4-2023

Unsupervised Progressive and Continual Learning of Disentangled Representations

Zhiyuan Li
zl7904@rit.edu

Unsupervised Progressive and Continual Learning of Disentangled Representations

by

Zhiyuan Li

A dissertation submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in Computing and Information Sciences

B. Thomas Golisano College of Computing and
Information Sciences

Rochester Institute of Technology
Rochester, New York
April 2023

Unsupervised Progressive and Continual Learning of Disentangled Representations

by

Zhiyuan Li

Committee Approval:

We, the undersigned committee members, certify that we have advised and/or supervised the candidate on the work described in this dissertation. We further certify that we have reviewed the dissertation manuscript and approve it in partial fulfillment of the requirements of the degree of Doctor of Philosophy in Computing and Information Sciences.

Dr. Linwei Wang — Date
Dissertation Advisor

Dr. Nathan Cahill — Date
Dissertation Committee Member

Dr. Rui Li — Date
Dissertation Committee Member

Dr. Travis Desell — Date
Dissertation Committee Member

Dr. Andreas Savakis — Date
Dissertation Defense Chairperson

Certified by:

Dr. Pengcheng Shi — Date
Ph.D. Program Director, Computing and Information Sciences

Unsupervised Progressive and Continual Learning of Disentangled Representations

by

Zhiyuan Li

Submitted to the
B. Thomas Golisano College of Computing and Information Sciences Ph.D.
Program in Computing and Information Sciences
in partial fulfillment of the requirements for the
Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

Unsupervised representation learning is an important task in machine learning that identifies and models underlying explanatory factors hidden in the observed data. In recent years, unsupervised representation learning has been attracting increasing attention for its abilities to improve interpretability, extract useful features without expert annotations, and enhance downstream tasks, which has been successful in many machine learning topics, such as Computer Vision, Natural Language Processing, and Anomaly Detection. Unsupervised representation learning has many desirable abilities, including disentangling generative factors, generalization between different domains, and incremental knowledge accumulation.

However, existing works had faced two critical challenges. First, the unsupervised representation learning models were often designed to learn and disentangle all representations of data at the same time, which obstructed the models from learning representations in a more *progressive* and reasonable way (like from easy to hard), resulting in bad (often blurry) generation quality with the loss of detailed information. Second, when it comes to a more realistic problem setting, *continual* unsupervised representation learning, existing works tended to suffer from catastrophic forgetting, including forgetting learned representations and how to disentangle them. The

continual disentangling problem was very difficult without modeling the relationship between data environments while the forgetting problem was often alleviated by generative-reply.

In this dissertation, we are interested in developing advanced unsupervised representation learning methods based on answering three research questions: (1) how to progressively learn representations such that it can improve the quality and the disentanglement of representations, (2) how to continually learn and accumulate the knowledge of representations from different data environments, and (3) how to continually reuse the existing representations to facilitate learning and disentangling representations given new data environments.

We first established a novel solution for resolving the first research question: progressively learn and disentangle representations and demonstrated the performance in a typical static data environment. And then, for answering the rest two research questions, we extended to study a more challenging and under-investigated setting: unsupervised continual learning and disentangling representations of dynamic data environments, where the proposed model is capable of not only remembering old representations but also reusing them to facilitate learning and disentangling representations in a sequential data stream.

In summary, in this dissertation, we proposed several novel unsupervised representation learning methods and their applications by drawing ideas from different well-studied areas such as auto-encoders, variational inference, mixture distribution, and self-organizing map. We demonstrated the presented methods on various benchmarks, such as dSprites, 3DShape, MNIST, Fashion-MNIST, and CelebA, to provide the quantitative and qualitative evaluation of the learned representations. We concluded by identifying the limitations of the proposed methods and discussing future research directions.

Acknowledgments

A time to weep, and a time to laugh. Yet no man can understand the work that God does from the beginning to the end. God does it, that men should fear before him.

It was a long and memorable journey for my Ph.D. study, and I thank God for making all these things happen. May Your name be glorified, Your kingdom come, Your will be done. May you lead my future path.

I want to thank my lovely family, especially my wife Siyu, for all the blessings she brought to my life. She is always giving me support and comfort when I am frustrated, encouraging me to do what I am supposed to do. I was a "boy" when we fell in love, and I really appreciate her patience and wisdom in helping me learn how to take responsibility as a man so I can be a better man. Practicing loving others is not easy, although we argue sometimes, thank God for helping us always come back and grow together. Our marriage started almost at the same time as my Ph.D. started, and now looking back at those treasured moments always makes me happy and grateful. My girl, Annie, you will always be my girl and daddy's best friend as you said daddy is your best friend haha. Ezra, my boy, you know little right now but daddy already looks forward to you being a man in the future and can't wait to play soccer with you. I am extremely grateful that God makes both of you join our family, making everything so special and wonderful!

I want to thank my advisor, Professor Linwei Wang. I can never emphasize too much how Professor Wang changed my life. She is a perfect example of hard-working talented people, and she made me truly realize that the more knowledgeable a person is, the more humble they are. She has great mentoring skills to find out the strength and weaknesses of students and makes them better based on that. She always carefully designed the tasks assigned to us such that we can gradually keep practicing, get better and better, and become independent in the end. It was pure enjoyment and grace to study and work under her advisement. She taught me how to be a researcher and how to know/evaluate myself appropriately, which is very valuable to me. Once I read a question: if you can restart and live a different life, what are the things and people you want to keep the same? Professor Wang

is absolutely in my answer. I am truly grateful to have her as my advisor in my memorable Ph.D. journey.

I want to thank Professor Shi for his great support and advice for my overall Ph.D. and career direction, his unique sense/style of humor, and the great "adventuring" stories that he shared with us! I want to thank Professor Ashbrook for his advice and leadership in my earlier year of Ph.D. study. I want to thank my committee members, Professor Nathan Cahill, Professor Rui Li, and Professor Travis Desell for their valuable feedback and discussions. I want to thank Professor Yin Pan for her continual heartwarming encouragement and care. I want to thank Lorrie, Min-hong, Siyuan, and Charles for their great support and services for all Ph.D. students, providing sush a friendly and welcoming environment that everyone greatly appreciated.

It is my pleasure to meet my lab friends. Especially, Jwala, the big sister of CBL, helped me so much in math and VAE foundations. And Prashnna, my best research partner, who contributed so much to my research works. I will always be happy to do research and play soccer with you (don't forget Kishan haha). I was so happy to have all of you, Xiajun, Ryan, Nilesh, Maryam, Pradeep, Sandesh, Omar, Mohammed, Ruby, Bipin, Jaideep, Junwen, Wentao, thank you so much for the company and the great time we have together in our big lab room.

Big thanks to my mother and father, uncle and aunt, laolao and laoye. I am very grateful that they raised me with their great love and for their heartwarming support and encouragement for my education. Amazingly, God also made family in Christ for me, especially sister Weiwei, Pastor Feng, Mitchell&Anna, and Yu Gu for their great support in my spiritual life. And so many brothers and sisters made it possible for me to witness the shadow of God's kingdom in Guangzhou and Rochester.

I praise my Lord for all of these during my Ph.D. journey. You are faithful, Your love and grace are beyond.

Author Publications

† indicates that a modified version of this publication is included in this dissertation.

† **Z Li**, X Jiang, R Missel, PK Gyawali, N Kumar, L Wang. Continual Unsupervised Disentangling of Self-Organizing Representations, *International Conference on Learning Representations*, (2023), (Notable-top-25%).

† **Z Li**, JV Murkute, PK Gyawali, L Wang, Progressive Learning and Disentanglement of Hierarchical Representations, *International Conference on Learning Representations*, (2020), (Spotlight-talk-top-15%).

- **Z Li***, PK Gyawali*, S Ghimire, L Wang, Semi-supervised Learning by Disentangling and Self-ensembling over Stochastic Latent Space, *Medical Image Computing and Computer-Assisted Intervention*, (2019).
- X Jiang, R Missel, **Z Li**, L Wang. Sequential Latent Variable Models for Few-Shot High-Dimensional Time-Series Forecasting, *International Conference on Learning Representations*, (2023), (Notable-top-25%).
- M Toloubidokhti, N Kumar, **Z Li**, PK Gyawali, B Zenger, WW Good, RS MacLeod, and L Wang. *Interpretable Modeling and Reduction of Unknown Errors in Mechanistic Operators, Medical Image Computing and Computer-Assisted Intervention*, (2022).
- X Jiang, **Z Li**, R Missel, MS Zaman, B Zenger, WW Good, RS MacLeod, JL Sapp, L Wang, Few-Shot Generation of Personalized Neural Surrogates for Cardiac Simulation via Bayesian Meta-learning, *Medical Image Computing and Computer-Assisted Intervention*, (2022).
- X Jiang, R Missel, M Toloubidokhti, **Z Li**, O Gharbia, JL Sapp, L Wang, Label-Free Physics-Informed Image Sequence Reconstruction with Disentangled Spatial-Temporal Modeling, *Medical Image Computing and Computer-Assisted Intervention*, (2021).
- R Missel, PK Gyawali, JV Murkute, **Z Li**, S Zhou, A AbdelWahab, J Davis, J Warren, J L Sapp, L Wang, A Hybrid Machine Learning Approach to Localizing the Origin of Ventricular Tachycardia Using 12-lead Electrocardiograms, *Computers in Biology and Medicine*, (2020).
- X Jiang, S Ghimire, J Dhamala, **Z Li**, PK Gyawali, L Wang, Learning Geometry-Dependent and Physics-Based Inverse Image Reconstruction, *Medical Image Computing and Computer-Assisted Intervention*, (2020).

- PK Gyawali, S Ghimire, P Bajracharya, **Z Li**, L Wang, Semi-supervised Medical Image Classification with Global Latent Mixing, *Medical Image Computing and Computer-Assisted Intervention*, (2020).
- R Ballagas, J Wei, M Vankipuram, **Z Li**, K Spies, H Horii, Exploring Pervasive Making Using Generative Modeling and Speech Input, *IEEE Pervasive Computing*, (2019).
- PK Gyawali, **Z Li**, C Knight, S Ghimire, BM Horacek, J Sapp, L Wang, Improving Disentangled Representation Learning with the Beta Bernoulli Process, *IEEE International Conference on Data Mining*, (2019).
- AK Yung, **Z Li**, D Ashbrook, Printy3D: In-situ Tangible Three-dimensional Design for Augmented Fabrication, *ACM Conference on Interaction Design and Children*, (2018).
- C Tejada, O Fujimoto, **Z Li**, D Ashbrook. Blowhole: Blowing-Activated Tags for Interactive 3D-Printed Models, *Graphics Interface Conference*, (2018).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

Learning useful representations of data with little or no supervision plays a vital role in deploying artificial intelligence for massive amounts of information in the real world. This makes unsupervised representation learning an important task in machine learning that has been attracting increasing attention for its capabilities to improve interpretability, extract useful features without expert annotations, and enhance downstream predictive tasks [4,21,53]. There are several iconic abilities for intelligent creatures to discover and utilize real-world representation. First, intelligent creatures tend to keep discovering and disentangling the generative factors of the real world. In the ancient world, people used to anticipate rain through general natural phenomena like the shape of clouds. However, the shape of the cloud actually contains and is entangled with many raining-related factors like air pressure, temperature, and humidity that humans discover later and greatly improve the accuracy of prediction by precisely identifying and treating different factors. Such disentangling ability is desired for it offers better interpretability of the model and tackles complicated problems by decomposing it into different sub-factors. Second, intelligent creatures won't forget what they learned and can continually learn new representations. It would be very strange that a skilled vehicle technician forgets his or her knowledge of cars after starting to learn web design on weekends. Even kids remember the animals they learned to recognize weeks ago from

books after watching new animals in the zoo later. However, machines do suffer from naively continually learning from stream data and this problem is often referred to as catastrophic forgetting [1, 44]. Third, intelligent creatures especially human beings are extremely good at building new concepts upon existing knowledge and reusing the existing ones to facilitate learning new representations. For example, once kids recognize dogs and cats and form a model that there is one head at the top and four legs at the bottom, they can quickly adopt this knowledge when they first see horses. The existing knowledge of heads and legs can be reused and help them focus on learning new features like bigger body sizes and different skin textures. However, for current machine learning methods, this is a very difficult problem that requires modeling the relationship among all data environments.

The above abilities are desired for machine learning, yet the gap at the moment leaves many difficult challenges and research opportunities for representation learning researchers. In this dissertation, we mainly focused on tackling the critical challenges of unsupervised representation learning in terms of two aspects: the models' ability to learn representations *progressively* and *continually*. First, for *progressive* representation learning, most existing unsupervised representation learning models lack this ability because they were designed to learn and disentangle all representations of data at the same time, which obstructed the models from learning representations from easy to hard. The inherent conflict between the bottom-up inference process (encoder) and the top-down generation process (decoder) additionally prevents deep models from fully utilizing the depth of neural networks. These often resulted in bad (often blurry) generation quality with the loss of detailed information. We will introduce and describe this in chapter 3. Second, when it comes to *continual* unsupervised representation learning, which is a more realistic and less-explored problem setting, existing works tended to suffer from catastrophic forgetting, including forgetting learned representations and how to disentangle them. The latter problem was very difficult without modeling the relationship between data environments while the former problem was often alleviated by generative-reply. We will provide in-depth discussions and solutions for this in chapter 4.

In summary, we approach the above challenges and research opportunities of unsupervised representation learning by asking three main research questions: (1) how to progressively learn representations such that it can improve the quality and the disentanglement of representations, (2) how to continually learn and accumulate the knowledge of representations from different data environments, and (3) how to continually reuse the existing representations to facilitate learning and disentangling representations given new data environments. To

answer these research questions, we investigated and divided unsupervised representation learning into two complementary scenarios: *static* data environment and *dynamic* data environment in this dissertation. In the first scenario, the data distribution is stationary, and training samples are drawn i.i.d. (often being done by shuffling training data) and then fed to the unsupervised representation learning model, which is the most typical setting for learning and disentangling representations of existing works. In the second scenario, the data distribution is non-stationary and is often fed to the model sequentially where old data is overwritten by new ones in data steam, making the old data no longer available at a certain time point.

We first present a novel solution for resolving the first research question: progressively learning and disentangling representations. And we design and demonstrate its performance in a typical *static* data environment, as we will describe in chapter 3. To address the second and third research questions, we then extend the idea of *progressive* learning to *continual* learning and investigated a more realistic and challenging problem setting: continual unsupervised representation learning in *dynamic* data environments, as will be explained in chapter 4. Specifically, we first establish a progressive representation learning network that gradually increases its capacity to learn representations from high-level abstraction to low-level abstraction to improve the quality and disentanglement of the learned representations given static data environments. Then, to solve the second research question -how to continually learn and accumulate the knowledge of representations - we turn to model the latent space with a topologically-connected mixture of distributions in the form of a Bayesian formulation of self-organizing map [28, 58], as a foundation model for accumulating relational structure of data and expanding the progressive representation learning to continual representation learning. Finally, for the third research question - how to continually reuse and disentangle representations - we propose novel methods for identifying active semantic factors of each data environment and computing the shared dimensions between different data environments, along with a continual disentangling framework that aims to guide and optimize the re-using of old representations and facilitate learning and disentangling new representations.

1.2 Contribution

The goal of this research is to develop advanced unsupervised representation learning methods to fill the gap between existing works and the current challenges and to improve the model's

progressive and continual learning ability. To this end, we presented the contributions of our research in progressive and continual unsupervised representation learning:

- Progressive learning and disentangling representation
  - We proposed a novel deep neural network for progressively learning and disentangling hierarchical representation from high-level abstraction to low-level abstraction. This allows the model to capture the most important and dominant features first and then progressively learn the rest representations to refine the remaining details [34].
  - We proposed the implementation strategy for progressive learning that greatly improved training stability [34].

- Continual learning and disentangling representation
  - We proposed a principled variational inference framework to learn and optimize a topologically-connected mixture model to continually reuse, expand, and disentangle representations [33].
  - We proposed a self-organized latent space with a Bayesian-SOM for continually accumulating the relational structure of data [33].
  - We proposed to distill the knowledge of active semantic factors across data environments by modeling each component of the SOM mixture model as a spike-and-slab distribution, for optimizing the reuse of shared representations across different data environments and disentangling new representations [33].

- Disentanglement metrics and evaluations
  - We proposed a new disentanglement metric *MIG-sup* to supplement existing metrics by specifically measuring how multiple generative factors are entangled in a single latent variable [34].
  - We proposed a mutual-information-based disentanglement metric for evaluating the performance of continual disentanglement, and reported the first comprehensive continual disentanglement experiments, to the best of our knowledge [33].

Chapter 2

Principles

This chapter serves as an introduction to the principles and the foundation literature of representation learning, disentangled representation, self-organized map, and continual learning, which are helpful to understand the remainder of the dissertation and the description of related works in later chapters.

2.1 Representation learning

Representation learning is an important task for machine learning. The choice of data representation (or features) plays an important role in the performance of machine learning methods. However, traditional feature engineering is labor-intensive and often relays on experts' prior knowledge [4]. Therefore, to expand the applicability of machine learning, representation learning has been attracting increasing attention in recent years for its ability to extract and organize the information from the observed data, and has been shown successfully in many machine learning fields [4]. Some general assumptions to guide representation learning include:

- data is generated by multiple explanatory factors, and there exists a hierarchical organization of them,

- certain factors are shared across tasks,
- representations are sparse, i.e., for any given observation, only a small number of factors are active and relevant,
- and representations (especially categorical representations) naturally form distinguishable clusters etc.

2.2 Disentangled representations

Learning disentangled representation is an important topic in representation learning for its desired interpretability of deep learning models and robustness analysis of complicated problems by investigating their sub-factors. Existing works mainly tackle this by promoting independence among the learned latent factors in VAE [9, 21, 25]. In [8, 21], a hyperparameter β is introduced for the KL-divergence term of VAE. Then a heavy posterior overlap caused by the large value of β can force a smooth and disentangled latent space. In [9], a decomposition of the KL term in VAE is proposed and the authors augured that it is the total-correlation term in it that plays an important role in disentanglement. A similar emphasis on using total correlation to promote disentanglement had been discussed in [25].

Although the one-to-one relationship between the generative factors and the latent variables is one intuitive understanding of disentanglement, it is not easy to measure disentanglement and different metrics are focused on different perspectives of disentanglement and sometimes hard to align with each other [37, 59], which leaves open questions that need further research.

2.3 Continual learning

Continual Learning (also often referred to as Incremental Learning or Life-long Learning) is a machine learning concept that learns a model for a number of tasks sequentially without forgetting knowledge obtained from the previous tasks, where the data in the old tasks are not available during training new ones.

For unsupervised continual learning, the term “unsupervised” can be referred to the task or the task boundary, or both. If it refers to the task, it means the task is unsupervised, e.g.,

unsupervised representation learning, clustering, and in this case the unsupervised tasks are learned continually. If it refers to the task boundary, it means there is no supervision on the task identity and/or when the data distributions change, and the model needs to infer that when necessary. In this dissertation, for clarity, "unsupervised" is referred to as in the former case, and we use the term "data environments" to describe the different data distributions during continual learning. In summary, we are interested in and focus on conducting unsupervised representation learning and disentangling for sequential-arrived data streams.

