Latent Space Interpretability for Autoencoders in Financial Crime Detection

by

Elisa Rossi

To obtain the degree of Master of Science in Applied Mathematics at TU Delft,
to be defended publicly on Friday October 3, 2025 at 11:00 AM.

Student number:	5974585	
Project duration:	January 6, 2025 – October 3, 2025	
Thesis committee:	N. Parolya,	TU Delft, Supervisor
	E. Haasdijk,	Deloitte, External Supervisor
	A. Papapantoleon,	TU Delft

TU Delft Deloitte.

Acknowledgements

I am truly grateful to the people I worked with at Deloitte and at the bank for their time, advice, and encouragement over these past months. In particular, my sincere thanks go to Danu Thung and Evert Haasdijk for trusting me and my choices, often more than I trusted myself. This made me braver and gave me the confidence to grow in ways that reach beyond this project.

I would also like to thank Professor Nestor Parolya for his support, and Professor Antonis Papapantoleon for taking the time to be part of my thesis committee.

Finally, I owe much to my friends and family for their patience and understanding, even when I've been hard to follow. Their presence mattered more than they realize.

Elisa Rossi
Delft, September 2025

Abstract

Financial crime is growing in scale and complexity, increasing the need for robust monitoring. Variational Autoencoders offer compact representations of transaction behavior that can support anomaly detection and related use cases in this domain, yet their adoption remains limited due to the lack of interpretability of their latent spaces.

To address this challenge, we formalize interpretability by quantifying the relationship between latent dimensions and aspects of transaction behavior, using explicitness, modularity, and compactness as complementary metrics. Based on these measures, we develop a framework and apply it in controlled experiments that vary regularization strength and latent dimensionality, in order to understand their influence on the structure and interpretability of latent spaces.

We find that weak regularization preserves detail but compromises modularity and compactness, intermediate values progressively improve these properties, until strong regularization forces the latent space to collapse. Latent dimensionality further shapes both the level of detail preserved and the conditions under which meaningful structure emerges. The results show that there is no universal optimum, but rather parameter regimes suited to different priorities. The framework offers a systematic way to align model design with these priorities, providing both empirical insights and a general tool to support the adoption of VAEs in financial crime detection.

Contents

List of Figures

List of Tables

1

Introduction

Financial crime has grown into a problem of global scale. Trillions of euros are estimated to be laundered each year, yet only a small fraction of these flows is ever detected [1]. Banks are expected to act as the first line of defense, investing heavily in compliance departments and monitoring systems. In the Netherlands alone, the banking sector spends nearly 1.4 billion euros annually on anti money laundering compliance, supported by tens of thousands of employees. Despite this massive investment, the effectiveness of these systems remains limited: detection rates are low, false positives overwhelm compliance teams, and criminals continuously adapt their methods to exploit new vulnerabilities.

These shortcomings open the door to new opportunities, and machine learning is one of the most promising directions. In particular, unsupervised methods that learn patterns directly from data are well suited to this context, as they adapt to evolving behaviors and are not limited by our current knowledge of financial crime.

Variational autoencoders (VAEs) are one such method [2]. They compress high-dimensional transaction data into a latent representation that captures the most relevant patterns. These representations can serve as compact descriptors of account behavior, making them useful for tasks such as risk profiling, clustering of similar accounts, and detection of suspicious activity. However, the latent variables learned by VAEs are not inherently interpretable, meaning that decisions cannot be directly linked to the aspects of behavior that drive them. This lack of interpretability makes banks hesitant to adopt such models: even effective tools cannot be used if their decisions cannot be understood, justified, and audited.

Disentangled representation learning aims to bridge this gap by structuring the latent space so that each latent variable corresponds to a distinct and meaningful aspect of the data [3]. This allows model outputs to be linked to interpretable behavioral factors, and helps identify which patterns matter most when distinguishing illicit from legitimate activity. For this idea to move from theory to practice, however, we need ways to measure disentanglement that allow us to refine models and steer them toward more interpretable representations.

Building on this motivation, the aim of this thesis is to explore how the interpretability of VAEs can be assessed and improved when applied to account-level transaction data. In particular, the work focuses on the β-Variational Autoencoder (β-VAE), a variant of the VAE that promotes disentanglement [4]. The research was conducted in collaboration with Deloitte and a Dutch bank, with access to real data, and is guided by two research objectives:

RO1: *To design a framework that quantifies the extent to which individual latent dimensions of a β-Variational Autoencoder capture specific behaviors in transaction data.*

RO2: *To analyze how changes to hyperparameters affect the interpretability of latent representations learned by a β-Variational Autoencoder.*

By addressing these objectives, the thesis makes two main contributions. First, it develops a framework for assessing how latent representations relate to any chosen set of behavioral descriptors, intended to guide model design and support adoption of VAEs in financial crime detection. Second, it applies this framework to a selected set of descriptors, revealing how different hyperparameter choices shape trade-offs in the organization and interpretability of latent spaces.

