[Men+18] Gonzalo E Mena, David Belanger, Scott W Linderman, and Jasper Snoek. "Learning Latent Permutations with Gumbel-Sinkhorn Networks". In: *International Conference on Learning Representations*. 2018.

[Mer] Merriam-Webster. *www.merriam-webster.com/dictionary/modality*.

[Met+53] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. "Equation of State Calculations by Fast Computing Machines". In: *The Journal of Chemical Physics* 21.6 (1953), 1087.

[Mik+13] Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S Corrado, and Jeffrey Dean. "Distributed Representations of Words and Phrases and their Compositionality". In: *Advances in Neural Information Processing Systems*. 2013, 3111.

[Mil21] Djordje Miladinovic. "On Training Deep Generative Models with Latent Variables". PhD thesis. ETH Zurich, 2021.

[MH18] Jeffrey W Miller and Matthew T Harrison. "Mixture models with a prior on the number of components". In: *Journal of the American Statistical Association* 113.521 (2018), 340.

[Mis36] Richard von Mises. "La distribution de la plus grande de n valeurs". In: *Rev. Math. Union Interbalcanique* 1 (1936), 141.

[Mit07] Tom M Mitchell. *Machine learning*. Vol. 1. McGraw-hill New York, 2007.

[Mon81] Gaspard Monge. "Mémoire sur la théorie des déblais et des remblais". In: *Mem. Math. Phys. Acad. Royale Sci.* (1781), 666.

[MHA14] Emilie Morvant, Amaury Habrard, and Stéphane Ayache. "Majority vote of diverse classifiers for late fusion". In: *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop*. 2014, 153.

[Mos+22] Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. "Relative representations enable zero-shot latent space communication". In: *arXiv preprint arXiv:2209.15430* (2022).

[MMG15] Youssef Mroueh, Etienne Marcheret, and Vaibhava Goel. "Deep multimodal learning for Audio-Visual Speech Recognition". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2015, 2130.

[MQR11] Peter Müller, Fernando A Quintana, and Gary L Rosner. "A Product Partition Model With Regression on Covariates". In: *Journal of Computational and Graphical Statistics* 20.1 (2011), 260.

[Mül+15] Peter Müller, Fernando Andres Quintana, Alejandro Jara, Tim Hanson, Peter Müller, Fernando Andrés Quintana, Alejandro Jara, and Tim Hanson. "Clustering and Feature Allocation". In: *Bayesian Nonparametric Data Analysis* (2015), 145.

[Mur07] Kevin P Murphy. "Conjugate Bayesian analysis of the Gaussian distribution". In: *def* $1.2\sigma 2$ (2007), 16.

[MR84] Fionn Murtagh and Adrian E Raftery. "Fitting straight lines to point patterns". In: *Pattern recognition* 17.5 (1984), 479.

[Nae+20] Muhammad F Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. "Reliable fidelity and diversity metrics for generative models". In: *Proceedings of the International Conference on Machine Learning*. 2020, 7176.

[Net+11] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. "Reading digits in natural images with unsupervised feature learning". In: (2011).

[Ngi+11] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. "Multimodal deep learning". In: *Proceedings of the International Conference on Machine Learning*. 2011, 689.

[NP05] Constantin Niculescu and Lars-Erik Persson. *Convex Functions and Their Applications: A Contemporary Approach*. Springer, 2005.

[Nie19] Frank Nielsen. "On the Jensen-Shannon symmetrization of distances relying on abstract means". In: *Entropy* (2019).

[OT03] Wlodzimierz Ogryczak and Arie Tamir. "Minimizing the sum of the k largest functions in linear time". In: *Information Processing Letters* 85.3 (2003), 117.

[OMT03] Seiji Ono, Kazuharu Misawa, and Kazuki Tsuji. "Effect of group selection on the evolution of altruistic behavior". In: *Journal of theoretical biology* 220.1 (2003), 55.

[OLV18] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).

[OCW14] Wanli Ouyang, Xiao Chu, and Xiaogang Wang. "Multi-source Deep Learning for Human Pose Estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, 2337.

[PBJ12] John Paisley, David M Blei, and Michael I Jordan. "Variational Bayesian Inference with Stochastic Search". In: *Proceedings of the International Coference on International Conference on Machine Learning*. ICML'12. 2012, 1363.

[PDV22] Emanuele Palumbo, Imant Daunhawer, and Julia E Vogt. "MMVAE+: Enhancing the generative quality of multimodal VAEs without compromises". In: *International Conference on Learning Representations*. 2022.

[Pan+15] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. "Librispeech: an asr corpus based on public domain audio books". In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. 2015, 5206.

[PCT06] Mario Paolucci, Rosaria Conte, and Gennaro Di Tosto. "A model of social organization and the evolution of food sharing in vampire bats". In: *Adaptive Behavior* 14.3 (2006), 223.

[Pap+02] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the Annual Meeting of the Association for Computational Linguistics*. 2002, 311.

[Pas+19] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *CoRR* abs/1912.0 (2019).

[Pau+20] Max B Paulus, Dami Choi, Daniel Tarlow, Andreas Krause, and Chris J Maddison. “Gradient estimation with stochastic softmax tricks”. In: *Advances in Neural Information Processing Systems* 33 (2020), 5691.

[Ped+11] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and others. “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12 (2011), 2825.

[PPY92] Mihael Perman, Jim Pitman, and Marc Yor. “Size-biased sampling of Poisson point processes and excursions”. In: *Probability Theory and Related Fields* 92.1 (1992), 21.

[PJS17] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. *Elements of causal inference: foundations and learning algorithms*. The MIT Press, 2017.