2.4 Variational auto-encoders (VAE)

VAE [26,47] are deep generative models that are very popular and successful for representation learning. It describe the generation of data $\mathbf{x}$ as $p_\theta(\mathbf{x}|\mathbf{z})$, involving a set of latent variables $\mathbf{z}$ that follows a prior distribution $p(\mathbf{z})$ (often described by an isotropic Gaussian $\mathcal{N}(0, \mathbf{I})$). A proposal distribution $q_\phi(\mathbf{z}|\mathbf{x})$ is introduced as the variational approximation of $p(\mathbf{z}|\mathbf{x})$, and optimized together with $p_\theta(\mathbf{x}|\mathbf{z})$ to maximize the evidence lower bound (ELBO) of the marginal data likelihood:

$$\log p(\mathbf{x}) \geq \mathcal{L} = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})), \tag{2.1}$$

where the first term is often interpreted as minimizing a reconstruction error and the second term minimizes the KL-divergence between the variational posterior and its prior.

2.5 Self-organizing map

A typical SOM [28] consists of a set of nodes on a two-dimensional gird, where a weight vector is learned for each node such as to minimize the distance between each input data and its distance to the closest weight vector (*i.e.*, Best Matching Unit (BMU)). In a Bayesian formulation of SOM [57], each node becomes one component of a mixture with the associated density parameters. The BMU to a given data point is then determined by a component's posterior probability given the data. A sequential Expectation-Maximization (EM) like algorithm was developed to incrementally update the SOM mixture to maximize the likelihood of sequentially-arrived data [57]. Once trained, SOM nodes provide prototypes of the input

data, where similar data will be mapped to neighboring prototypes. These allow Bayesian SOM to formally model the joint distribution of input data and SOM prototypes. a classic unsupervised neural network for learning a topologically interpretable relationship of data, in the form of learned prototypes that each represents a summary of similar data and is topologically connected to neighboring similar prototypes with distance measures. The joint-probability density of Bayesian-SOM is:

$$p(\mathbf{z}|\mathbf{\Theta}) = \sum_{k=1}^{K} p(\mathbf{z}|v_k, \theta_k)p(v_k), \tag{2.2}$$

where $p(\mathbf{z}|v_k, \theta_k) = (2\pi)^{-d/2}|\Sigma_k|^{-1/2}\exp\{-0.5(\mathbf{z} - \boldsymbol{\mu}_k)^T\Sigma_k^{-1}(\mathbf{z} - \boldsymbol{\mu}_k)\}$ is the component conditional density, $\theta_k = \{\boldsymbol{\mu}_k, \Sigma_k\}$ is the component parameters with $\Sigma_k = diag(\boldsymbol{\sigma}_k^2)$, $p(v_k) = w_k$ is the mixing prior, for the k-th component. $\Theta = \{\theta_1, \theta_2, ..., \theta_k\}$.

Chapter 3

Progressive learning and disentangling of representation

In this chapter, we describe our contributions to answering the first research question: how to progressively learn representations such that it can improve the quality and the disentanglement of representations. It includes a novel VAE-based neural network framework and the corresponding progressive training strategy.

3.1 Introduction

In a typical unsupervised representation learning setting, i.e., stationary data environment and i.i.d. training samples, most existing works are designed to learn and disentangle all representations at the same time [9, 19, 21, 25]. However, there exist challenges for extracting and disentangling generative factors all at once, especially at different abstraction levels [34]. To solve this, inspired by the human cognition system, [14] suggested the importance of "starting small" in two aspects of the learning process of neural networks: *incremental input* in which a network is trained with data and tasks of increasing complexity, and *incremental memory* in which the network capacity undergoes developmental changes given fixed external data and tasks — both pointing to an incremental learning strategy for simplifying a complex

final task. Indeed, the former concept of *incremental input* has underpinned the success of curriculum learning [5]. In the context of deep generative models (DGMs), various stacked versions of generative adversarial networks (GANs) have been proposed to decompose the final task of high-resolution image generation into progressive sub-tasks of generating small to large images [11, 61]. The latter aspect of "starting small" with incremental growth of network capacity is less explored, although recent works have demonstrated the advantage of progressively growing the depth of GANs for generating high-resolution images [24, 54]. These works, so far, have focused on progressive learning as a strategy to improve image generation.

Meanwhile, DGMs have not been able to fully utilize the depth of neural networks like their supervised counterparts, for which a fundamental cause lies in the inherent conflict between the bottom-up inference and top-down generation process [32, 62]: while the bottom-up abstraction is able to extract high-level representations helpful for discriminative tasks, the goal of generation requires the preservation of all generative factors that are likely at different abstraction levels. This issue was addressed in recent works by allowing VAEs to generate from details added at different depths of the network, using either memory modules between top-down generation layers [32], or hierarchical latent representations extracted at different depths via a variational ladder autoencoder VLAE, [62].

We are motivated to investigate the possibility to use progressive learning strategies to improve learning and disentangling of hierarchical representations. At a high level, the idea of progressively or sequentially learning latent representations has been previously considered in VAE. In [18], the network learned to sequentially refine generated images through recurrent networks. In [31], a teacher-student training strategy was used to progressively increase the number of latent dimensions in VAE to improve the generation of images while preserving the disentangling ability of the teacher model. However, these works primarily focus on progressively growing the capacity of VAE to generate, rather than to extract and disentangle hierarchical representations.

In comparison, in this chapter, we focus on 1) progressively growing the capacity of the network to extract hierarchical representations, and 2) these hierarchical representations are extracted and used in generation from different abstraction levels. We present a simple progressive training strategy that grows the hierarchical latent representations from different depths of the inference and generation model, learning from high- to low-levels of abstractions as the capacity of the model architecture grows. Because it can be viewed as a progressive

strategy to train the VLAE presented in [62], we term the presented model *pro*-VLAE. We quantitatively demonstrate the ability of *pro*-VLAE to improve disentanglement on two benchmark data sets using three disentanglement metrics, including a new metric we proposed to complement the metric of mutual information gap (MIG) previously presented in [9]. These quantitative studies include comprehensive comparisons to β-VAE [21], VLAE [62], and the teacher-student strategy as presented in [31] at different values of the hyperparameter β. We further present both qualitative and quantitative evidence that *pro*-VLAE is able to first learn the most abstract representations and then progressively disentangle existing factors or learn new factors at lower levels of abstraction, improving disentangling of hierarhical representations in the process.

3.2 Related works

A hierarchy of feature maps can be naturally formed in stacked discriminative models ([60]). Similarly, in DGM, many works have proposed stacked-VAEs as a common way to learn a hierarchy of latent variables and thereby improve image generation ($[2, 27, 50]$). However, this stacked hierarchy is not only difficult to train as the depths increases ($[2, 50]$), but also has an unclear benefit for learning either hierarchical or disentangled representations: as shown in [62], when fully optimized, it is equivalent to a model with a single layer of latent variables. Alternatively, instead of a hierarchy of latent variables, independent hierarchical representations at different abstraction levels can be extracted and used in generation from different depths of the network ([47,62]). A similar idea was presented in [32] to generate lost details from memory and attention modules at different depths of the top-down generation process. The presented work aligns with existing works ($[47, 62]$) in learning independent hierarchical representation from different levels of abstraction, and we look to facilitate this learning by progressively learning the representations from high- to low-levels.

Progressive learning has been successful for high-quality image generation, mostly in the setting of GANs. Following the seminar work of [14], these progressive strategies can be loosely grouped into two categories. Mostly, in line with *incremental input,* several works have proposed to divide the final task of image generation into progressive tasks of generating low-resolution to high-resolution images with multi-scale supervision ([11,61]). Alternatively, in line with *incremental memory,* a small number of works have demonstrated the ability to simply grow the architecture of GANs from a shallow network with limited capacity for

generating low-resolution images, to a deep network capable of generating super-resolution images ([24, 54]). This approach was also shown to be time-efficient since the early-stage small networks require less time to converge comparing to training a full network from the beginning. This latter group of works provided compelling evidence for the benefit of progressively growing the capacity of a network to generate images, although its extension for growing the capacity of a network to learn hierarchical representations has not been explored.

Limited work has considered incremental learning of representations in VAE. In [18], recurrent networks with attention mechanisms were used to sequentially refines the details in generated images. It however focused on the generation performance of VAE without considering the learned representations. In [31], a teacher-student strategy was used to progressively grow the dimension of the latent representations in VAE. Its fundamental motivation was that, given a teacher model that has learned to effectively disentangle major factors of variations, progressively learning additional nuisance variables will improve generation without compromising the disentangling ability of the teacher – the latter accomplished via a newly-proposed Jacobian supervision. The capacity of this model to grow, thus, is by design limited to the extraction of nuisance variables. In comparison, we are interested in a more significant growth of the VAE capacity to progressively learn and improve disentangling of important factors of variations which, as we will later demonstrate, is not what the model in [31] is intended for. In addition, neither of these works considered learning different levels of abstractions at different depths of the network, and the presented *pro*-VLAE provides a simpler training strategy to achieve progressive representation learning.

Learning disentangled representation is a primary motivation of our work, and an important topic in VAE. Existing works mainly tackle this by promoting the independence among the learned latent factors in VAE ([9, 21, 25]). The presented progressive learning strategy provides a novel approach to improve disentangling that is different to these existing methods and a possibility to augment them in the future.

3.3 Methods

3.3.1 Model: VAE with hierarchical representations

We assume a generative model $p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ for observed $\mathbf{x}$ and its latent variable $\mathbf{z}$. To learn hierarchical representations of $\mathbf{x}$, we decompose $\mathbf{z}$ into $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_L\}$ with $\mathbf{z}_l (l = 1, 2, 3, ..., L)$ from different abstraction levels that are loosely guided by the depth of neural network as in [62]. We define the hierarchical generative model p_θ as:

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{x}|\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_L) \prod_{l=1}^{L} p(\mathbf{z}_l). \tag{3.1}$$

Note that there is no hierarchical dependence among the latent variables as in common hierarchical latent variable models. Rather, similar to that in [47] and [62], $\mathbf{z}_l$'s are independent and each represents generative factors at an abstraction level not captured in other levels. We then define an inference model q_ϕ to approximate the posterior as:

$$q(\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_L|\mathbf{x}) = \prod_{l=1}^{L} q(\mathbf{z}_l|\mathbf{h}_l(\mathbf{x})), \tag{3.2}$$

where $\mathbf{h}_l(\mathbf{x})$ represents a particular level of bottom-up abstraction of $\mathbf{x}$. We parameterize p_θ and q_ϕ with an encoding-decoding structure and, as in [62], we approximate the abstraction level with the network depth. The full model is illustrated in Fig. 3.1(c), with a final goal to maximize a modified evidence lower bound (ELBO) of the marginal likelihood of data $\mathbf{x}$:

$$\log p(\mathbf{x}) \geq \mathcal{L} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z})] - \beta KL(q(\mathbf{z}|\mathbf{x})||p(\mathbf{z})), \tag{3.3}$$

where KL denotes the Kullback-–Leibler divergence, prior $p(\mathbf{z})$ is set to isotropic Gaussian $\mathcal{N}(0, \mathbf{I})$ according to standard practice, and β is a hyperparameter introduced in [21] to promote disentangling, defaulting to the standrd ELBO objective when $\beta = 1$.

3.3.2 Progressive learning of hierarchical representation

We present a progressive learning strategy, as illustrated in Fig. 3.1, to achieve the final goal in equation (3.3) by learning the latent variables $\mathbf{z}_l$ progressively from the highest ($l = L$) to the lowest $l = 1$) level of abstractions. We start by learning the most abstraction

representations at layer L as show in Fig. 3.1(a). In this case, our model degenerates to a vanilla VAE with latent variables $\mathbf{z}_L$ at the deepest layer. We keep the dimension of $\mathbf{z}_L$ small to start small in terms of the capacity to learn latent representations, where we define the inference model at progressive step $s = 0$ as:

$$\mathbf{z}_L \sim \mathcal{N}(\mu_L(\mathbf{h}_L), \sigma_L(\mathbf{h}_L)),\ \mathbf{h}_l = f_l^e(\mathbf{h}_{l-1}),\ \text{for } l = 1, 2, ..., L, \text{and } \mathbf{h}_0 \equiv \mathbf{x}, \tag{3.4}$$

and the generative model as:

$$\mathbf{g}_L = f_L^d(\mathbf{z}_L),\ \mathbf{g}_l = f_l^d(\mathbf{g}_{l+1}),\ \mathbf{x} = D(\mathbf{x}; f_0^d(\mathbf{g}_0)), \tag{3.5}$$

where f_l^e, μ_L, and σ_L are parts of the encoder architecture, f_l^d are parts of the decoder architecture, and D is the distribution of $\mathbf{x}$ parametrized by $f_0^d(\mathbf{g}_0)$, which can be either Bernoulli or Gaussian depending on the data. Next, as shown in Fig. 3.1, we progressively grow the model to learn $z_{L-1}, ..., z_2, z_1$ from high to low abstraction levels. At each progressive step $s = 1, 2, ..., L-1$, we move down one abstraction level, and grow the inference model by introducing new latent code:

$$\mathbf{z}_l \sim \mathcal{N}(\mu_l(\mathbf{h}_l), \sigma_l(\mathbf{h}_l)),\ l = L - s. \tag{3.6}$$

Simultaneously, we grow the decoder such that it can generate with the new latent code as:

$$\mathbf{g}_l = f_l^d([m_l(\mathbf{z}_l); \mathbf{g}_{l+1}]),\ l = L - s, \tag{3.7}$$

where m_l includes transposed convolution layers outputting a feature map in the same shape as $\mathbf{g}_{l+1}$, and $[\cdot;\cdot]$ denotes a concatenation operation. The training objective at progressive step s is then:

$$\mathcal{L}_{pro} = \mathbb{E}_{q(\mathbf{z}_L, \mathbf{z}_{L-1}, ..., \mathbf{z}_{L-s}|\mathbf{x})}[\log p(\mathbf{x}|\mathbf{z}_L, \mathbf{z}_{L-1}, ..., \mathbf{z}_{L-s})] - \beta \sum_{L-s}^{L} KL(q(\mathbf{z}_l|\mathbf{x})||p(\mathbf{z}_l)), \tag{3.8}$$

By replacing the full objective in equation (3.3) with a sequence of the objectives in equation (3.8) as the training progresses, we incrementally learn to *extract* and *generate with* hierarchical latent representations $\mathbf{z}_l$'s from high to low levels of abstractions. Once trained, the full model as shown in Fig. 3.1(c) will be used for inference and generation, and progressive processes are no loner needed.

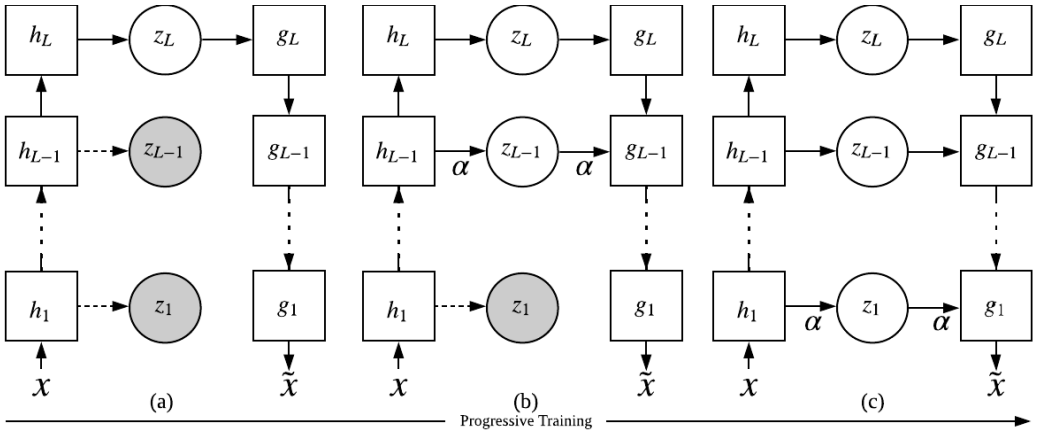

Figure 3.1: Progressive learning of hierarchical representations. White blocks and solid lines are VAE models at the current progression. α is a fade-in coefficient for blending in the new network component. Gray circles and dash line represents (optional) constraining of the future latent variables.

3.3.3 Implementation strategies

Two important strategies are utilized to implement the proposed progressive representation learning. First, directly adding new components to a trained network often introduce a sudden shock to the gradient: in VAEs, this often leads to the explosion of the variance in the latent distributions. To avoid this shock, we adopt the popular method of "fade-in" [24] to smoothly blend the new and existing network components. In specific, we introduce a "fade-in" coefficient α to equations (3.6) and (3.7) when growing new components in the encoder and the decoder:

$$\mathbf{z}_l \sim \mathcal{N}(\mu_l(\alpha \mathbf{h}_l),\ \sigma_l(\alpha \mathbf{h}_l)), \mathbf{g}_l = f_l^d([\alpha m_l(\mathbf{z}_l); \mathbf{g}_{l+1}]), \tag{3.9}$$

where α increases from 0 to 1 within a certain number of iterations (5000 in our experiments) since the addition of the new network components μ_l,σ_l, and m_l.

Second, we further stabilize the training by weakly constraining the distribution of $\mathbf{z}_l$'s before they are added to the network. This can be achieved by a applying a KL penalty, modulated by a small coefficient γ, to all latent variables that have not been used in the generation at

progressive step s:

$$\mathcal{L}_{pre-trained} = \gamma \sum_{l=1}^{L-s-1} \big[- KL(q(\mathbf{z}_l|\mathbf{x})||p(\mathbf{z}_l)) \big], \tag{3.10}$$

where γ is set to 0.5 in our experiments. The final training objective at step s then becomes:

$$\mathcal{L} = \mathcal{L}_{pro} + \mathcal{L}_{pre-trained} \tag{3.11}$$

Note that the latent variables at the hierarchy lower than $L - s$ are neither meaningfully inferred nor used in generation at progressive step s, and $\mathcal{L}_{pre-trained}$ merely intends to regularize the distribution of these latent variables before they are added to the network. In the experiments below, we use both "fade-in" and $\mathcal{L}_{pre-trained}$ when implementing the progressive training strategy.

3.3.4 Disentanglement metric

Various quantitative metrics for measuring disentanglement have been proposed [9, 21, 25]. For instance, the recently proposed MIG metrics [9] measures the gap of mutual information between the top two latent dimensions that have the highest mutual information with a given generative factor. A low MIG score, therefore, suggests an undesired outcome that the same factor is split into multiple dimensions. However, if different generative factors are entangled into the same latent dimension, the MIG score will not be affected.