The remainder of this thesis is organized as follows. Chapter 2 introduces the challenge of detecting financial crime and reviews existing methods, with a focus on autoencoders and their interpretability. Chapter 3 covers the foundations of variational autoencoders. It introduces latent variable models, explains how variational inference makes them tractable, and shows how VAEs combine these ideas in a neural network framework. Chapter 4 focuses on disentangled representation learning, describing its motivation, formal properties, and the role of β-VAEs. It further introduces metrics for evaluating disentanglement, with particular attention to DCI, which serves as a starting point for this thesis. Chapter 5 describes the experimental setup, covering data, preprocessing, model design, and the interpretability framework. Chapter 6 reports the results, including an analysis of training dynamics, activity of latent dimensions, and interpretability. Chapter 7 concludes with the main findings, limitations, and opportunities for future research.

2

Financial Crime Detection

In this chapter, we provide an overview of financial crime detection, tracing the problem from its broader context to the technical methods used in practice. Section 2.1 introduces financial crime, outlining its scale, impact, and the regulatory responsibilities of financial institutions. Section 2.2 reviews detection approaches, beginning with traditional rule-based systems and moving toward machine learning methods. Section 2.3 narrows the focus to autoencoders and related models, emphasizing how their latent spaces can be used in financial crime detection. Finally, Section 2.4 highlights the importance of interpretability in latent representations, a theme that motivates the methods explored in later chapters.

2.1. Financial Crime

Financial crime is a broad term that refers to illegal activities carried out for financial gain through the misuse of financial systems [5]. Examples include fraud, bribery, corruption and cybercrime. While these crimes take many different forms, they share the common goal of generating illicit profits that can be hidden, moved, and reinvested in the economy.

At the center of this lies money laundering, the process of disguising the criminal origin of funds so they can be used without raising suspicion. Laundering is more than just illegal money changing hands; it is what makes crime profitable. By giving criminals a way to legitimize their earnings, it fuels and sustains activities such as drug trafficking, human trafficking, and organized crime [6]. Its impact has intensified in recent years with advances in technology and global connectivity [7]. Mobile banking and digital payments have opened up financial services to billions of people, making transactions faster, cheaper, and more convenient than ever before. At the same time, these innovations have created new vulnerabilities that criminals can exploit. According to the United Nations, between 800 billion and 2 trillion US dollars are laundered globally each year, equivalent to 2 to 5 percent of global GDP, yet less than 1 percent of this is ever intercepted [1, 8].

2.1.1. Banks as Gatekeepers of the Financial System

Because of their central role in the financial system, financial institutions are seen as the first line of defense against financial crime. In the Netherlands, this responsibility is formalized in the Money Laundering and Terrorist Financing (Prevention) Act (Wwft) [9]. The Wwft applies to a wide range of institutions, including insurers and payment service providers, but this thesis focuses specifically on banks. Their obligations extend across the entire client life cycle and require the implementation of comprehensive controls into their daily operations.

The process begins with client onboarding, where banks verify the identity of customers and

assess the risk level associated with the client relationship. Screening tools are used to check clients against sanction lists, politically exposed persons databases, and adverse media sources.

Once a client relationship is established, banks must conduct ongoing monitoring. Transactions are continuously reviewed to make sure they fit the customer's expected profile, and unusual patterns are flagged for further investigation. Clients are also subject to periodic reviews, where their risk exposure is reassessed, with higher risk clients reviewed more frequently. In addition, event-driven reviews may be triggered by sudden changes in behavior or new information, such as adverse media coverage.

If suspicious activity is detected, banks are required to file a Suspicious Activity Report (SAR) to the Financial Intelligence Unit. In severe cases, such as confirmed involvement in financial crime, banks may terminate the client relationship altogether.

To meet these obligations, Dutch banks have built large compliance departments and invested heavily in monitoring systems. According to De Nederlandsche Bank, the sector spends roughly 1.4 billion euros each year on anti-money laundering compliance [10]. Despite this effort, financial institutions continue to face major challenges, as criminals adjust their methods to take advantage of new vulnerabilities.

2.2. Detection Approaches in Practice

Ongoing monitoring is carried out through detection systems that review transactions and client behavior to spot signs of financial crime. These systems can take different forms: some are built on fixed rule sets, others rely on machine learning, and many combine both approaches in hybrid frameworks.