[Pet+21] Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. “Monotonic Differentiable Sorting Networks”. In: *International Conference on Learning Representations*. 2021.

[PP008] Kaare Brandt Petersen, Michael S Pedersen, and others. “The matrix cookbook”. In: *Technical University of Denmark* 7.15 (2008), 510.

[Pit96] Jim Pitman. “Some Developments of the Blackwell-Macqueen URN Scheme”. In: *Lecture Notes-Monograph Series* 30 (1996), 245.

[PY97] Jim Pitman and Marc Yor. “The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator”. In: *The Annals of Probability* (1997), 855.

[Pla68] Robin L Plackett. “Random permutations”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 30.3 (1968), 517.

[Pla75] Robin L Plackett. “The analysis of permutations”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 24.2 (1975), 193.

[PE20] Sebastian Prillo and Julian Eisenschlos. “SoftSort: A Continuous Relaxation for the argsort Operator”. In: *Proceedings of the International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. 2020, 7793.

[Qui06] Fernando A Quintana. “A predictive view of Bayesian clustering”. In: *Journal of Statistical Planning and Inference* 136.8 (2006), 2407.

[RHS08] Sophia Rabe-Hesketh and Anders Skrondal. “Classical latent variable models for medical research”. In: *Statistical methods in medical research* 17.1 (2008), 5.

[Rad+18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and others. “Improving language understanding by generative pre-training”. In: (2018).

[RR45] C Radhakrishna Rao. “Information and accuracy attainable in the estimation of statistical parameters”. In: *Bulletin of the Calcutta Mathematical Society* 37.3 (1945), 81.

[Ram+22] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. “Hierarchical text-conditional image generation with clip latents”. In: *arXiv preprint arXiv:2204.06125* (2022).

[Ram+21] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. “Zero-shot text-to-image generation”. In: *International Conference on Machine Learning*. 2021, 8821.

[Ran71] William M Rand. “Objective criteria for the evaluation of clustering methods”. In: *Journal of the American Statistical association* 66.336 (1971), 846.

[RGB14] Rajesh Ranganath, Sean Gerrish, and David M Blei. “Black Box Variational Inference”. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics*. Vol. 33. JMLR Workshop and Conference Proceedings. 2014, 814.

[RKF19] Kevin Reuning, Michael R Kenwick, and Christopher J Fariss. “Exploring the dynamics of latent variable models”. In: *Political Analysis* 27.4 (2019), 503.

[RMW14] Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic backpropagation and approximate inference in deep generative models". In: *Proceedings of the International Conference on Machine Learning*. 2014, 1278.

[Rib+20] Antônio H Ribeiro, Manoel H Ribeiro, Gabriela M M Paixão, Derick M Oliveira, Paulo R Gomes, Jéssica A Canazart, Milton P S Ferreira, Carl R Andersson, Peter W Macfarlane, Wagner Meira Jr, and others. "Automatic diagnosis of the 12-lead ECG using a deep neural network". In: *Nature communications* 11.1 (2020), 1760.

[RM51] Herbert Robbins and Sutton Monro. "A Stochastic Approximation Method". In: *The Annals of Mathematical Statistics* 22.3 (1951), 400.

[Rom+22] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. "High-resolution image synthesis with latent diffusion models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, 10684.

[Rot64] Gian-Carlo Rota. "The Number of Partitions of a Set". In: *The American Mathematical Monthly* 71.5 (1964), 498.

[RTG00] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. "The earth mover's distance as a metric for image retrieval". In: *International Journal of Computer Vision* 40.2 (2000), 99.

[SFH17] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. "Dynamic Routing Between Capsules". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.

[Sah+22] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, and others. "Photorealistic text-to-image diffusion models with deep language understanding". In: *Advances in Neural Information Processing Systems* 35 (2022), 36479.

[Saj+18] Mehdi S M Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. "Assessing generative models via precision and recall". In: *Advances in Neural Information Processing Systems* 31 (2018).

[Sal+16] Tim Salimans, Ian J Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved Techniques for Training GANs". In: *Advances in Neural Information Processing Systems*. 2016, 2226.

[SC+17] Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian, and Stephen Gould. "Deeppermnet: Visual permutation learning". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, 3949.

[Sat01] Masa-aki Sato. "Online Model Selection Based on the Variational Bayes". In: *Neural Computation* 13.7 (2001), 1649.

[Sch92] Jürgen Schmidhuber. "Learning factorial codes by predictability minimization". In: *Neural computation* 4.6 (1992), 863.

[SK18] Ozan Sener and Vladlen Koltun. "Multi-task learning as multi-objective optimization". In: *Advances in Neural Information Processing Systems* 31 (2018).

[Shi+20] Yuge Shi, Brooks Paige, Philip H S Torr, and N Siddharth. "Relating by Contrasting: A Data-efficient Framework for Multimodal Generative Models". In: (2020).

[Shi+19] Yuge Shi, N Siddharth, Brooks Paige, and Philip Torr. "Variational Mixture-of-Experts Autoencoders for Multi-Modal Deep Generative Models". In: *Advances in Neural Information Processing Systems*. 2019, 15692.

[Shu+19] Rui Shu, Yining Chen, Abhishek Kumar, Stefano Ermon, and Ben Poole. "Weakly supervised disentanglement with guarantees". In: *arXiv preprint arXiv:1910.09772* (2019).

[Sil+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, and others. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), 484.

[Sil+17] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, and others. "Mastering the game of go without human knowledge". In: *nature* 550.7676 (2017), 354.

[SWR19] Loic Simon, Ryan Webster, and Julien Rabin. "Revisiting precision recall definition for generative modeling". In: *Proceedings of the International Conference on Machine Learning*. 2019, 5799.