Therefore, we propose a new disentanglement metric to supplement MIG by recognizing the entanglement of multiple generative factors into the same latent dimension. We define MIG-sup as:

$$\frac{1}{J} \sum_{1}^{J} \big(I_{norm}(\mathbf{z}_j; v_{k^{(j)}}) - \max_{k \neq k^{(j)}} I_{norm}(\mathbf{z}_j; v_k) \big), \tag{3.12}$$

where $\mathbf{z}$ is the latent variables and v is the ground truth factors, $k^{(j)} = \text{argmax}_k I_{norm}(\mathbf{z}_j; v_k)$, J is the number of meaningful latent dimensions, and $I_{norm}(\mathbf{z}_j; v_k)$ is normalized mutual information $I(\mathbf{z}_j; v_k)/H(v_k)$. Considering MIG and MIG-sup together will provide a more complete measure of disentanglement, accounting for both the splitting of one factor into multiple dimensions and the encoding of multiple factors into the same dimension. In an ideal disentanglement, both MIG and MIG-sup should be 1, recognizing a one-to-one relationship between a generative factor and a latent dimension. This would have a similar effect to the

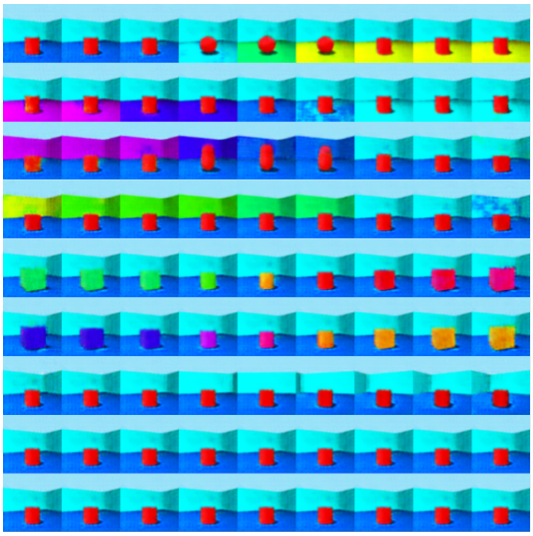

Figure 3.2: An example of one factor being encoded in multiple dimensions. Each row is a traverse for one dimension (dimension order adjusted for better visualization). Notice that both the 1st and the 2nd rows are encoding floor-color, both the 3rd and 4th rows are encoding wall-color, and both the 5th and 6th rows are encoding object color. Therefore, the MIG is very low since it penalizes splitting one factor into multiple dimensions. On the other hand, the MIG-sup and factor-metric are not too bad since one dimension mainly encodes one factor, even though there is some entanglement of color-vs-shape and color-vs-scale.

metric that was proposed in [12], although MIG-based metrics do not rely on training extra classifiers or regressors and are unbiased for hyperparameter settings. The factor metric [25] also has similar properties with MIG-sup, although MIG-sup is stricter on penalizing any amount of other minor factors in the same dimension.

As shown in Fig. 3.2 and Fig. 3.3, we presented examples of how different disentanglement metrics (MIG, MIG-sup, and the factor metric) correspond to the visual traversing results to help the reader understand how different metric evaluates disentanglement.

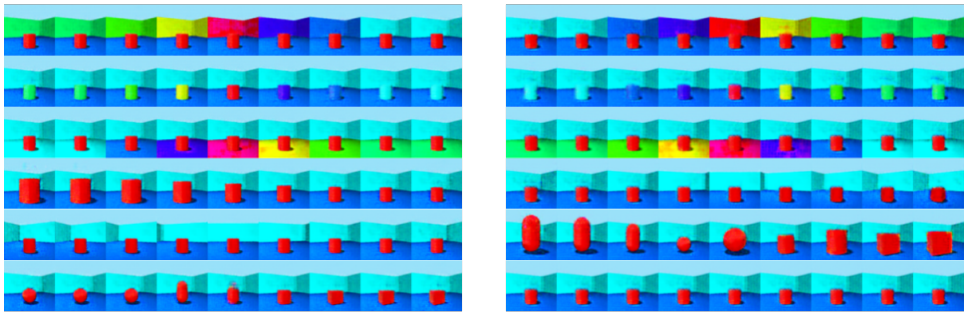

Figure 3.3: An example of one dimension containing multiple factors. Each row is a traverse for one dimension (dimension order adjusted for better visualization). Notice that both models achieve high and similar MIG because all 6 factors are encoded and no splitting to multiple dimensions. However, the right-hand side model has much lower MIG-sup and factor-metric than the left-hand side model. Because both scale and shape are encoded in the 5th row, while the 6th row has no factor. Both MIG-sup and factor-metric penalize encoding multiple factors in one dimension. Besides, our MIG-sup is lower and drops more than factor-metric because MIG-sup is stricter in this case.

3.4 Experiment

We tested the presented *pro*-VLAE on four benchmark data sets: dSprites ([40]), 3DShapes ([7]), MNIST ([30]), and CelebA ([35]), where the first two include ground-truth generative factors that allow us to carry out comprehensive quantitative comparisons of disentangling metrics with existing models. In the following, we first quantitatively compare the disentangling ability of *pro*-VLAE in comparison to three existing models using three disentanglement metrics. We then analyze *pro*-VLAE from the aspects of how it learns progressively, its ability to disentangle, and its ability to learn abstractions at different levels.

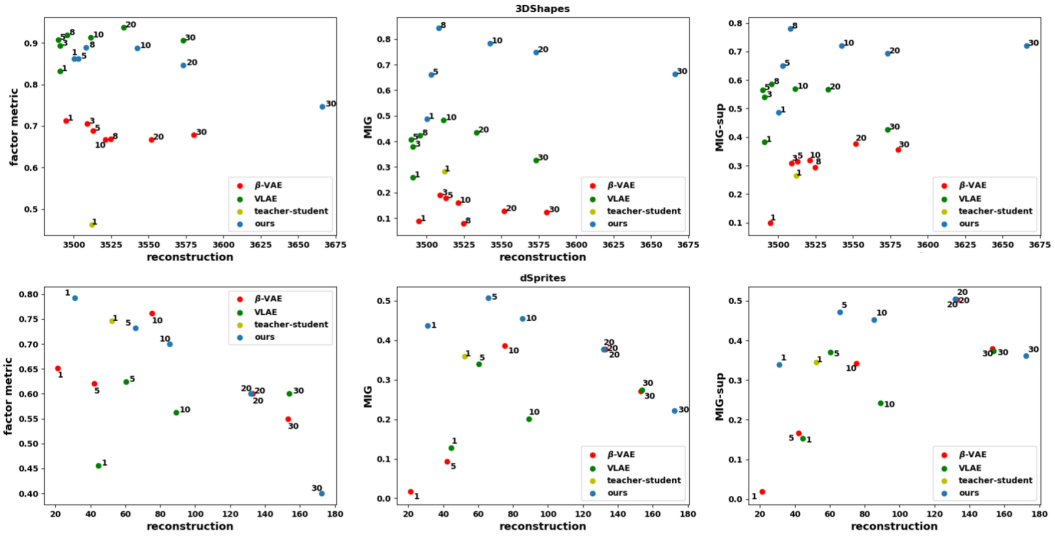

Figure 3.4: Quantitative comparison of disentanglement metrics. Each point is annotated by the β value and averaged over top three best random seeds for the given β on the give model. **Left to right:** reconstruction errors *vs.* disentanglement metrics of factor, MIG, and MIG-sup, a higher value indicating a better disentanglement in each metric.

3.4.1 Comparisons in quantitative disentanglement metrics

For quantitative comparisons, we considered the factor metric in [25], the MIG in [9], and the MIG-sup presented in this work. We compared *pro*-VLAE (changing β) with beta-VAE ([21]), VLAE ([62]) as a hierarchical baseline without progressive training, and the teacher-student model ([31]) as the most related progressive VAE without hierarchical representations. All models were considered at different values of β except the teacher-student model: the comparison of β-VAE, VLAE, and the presented *pro*-VLAE thus also provides an ablation study on the effect of learning hierarchical representations and doing so in a progressive manner.

For fair comparisons, we strictly required all models to have the same number of latent variables and the same number of training iterations. For instance, if a hierarchical model has three layers that each has three latent dimensions, a non-hierarchical model will have

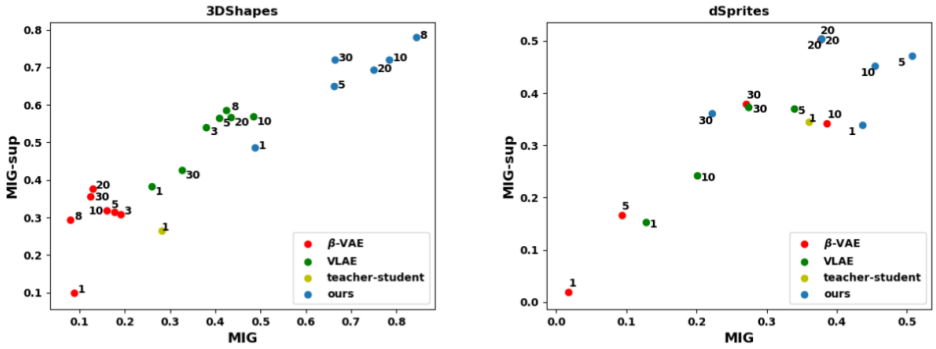

Figure 3.5: MIG *vs.* MIG-sup following a similar presentation in Fig. 3.4. A better disentanglement should have higher MIG and higher MIG-sup, locating at the top-right quadrant of the plot.

nine latent dimensions; if a progressive method has three progressive steps with 15 epochs of training each, a non-progressive method will be trained for 45 epochs. Three to five experiments were conducted for each model at each β value, and the average of the top three is used for reporting the quantitative results in Fig. 3.4.

As shown, for MIG and MIG-sup, VLAE generally outperformed β-VAE at most β values, while *pro*-VLAE showed a clear margin of improvement over both methods. With the factor metric, *pro*-VLAE was still among the top performers, although with a smaller margin and a larger overlap with VLAE on 3DShapes, and with β-VAE ($\beta = 10$) on dSprites. The teacher-student strategy with Jacobian supervision in general had a low to moderate disentangling score, especially on 3DShapes. This is consistent with the original motivation of the method for progressively learning nuisance variables after the teacher learns to disentangle effectively, rather than progressively disentangling hierarchical factors of variations as intended by *pro*-VLAE. Note that *pro*-VLAE in general performed better with a smaller value of β ($\beta < 20$), suggesting that progressive learning already had an effect of promoting disentangling and a high value of β may over-promote disentangling at the expense of reconstruction quality.

Fig. 3.5 shows MIG *vs.* MIG-sup scores among the tested models. As shown, results from *pro*-VLAE were well separated from the other three models at the right top quadrant of the plots, obtaining simultaneously high MIG and MIG-sup scores as a clear evidence for improved disentangling ability.

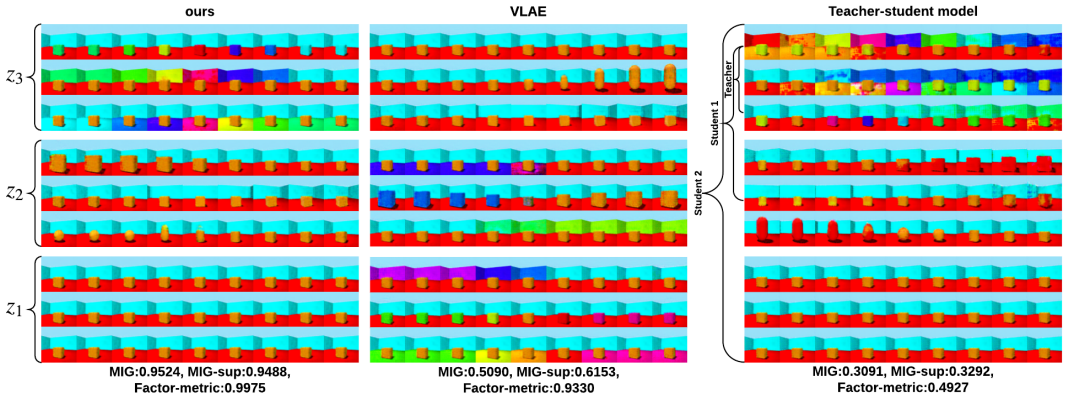

Figure 3.6: Traversing each latent dimension in *pro*-VLAE ($\beta = 8$), VLAE ($\beta = 10$), and teacher-student model. The hierarchy of the latent variables is noted by brackets on the side.

Fig. 3.6 provides images generated by traversing each latent dimension using the best *pro*-VLAE ($\beta = 8$), the best VLAE ($\beta = 10$), and the teacher-student model on 3DShapes data. As shown, *pro*-VLAE learned to disentangle the object, wall, and floor color in the deepest layer; the following hierarchy of representations then disentangled objective scale, orientation, and shape, while the lowest-level of abstractions ran out of meaningful generative factors to learn. In comparison, the VLAE distributed six generative factors over the nine latent dimensions, where color was split across the hierarchy and sometimes entangled with the object scale (in $\mathbf{z}_2$). The teacher-student model was much less disentangled, which we will delve into further in the following section.

3.4.2 Information flow during progressive learning

To further understand what happened during progressive learning, we use mutual information $I(\mathbf{x}, \mathbf{z}_l)$ as a surrogate to track the amount of information learned in each hierarchy of latent variables $\mathbf{z}_l$ during the progressive learning. We adopted the approach in [9] to empirically estimate the mutual information by stratified sampling.

Fig. 3.7 shows an example from 3DShapes. At progressive step 0, *pro*-VAE was only learning

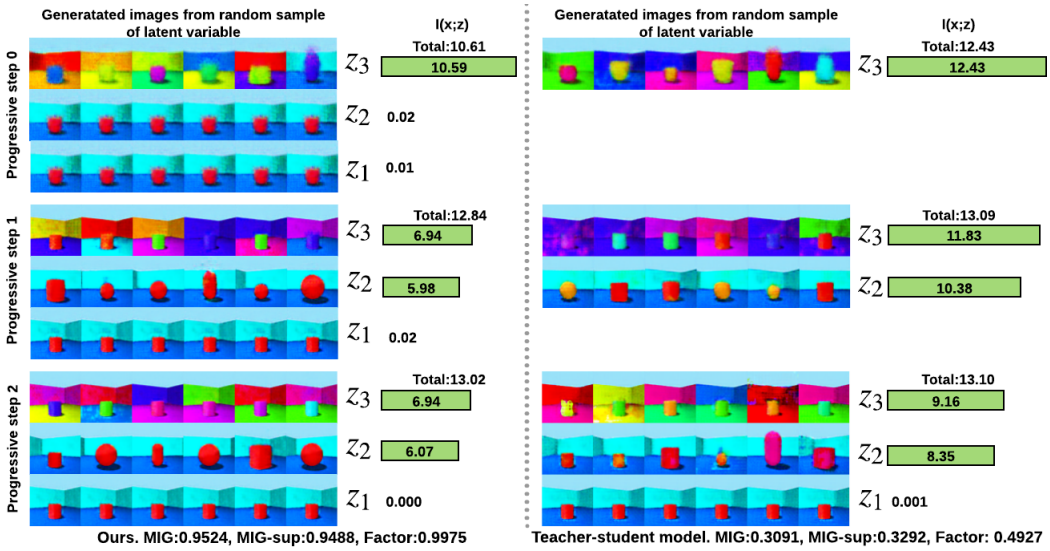

Figure 3.7: Progressive learning of hierarchical representations. At each progression and for each $\mathbf{z}_l$, the row of images are generated by randomly sampling from its prior distributions while fixing the other latent variables (this is NOT traversing). The green bar at each row tracks the mutual information $I(\mathbf{x}; \mathbf{z}_l)$, while the total mutual information $I(\mathbf{x}; \mathbf{z})$ is labeled on top.

the deepest latent variables in $\mathbf{z}_3$, discovering most of the generative factors including color, objective shape, and orientation entangled within $\mathbf{z}_3$. At progressive step 1, interestingly, the model was able to "drag" out shape and rotation factors from $\mathbf{z}_3$ and disentangle them into $\mathbf{z}_2$ along with a new scale factor. Thus $I(\mathbf{x}; \mathbf{z}_3)$ decreased from 10.59 to 6.94 while $I(\mathbf{x}; \mathbf{z}_2)$ increased from 0.02 to 5.98 in this progression, while the total mutual information $I(\mathbf{x}; \mathbf{z})$ increased from 10.61 to 12.84, suggesting the overall learning of more detailed information. Since 3DShapes only has 6 factors, the lowest-level representation $\mathbf{z}_1$ had nothing to learn in progressive step 2, and the allocation of mutual information remained nearly unchanged. Note that the sum of $I(\mathbf{x}, \mathbf{z}_l)$'s does not equal to $I(\mathbf{x}, \mathbf{z})$ and $I_{over} = \sum_1^L I(\mathbf{x}, \mathbf{z}_l) - I(\mathbf{x}, \mathbf{z})$ suggests the amount of information that is entangled.

In comparison, the teacher-student model was less effective in progressively dragging entangled representations to newly added latent dimensions, as suggested by the slowing changing

of $I(\mathbf{x}, \mathbf{z}_l)$'s during progression and the larger value of I_{over}. This suggests that, since the teacher-student model was motivated for progressively learning nuisance variables, the extent to which its capacity can grow for learning new representations is limited by two fundamental causes: 1) because it increases the dimension of the same latent vectors at the same depth, the growth of the network capacity is limited in comparison to *pro*-VLAE, and 2) the Jacobian supervision further restricts the student model to maintain the same disentangling ability of the teacher model.

3.4.3 Disentangling hierarchical representations

We also qualitatively examined *pro*-VLAE on data with both relatively simple (MNIST) and complex (CelebA) factors of variations, all done in unsupervised training. On MNIST (Figure 3.8), while the deepest latent representations encoded the highest-level features in terms of digit identity, the representations learned at shallower levels encoded changes in writing styles. In Figure 3.9, we show the latent representation progressively learned in CelebA from the highest to lowest levels of abstractions, along with disentangling within each level demonstrated by traversing one selected dimension at a time. These dimensions are selected as examples associated with clear semantic meanings. As shown, while the deepest latent representation $\mathbf{z}_4$ learned to disentangle high-level features such as gender and race, the shallowest representation $\mathbf{z}_1$ learned to disentangle low-level features such as eye-shadow. Moreover, the number of distinct representations learned decreased from deep to shallow layers. While demonstrating disentangling by traversing each individual latent dimension or by hierarchically-learned representations has been separately reported in previous works [21, 62], to our knowledge, this is the first time the ability of a model to disentangle individual latent factors in a hierarchical manner has been demonstrated. This provides evidence that the presented the progressive strategy of learning can improve the disentangling of first, the most abstract representations followed by progressively lower levels of abstractions.

3.4.4 A closer comparison with VLAE

Here we presented a closer comparison with the baseline VLAE. Fig 3.10 showed the two-dimensional traversing on MNIST dataset of pro-VLAE following the same generation

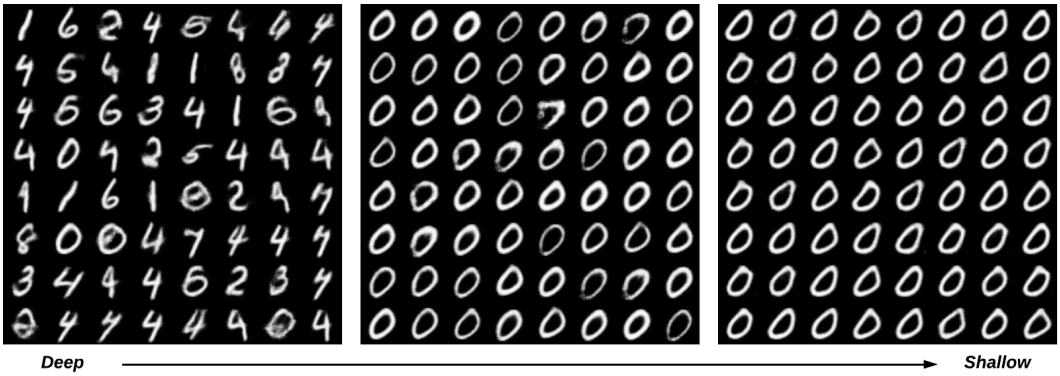

Figure 3.8: Visualization of hierarchical features learned for MNIST data. Each sub-figure is generated by randomly sampling from the prior distribution of $\mathbf{z}_l$ at one abstraction level while fixing the others. The original latent code is inferred from an image with the digit "0". **From left to right:** $\mathbf{z}_3$ encodes the highest abstraction: digit identity; $\mathbf{z}_2$ encodes stroke width; and $\mathbf{z}_1$ encodes other digit styles.

strategy of VLAE. Table 3.1 showed the information allocation in each layer of VLAE and pro-VLAE. Compared to VLAE, the pro-VLAE allocated information in a more clear pattern of descending order due to the progressive learning strategy.

3.4.5 Ablation study on implementation strategies

As shown in Table 3.2, the proposed training strategies ("fade-in" and pre-trained KL penalty) greatly improved the stableness of progressive training.

3.4.6 Closer investigation of information flow over latent variables

Finally, we presented additional quantitative results on how information flow among the latent variables during progressive training. We conducted experiments on both 3DShapes and MNIST data sets, considering different hierarchical architectures including a combination of different numbers of latent layers L and different numbers of latent dimensions z_{dim} for

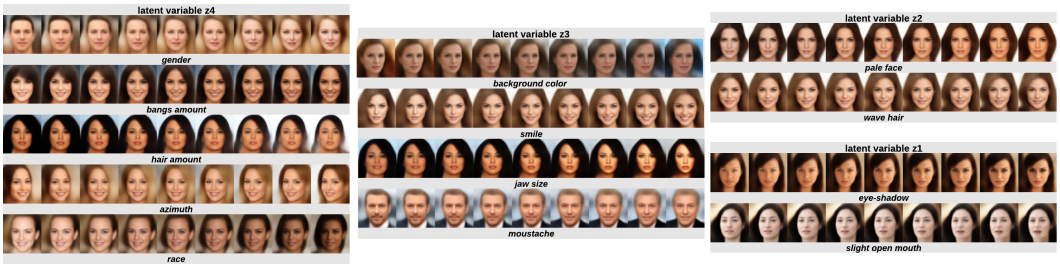

Figure 3.9: Visualization of hierarchical features learned for CelebA data. Each subfigure is generated by traversing along a selected latent dimension in each row within each hierarchy of $\mathbf{z}_l$'s. **From left to right:** latent variables $\mathbf{z}_4$ to $\mathbf{z}_1$ progressively learn major (*e.g.*, gender in $\mathbf{z}_4$ and smile in $\mathbf{z}_3$) to minor representations (*e.g.* wavy-hair in $\mathbf{z}_2$ and eye-shadow in $\mathbf{z}_1$) in a disentangled manner.

each layer. Each experiment was repeated three times with random initializations, from which the mean and the standard deviation of mutual information $I(x; \mathbf{z}_l)$ were computed.

As shown in Tables 3.3 and 3.4, for all hierarchical architectures, the information amount in each layer is captured in a clear descending order, which aligns with the motivation of the presented progressive learning strategy. Generally, the information also tends to flow from previous layers to new layers, suggesting a disentanglement of latent factors as new latent layers are added. This is especially obvious for 3DShapes data where the generative factors are better defined. In addition, models with small latent codes ($z_{dim} = 1$) are not able to learn the same amount of information (total $I(x, \mathbf{z})$) as those with larger latent codes ($z_{dim} = 3$). The variance of information in each layer in the former also appears to be high. We reason that it may be because the model is trying to squeeze too much information into a small code, resulting in large vibrations during progressive learning. On the other hand, while a model has large latent codes ($L = 4, z_{dim} = 3$), the information flow becomes less clear after the addition of certain layers. Overall, assuming there are K generative factors and there are D dimensions in total available in the model, ideally, we would like to design the model such that $D = K$. However, since K is unknown in most data, L and z_{dim} become hyperparameters that need to be tuned for different data sets.

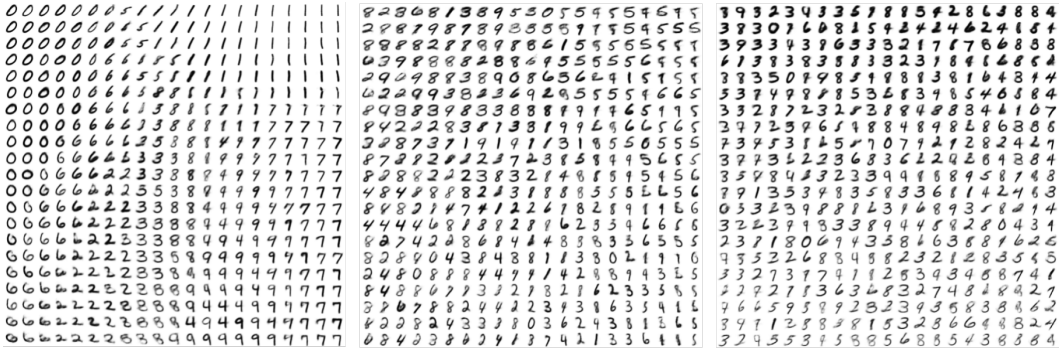

Figure 3.10: MNIST traversing results following the same generation strategy and network hierarchy as those presented in Figure 5 of ([62]). The network has 3 layers and 2-dimensional latent codes at each layer. Each image is generated by traversing each of the two-dimensional latent codes in one layer, while randomly sampling from the other layers. **From left to right:** The top layer $\mathbf{z}_3$ encodes the digit identity and tilt; $\mathbf{z}_2$ encodes digit width (digits around top-left are thicker than digits around bottom-right); and the bottom layer $\mathbf{z}_1$ encodes stroke width. Compared to VLAE, the representation learned in the presented method suggests smoother traversing on digits and similar results for digit width and stroke width.

3.5 Conclusion

In this chapter, we presented a progressive strategy for learning and disentangling hierarchical representations, for answering the first research question: how to progressively learn representations such that it can improve the quality and the disentanglement of representations. Starting from a simple VAE, the model first learns the most abstract representation. Next, the model learns independent representations from high- to low-levels of abstraction by progressively growing the capacity of the VAE from deep to shallow. Experiments on several benchmark data sets demonstrated the advantages of the presented method. Based on these foundation methods and results, to address the rest research questions, we expand progressive representation learning to a continual manner of learning and disentangling representations in dynamic data environments, as we will describe in the next chapter.

Table 3.1: Mutual information $I(x; \mathbf{z}_l)$ between data x and latent codes $\mathbf{z}_l$ at each l-th depth of the network, corresponding to the qualitative results presented in Fig. 3.6 and Fig. 3.8 on 3Dshapes and MNIST data sets. Both VLAE and the presented *pro*-VLAE models have the same hierarchical architecture with 3 layers and 3 latent dimensions for each layer. Compared to VLAE, the presented method allocated information in a more clear descending order owing to the progressive learning.

3DShapes	$I(x; \mathbf{z}_3)$	$I(x; \mathbf{z}_2)$	$I(x; \mathbf{z}_1)$	total $I(x; \mathbf{z})$
VLAE	4.41	4.69	5.01	12.75
pro-VLAE	6.94	6.07	0.00	13.02

MNIST	$I(x; \mathbf{z}_3)$	$I(x; \mathbf{z}_2)$	$I(x; \mathbf{z}_1)$	total $I(x; \mathbf{z})$
VLAE	8.28	8.89	7.86	11.04
pro-VLAE	9.83	8.24	6.28	10.93

Table 3.2: The effect of progressive implementation strategies, *i.e.*, "fade-in" and pre-trained KL penalty, on successful training rates. We conducted 15 ablations experiments that each have 4 sub-experiments. As shown, the *pro*-VLAE cannot be trained successfully without the presented implementation strategies, while each of the strategies helps stabilize the progressive training.

no strategies	pre-trained KL only	fade-in only	Both
0.0	0.667	0.733	0.867

Table 3.3: Mutual information flow on progressive learning 3DShapes dataset, with $L = 2, z_{dim} = 3$, $L = 3, z_{dim} = 2$, and $L = 4, z_{dim} = 1$ accordinginly.

progressive step	$I(x;\mathbf{z}_2)$	$I(x;\mathbf{z}_1)$	total $I(x;\mathbf{z})$
0	10.68 ± 0.19		10.68 ± 0.19
1	7.22 ± 0.30	5.94 ± 0.26	12.88 ± 0.20

progressive step	$I(x;\mathbf{z}_3)$	$I(x;\mathbf{z}_2)$	$I(x;\mathbf{z}_1)$	total $I(x;\mathbf{z})$
0	10.16 ± 0.13			10.16 ± 0.13
1	9.76 ± 0.05	7.36 ± 0.10		13.00 ± 0.02
2	6.83 ± 1.37	6.66 ± 0.17	5.80 ± 0.41	13.07 ± 0.02

progressive step	$I(x;\mathbf{z}_4)$	$I(x;\mathbf{z}_3)$	$I(x;\mathbf{z}_2)$	$I(x;\mathbf{z}_1)$	total $I(x;\mathbf{z})$
0	4.89 ± 0.03				4.89 ± 0.03
1	4.77 ± 0.04	3.55 ± 0.04			8.14 ± 0.09
2	4.66 ± 0.04	3.75 ± 0.04	2.70 ± 0.10		10.67 ± 0.09
3	4.55 ± 0.11	3.53 ± 0.35	2.80 ± 0.19	2.17 ± 0.14	11.72 ± 0.03

Table 3.4: Mutual information flow on progressive learning MNIST dataset, with $L = 3, z_{dim} = 1$, $L = 3, z_{dim} = 3$, and $L = 4, z_{dim} = 3$ accordinginly.

progressive step	$I(x;\mathbf{z}_3)$	$I(x;\mathbf{z}_2)$	$I(x;\mathbf{z}_1)$	total $I(x;\mathbf{z})$
0	5.86 ± 1.19			5.86 ± 1.19
1	3.62 ± 1.04	4.64 ± 2.83		7.62 ± 2.63
2	3.88 ± 0.75	4.99 ± 0.98	2.37 ± 0.77	8.25 ± 1.65

progressive step	$I(x;\mathbf{z}_3)$	$I(x;\mathbf{z}_2)$	$I(x;\mathbf{z}_1)$	total $I(x;\mathbf{z})$
0	10.08 ± 0.10			10.08 ± 0.10
1	$9,97 \pm 0.05$	8.03 ± 0.17		11.01 ± 0.04
2	9.91 ± 0.04	8.09 ± 0.07	6.27 ± 0.02	11.02 ± 0.02

progressive step	$I(x;\mathbf{z}_4)$	$I(x;\mathbf{z}_3)$	$I(x;\mathbf{z}_2)$	$I(x;\mathbf{z}_1)$	total $I(x;\mathbf{z})$
0	10.06 ± 0.22				10.06 ± 0.22
1	10.11 ± 0.06	7.95 ± 0.08			10.98 ± 0.02
2	10.06 ± 0.08	8.1 ± 0.04	6.39 ± 0.12		10.98 ± 0.06
3	9.99 ± 0.09	8.11 ± 0.03	6.45 ± 0.15	3.52 ± 0.07	11.03 ± 0.03

Chapter 4

Continual Unsupervised Disentangling of Self-Organizing Representations

In this chapter, we introduced an under-investigated research direction that extends unsupervised representation learning to continual learning settings. Unlike learning and disentangling representations in a static data environment where the model sees all semantic factors at once, which is a typical setting, continually learning and disentangling sequentially-arrived semantic factors is fundamentally different. In this scenario, the model needs to confirm that it 1) does not forget previously learned factors, 2) is able to re-use latent dimensions corresponding to shared factors, and 3) prevents new presentations to be entangled into the old ones. We argue that a fundamental bottleneck is that the existing models often treat continually-arrived data independently with little knowledge of how they are related based on the representations. To address these challenges, we build our work from two main components. In the first component, we developed a foundation method, a VAE with self-organizing latent space, to learn the relational structure of data, for answering the second research question: how to continually learn and accumulate knowledge of representations from different data environments, as described in section 4.3.1. In the second component, we use knowledge learned by the foundation model for continual representation learning,

including data-summary-based generative reply, and use the relational structure to improve continual disentanglement, for answering the third research question: how to continually reuse the existing representations to facilitate learning and disentangling representations given new data environments, as detailed explained in section 4.3.2.

4.1 Introduction

The progress in continual learning has been mostly made for supervised discriminative learning, whereas continual unsupervised representation learning remains relatively under-explored [1, 45, 46]. The few existing works have primarily focused on battling *catastrophic forgetting* in the generative performance of a model: for instance, a common approach known as *generative-replay* synthesizes *past* samples using a snapshot of the generative model trained from past data, and then continually trains the model to generate both new data and synthesized past samples [1, 45, 46].

There is however another important yet under-explored question in continual unsupervised representation learning: how to *reuse, expand, and continually disentangle* latent semantic factors across different data environments? These are inherent in the human learning process: while learning from new data (*e.g.*, learning cars after bicycles), we are naturally able to reuse shared semantic factors without re-learning (*e.g.*, wheels), expand and disentangle new semantic factors (*e.g.*, the shape of cars), while accumulating knowledge about the relationship among data environments based on these semantic factors (*e.g.*, bicycles and cars both have wheels but are different in shapes). Disentangled representation learning, as a long-standing research topic, has demonstrated various benefits in generative modeling and downstream tasks [21, 22, 25, 29, 36, 48]. With increasing recent interests in unsupervised representation learning in a continual learning setting [38, 46], it is important to investigate the challenges and solutions to achieve disentanglement of sequentially-arrived semantic factors in streaming data.

Reusing latent dimensions for learned semantic factors has mainly been attempted by a teacher-student like approach where the student model is taught to infer and generate similarly to a snapshot of the past models (teacher) on *replayed* data [1, 45]. In [1], this is further facilitated by explicitly masking out latent dimensions that are not actively used in a data environment. Such masks however have to be heuristically defined *before* training on

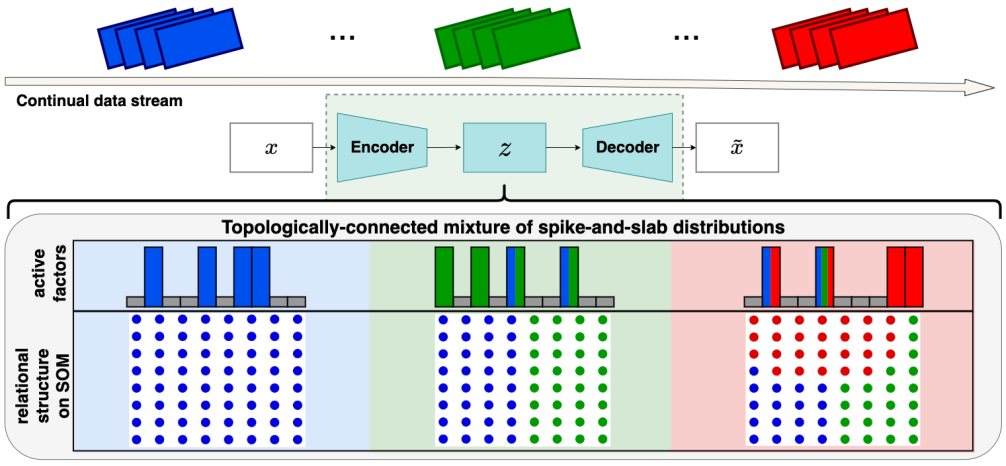

Figure 4.1: Through a self-organizing spike-and-slab mixture, CUDOS continually distills knowledge about the relational structure of data environments with their shared and distinct semantic factors.

the new environment. How to automatically discover latent dimensions explaining *active* semantic factors underlying each data environment, in a continual fashion, remains an open question.

Expanding learned semantic factors, in part, results naturally from continually optimizing a generative model to newly arrived data [1, 45], or even progressively increasing the model's latent capacity on the same data [8,34]. These alone however would not uncover the relational structure among data environments in the latent space. In [46], a mixture of Gaussians has been used and continually expanded such that newly arrived data are clustered to an existing or new mixture component. While this captures the expanding data distributions, it does not consider the reuse of semantic factors among data clusters.

Continually disentangling semantic factors, until now, is limited to the native disentangling ability inherent in VAE, or promoting the reusing of shared semantic factors [1, 45]. While the common strategy of generative replay teaches a model what latent dimensions to use for shared semantic factors on the replayed data [1, 45], no such guidance is available on new data. As a result, as we will show, none of the existing approaches can prevent newly-learned semantic factors to be entangled with re-used ones.

In this work, we show that the above limitations boil down to a fundamental bottleneck in existing continual unsupervised learning of representations: that the learner is asked to treat continually-arrived data independently, *without knowing how they are related based on the underlying semantic factors.* To overcome this, we argue that the model needs to learn two critical knowledge: latent dimensions explaining *active* semantic factors underlying each data environment, and the relationship among the latter based on the former. We present Continual Unsupervised Disentangling of self-Organizing representations (CUDOS) that is able to accumulate the relational structure of continually-arrived data based on their underlying *active* semantic factors, and exploiting this knowledge to guide disentangling of sequentially-arrived semantic factors. As illustrated in Fig. 4.1, to accumulate the relational structure of the data, we model the latent representations with a topologically-connected mixture of distributions via Bayesian self-organizing maps (SOM) [28, 57]. To automatically discover *active* semantic factors underlying each data environment, we model each component of the SOM mixture with a spike-and-slab distribution [51, 52], such that the sparse spike variable identifies latent dimensions explaining active semantic factors. This results in a generative model with a self-organizing mixture of slab-and-spike distributions, where the distilled knowledge – the relational structure of data environments and their associated active semantic factors – supports 1) mixture-based generative replay and 2) continual disentangling of sequentially-arrived semantic factors.

We evaluated CUDOS on both benchmark datasets for continual representation learning, and a split version of 3DShapes [7] designed for quantitative evaluation of disentangling sequentially-arrived semantic factors. In comparison to existing works, we showed that CUDOS not only addressed catastrophic forgetting, but also improved – both quantitatively and qualitatively – continual disentanglement of latent semantic factors and thereby downstream discriminative tasks.

4.2 Related works

Deep Learning with SOM: The use of SOM has been explored within deep learning of image classification [35], image clustering [15, 39], and time-series prediction [16, 23]. None of these works, however, considered continual learning of representations. In the context of continual learning, SOM was used to first learn the relationship among all discriminative tasks, and then used to mask the neurons of a fully-connected layer for each task [3]. A

dual-memory self-organizing architecture was also presented for learning object instances and categories in life-long object recognition [43]. While both works shared our motivation to leverage the ability of SOM to maintain and expand a memory of data distributions across environments, neither considered continual unsupervised representation learning, or utilized SOM as a topologically-connected Bayesian mixture model.

Inferring Active Semantic Factors in VAE: Learning meaningful representations is a vital task for VAE, where large latent space often leads to latent dimensions that carry little information [10]. Sparse coding and discrete latent space have proved to be elegant solutions [42, 52]. Non-parametric discrete densities have been explored for stochastic latent activation to improve disentanglement [19]. A discrete latent space based on vector quantization was shown to solve posterior collapse [42], whereas sparsity was directly modeled in a continuous latent space using spike-and-slab priors [52]. None of these concepts has been extended to continual unsupervised learning of representations.