[Sin64] Richard Sinkhorn. "A relationship between arbitrary positive matrices and doubly stochastic matrices". In: *The Annals of Mathematical Statistics* 35.2 (1964), 876.

[Smi39] Nikolai V Smirnov. "On the estimation of the discrepancy between empirical curves of distribution for two independent samples". In: *Bull. Math. Univ. Moscou* 2.2 (1939), 3.

[Spi19] Michael Spivak. *Calculus*. Reverté, 2019.

[Sto79] Robert Roth Stoll. *Set theory and logic*. Courier Corporation, 1979.

[ST10] Kyle Strimbu and Jorge A Tavel. "What are biomarkers?" In: *Current Opinion in HIV and AIDS* 5.6 (2010), 463.

[SDV20] Thomas M Sutter, Imant Daunhawer, and Julia E Vogt. "Multimodal Generative Learning Utilizing Jensen-Shannon Divergence". In: *Advances in Neural Information Processing Systems* (2020).

[SDV21] Thomas M Sutter, Imant Daunhawer, and Julia E Vogt. "Generalized Multimodal ELBO". In: *International Conference on Learning Representations* (2021).

[Sut+23a] Thomas M Sutter, Laura Manduchi, Alain Ryser, and Julia E Vogt. "Learning Group Importance using the Differentiable Hypergeometric Distribution". In: *International Conference on Learning Representations* (2023).

[Sut+23b] Thomas M Sutter, Alain Ryser, Joram Liebeskind, and Julia E Vogt. "Differentiable Random Partition Models". In: *Under Submission*. 2023.

[SV21] Thomas M Sutter and Julia E Vogt. "Multimodal Relational VAE". In: *Neurips Workshop on Bayesian Deep Learning*. 2021.

[SM22] Masahiro Suzuki and Yutaka Matsuo. "A survey of multimodal deep generative models". In: *Advanced Robotics* 36.5-6 (2022), 261.

[SNM16] Masahiro Suzuki, Kotaro Nakayama, and Yutaka Matsuo. "Joint Multimodal Learning with Deep Generative Models". In: *arXiv preprint arXiv:1611.01891* (2016), 1.

[Sze+16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. "Rethinking the inception architecture for computer vision". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, 2818.

[TOB15] Lucas Theis, Aäron van den Oord, and Matthias Bethge. "A note on the evaluation of generative models". In: *arXiv preprint arXiv:1511.01844* (2015), 1.

[Thu27] Louis L Thurstone. "A law of comparative judgment". In: *Scaling*. Routledge, 1927, 81.

[TE19] Yingtao Tian and Jesse Engel. "Latent translation: Crossing modalities by bridging generative models". In: *arXiv preprint arXiv:1902.08261* (2019).

[TLG15] Michalis K Titsias and Miguel Lázaro-Gredilla. "Local expectation gradients for black box variational inference". In: *Advances in Neural Information Processing Systems* 28 (2015).

[Tom22] Jakub M Tomczak. *Deep Generative Modeling*. Springer, 2022.

[Tsa+19] Yao-Hung Hubert Tsai, Paul Pu Liang, Amir Zadeh, Louis-Philippe Morency, and Ruslan Salakhutdinov. "Learning Factorized Multimodal Representations". In: *International Conference on Learning Representations*. 2019.

[TBL18] Michael Tschannen, Olivier Bachem, and Mario Lucic. "Recent advances in autoencoder-based representation learning". In: *arXiv preprint arXiv:1812.05069* (2018).

[VK20] Arash Vahdat and Jan Kautz. "NVAE: A deep hierarchical variational autoencoder". In: *Advances in Neural Information Processing Systems* 33 (2020), 19667.

[VDO+16] Aäron Van Den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. "Wavenet: A generative model for raw audio". In: *arXiv preprint arXiv:1609.03499* (2016).

[VDOVo17] Aäron Van Den Oord, Oriol Vinyals, and others. "Neural discrete representation learning". In: *Advances in Neural Information Processing Systems* (2017).

[Vap91] Vladimir Vapnik. “Principles of risk minimization for learning theory”. In: *Advances in Neural Information Processing Systems* (1991).

[Vas+17] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. “Attention is all you need”. In: *Advances in Neural Information Processing Systems* (2017).

[Ved+18] Ramakrishna Vedantam, Ian Fischer, Jonathan Huang, and Kevin Murphy. “Generative Models of Visually Grounded Imagination”. In: *International Conference on Learning Representations*. 2018.

[VLZP15] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. “Cider: Consensus-based image description evaluation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, 4566.

[VEB09] Nguyen Xuan Vinh, Julien Epps, and James Bailey. “Information theoretic measures for clusterings comparison: is a correction for chance necessary?” In: *Proceedings of the International Conference on Machine Learning*. 2009, 1073.

[Vir+20] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, and others. “SciPy 1.0: fundamental algorithms for scientific computing in Python”. In: *Nature methods* 17.3 (2020), 261.

[WJ08] Martin J Wainwright and Michael I Jordan. *Graphical Models, Exponential Families, and Variational Inference*. Foundations and trends in machine learning. Now Publishers, 2008.

[Wal63] Kenneth T Wallenius. *Biased sampling; the noncentral hypergeometric probability distribution*. Tech. rep. Stanford Univ Ca Applied Mathematics And Statistics Labs, 1963.

[Wan+19] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. “Superglue: A stickier benchmark for general-purpose language understanding systems”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[Wan+18] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. "GLUE: A multi-task benchmark and analysis platform for natural language understanding". In: *arXiv preprint arXiv:1804.07461* (2018).