Continual Unsupervised Representation Learning: Most existing works in continual unsupervised representation learning relied on enhanced generative replay to combat catastrophic forgetting of generation using learned semantic factors [1, 45, 56]. The two most related works are those presented in [1] and [46]. In [1], to reuse latent dimensions explaining semantic factors shared with past data, a mask for active latent dimensions specific to each data environment was used [1]. This mask was heuristically determined *before* training on the new data, which will affect the learning and sharing of latent dimensions on the new data. Furthermore, while this strategy protects latent dimensions specific to past data environments (not shared and thus turned off), it does not prevent new semantic factors from being entangled with the shared dimensions. CUDOS presents fundamental solutions to these problems by 1) principled variational inference of *active* latent dimensions leveraging slab-and-spike priors, and 2) guiding continual disentangling of sequentially-arrived semantic factors by exploiting the relational structure of data.

In [46], a mixture of Gaussians was continually expanded as new data arrive. This shares our intuition that continual learning of representations can benefit from accumulating an evolving summary of data in the latent space. How the different data clusters are related in terms of shared and distinct semantic factors, however, was not exploited in [46]. In contrast, CUDOS exploits the relationship among data clusters based on their underlying active semantic factors, and uses that to facilitate continual reuse and disentangling of sequentially-arrived semantic factors.

4.3 Methods

We first establish the foundation for CUDOS: a VAE with a self-organizing mixture of spike-and-slab priors to learn the relational structure of data based on their *active* semantic factors (Section 4.3.1). We then describe how to use this data summary for generative replay (Section 4.3.2), and use the relation between new and past data to improve continual disentangling of representations (Section 4.3.2).

4.3.1 Learning the Relational Structure of Data via Active Semantic Factors

Fig. 4.1 outlines the foundation model underlying CUDOS. Given data $\mathbf{x}$, we are interested in learning representations of meaningful semantic factors within a latent vector $\mathbf{z} \in \mathbb{R}^J$. While doing so we encourage sparse coding to discover latent dimensions explaining *active* semantic factors in VAE [52], while learning the relational structure of data based on these semantic factors.

Generative Model: To accumulate the relational structure of data, we model the latent space by a mixture of distributions using Bayesian-SOM [57] with K nodes. Additionally, to discover the active semantic factors for each data environment, we model each mixture component as a spike-and-slab distribution that encourages sparsity in the latent dimensions [52]. This gives rise to the following generative process:

$$\begin{gathered} w \sim \mathrm{Cat}(\boldsymbol{\pi}), \quad p_{\psi_k}(\mathbf{z}|w_k = 1) = \prod_{j=1}^{J} [\alpha_k^j \mathcal{N}(z^j; \mu_k^j, (\sigma_k^j)^2) + (1 - \alpha_k^j)\delta(z^j)], \\ p_\psi(\mathbf{z}) = \sum_{k=1}^{K} p(\mathbf{z}|w_k)p(w_k), \quad \mathbf{x} \sim p_\theta(\mathbf{x}|\mathbf{z}), \quad p_{\theta,\psi}(\mathbf{x}, \mathbf{z}, w) = p_\theta(\mathbf{x}|\mathbf{z})p_\psi(\mathbf{z}|w)p(w). \end{gathered} \tag{4.1}$$

where $*^j$ denotes the j-th element of $*$, and $\delta(*)$ Dirac delta function centered at zero. The mixing prior $p(w)$ is parameterized by $\boldsymbol{\pi}$, $\pi_i \geq 0$ and $\sum \pi_i = 1$. The latent variable $\mathbf{z}$ from the k-th mixture component is parametrized by $\psi_k = \{\boldsymbol{\mu}_k, \boldsymbol{\sigma}_k^2, \boldsymbol{\alpha}_k, \pi_k\}$, where parameter $\boldsymbol{\alpha}_k$ introduces sparsity to mask out inactive dimensions. Data $\mathbf{x}$ are generated from $\mathbf{z}$ via a neural network parametrised by θ.

Inference Model: We define variational approximations of the posterior density $p(\mathbf{z}, w|\mathbf{x})$

as:

$$q(\mathbf{z}, w|\mathbf{x}) = q_\phi(\mathbf{z}|\mathbf{x})p_\psi(w|\mathbf{z}), \quad q_\phi(\mathbf{z}|\mathbf{x}) = \prod_{j=1}^{J}[\tilde{\alpha}^j \mathcal{N}(z^j; \tilde{\mu}^j, (\tilde{\sigma}^j)^2) + (1-\tilde{\alpha}^j)\delta(z^j)],$$
$$p_\psi(w_k = 1|\mathbf{z}) = \frac{p(\mathbf{z}|w_k, \psi_k)p(w_k = 1)}{\sum_{k'=1}^{K} p(\mathbf{z}|w_{k'}, \psi_{k'})p(w_{k'} = 1)}. \tag{4.2}$$

where $p_\psi(w_k = 1|\mathbf{z})$ is the posterior probability of the k-th mixture component. For parameters $\tilde{\boldsymbol{\mu}}$ and $\log\tilde{\boldsymbol{\sigma}}^2$ of the spike-and-slab distribution, their inference is amortized as the output of an encoding network parameterized by ϕ. For parameter $\tilde{\boldsymbol{\alpha}}$, considering that similar data should share latent dimensions for common semantic factors, we infer it at a set level as inspired by the *neural statistician* [13, 20]. We discuss how to determine the set in Section 4.3.2.

Variational Inference: The parameters ψ, θ, and ϕ are optimized by the ELBO loss as:

$$\begin{aligned}
&\log p(\mathbf{x}) \geq \mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(\mathbf{z},w|\mathbf{x})}[\log \frac{p_{\theta,\psi}(\mathbf{z},\mathbf{x},w)}{q_\phi(\mathbf{z},w|\mathbf{x})}] \\
&= \mathbb{E}_{q_\phi(\mathbf{z},w|\mathbf{x})}[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p_\psi(\mathbf{z}|w)p(w)}{q_\phi(\mathbf{z}|\mathbf{x})p_\psi(w|\mathbf{z})}] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] \\
&\quad - \mathbb{E}_{p_\psi(w|\mathbf{z})}[D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\psi(\mathbf{z}|w))] \\
&\quad - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[D_{KL}(p_\psi(w|\mathbf{z})||p(w))].
\end{aligned} \tag{4.3}$$

The first reconstruction term is similar to that in the vanilla VAE. The second term measures the KL-divergence between $q_\phi(\mathbf{z}|\mathbf{x})$ and its conditional prior, measured over the posterior distribution of the SOM mixture $p_\psi(w|\mathbf{z})$. Specifically, we estimate the expectation over $p_\psi(w|\mathbf{z})$ as:

$$\mathbb{E}_{p_\psi(w|\mathbf{z})}[D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\psi(\mathbf{z}|w))] = \sum_{k=1}^{K} p_\psi(w_k = 1|\mathbf{z}) D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\psi(\mathbf{z}|w_k = 1)], \tag{4.4}$$

where $p_\psi(w_k|\mathbf{z})$ can be computed in a batch during forward propagation, and $D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p_\psi(\mathbf{z}|w_k = 1)$ can be derived following [52] as:

$$\sum_{j}^{J}[\tilde{\alpha}^j(\log\frac{\sigma_k^j}{\tilde{\sigma}^j} + \frac{(\tilde{\mu}^j - \mu_k^j)^2 + (\tilde{\sigma}^j)^2}{2(\sigma_k^j)^2} - \frac{1}{2}) + (1-\tilde{\alpha}^j)\log\frac{1-\tilde{\alpha}^j}{1-\alpha_k^j} + \tilde{\alpha}^j \log\frac{\tilde{\alpha}^j}{\alpha_k^j}] \tag{4.5}$$

This KL loss encourages $q_\phi(\mathbf{z}|\mathbf{x})$ to follow a mixture density with the mixing probability determined by the posterior probability of each component given $\mathbf{z}$. Note that the latter automatically considers sharing of semantic factors between the inferred $\mathbf{z}$ and each mixture component k (via spike variable $\tilde{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}_k$). The third term in Eqn. (4.3) measures the KL-divergence between the posterior density of w and its prior (set to be uniform in this work). The expectation is estimated by Monte Carlo samples.

Iterative Optimization: We maximize the ELBO loss as defined in Eqn. (4.3) by iterative optimization. In each iteration, we first fix ψ of the SOM mixture and maximize Eqn. (4.3) with respect to the VAE's parameters θ and ϕ by stochastic gradient descent with reparameterization trick [26, 52]: at the first iteration, the SOM-mixture is initialized as a uniform mixture of $\psi = \{\mathbf{0}, I, \mathbf{0.2}\}$ and the optimization becomes a standard ELBO with spike-and-slab priors.

With the updated θ and ϕ, we then maximize Eqn. (4.3) with respect to the SOM-mixture parameter ψ which, as derived in Eqn. (4.6), amounts to maximizing the expectation of the log-likelihood of $p_\psi(\mathbf{z})$ over the variational posterior distribution of $q_\phi(\mathbf{z}|\mathbf{x})$: $\psi^* = \text{argmax}_\psi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\psi(\mathbf{z})]$. To make the partial derivative of ψ clearer, we first rewrite the last two terms of ELBO in Eqn. (4.3) as:

$$
\begin{aligned}
&- \mathbb{E}_{p_\psi(w|\mathbf{z})}[D_{KL}(q_\phi(\mathbf{z}|\mathbf{x})||p_\psi(\mathbf{z}|w))] - \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[D_{KL}(p_\psi(w|\mathbf{z})||p(w))] \\
&= \int_{p_\psi(w|\mathbf{z})} p_\psi(w|\mathbf{z}) \int q_\phi(\mathbf{z}|\mathbf{x}) log \frac{p_\psi(\mathbf{z}|w)}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z} dw + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \int p_\psi(w|\mathbf{z}) log \frac{p(w)}{p_\psi(w|\mathbf{z})} dw \\
&= \int_{q_\phi(\mathbf{z}|\mathbf{x})} q_\phi(\mathbf{z}|\mathbf{x}) \int p_\psi(w|\mathbf{z}) log p_\psi(\mathbf{z}|w) d\mathbf{z} dw \\
&\quad - \int_{p_\psi(w|\mathbf{z})} p_\psi(w|\mathbf{z}) \int q_\phi(\mathbf{z}|\mathbf{x}) log q_\phi(\mathbf{z}|\mathbf{x}) d\mathbf{z} dw \\
&\quad + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \int p_\psi(w|\mathbf{z}) log \frac{p(w)}{p_\psi(w|\mathbf{z})} dw \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \int p_\psi(w|\mathbf{z}) log \frac{p_\psi(\mathbf{z}|w) p(w)}{p_\psi(w|\mathbf{z})} dw - \mathbb{E}_{p_\psi(w|\mathbf{z}), q_\phi(\mathbf{z}|\mathbf{x})}[log q_\phi(\mathbf{z}|\mathbf{x})] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[log p_\psi(\mathbf{z})] - \mathbb{E}_{p_\psi(w|\mathbf{z}), q_\phi(\mathbf{z}|\mathbf{x})}[log q_\phi(\mathbf{z}|\mathbf{x})]
\end{aligned}
\tag{4.6}
$$

Therefore the optimized ψ^* can be solved by:

$$
\psi^* = \text{argmax}_\psi \text{ELBO} = \text{argmax}_\psi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[log p_\psi(\mathbf{z})] \tag{4.7}
$$

We follow the theory in [17] to optimize ψ using stochastic gradient descent.

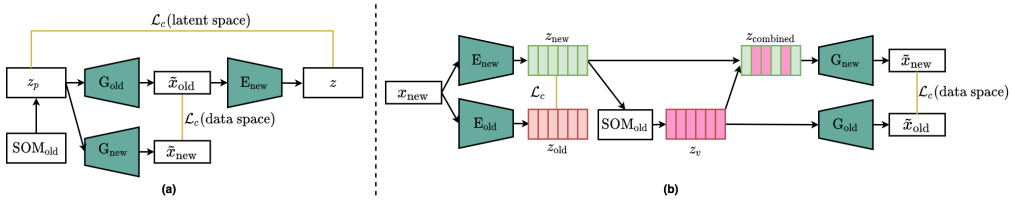

Figure 4.2: (a) Combating catastrophic forgetting by generative replays from SOM-mixture, and (b) Continually disentangling by using the relation between new and past data based on their underlying shared semantic dimensions.

4.3.2 Continual Learning with CUDOS

We now consider a setting of continual unsupervised learning where the label of the underlying data environments is unknown. Following existing approaches [1, 45, 46], we maintain a snapshot of the model parameters $[\psi_{\text{old}}, \theta_{\text{old}}, \phi_{\text{old}}]$ every τ training steps. These model snapshots are used to guide (1) synthesizing replayed samples to teach the model to perform consistently on past data (Section 4.3.2), and (2) continually disentangling sequentially-arrived semantic factors using the relationship between past and new data (4.3.2).

Inferring α for streaming data

As mentioned in Section 4.3.1, we choose to infer $\tilde{\boldsymbol{\alpha}}$ at a set level to leverage shared information underpinning a set of data. If the boundary between data environments is known, $\tilde{\boldsymbol{\alpha}}$ can be shared for all data within the same data environment. Alternatively, if the boundary of data environments is not known, we assume $\tilde{\boldsymbol{\alpha}}$ to be shared within each mini-batch $\boldsymbol{x}_{\text{batch}}$. Specifically, we maintain $\phi_\alpha = \{\tilde{\boldsymbol{\alpha}}_1, \tilde{\boldsymbol{\alpha}}_2, \cdots, \tilde{\boldsymbol{\alpha}}_d\}$ for d number of sets with distinct underlying semantic factors. Given a new batch of $\boldsymbol{x}_{\text{batch}}$, we first determine if it can be described by an existing $\tilde{\boldsymbol{\alpha}}$, or a new $\tilde{\boldsymbol{\alpha}}_{d+1}$ has to be allocated if the existing ϕ_α fails to describe $\boldsymbol{x}_{\text{batch}}$ well:

$$\tilde{\boldsymbol{\alpha}}_{\text{new}} = \begin{cases} \hat{\boldsymbol{\alpha}}, & \text{if } \mathbb{E}_{p(\boldsymbol{z}|\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\sigma}}^2,\hat{\boldsymbol{\alpha}})}[\mathcal{L}_r] \leq T_\alpha, \\ \tilde{\boldsymbol{\alpha}}_{d+1}, & \text{otherwise,} \end{cases} \tag{4.8}$$

where $\hat{\boldsymbol{\alpha}} = \text{argmin}_{\tilde{\boldsymbol{\alpha}} \in \phi_\alpha} \mathbb{E}_{p(\boldsymbol{z}|\tilde{\boldsymbol{\mu}},\tilde{\boldsymbol{\sigma}}^2,\tilde{\boldsymbol{\alpha}})}[\mathcal{L}_r]$, T_α is a threshold for the reconstruction error $\mathcal{L}_r$ of $\boldsymbol{x}_{\text{batch}}$ averaging over pixels. Note that Equation (4.8) only determines how a data batch

$\boldsymbol{x}_{\text{batch}}$ is associated with a variable $\tilde{\boldsymbol{\alpha}}$. The values of these variables are optimized during variational inference.

Generative replay with CUDOS

To combat forgetting, we synthesize data samples following the generation process as defined in Eqn. (4.1), using the snapshot of the SOM mixture with parameters ψ_{old}. As illustrated in Fig. 4.2(a), on the synthesized samples $\tilde{\mathbf{x}}_{\text{old}}$ and their corresponding latent samples $\mathbf{z}_{\text{p}}$, we encourage the model to be consistent with its past snapshot [1, 31, 45]:

$$\mathcal{L}_{\text{old}} = \mathcal{L}_c[S(p_\theta(\mathbf{x}|\mathbf{z}_{\text{p}})), \tilde{\mathbf{x}}_{\text{old}}] + \mathcal{L}_c[S(q_\phi(\mathbf{z}|\tilde{\mathbf{x}}_{\text{old}})), \mathbf{z}_{\text{p}}], \tag{4.9}$$

where $\mathcal{L}_c$ is mean squared error and $S(\cdot)$ is a sampling process. This strategy combines the two key existing concepts in *replay mechanism*: as in generative replay [45, 49], data are synthesized quickly and readily with limited burden on storage; as in core-set methods [6, 41], the SOM-mixture provides a representative summary of data ensuring that more important mixture components are more frequently re-used in future training.

Guiding Continual Disentanglement

Generative replay lacks mechanisms to teach the model how to disentangle newly-arrived semantic factors from latent dimensions already used for previously-learned semantic factors. CUDOS provides a unique opportunity to address this issue by its ability to describe the relation between new and past data. For a new data $\mathbf{x}$, its shared latent dimensions with SOM summary of past data (parameterized by ψ_{old}) is determined by a mask $\mathbf{s}_{\psi_{\text{old}}}$ computed by a scaled and displaced Sigmoid function:

$$s^j_{\psi_{old}} = \text{Sigmoid}(b(\tilde{\alpha}^j \cdot \alpha^j_{v_{\text{BMU}},\psi_{\text{old}}} - 0.5)), \tag{4.10}$$

where $\tilde{\boldsymbol{\alpha}}$ is the spike variable inferred from $\mathbf{x}$ and $\boldsymbol{\alpha}_{v_{\text{BMU}},\psi_{\text{old}}}$ the spike variable of the best-matching unit v_{BMU} (the component with the largest posterior) on the snapshot of the past SOM. b scales and sharpens the Sigmoid function towards a gated function [52]. Note that neither $\mathbf{s}_{\psi_{\text{old}}}$ or v_{BMU} is fixed; they are functions of the unknown spike variables $\tilde{\boldsymbol{\alpha}}$ and $\boldsymbol{\alpha}_{v_{\text{BMU}},\psi_{\text{old}}}$. We now use this to (1) maintain consistency on latent dimensions for shared semantic factors (if any), and (2) prevent entangling new semantic factors into the shared latent dimensions.

Reusing Shared Semantic Factors: To teach the model to reuse latent dimensions corresponding to previously-learned semantic factors on new data $\mathbf{x}$, we ask the VAE encoder q_ϕ to be consistent with its past snapshot $q_{\phi_{\text{old}}}$ in the shared dimensions when inferring from the new data:

$$\mathcal{L}_{\text{newz}} = \mathcal{L}_c[S(q_\phi(\mathbf{z}|\mathbf{x})) \odot \mathbf{s}_{\psi_{old}}, S(q_{\phi_{\text{old}}}(\mathbf{z}|\mathbf{x})) \odot \mathbf{s}_{\psi_{old}}], \tag{4.11}$$

where $\odot$ is element-wise multiplication.

Disentangling New Semantic Factors: Now since the past model does not necessarily know how to generate new data $\mathbf{x}$, how does it teach the new model? Intuitively, given the relation between $\mathbf{x}$ and the past data as determined in Eqn. (4.10), even though the past model does not know how to generate the new $\mathbf{x}$, it would know how to generate from those shared semantic dimensions in $\mathbf{x}$. Therefore, as illustrated in Fig. 4.2(b), if we take a sample $\mathbf{z}_{v_{\text{BMU}},\psi_{\text{old}}}$ from the mixture component v_{BMU}, and replace its latent values with those inferred from $\mathbf{x}$ at the shared latent dimensions, the model should know how to generate from this combined latent vector $\mathbf{z}_{\text{com}}$ in a way consistent with the past generator:

$$\mathbf{z}_{v_{\text{BMU}},\psi_{\text{old}}} = S(p_{\psi_{\text{old}}}(\mathbf{z}|v_{\text{BMU}})), \quad \mathbf{z}_{\text{com}} = \mathbf{z}_{v_{\text{BMU}},\psi_{\text{old}}} \odot (\mathbf{1} - \mathbf{s}_{\psi_{old}}) + S(q_\phi(\mathbf{z}|\mathbf{x})) \odot \mathbf{s}_{\psi_{old}}, \tag{4.12}$$

$$\mathcal{L}_{\text{newx}} = \mathcal{L}_c[S(p_{\theta_{\text{old}}}(\mathbf{x}|\mathbf{z}_{\text{com}})), S(p_\theta(\mathbf{x}|\mathbf{z}_{\text{com}}))]. \tag{4.13}$$

Intuitively, if the model entangles new semantic factors into the shared dimensions, it will be penalized as the past decoder does not know how to generate from these new factors. As we will demonstrate, this constraint – uniquely made possible by CUDOS – is critical in continual disentanglement.

Summary: In summary, the overall loss for CUDOS is:

$$\mathcal{L} = \mathcal{L}_{\text{ELBO}} + \gamma_1 \mathcal{L}_{\text{old}} + \gamma_2 \mathcal{L}_{\text{newz}} + \gamma_3 \mathcal{L}_{\text{newx}}, \tag{4.14}$$

where γ_1, γ_2, and γ_3 are weighting hyperparameters. This extends the foundation objective function to continual learning settings, which promotes the sharing of shared semantic factors between new and past data environments, and the disentangling of new semantic factors.

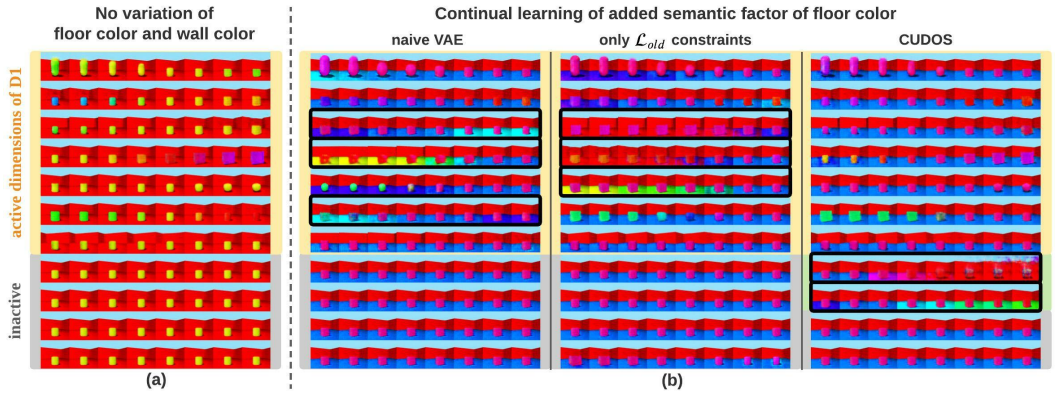

Figure 4.3: Continually learning a split version of 3DShapes where the variations of floor colors were absent initially and appeared later. Black bounding boxes annotate in which latent dimensions the new semantic factors are learnt. (a): Traversing each latent dimension after training on data with no floor color variations. (b): Traversing each dimension after training on the data with floor color variations, where comparison of CUDOS with baseline methods shows the improved ability to disentangle new semantic factors into previously inactive dimensions.

4.4 Experiments and Results

We evaluated CUDOS on (1) a split version of 3DShapes [7] for quantitative evaluation of continual disentanglement, (2) MNIST [30], Fashion-MNIST [55], and their moving versions in [1], and (3) split-CelebA [35].

Split-3DShapes: We quantitatively evaluated the continual disentanglement of past and new representations in a split version of 3Dshapes with two sub-sets: The first one only had red floor and wall, and the second added all floor colors except red. A successful continual learning of representations is expected to continually learn the new factor of floor color while reusing the others learned in the first set. We compared CUDOS to four groups of baselines: (1) naive VAE [26], naive TC-VAE [9], and VAE with gradually increased capacity [8] without an explicit mechanism to combat catastrophic forgetting, (2) unsupervised continual learning reliant on generative replay [1, 45] and heuristically-defined masks of active latent dimensions [1], (3) unsupervised continual learning using a mixture of Gaussian in the latent space [46], and (4) a continual learning version of VQ-VAE [42] (a prototype-based

Table 4.1: Quantitative metrics for disentanglement. MIG: mutual information gap [9]. MIG-sup: supplement mutual information gap [34]. $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$: mutual information between already active dimensions and the new factors. Δ recon-loss: the relative loss change for reconstructing past data.

Method	MIG ↑	MIG-sup ↑	$I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$ ↓	Δ recon-loss ↓
Naive VAE	0.23±0.05	0.23±0.09	1.22±0.30	0.35±0.04
Naive TC-VAE	0.30±0.14	0.38±0.17	1.11±0.16	0.36±0.14
Burgess2018	0.19±0.03	0.13±0.02	1.15±0.10	0.30±0.04
Continual TC-VAE	0.14±0.11	0.25±0.11	1.31±0.67	0.01±0.00
Continual VQ-VAE	0.12±0.05	0.23±0.05	1.47±1.24	0.01±0.00
Achille2018	0.16±0.06	0.23±0.08	0.64±0.20	0.03±0.01
Rao2019	0.10±0.08	0.09±0.04	0.89±0.08	0.04±0.01
Ramapuram2020	0.20±0.06	0.30±0.07	1.12±0.22	**0.00±0.00**
CUDOS	**0.24±0.05**	**0.33±0.05**	**0.02±0.03**	0.02±0.01

method similar to SOM) and TC-VAE [9] (a representative disentangling VAE) with replay mechanism $\mathcal{L}_{\text{old}}$. Experiments on each model were executed at least 5 times.

Choosing suitable disentanglement metrics is vital as different metrics may measure different aspects of the disentanglement [37, 59]. Here we consider two types of metrics. First, to form a complete measurement of the one-to-one relationship between latent dimensions and semantic factors, we chose MIG-sup [34] that penalizes learning multiple semantic factors into the same dimensions in combination with the complementary MIG [9]. Second, to focus on the disentanglement of sequentially-arrived semantic factors, we compute the mutual information $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$ between active dimensions for past data, and new factors in the new data. Ideally, if a model manages to disentangle new semantic factors into dimensions not used by previous data, $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$ should be close to 0. Finally, we also compute the relative change of reconstruction loss for past data to measure forgetting.

As shown in Fig.4.3(a), major semantic factors such as object shape, size, and color were learned from the first subset. When a new data stream is introduced (shown in Fig.4.3(b)), while all models were able to reuse most of the latent dimensions corresponding to previously-learned semantic factors, all baseline models entangled new semantic factors – the floor

color – with these dimensions. In contrast, CUDOS was able to not only reuse shared latent dimensions, but also disentangle new ones into previously inactive dimensions. Note that we include results of [45] in Fig.4.3 as an example of the baseline models that have constraints on replayed data, annotated by $\mathcal{L}_{\text{old}}$. Visual traversing results of other comparison models can be found in Section.4.4.3. Additional results on different sequences of split-3DShapes can be found in Section 4.4.7.

Quantitatively, as summarized in Table. 4.1, CUDOS witnessed the lowest mutual information $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$ among all comparison models, as well as the best disentanglement performance as measured by MIG and MIG-sup. This carefully designed experiment provided clear evidence that CUDOS is able to continually disentangle new semantic factors without entangling them with shared ones learned from the past. It also showed that this cannot be achieved by naively using generative replay to extend existing disentangling or prototype-based VAE (*e.g.*, TC-VAE and VQ-VAE) into continual versions. More discussion of differences between static and continual disentanglement can be found in Section 4.4.8.

Fig. 4.4 (a) shows that the SOM-mixture continually updated a summary of the relation between past (blue box) and new data environments (black box) based on their active latent dimensions (b). T-SNE plots of the learned latent representations after continual learning as an alternative way to visualize the relation among data are provided in Section 4.4.6.

Moving MNIST & Fashion-MNIST: Based on the original MNIST and fashion-MNIST dataset, we created the moving version of them by: (1) Resize the original 28*28 images to 36*36 images, (2) Place the original 36*36 images image at the top-left of a 64*64 black background. (2) Apply translation for both x and y axis with values [5,10,15,20,25]. We then tested CUDOS on sequences of Fashion-MNIST, MNIST, and moving versions of them similar to that presented in [1]. Fig.5 presented results for one sequence: Moving Fashion-MNIST $\rightarrow$ Moving MNIST $\rightarrow$ Fashion-MNIST. As shown, CUDOS was able to disentangle new semantic factors (green boxes), while re-using those learned from the past (red boxes). For instance, the positional semantic factors learned from Moving Fashion-MNIST were reused while learning Moving MNIST.

CelebA: We further designed experiments for continually learning over split versions of CelebA data. Fig.4.6 provide examples of results obtained by two different splits: by age (top), and by bangs (bottom). Traversing results on selected dimensions showed that CUDOS was able to reuse latent dimensions for previous semantic factors, while learning new ones

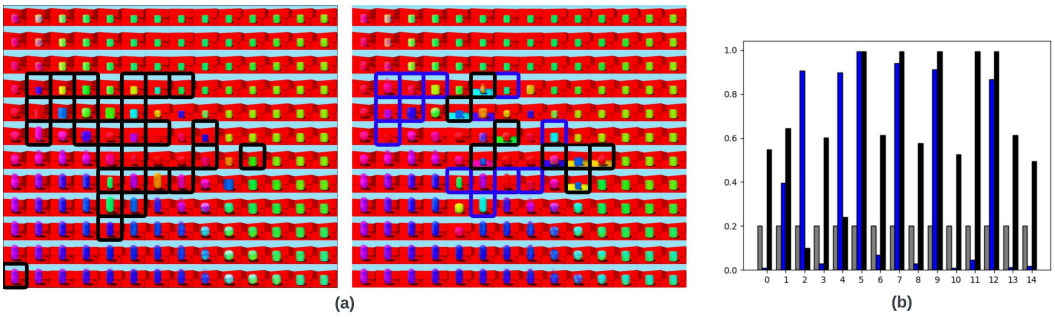

Figure 4.4: SOM results on split-3DShapes. (a) SOM prototypes learnt after the first (left) and second (right) data environments. Black boxes label prototypes associated with current (black) and replayed data samples (blue), while the rest of the prototypes are mixture-interpolated. (b) Average spike parameter α_k. Prototypes of the second data environment (black) shared active dimensions from the first data environment (blue) with added dimensions. Gray represents inactive dimensions.

related to the new attributes.

4.4.1 Benefits on downstream tasks

To evaluate the usefulness of the disentangled representations learned by CUDOS, we focused on shared tasks related to the continually-learned semantic factors shared among past and new data environments, including predicting the scale and orientation of 3DShapes, and predicting the X- and Y- positions of Moving-MNIST. For each task, we identified the corresponding active latent dimensions after learning in data environment 1, and continually trained linear regressors to predict the ground-truth labels using these active latent dimensions: the intuition is that, if new semantic factors are entangled into these shared dimensions, the performance of the regressor will decrease during continual learning. For comparisons, [1] and [45] were used as baselines and their results aggregated for 3Dshapes, and [45] was used as the baseline for Moving-MNIST. Table 4.2 summarizes the R^2 scores of each task obtained on new data along with their changes Δ from the R^2 scores obtained on old data prior to continual learning. As shown, CUDOS achieved the best final R^2 score for scale and orientation regression, along with a minimum drop in performance over the course of

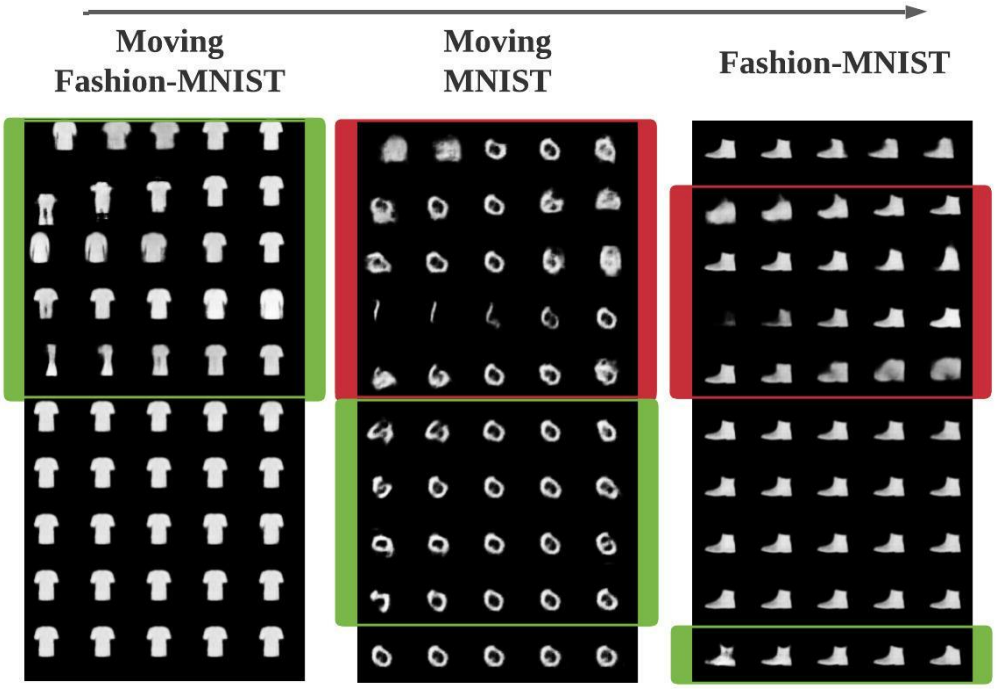

Figure 4.5: Continual learning of Moving Fashion-MNIST and MNIST.

continual learning. For Moving-MNIST, CUDOS and baselines achieved similar results on X-Y position regression, with in general less significant performance drop in comparison to 3DShapes. We argue that this is because X-Y position regression on a clean background is an easier task, than scale and orientation regression on a more complex data environment like 3DShapes. Additionally, Table 4.3 shows digit-classification performance on Moving-MNIST using active and inactive dimensions, which suggested that active semantic factors for the digit data environments are learned properly.

4.4.2 Ablation study

Table. 4.4 presents a detailed ablation study on the contribution brought by the different ingredients within CUDOS. We did not include results from VAE + $\mathcal{L}_{\text{old}}$ because they are

3DShapes	baselines	CUDOS
Scale $R^2 \uparrow$	0.05±0.07	**0.58±0.13**
Orientation $R^2 \uparrow$	0.08±0.05	**0.93±0.02**
Δ Scale $\uparrow$	-0.92±0.10	**-0.13±0.13**
Δ Orientation $\uparrow$	-0.81±0.15	**-0.00±0.01**
Moving-MNIST	baselines	CUDOS
X-pos $R^2 \uparrow$	0.67±0.09	**0.70±0.06**
Y-pos $R^2 \uparrow$	**0.75±0.05**	0.66±0.23
Δ X-pos $\uparrow$	0.053±0.13	**0.054±0.08**
Δ Y-pos $\uparrow$	**0.01±0.06**	-0.15±0.27

Table 4.2: R^2 score on new data & its change Δ from that on old data prior to continual learing.

Table 4.3: Continual digit classification accuracy (testing) based on active or inactive dimensions.

	moving fashion	moving MNIST	fashion
active	**0.74**	**0.85**	**0.86**
inactive	0.4	0.17	0.35

represented by the work of [45] as reported in Table. 4.1. As shown, the sparsity introduced by spike-and-slab distribution plays a significant role in improving the disentanglement ability of CUDOS. While SOM alone does not appear to improve the general disentangling ability of the model, it does seem to help disentangle new semantic factors from previously learned ones; more importantly, it is a necessary component for learning the relational structure of data to guide disentanglement, *i.e.*, for enabling $\mathcal{L}_{\text{newx}}$ in Eqn. (4.13). Indeed, the combined introduction of SOM and $\mathcal{L}_{\text{newx}}$ brings significant improvement in the ability of CUDOS to disentangle new semantic factors from previously learned ones, as it is evident in the improvement achieved in MIG-sup and $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$. More implementation details can be found in the following section.

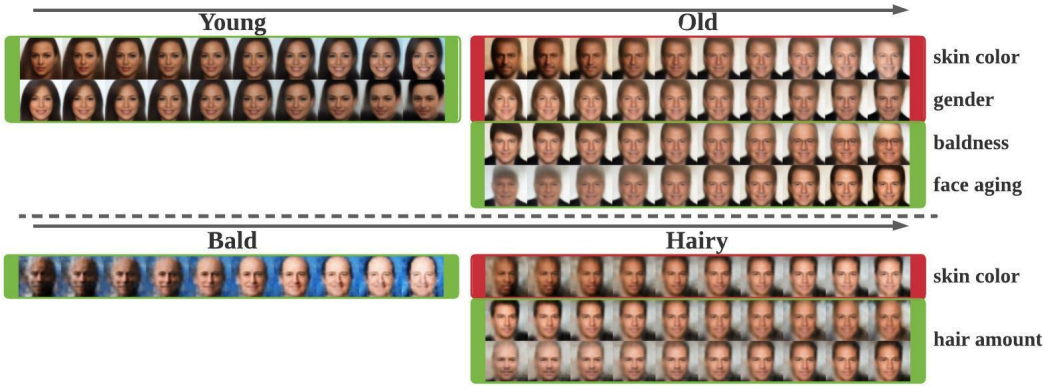

Figure 4.6: CelebA split by age (top row) and bangs (bottom row). The green boxes highlight the newly learned semantic factors, and the red boxes highlight the reused ones.

4.4.3 Traversing results on split-3DShapes and Moving-MNIST

Here we presented additional traversing results for baseline methods, along with their $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$ value for 3DShapes, and additional traversing results for baseline methods on Moving-MNIST.

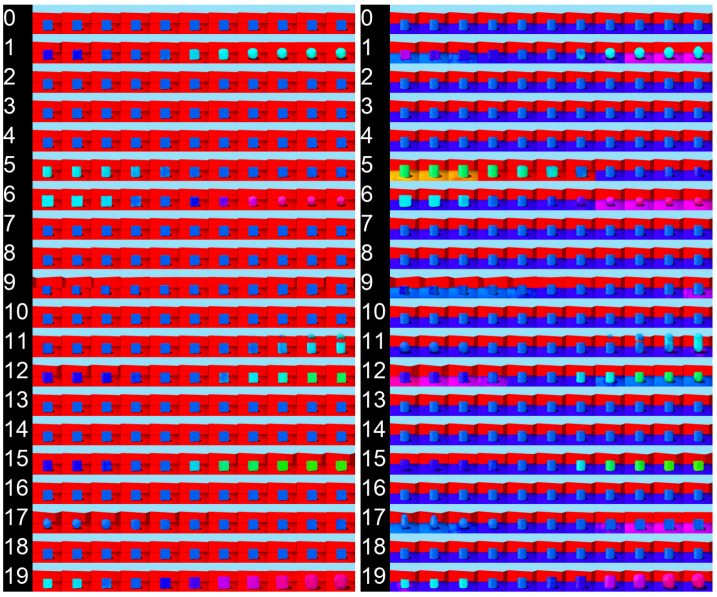

Figure 4.7: Burgess2018, $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$: 1.185

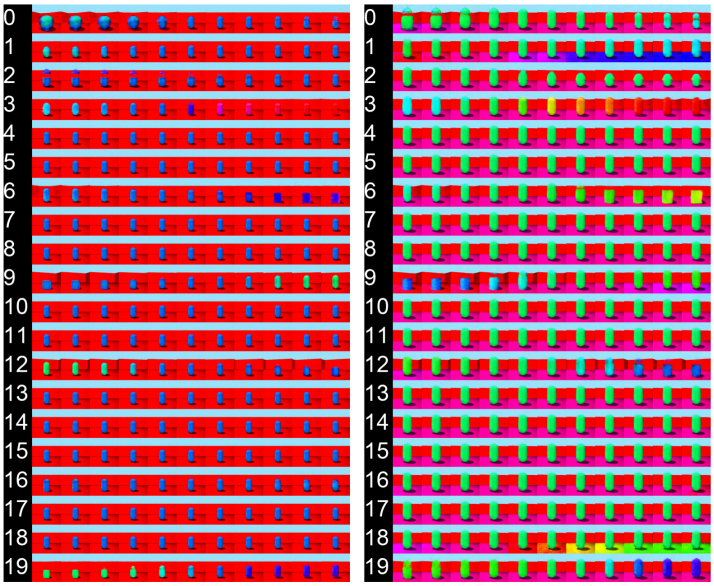

Figure 4.8: Achille2018, $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$: 0.602.