[Wan+15] Daixin Wang, Peng Cui, Mingdong Ou, and Wenwu Zhu. "Deep Multimodal Hashing with Orthogonal Regularization". In: *Proceedings of the International Joint Conference on Artificial Intelligence*. 2015, 2291.

[WW96] Edmund T Whittaker and George N Watson. *A Course of Modern Analysis*. 4th ed. Cambridge Mathematical Library. Cambridge University Press, 1996.

[Wil92] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3 (1992), 229.

[WG18] Mike Wu and Noah Goodman. "Multimodal Generative Models for Scalable Weakly-Supervised Learning". In: *Advances in Neural Information Processing Systems*. 2018, 5580.

[WG19] Mike Wu and Noah Goodman. *Multimodal Generative Models for Compositional Representation Learning*. 2019.

[Wu+14] Zuxuan Wu, Yu-Gang Jiang, Jun Wang, Jian Pu, and Xiangyang Xue. "Exploring Inter-feature and Inter-class Relationships with Deep Neural Networks for Video Classification". In: *Proceedings of the ACM International Conference on Multimedia*. 2014, 167.

[XRV17] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017.

[XE19] Sang M Xie and Stefano Ermon. "Reparameterizable subset sampling via continuous relaxations". In: *arXiv preprint arXiv:1901.10517* (2019).

[Xin+22] Derrick Xin, Behrooz Ghorbani, Justin Gilmer, Ankush Garg, and Orhan Firat. "Do Current Multi-Task Optimization Methods in Deep Learning Even Help?" In: *Advances in Neural Information Processing Systems* (2022), 13597.

[XTX13] Chang Xu, Dacheng Tao, and Chao Xu. "A survey on multiview learning". In: *arXiv preprint arXiv:1304.5634* (2013).

[Yan+17] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. "Improved variational autoencoders for text modeling using dilated convolutions". In: *Proceedings of the International Conference on Machine Learning*. 2017, 3881.

[Yel77] John I Yellott. "The relationship between Luce's choice axiom, Thurstone's theory of comparative judgment, and the double exponential distribution". In: *Journal of Mathematical Psychology* 15.2 (1977), 109.

[Yel01] John I Yellott. "Luce's Choice Axiom". In: 2001.

[Zha+18] Cheng Zhang, Judith Butepage, Hedvig Kjellstrom, and Stephan Mandt. "Advances in Variational Inference". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2018), 1.

A

APPENDIX

A.1 GRADIENT-BASED OPTIMIZATION OF DISCRETE STRUCTURES

A.1.1 *Proof of Gumbel-Max Trick*

In this section, we provide the proof of the Gumbel-Max trick in Theorem 2.4.1.

Proof. In the proof, we show that $P(I = k) = \alpha_k$, i. e. sampling using the Gumbel-Max as described in Theorem 2.4.1 results in exact samples from $Cat(\boldsymbol{\alpha})$.

$$P(I = k) = \mathbb{E}_{G(\log s_k, 1)} \left[\prod_{j \le K, j \ne k} p(G(\log s_j, 1) < G(\log s_k, 1))) \right] \tag{A.1}$$

where we start the proof by understanding that $P(I = k) = \alpha_k$ holds if and only if $G(\log s_k, 1) > G(\log s_j, 1), \forall\, j \le K, j \ne k$. The product on the right hand side in Equation (A.1) follows from the independence assumption of the gumbel variables g_k. In the following, we use $f_k(\cdot)$ to denote the PDF of the distribution $G(\log s_k, 1))$. We continue the proof with

$$P(I = k) = \int f_k(m) \prod_{j \le K, j \ne k} p(G(\log s_j, 1) < m))dm \tag{A.2}$$

$$= \int f_k(m) \prod_{j \le K, j \ne k} \exp\left(-\exp\left(\log s_j - m\right)\right) dm \tag{A.3}$$

$$= \int f_k(m) \exp\left(- \sum_{j \le K, j \ne k} \exp\left(\log s_j - m\right)\right) dm \tag{A.4}$$

$$= \int \exp\left(\log s_k - m - \exp\left(\log s_k - m\right)\right) \cdot \exp\left(- \sum_{j \le K, j \ne k} \exp\left(\log s_j - m\right)\right) dm \tag{A.5}$$

$$= \int \exp\left(\log s_k - m\right) \exp\left(- \sum_{j \le K} \exp\left(\log s_j - m\right)\right) dm \tag{A.6}$$

$$= \int s_k \exp(-m) \exp\left(-\exp(-m) \cdot \sum_{j \leq K} s_j\right) dm \tag{A.7}$$

We have $Z = \sum_{j=1}^{K} s_j$ and $\alpha_k = s_k Z$ by definition:

$$P(I = k) = \alpha_k Z \int \exp(-m) \exp(-Z \exp(-m)) \, dm \tag{A.8}$$

Using $\int \exp(-m) \exp(-Z \exp(-m)) \, dm = \frac{1}{Z}$, it follows

$$P(I = k) = \alpha_k \tag{A.9}$$

□

A.1.2 *Proof of the equality between Thurstonian and Plackett-Luce models*

In this section, we provide the proof of the Gumbel-Max trick in Theorem 2.4.2.

Proof. We take the proof from Grover et al. [Gro+19], which again follows a result from Yellott [Yel77], and also provide a proof sketch only and refer the reader to Yellott [Yel77] for more details.