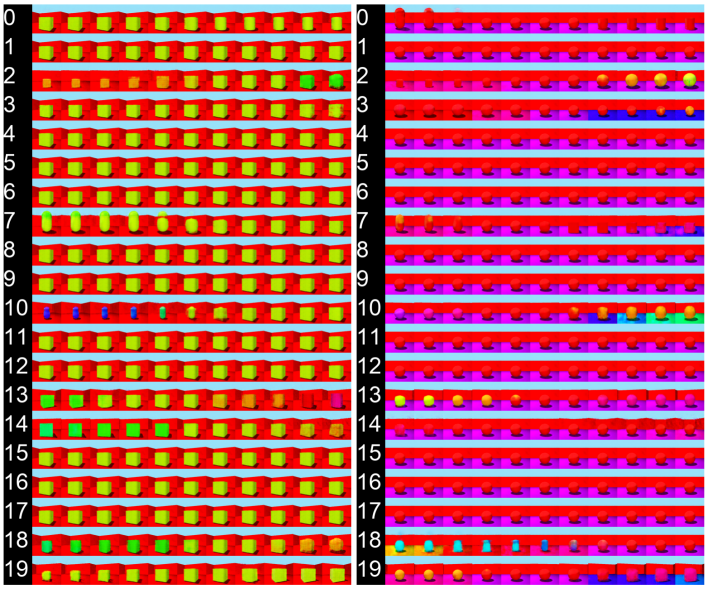

Figure 4.9: Ramapuram2020, $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$: 0.877

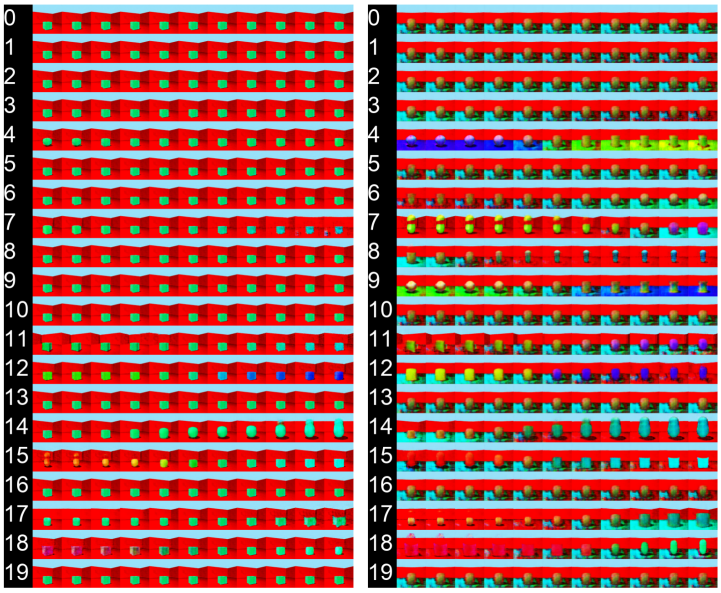

Figure 4.10: Continual VQ-VAE, $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$: 1.402

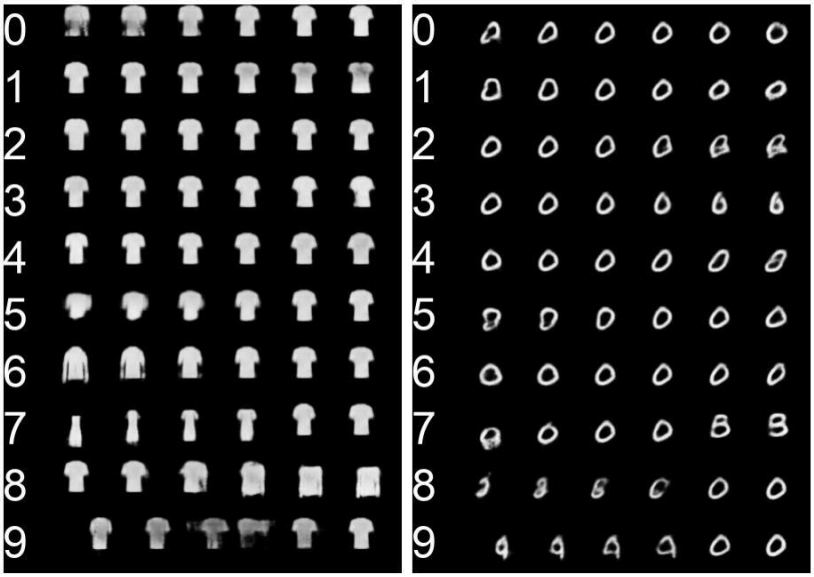

Figure 4.11: Burgess2018.

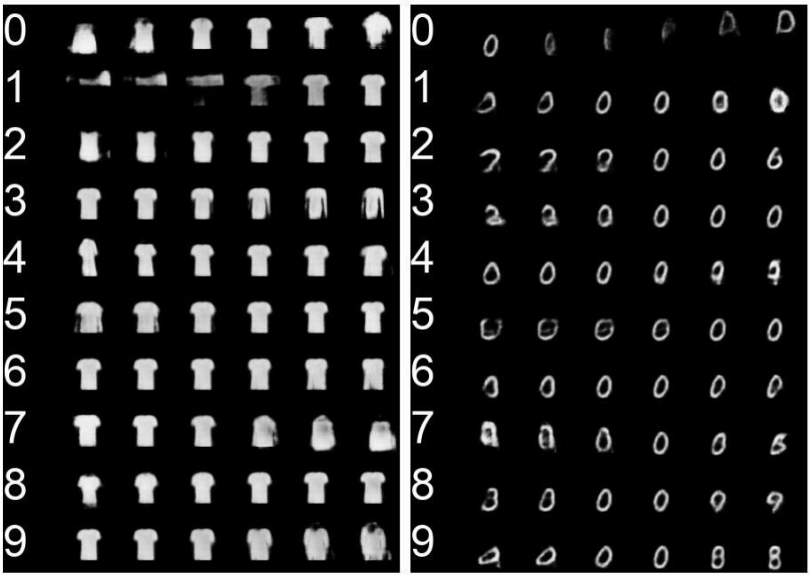

Figure 4.12: Achille2018.

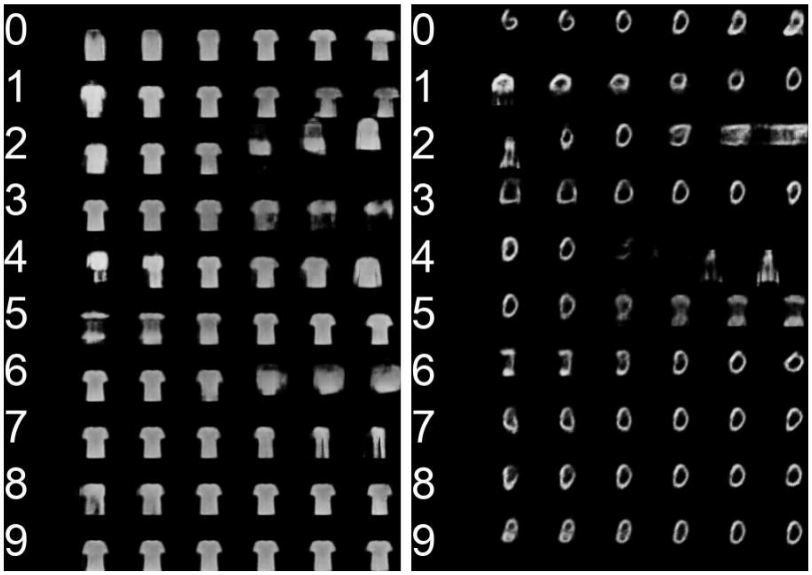

Figure 4.13: Ramapuram2020.

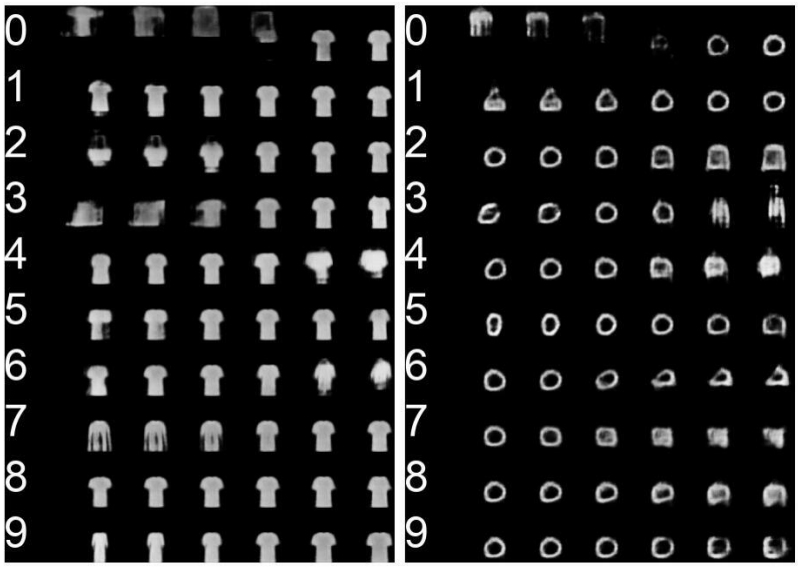

Figure 4.14: Continual VQ-VAE.

Table 4.4: Top: Continual digit classification accuracy (testing) based on active or inactive dimensions. Bottom: Ablation study. SS: spike-and-slab density. $\mathcal{L}_{\text{old}}/\mathcal{L}_{\text{new}}$: constraints on replayed/new data. *: VAE+SOM

Method	MIG ↑	MIG-sup ↑	$I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$ ↓	Δ recons ↓
*+$\mathcal{L}_{\text{old}}$	0.13±0.03	0.13±0.03	0.97±0.24	0.01±0.00
VAE+SS+$\mathcal{L}_{\text{old}}$	**0.30±0.12**	0.29±0.12	0.78±0.77	**0.01±0.10**
*+SS+$\mathcal{L}_{\text{old}}$	0.22±0.07	0.24±0.08	0.65±0.27	0.02±0.00
*+SS+$\mathcal{L}_{\text{old}}$+$\mathcal{L}_{\text{newz}}$	0.22±0.04	0.32±0.09	0.07±0.05	0.02±0.00
CUDOS (above+$\mathcal{L}_{\text{newx}}$)	0.24±0.05	**0.33±0.05**	**0.02±0.03**	0.02±0.01

4.4.4 SOM Prototypes

Additional SOM prototypes learnt for Moving-MNIST are shown in Fig 4.15. In SOM, the presented model was able to remember and accumulate old representations, e.g., fashion digits, while learning new representations, e.g., number digits, Additionally, the shared representations, e.g., the x-y translation, were changing smoothly among prototypes.

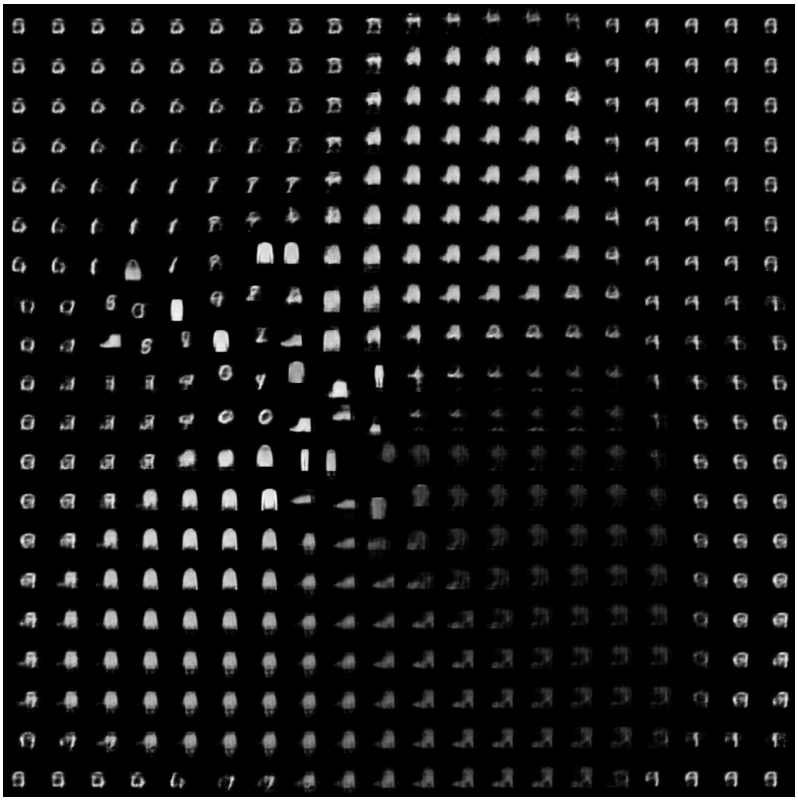

Figure 4.15: Prototypes learnt for continual learning on Moving-Fanshion-MNIST to Moving-MNIST to MNIST.

4.4.5 Data environments mapping on SOM during training

In Fig 4.16 and Fig 4.17 we presented additional data environments' mapping density on SOM during continual training. As shown in Fig 4.17, the presented model is able to reuse existing SOM prototypes while creating new prototypes.

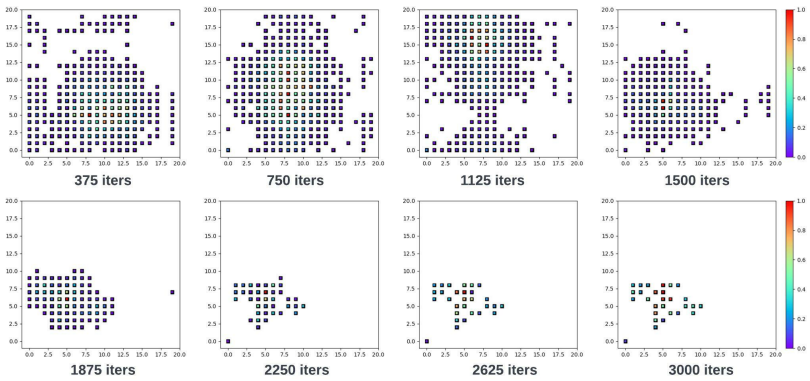

Figure 4.16: Mapping density on SOM of the first data environment of split-3DShapes during training.

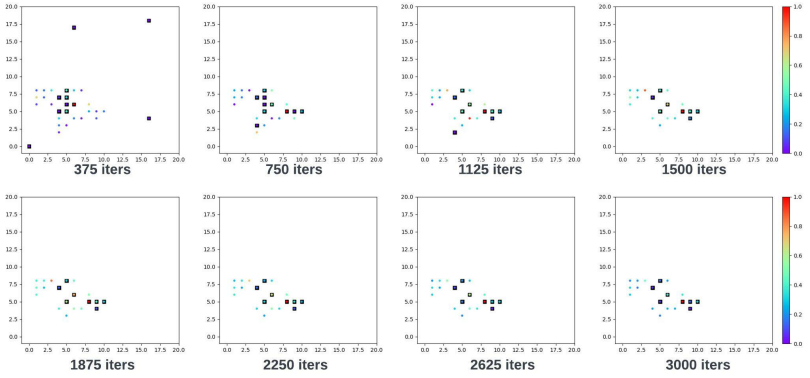

Figure 4.17: Mapping density on SOM of the second data environment of split-3DShapes during training. Dots without black-boundaries are replay data.

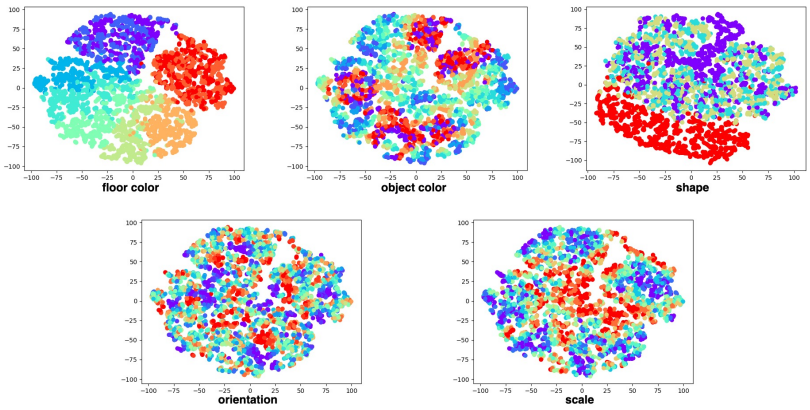

Figure 4.18: T-SNE plots for 3DShapes factors.

4.4.6 T-SNE for 3DShapes factors

Here we presented additional T-SNE plots for the learned latent representations of our model (colored by the generative factors of 3DShape) after continual learning of split-3DShape as shown in Fig 4.3. As shown in Fig.4.18, some factors such as floor color (10 classes) and shape (4 classes) formed good clusters while some more difficult factors such as orientation (15 classes) and scale (8 classes) formed fewer discriminative clusters.

4.4.7 Additional experiments of split-3DShapes

Here we first presented additional experiments of split-3DShapes with two splitting versions that each has three data environments. The first version, as shown in Fig 4.19, is starting with no floor and wall color variations and then continually added them. The second version, as shown in Fig 4.20, was starting with no scale and wall color variations and then continually added them. For most latent dimensions during continual learning, CUDOS was able to reuse those corresponding to previously learned semantic factors and disentangle new ones into previously inactive dimensions.

Next, we presented a reversed sequence of split-3DShape, where the first data environment includes all semantic factors and the second one includes a subset that doesn't have the floor

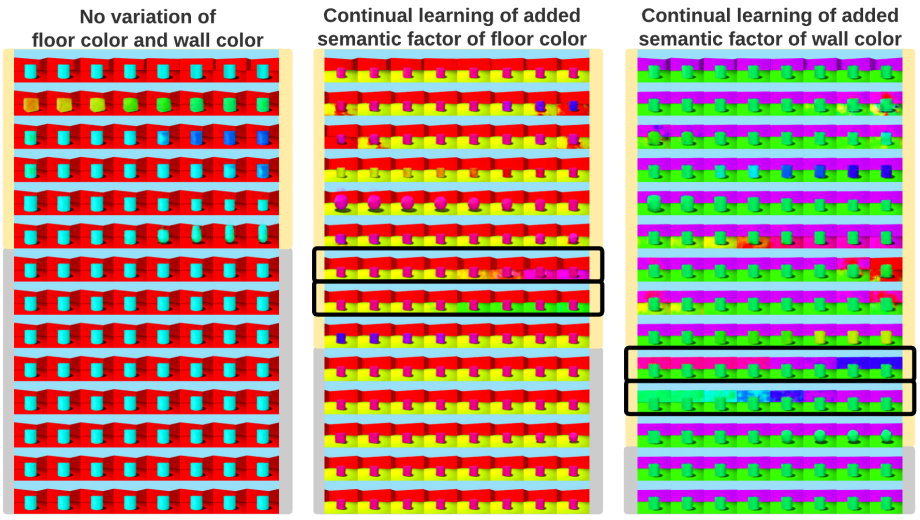

Figure 4.19: Continually learning a split version of 3DShapes where the variations of floor and wall colors were absent initially and appeared later. Each row of images is traversing each latent dimension after training on data environment titled above. Black bounding boxes annotate where the new semantic factors are learnt.

color variations (only red remained). As shown in fig 4.21, our model was able to reuse all previous semantic factors without any expansion of latent space. our model was able to reuse all previous semantic factors without any expansion of latent space. Furthermore, our model was able to remember how to generate the variations of the missing floor color in the second data environment after continual training.