Consider random variables $\{X_i\}_{i=1}^{n}$ such that $X_i \sim \exp(s_i)$. We may prove by induction a generalization of the memoryless property:

$$q(X_{1\ldots n} \mid x \leq \min_i X_i) = q(X_1 \leq \cdots \leq X_n \mid x \leq \min_i X_i) \tag{A.10}$$

$$= \int_0^\infty q(x \leq X_1 \leq X_1 + t \mid x \leq \min_i X_i) \tag{A.11}$$

$$q(X_2 \leq \cdots \leq X_n \mid x + t \leq \min_{i \geq 2} X_i) dt$$

$$= \int_0^\infty q(0 \leq X_1 \leq t) \tag{A.12}$$

$$q(X_2 \leq \cdots \leq X_n \mid x + t \leq \min_{i \geq 2} X_i) dt$$

If we assume as inductive hypothesis that $q(X_2 \leq \cdots \leq X_n \mid x + t \leq \min_{i \geq 2} X_i) = q(X_2 \leq \cdots \leq X_n \mid t \leq \min_{i \geq 2} X_i)$, we complete the induction as:

$$q(X_{1\ldots n} \mid x \leq \min_i X_i) = q(X_1 \leq \cdots \leq X_n \mid x \leq \min_i X_i) \tag{A.13}$$

$$= \int_0^\infty q(0 \leq X_1 \leq t) \tag{A.14}$$

$$q(X_2 \leq \cdots \leq X_n \mid t \leq \min_{i\geq 2} X_i)dt$$

$$=q(X_1 \leq X_2 \leq \cdots \leq X_n \mid 0 \leq \min_i X_i) \qquad \text{(A.15)}$$

It follows from a familiar property of argmin of exponential distributions that:

$$q(X_{1\ldots n} \mid x \leq \min_i X_i) = q(X_1 \leq \cdots \leq X_n \mid x \leq \min_i X_i) \qquad \text{(A.16)}$$

$$=q(X_1 \leq \min_i X_i)q(X_2 \leq \cdots \leq X_n \mid X_i \leq \min_i X_i) \qquad \text{(A.17)}$$

$$=\frac{s_i}{Z}q(X_2 \leq \cdots \leq X_n \mid X_i \leq \min_i X_i) \qquad \text{(A.18)}$$

$$=\frac{s_i}{Z}\int_0^\infty q(X_1 = x)q(X_2 \leq \cdots \leq X_n \mid x \leq \min_{i\geq 2} X_i)dx \qquad \text{(A.19)}$$

$$=\frac{s_i}{Z}q(X_2 \leq \cdots \leq X_n) \qquad \text{(A.20)}$$

and by another induction, we have

$$q(X_1 \leq \cdots \leq X_n) = \prod_{i=}^{n} \frac{s_i}{Z - \sum_{j=1}^{i-1} s_j}. \qquad \text{(A.21)}$$

Finally, following the argument of Balog et al. [Bal+17], we apply the strictly decreasing function $g(x) = -\beta \log x$ to this identity, which from the definition of the Gumbel distribution implies

$$q(\tilde{s}_1 \leq \cdots \leq \tilde{s}_n) = \prod_{i=}^{n} \frac{s_i}{Z - \sum_{j=1}^{i-1} s_j}. \qquad \text{(A.22)}$$

We follow Grover et al. [Gro+19] for this proof. □

A.1.3 *Derivation of differentiable Sorting Operator*

Grover et al. [Gro+19] make use of the following lemma [OT03] in the derivation of their differentiable sorting procedure. For completeness, we provide the lemma here.

Lemma A.1.1 (Lemma 1 from Ogryczak and Tamir [OT03]). *Given a vector of scores $\boldsymbol{s} = [s_1, \ldots, s_n]^T \in \mathbb{R}_+^n$ and a permutation matrix $\pi \in \Pi_n$, which sorts* *the scores $\boldsymbol{s}$ in decreasing order, i. e. $(\pi \boldsymbol{s}))_1 \geq (\pi \boldsymbol{s}))_2 \geq \ldots \geq (\pi \boldsymbol{s}))_n$, the sum of the k-largest elements is given as*

$$\sum_{i=1}^{k} (\pi \boldsymbol{s}))i = \min_{\lambda \in \boldsymbol{s}} \lambda k + \sum_{i=1}^{n} \max(s_i - \lambda, 0) \tag{A.23}$$

Proof. For any value of λ, the following inequalities hold

$$\sum_{i=1}^{k} (\pi \boldsymbol{s})_i = \lambda k + \sum_{i=1}^{k} ((\pi \boldsymbol{s})_i - \lambda) \tag{A.24}$$

$$\leq \lambda k + \sum_{i=1}^{k} \max((\pi \boldsymbol{s})_i - \lambda, 0) \tag{A.25}$$

$$\leq \lambda k + \sum_{i=1}^{n} \max((\pi \boldsymbol{s})_i - \lambda, 0) \tag{A.26}$$

Furthermore, for $\lambda = (\pi \boldsymbol{s})_k$:

$$\lambda k + \sum_{i=1}^{n} \max((\pi \boldsymbol{s})_i - \lambda, 0) = (\pi \boldsymbol{s})_k k + \sum_{i=1}^{n} \max((\pi \boldsymbol{s})_i - (\pi \boldsymbol{s})_k, 0) \tag{A.27}$$

$$= (\pi \boldsymbol{s})_k k + \sum_{i=1}^{k} ((\pi \boldsymbol{s})_i - (\pi \boldsymbol{s})_k) \tag{A.28}$$

$$= \sum_{i=1}^{k} (\pi \boldsymbol{s})_i \tag{A.29}$$

□

A.1.4 *Proof of Differentiable Permutation Matrix*

In this section, we provide the proof of Corollary 2.4.2.