4.4.8 Discussion of differences between static and continual disentanglement

Continually disentangling sequentially-arrived semantic factors is fundamentally different from disentangling a static dataset where the model sees all semantic factors at once. The challenges cannot be addressed by state-of-the-art (SOTA) disentangling VAEs such as FactorVAE [25] and TC-VAE [9]. The continual setting sees significant challenges such as

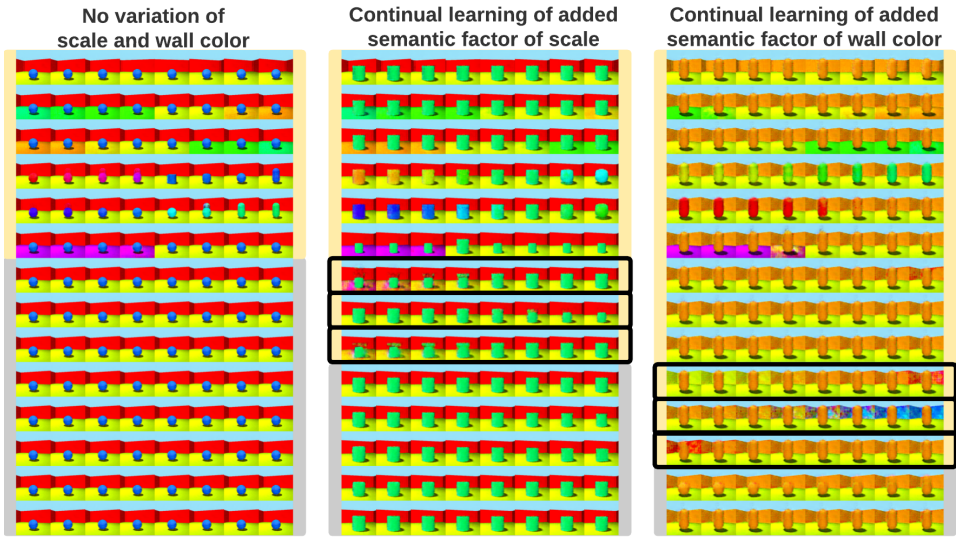

Figure 4.20: Continually learning a split version of 3DShapes where the variations of scale and wall colors were absent initially and appeared later. Each row of images is traversing each latent dimension after training on data environment titled above. Black bounding boxes annotate where the new semantic factors are learned.

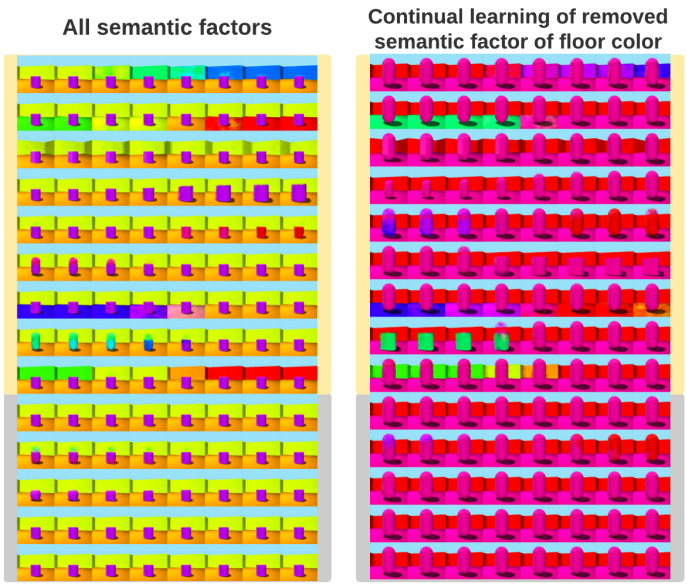

Figure 4.21: Continually learning a reversed sequence of split-3DShape where all semantic factors were presented in the beginning but later the variation of floor color was removed (only red remained). Each row of images is traversing each latent dimension after training on data environment titled above.

forgetting (including forgetting learned semantic factors) and, as in other continual learning problems, cannot be expected to see similar disentanglement performance to those reported on static settings. That's why the non-continual baselines in Table 4.1, including naive VAE, naive TC-VAE, and [8], all showed weaker performance than what would have been expected in a static learning setting. Furthermore, because these models are not designed for continual learning, they will also face challenges such as catastrophic forgetting. We can take a closer look at this by taking naive TC-VAE versus continual TC-VAE (with generative replay mechanism) as an example. As shown in Table 4.1, naive TC-VAE achieved overall better MIG and MIG-sup scores because they can forget about previous data and focus on disentangling the new data (in split 3DShapes, the two sequentially-presented datasets share many latent factors but one). Therefore, forgetting previous semantic factors did not create a large performance drop in MIG and MIG-sup scores. This catastrophic forgetting however is reflected in the reconstruction loss, which increased from 2365 to 3207 (around 35% change) on the previous data. In addition, the $I(\mathbf{z}_{\text{past}}; \text{factors}_{\text{new}})$ metric further shows that naive TC-VAE is not able to separate new semantic factors from previously-learned latent dimensions, but rather achieved relatively high disentangling by simply forgetting previous factors and learning on the new data itself. This is similar to naive VAE's performance.

By extending TC-VAE to a continual learning setting with generative replay, the continual TC-VAE was able to remember how to reconstruct the previous data, where the reconstruction loss only increased from 2376 to 2395 (0.7% change). However, its disentanglement performance including MIG and MIG-sup dropped. This demonstrates the aforementioned challenges that disentangling sequentially-arrived semantic factors (while remembering the previous factors at the same time) is fundamentally different from disentangling all seen factors at once. It also shows that such challenges cannot be addressed by either naive SOTA disentangling VAEs, or simply extending these SOTA disentangling VAEs into a continual setting (via standard techniques such as generative replay). We believe these provide further evidence for the contribution of the presented work.

4.4.9 Hyper-parameters settings and implementation strategy

We set $\gamma_1 = 0.25, \gamma_2 = 1, \gamma_1 = 0.35$, $b = 10$ in all experiments. Snapshot of the model is updated every $\tau = 1500$ iteration step. Regarding weights in Eqn. (4.14), in our experiments, generally, we are trying to avoid certain parts of the loss function becoming too large or small, and we found a rule-of-thumb weight value described above across different datasets.

The most tricky part is the weighting for the constraints of old data and new data. We found a slightly higher weight on new data can achieve better continual disentanglement results. We reason that the constraint on new factors is more difficult compared with reconstructing old data, and therefore emphasizing more on that (higher weight) can be helpful. In general, we didn't find obvious differences within around 30% percent changes of each weight. Putting them too high (larger than the ELBO term) will affect the original continual learning of new data. Metrics in Table. 4.1 are calculated in a setting where the boundary of data environments is known.

4.5 Conclusion

In this chapter, we demonstrated that an overlooked key ingredient to continual unsupervised learning of representations is to exploit the relational structure of data based on their underlying active semantic factors. We extended our previous research on progressive presentation learning to continual representation learning. We explained and demonstrated the progressive learning strategies proposed in chapter 3 and other VAEs for static data environments failed to naively apply to continual learning scenarios. Specifically, we enhanced the idea of growing the capacity of the model from progressive learning by developing a Bayesian-SOM latent space to not only continual learning but also accumulating representations. We presented CUDOS, a novel VAE with self-organizing spike-and-slab mixtures, to address the challenges in continual representation learning and disentangling.

Chapter 5

Conclusion and Future Works

In this dissertation, we pointed out and reviewed the important aspects of unsupervised representation learning and its challenges. We approached the challenges by asking three research questions for improving unsupervised representation learning, especially in the aspects of progressive and continual learning and disentangling. We presented several advanced unsupervised representation learning methods with comprehensive experiments to demonstrate their performance in both typical static data environments and dynamic data environments.

5.0.1 Future Works

Robust optimization of continual representation learning: We plan to advance our research progress for continual unsupervised learning of representations, especially in reusing, expanding, and continually disentangling learned semantic factors across data environments. In this dissertation, we tackled this problem via a novel definition of the shared representations of the streaming data environment based on their active dimensions, such that the shared representations can be optimized instead of heuristically determined. In the future, we plan to advance this direction by investigating information-controllable representation learning by explicitly defining and optimizing the amount of information during progressive and continual representation learning for better interpretability and trustworthy AI.

Fair and distributionally-robust representation learning: Underrepresented subgroups are often experiencing the biased performance of machine learning models, and the differential performance of deep learning models in well- and under-represented subgroups remains relatively under-explored. Distributionally robust optimization (DRO) was shown to be an effective approach that minimizes the worst-case performance of the model. Most existing approaches however require the subgroups to be known. Furthermore, the coexistence of multiple subgroup attributes (e.g., gender and ethnicity) in the same data leaves an open question as to how addressing disparity for one subgroup may benefit or harm other types of under-represented subgroups. We plan to explore the role of unsupervised representation learning in improving subgroup robustness without subgroup supervision.

Comprehensive metrics and evaluations for continual disentanglement: The evaluations of disentanglement have been a long-standing problem and, as far as we know, all existing disentanglement metrics are each focused on a specific perspective of disentanglement. In this dissertation, we proposed a new metric MIG-sup that supplements information-based metrics and reported the first disentanglement performances in continual settings. In the future, we plan to investigate novel metrics and experimental protocols for more comprehensive evaluations of continual representation learning and disentangling, such as how existing representations remain the same, and how new representations are separated from old ones.

Learn minor disease-specific representations: Compared to many visual benchmarks, disease-specific factors in medical images may be buried by other more significant factors in terms of contribution to pixels distribution (e.g., a lung nodule area vs. torso shape & orientation). This, we believe, may explain the relatively limited progress of unsupervised representation learning in medical images despite its recent traction in other visual domains. In the future, we plan to extend our works of representation learning to resolve the unique challenges of medical areas, especially learning and capturing minor disease-specific features.

Bibliography

[1] Alessandro Achille, Tom Eccles, Loic Matthey, Chris Burgess, Nicholas Watters, Alexander Lerchner, and Irina Higgins. Life-long disentangled representation learning with cross-domain latent homologies. In Advances in Neural Information Processing Systems, pages 9873–9883, 2018.

[2] Philip Bachman. An architecture for deep, hierarchical generative models. In Advances in Neural Information Processing Systems, pages 4826–4834, 2016.

[3] Pouya Bashivan, Martin Schrimpf, Robert Ajemian, Irina Rish, Matthew Riemer, and Yuhai Tu. Continual learning with self-organizing maps. In Continual Learning Workshop, 32nd Conference on Neural Information Processing Systems, 2018.

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8):1798–1828, 2013.

[5] Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In International Conferences on Machine Learning, 2015.

[6] Zalán Borsos, Mojmír Mutnỳ, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. In Advances in Neural Information Processing Systems, 2020.

[7] Chris Burgess and Hyunjik Kim. 3d shapes dataset. https://github.com/deepmind/3dshapes-dataset/, 2018.

[8] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in beta-vae. In Advances in Neural Information Processing Systems, 2017.

[9] Ricky TQ Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In Advances in Neural Information Processing Systems, 2018.

[10] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. In International Conference on Learning Representations, 2017.

[11] Emily L Denton, Soumith Chintala, Arthur Szlam, and Rob Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. In Advances in Neural Information Processing Systems, pages 1486–1494, 2015.

[12] Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In International Conference on Learning Representations, 2018.

[13] Harrison Edwards and Amos Storkey. Towards a neural statistician. In International Conference on Learning Representations, 2017.

[14] Jeffrey L Elman. Learning and development in neural networks: The importance of starting small. Cognition, 48(1):71–99, 1993.

[15] Florent Forest, Mustapha Lebbah, Hanene Azzag, and Jérôme Lacaille. Deep embedded som: Joint representation learning and self-organization. Reconstruction, 500:500, 2000.

[16] Vincent Fortuin, Matthias Hüser, Francesco Locatello, Heiko Strathmann, and Gunnar Rätsch. Som-vae: Interpretable discrete representation learning on time series. In International Conference on Learning Representations, 2019.

[17] Alexander Gepperth and Benedikt Pfülb. Gradient-based training of gaussian mixture models for high-dimensional streaming data. Neural Processing Letters, 53(6):4331–4348, 2021.

[18] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. Draw: A recurrent neural network for image generation. In International Conference on Machine Learning, 2015.

[19] Prashnna Gyawali, Zhiyuan Li, Cameron Knight, Sandesh Ghimire, B Milan Horacek, John Sapp, and Linwei Wang. Improving disentangled representation learning with the beta bernoulli process. In 2019 IEEE International Conference on Data Mining (ICDM), pages 1078–1083. IEEE, 2019.

[20] Luke B Hewitt, Maxwell I Nye, Andreea Gane, Tommi Jaakkola, and Joshua B Tenenbaum. The variational homoencoder: Learning to learn high capacity generative models from few examples. In The Conference on Uncertainty in Artificial Intelligence, 2018.

[21] 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, 2017.

[22] Daniella Horan, Eitan Richardson, and Yair Weiss. When is unsupervised disentanglement possible? In Advances in Neural Information Processing Systems, volume 34, pages 5150–5161, 2021.

[23] Cheng Hao Jin, Gouchol Pok, Yongmi Lee, Hyun-Woo Park, Kwang Deuk Kim, Unil Yun, and Keun Ho Ryu. A som clustering pattern sequence-based next symbol prediction method for day-ahead direct electricity load and price forecasting. Energy conversion and management, 90:84–92, 2015.

[24] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In International Conference on Learning Representations, 2018.

[25] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In International Conference on Machine Learning, 2018.

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In International Conference on Learning Representations, 2014.

[27] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In Advances in neural information processing systems, pages 4743–4751, 2016.

[28] Teuvo Kohonen. The self-organizing map. Proceedings of the IEEE, 78(9):1464–1480, 1990.

[29] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. Variational inference of disentangled latent concepts from unlabeled observations. arXiv preprint arXiv:1711.00848, 2017.

[30] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[31] José Lezama. Overcoming the disentanglement vs reconstruction trade-off via jacobian supervision. In International Conference on Learning Representations, 2019.

[32] Chongxuan Li, Jun Zhu, and Bo Zhang. Learning to generate with memory. In International Conference on Machine Learning, pages 1177–1186, 2016.

[33] Zhiyuan Li, Xiajun Jiang, Ryan Missel, Prashnna Kumar Gyawali, Nilesh Kumar, and Linwei Wang. Continual unsupervised disentangling of self-organizing representations. In International Conference on Learning Representations, 2023.

[34] Zhiyuan Li, Jaideep Vitthal Murkute, Prashnna Kumar Gyawali, and Linwei Wang. Progressive learning and disentanglement of hierarchical representations. In International Conference on Learning Representations, 2020.

[35] Nan Liu, Jinjun Wang, and Yihong Gong. Deep self-organizing map for visual classification. In 2015 international joint conference on neural networks (IJCNN), pages 1–6. IEEE, 2015.

[36] Ran Liu, Mehdi Azabou, Max Dabagia, Chi-Heng Lin, Mohammad Gheshlaghi Azar, Keith Hengen, Michal Valko, and Eva Dyer. Drop, swap, and generate: A self-supervised approach for generating neural activity. In Advances in Neural Information Processing Systems, volume 34, pages 10587–10599, 2021.

[37] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. A sober look at the unsupervised learning of disentangled representations and their evaluation. Journal of Machine Learning Research, 2020.

[38] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In International Conference on Learning Representations, 2021.

[39] Laura Manduchi, Matthias Hüser, Julia Vogt, Gunnar Rätsch, and Vincent Fortuin. Dpsom: Deep probabilistic clustering with self-organizing maps. arXiv preprint arXiv:1910.01590, 2019.

[40] Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

[41] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In International Conference on Learning Representations, 2018.

[42] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Advances in Neural Information Processing Systems, 2017.

[43] German I Parisi, Jun Tani, Cornelius Weber, and Stefan Wermter. Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization. Frontiers in Neurorobotics, 12:78, 2018.

[44] Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis. Lifelong generative modeling. arXiv preprint arXiv:1705.09847, 2017.

[45] Jason Ramapuram, Magda Gregorova, and Alexandros Kalousis. Lifelong generative modeling. Neurocomputing, 404:381–400, 2020.

[46] Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. In Advances in Neural Information Processing Systems, pages 7647–7657, 2019.

[47] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In International conference on machine learning, pages 1278–1286. PMLR, 2014.

[48] Travers Rhodes and Daniel Lee. Local disentanglement in variational auto-encoders using jacobian l_1 regularization. In Advances in Neural Information Processing Systems, volume 34, pages 22708–22719, 2021.

[49] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In Advances in Neural Information Processing Systems, pages 2990–2999, 2017.

[50] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In Advances in neural information processing systems, pages 3738–3746, 2016.

[51] Michalis Titsias and Miguel Lazaro-Gredilla. Spike and slab variational inference for multi-task and multiple kernel learning. In Advances in Neural Information Processing Systems, 2011.

[52] Francesco Tonolini, Bjørn Sand Jensen, and Roderick Murray-Smith. Variational sparse coding. In Uncertainty in Artificial Intelligence, pages 690–700. PMLR, 2020.

[53] Michael Tschannen, Olivier Bachem, and Mario Lucic. Recent advances in autoencoder-based representation learning. In Bayesian Deep Learning Workshop, NeurIPS, 2018.

[54] Yifan Wang, Federico Perazzi, Brian McWilliams, Alexander Sorkine-Hornung, Olga Sorkine-Hornung, and Christopher Schroers. A fully progressive approach to single-image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 864–873, 2018.

[55] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747, 2017.

[56] Fei Ye and Adrian G Bors. Learning latent representations across multiple data domains using lifelong vaegan. In European Conference on Computer Vision, pages 777–795. Springer, 2020.

[57] H Yin and NM Allinson. Bayesian self-organising map for gaussian mixtures. IEE Proceedings-Vision, Image and Signal Processing, 148(4):234–240, 2001.

[58] Hufun Yin and Nigel M Allinson. Bayesian learning for self-organising maps. Electronics letters, 33(4):304–305, 1997.

[59] Julian Zaidi, Jonathan Boilard, Ghyslain Gagnon, and Marc-André Carbonneau. Measuring disentanglement: A review of metrics. IEEE Transactions on Neural Networks and Learning Systems, 2022.

[60] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In European Conference on Computer Vision, pages 818–833. Springer, 2014.

[61] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(8):1947–1962, 2018.

[62] Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. In International Conference on Machine Learning, pages 4091–4099, 2017.