Proof. We first consider at exactly what values of λ the sum in Lemma A.1.1 is minimized. Please note that the proof only holds if all values $s_i \in \boldsymbol{s}$ are distinct.

The equality $\sum_{i=1}^{k} (\pi \boldsymbol{s})_i = \lambda k + \sum_{i=1}^{n} \max(s_i - \lambda, 0)$ holds only when $(\pi \boldsymbol{s})_k \leq \lambda \leq (\pi \boldsymbol{s})_{k+1}$. Following Lemma A.1.1, these values of λ also minimize the right-hand side of the equality.

Symmetrically, if we consider the scores $\boldsymbol{t} = -\boldsymbol{s}$, then $\lambda(n - k + 1) + \sum_{i=1}^{n} \max(t_i - \lambda, 0)$ is minimized at $(\pi \boldsymbol{t})_{n-k+1} \leq \lambda \leq (\pi \boldsymbol{t})_k$.

Replacing λ with $-\lambda$ and using the definition of $\boldsymbol{t}$ implies that $\lambda(k-1-n)+\sum_{i=1}^{n}\max(\lambda-s_i,0)$ is minimized at $(\pi\boldsymbol{s})_{k-1}\leq\lambda\leq(\pi\boldsymbol{s})_k$.

It follows

$$\begin{aligned}(\pi\boldsymbol{s})_k &= \underset{\lambda\in\boldsymbol{s}}{\arg\min}\left[\left(\lambda k+\sum_{i=1}^{n}\max(s_i-\lambda,0)\right)\right.\\ &\quad\left.+\left(\lambda k(k-1-n)+\sum_{i=1}^{n}\max(\lambda-s_i,0)\right)\right] &\text{(A.30)}\\ &= \underset{\lambda\in\boldsymbol{s}}{\arg\min}\left[\lambda(2k-1-n)+\sum_{i=1}^{n}|s_i-\lambda|\right] &\text{(A.31)}\end{aligned}$$

Therefore, it follows that if $s_i=(\pi\boldsymbol{s})_k$, then $i=\arg\min(2k-1-n)\boldsymbol{s}+A_{\boldsymbol{s}}\mathbb{1}$. □

A.2 JOINT LATENT SPACE MODELS

A.2.1 *PolyMNIST Background Images*

We use the following background images $\boldsymbol{BG}_m$ to generate the PolyMNIST dataset:

1. John Burkardt. Licensed under GNU LGPL.
   `https://people.sc.fsu.edu/~jburkardt/data/jpg/fractal_tree.jpg`
   [Online; retrieved 27.09.2020]

2. Edvard Munch. The Scream. Public domain.
   `https://upload.wikimedia.org/wikipedia/commons/f/f4/The_Scream.jpg`
   [Online; retrieved 27.09.2020]

3. The Waterloo Image Repository. Lena. Copyright belongs to the author.
   `http://links.uwaterloo.ca/Repository/TIF/lena3.tif`
   [Online; retrieved 27.09.2020]

4. John Burkardt. Licensed under GNU LGPL.
   `https://people.sc.fsu.edu/~jburkardt/data/jpg/star_field.jpg`
   [Online; retrieved 27.09.2020]

5. John Burkardt. Licensed under GNU LGPL.
   `https://people.sc.fsu.edu/~jburkardt/data/jpg/shingles.jpg`
   [Online; retrieved 27.09.2020]

A.2.2 *Network Architectures*

This section describes the network architectures of the encoders and decoders used in Chapters 3 and 4.

Layer	Type	# Features In	# Features Out
1	linear	784	400
2a	linear	400	20
2b	linear	400	20

(a) MNIST Encoder

Layer	Type	# Features In	# Features Out
1	linear	20	400
2	linear	400	784

(b) MNIST Decoder

TABLE A.1: MIST: Encoder and Decoder Layers. The ReLU activation function follows every layer. The only exceptions are the linear layers 2a and 2b, which we use to map the encodings to μ and $\sigma^2 \boldsymbol{I}$ of the approximate posterior distribution.

A.3 MODALITY-SPECIFIC LATENT SUBSPACES

A.3.1 *Network Architectures for Bimodal CelebA*

This section describes the network architectures used in the bimodal CelebA experiment in Chapter 4.

Layer	Type	#F. In	#F. Out	Spec.
1	conv_{2d}	3	32	(4, 2, 1, 1)
2	conv_{2d}	32	64	(4, 2, 1, 1)
3	conv_{2d}	64	64	(4, 2, 1, 1)
4	conv_{2d}	64	128	(4, 2, 0, 1)
5a	linear	128	20	
5b	linear	128	20	

(a) SVHN Encoder

Layer	Type	#F. In	#F. Out	Spec.
1	linear	20	128	
2	conv_{2d}^{T}	128	64	(4, 2, 0, 1)
3	conv_{2d}^{T}	64	64	(4, 2, 1, 1)
4	conv_{2d}^{T}	64	32	(4, 2, 1, 1)
5	conv_{2d}^{T}	32	3	(4, 2, 1, 1)

(b) SVHN Decoder

Table A.3: SVHN: Encoder and Decoder Layers. The specifications name kernel size, stride, padding, and dilation. A ReLU activation function follows all layers.

Layer	Type	#F. In	#F. Out	Spec.
1	conv_{1d}	71	128	(1, 1, 0, 1)
2	conv_{1d}	128	128	(4, 2, 1, 1)
3	conv_{1d}	128	128	(4, 2, 0, 1)
4a	linear	128	20	
4b	linear	128	20	

(a) Text Encoder

Layer	Type	#F. In	#F. Out	Spec.
1	linear	20	128	
2	conv_{1d}^{T}	128	128	(4, 1, 0, 1)
3	conv_{1d}^{T}	128	128	(4, 2, 1, 1)
4	conv_{1d}^{T}	128	71	(1, 1, 0, 1)

(b) Text Decoder

Table A.5: Text for MNIST-SVHN-Text: Encoder and Decoder Layers. The specifications name kernel size, stride, padding, and dilation. A ReLU activation function follows all layers.

Layer	Type	#F. In	#F. Out	Spec.
1	conv_{2d}	3	32	(4, 2, 1, 1)
2	conv_{2d}	32	64	(4, 2, 1, 1)
3	conv_{2d}	64	64	(4, 2, 1, 1)
4	conv_{2d}	64	128	(4, 2, 0, 1)
5	Flatten	128	2048	
6a	linear	2048	512	
6b	linear	2048	512	

(a) PolyMNIST Encoder

Layer	Type	#F. In	#F. Out	Spec.
1	linear	512	2048	
2	Unflatten	2048	128	
3	conv_{2d}^{T}	128	64	(4, 2, 0, 1)
4	conv_{2d}^{T}	64	64	(4, 2, 1, 1)
5	conv_{2d}^{T}	64	32	(4, 2, 1, 1)
6	conv_{2d}^{T}	32	3	(4, 2, 1, 1)

(b) PolyMNIST Decoder

Table A.7: PolyMNIST network architectures. We use the same basic structure for all modalities $\boldsymbol{x}_m \in \boldsymbol{X}$, but with different initialization such that the networks represent different encoding and decoding functions. The specifications name kernel size, stride, padding, and dilation. A ReLU activation function follows all layers.

Layer	Type	#F. In	#F. Out	Spec.
1	conv_{2d}	3	128	(3, 2, 1, 1)
2	res_{2d}	128	256	(4, 2, 1, 1)
3	res_{2d}	256	384	(4, 2, 1, 1)
4	res_{2d}	384	512	(4, 2, 1, 1)
5	res_{2d}	512	640	(4, 2, 1, 1)
6a	linear	640	32	
6b	linear	640	32	

(a) Image Encoder

Layer	Type	#F. In	#F. Out	Spec.
1	linear	64	640	
2	res_{2d}^{T}	640	512	(4, 1, 0, 1)
3	res_{2d}^{T}	512	384	(4, 1, 1, 1)
4	res_{2d}^{T}	384	256	(4, 1, 1, 1)
5	res_{2d}^{T}	256	128	(4, 1, 1, 1)
6	conv_{2d}^{T}	128	3	(3, 2, 1, 1)

(b) Image Decoder

Table A.9: CelebA Image: Encoder and Decoder Layers. The specifications name kernel size, stride, padding, and dilation. res names a residual block.

Layer	Type	#F. In	#F. Out	Spec.
1	conv$_{1d}$	71	128	(3, 2, 1, 1)
2	res$_{1d}$	128	256	(4, 2, 1, 1)
3	res$_{1d}$	256	384	(4, 2, 1, 1)
4	res$_{1d}$	384	512	(4, 2, 1, 1)
5	res$_{1d}$	512	640	(4, 2, 1, 1)
6	res$_{1d}$	640	640	(4, 2, 1, 1)
7	res$_{1d}$	640	640	(4, 2, 0, 1)
8a	linear	640	32	
8b	linear	640	32	

(a) Text Encoder

Layer	Type	#F. In	#F. Out	Spec.
1	linear	64	896	
2	res$_{1d}^{T}$	640	640	(4, 2, 0, 1)
3	res$_{1d}^{T}$	640	640	(4, 2, 1, 1)
4	res$_{1d}^{T}$	640	512	(4, 2, 1, 1)
5	res$_{1d}^{T}$	512	384	(4, 2, 1, 1)
6	res$_{1d}^{T}$	384	256	(4, 2, 1, 1)
7	res$_{1d}^{T}$	256	128	(4, 2, 1, 1)
8	conv$_{1d}^{T}$	128	71	(3, 2, 1, 1)

(b) Text Decoder

Table A.11: CelebA Text: Encoder and Decoder Layers. The specifications name kernel size, stride, padding, and dilation. res names a residual block.

A.4 LEARNING THE RELATIONSHIP BETWEEN GROUPS OF SAMPLES

A.4.1 *Clustering*

Since we applied an upper bound on the KL-divergence $D_{\mathrm{KL}}[q_\phi(Y \mid X) \mid\mid p_\theta(Y)]$, we need to weight the KL divergence terms similarly as in the β-VAE [Hig+16] or in the multimodal VAEs (Sections 3.3 and 4.2). To balance regularization for balanced clusters and randomness of the permutations independently, we reshuffle the terms in Equations (5.70) and (5.71) to come up with the following formulation:

$$\begin{aligned}\mathcal{L}(\theta, \phi; X) = & \sum_{\boldsymbol{x} \in X} \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\log p_\theta(\boldsymbol{x} \mid \boldsymbol{z})\right] \\ & - \sum_{\boldsymbol{x} \in X} \mathbb{E}_{q_\phi(Y|X)} \left[\beta \cdot D_{\mathrm{KL}}[q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p_\theta(\boldsymbol{z} \mid Y)]\right] \\ & - \mathbb{E}_{q_\phi(Y|X)} \left[\gamma \cdot \log \left(\frac{|\Pi_Y| \cdot q_\phi(\boldsymbol{n}; \boldsymbol{\omega}(X))}{p_\theta(\boldsymbol{n}; \boldsymbol{\omega})} \right) \right. \\ & \left. + \delta \cdot \log \left(\frac{\max_\pi q_\phi(\pi; \boldsymbol{s}(X))}{p_\theta(\pi_Y; \boldsymbol{s})} \right) \right] \end{aligned} \tag{A.32}$$

As in vanilla VAEs, we can estimate the reconstruction term in Equation (5.68) with MCMC by applying the reparametrization trick [KW14] to $q(\boldsymbol{z} \mid \boldsymbol{x})$ to sample L samples $\boldsymbol{z}^{(l)} \sim q_\phi(\boldsymbol{z} \mid \boldsymbol{x})$ and compute their reconstruction error to estimate Equation (5.68). Similarly, as described in Section 5.2.2, we apply the reparametrization trick to $p(\boldsymbol{n}; \boldsymbol{s})$ and $p(\pi; \boldsymbol{s})$ to attain L reparametrized samples $Y^{(l)} \sim q_\phi(Y \mid X)$. These can then be used to compute the KL divergences in Equations (5.69) to (5.71) in closed form. Finally, this leads to the following loss during training:

$$\begin{aligned}\mathcal{L}(\theta, \phi; X) = & \sum_{\boldsymbol{x} \in X} \frac{1}{L} \sum_{l=1}^{L} \log p_\theta(\boldsymbol{x} \mid \boldsymbol{z}^{(l)}) \\ & - \sum_{\boldsymbol{x} \in X} \frac{1}{L} \sum_{l=1}^{L} \beta \cdot D_{\mathrm{KL}}[q_\phi(\boldsymbol{z} \mid \boldsymbol{x}) \mid\mid p_\theta(\boldsymbol{z} \mid Y^{(l)})] \qquad \text{(A.33)} \\ & - \frac{1}{L} \sum_{l=1}^{L} \left(\gamma \cdot \log \left(\frac{|\Pi_{Y^{(l)}}| \cdot q(\boldsymbol{n}^{(l)}; \boldsymbol{\omega}(\mathcal{X}))}{p(\boldsymbol{n}^{(l)}; \boldsymbol{\omega})} \right) \right. \\ & \left. + \delta \cdot \log \left(\frac{\max_\pi q_\phi(\pi; \boldsymbol{s}(X))}{p_\theta(\pi^{(l)}; \boldsymbol{s})} \right) \right) \qquad \text{(A.34)} \end{aligned}$$

In our experiments, we set $L = 100$ since the MVHG and PL distributions are not concentrated around their mean very well, such that more Monte

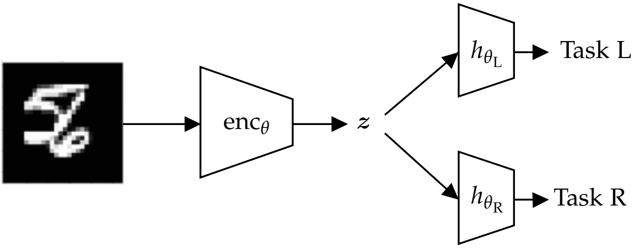


FIGURE A.1: Overview of the multitask learning pipeline of the ULS method.

Carlo samples lead to much better approximations of the expectation terms. Please note that we could increase efficiency by reducing the number of MC samples of the generative model $p_theta(\boldsymbol{x} \mid \boldsymbol{z}^{(l)})$.

In addition to the temperature annealing (see Section 5.3.1.2), we also annealed the weights β, γ, and δ with the same schedule, setting $\beta_{init} = 0.1 \cdot \beta_{final}$, $\beta_{final} = 0.1$, $\gamma_{init} = 0.1 \cdot \gamma_{final}$, $\gamma_{final} = 1$, $\delta_{init} = 0.1 \cdot \delta_{final}$, and $\delta_{final} = 0.001$.

A.4.2 *Multitask Learning*

Figure A.1 show the baseline architecture for the unitary loss scaling (ULS) multitask learning method.

A.4.3 *Weakly Supervised Learning*

The ELBO $\mathcal{L}(\theta, \phi; \boldsymbol{X})$ to be optimized is re-written accordingly as

$$\begin{aligned}\mathcal{L}(\theta, \phi; \boldsymbol{X}) = {} & \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log p_\theta(\boldsymbol{x}_1 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1)\right] + \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log p_\theta(\boldsymbol{x}_2 \mid \boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_2)\right] \\ & - \beta \cdot \mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log \frac{q_\phi(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2 \mid Y, \boldsymbol{X})}{p_\theta(\boldsymbol{z}_{\bar{s}}, \boldsymbol{z}_1, \boldsymbol{z}_2)}\right] \\ & - \gamma \cdot \Bigg(\mathbb{E}_{q_\phi(\boldsymbol{z}, Y \mid \boldsymbol{X})} \left[\log \frac{q_\phi(\boldsymbol{n} \mid \boldsymbol{X}; \boldsymbol{\omega}) \cdot |\Pi_Y|}{p_\theta(\boldsymbol{n}; \boldsymbol{\omega}_p) p_\theta(\pi \mid \boldsymbol{X}; \boldsymbol{s}_p)}\right] \\ & + \log \max_{\pi \in \Pi_Y} q_\phi(\pi \mid \boldsymbol{X}; \boldsymbol{s})\Bigg) \qquad \text{(A.35)}\end{aligned}$$

We did not change any hyperparameters or network details. All experiments were performed using $\beta = 1$, which is the best performing β (according to Locatello et al. [Loc+20]). For DRPM-VAE, we choose $\gamma = 0.25$ for all runs. We use the same parameters for the prior distributions for all dataset versions. To not introduce any a priori knowledge on the number of shared factors, we set all $s_i = 1.0$ and all $\omega_k = 1.0$.