2.2.1. Rule-Based Systems

Rule-based systems have long been the standard in financial crime detection and still serve as the backbone of many frameworks today. These systems are designed to translate regulatory requirements, historical data, and known criminal patterns into a set of if-then conditions. Every transaction is checked against these conditions, and an alert is raised whenever one is violated. Typically, rules target well-known risk indicators such as unusually large transactions, payments involving high risk jurisdictions, or connections to sanctioned entities. Rules can also capture unusual account behavior, for example when a dormant account suddenly becomes active or when transactions are inconsistent with the customer's normal financial profile.

A key strength of these systems is their transparency. Because alerts are generated from predefined rules, they can be explained clearly, and the resulting audit trail allows banks to demonstrate to regulators which risks are being addressed and how. They are also valued for their simplicity, relatively low cost, and flexibility, since thresholds and scenarios can be adjusted to keep up with new regulations or emerging typologies.

These strengths, however, come with significant trade-offs that limit their effectiveness. Static, expert defined rules can only detect patterns that are already known. Criminals take advantage of this by structuring their activity to remain just below thresholds or by combining methods in ways that rules do not anticipate. Equally important is their lack of contextual understanding. Because they rely on simple binary logic, these systems struggle to capture subtle changes in behavior or broader relationships between transactions. This narrow focus means they cannot connect weak signals that, when viewed together, reveal criminal activity. Another well-known drawback is the high rate of false positives. Legitimate transactions may trigger alerts simply because they breach a threshold. For large institutions processing millions of transactions each day, this results in an enormous volume of alerts, most of which are harmless. Investigating them consumes significant compliance resources and reduces the capacity to focus on cases that truly matter.

These challenges highlight the need for data driven approaches that are more flexible and capable

of adapting to evolving risks.

2.2.2. Machine Learning Systems

Financial crime teams today have access to more information than ever before, collected across the entire customer lifecycle and enriched with external data sources. The real challenge is not the lack of data, but the difficulty of extracting meaningful insight from it.

Industry analyses, including Deloitte's NextGen AML report [11], highlight that the future of financial crime detection requires moving beyond static, rule-based systems toward more adaptive and data driven approaches. Machine learning offers a promising path forward, as it can learn directly from data rather than relying solely on predefined rules, making it possible to detect subtle and evolving patterns of behavior that traditional approaches often miss.

There are different types of machine learning, suited to different kinds of data and objectives. *Supervised learning* relies on labeled data from past investigations, where transactions or accounts have been classified as either suspicious or legitimate. By learning from these examples, models capture patterns that separate normal from illicit activity and can then use them to assess new cases. *Unsupervised learning* offers a different perspective by searching for structure in data rather than relying on labels. Clustering methods group customers or transactions with similar characteristics, making unusual cases easier to detect. Outlier detection algorithms highlight cases that deviate from typical behavior, while graph-based models analyze transaction networks to identify suspicious flows. These methods are particularly useful for identifying new or emerging forms of financial crime that supervised models, trained only on past cases, would fail to capture, and they are especially valuable in settings where labeled examples are scarce and difficult to obtain.

Beyond these approaches, deep learning models such as autoencoders and variational autoencoders have gained attention. While also unsupervised, they differ from traditional methods by focusing on learning compact latent representations of high-dimensional data. These representations can then be used in multiple applications in financial crime, including anomaly detection, clustering, and risk profiling. The mathematical formulation and architecture of these models will be introduced in Chapter 3, while the next section reviews how they have been applied in the financial crime literature.

2.3. Autoencoders and Variational Autoencoders

An autoencoder (AE) is composed of an encoding and a decoding network. The encoder compresses the input data into a lower dimensional latent representation, and the decoder tries to reconstruct the original input from this compressed form. The model is trained to minimize reconstruction error, that is, the difference between each input and its reconstruction. Because the latent space is smaller than the input space, only the most relevant information can be preserved while noise and irrelevant variation are ignored, so the latent representation serves as a concise summary of the original data. Principal component analysis (PCA) is a simple linear version of this idea, while autoencoders extend it with nonlinear mappings that capture more complex patterns.

Several regularized autoencoder variants have been introduced to learn richer and more expressive latent representations. A sparse autoencoder (SAE) promotes sparsity in hidden activations by keeping only the top K active units [12]. A denoising autoencoder (DAE) trains on corrupted inputs and learns to reconstruct the clean versions, yielding representations that are robust to small variations [13]. A contractive autoencoder (CAE) enforces local stability by including a penalty that limits variation in the code under small changes of the input [14]. A variational autoencoder (VAE) extends this family by introducing a probabilistic formulation. By imposing a prior distribution on the latent variables, VAEs encode inputs as probability distributions rather than fixed points [2]. This regularization encourages a smooth and meaningfully organized latent space that can be directly useful for downstream tasks, and for this reason VAEs are the main

focus of this thesis.

In practice, applications of AEs and VAEs in financial crime detection can be grouped into two main directions: anomaly detection based on reconstruction error, and the use of latent representations for downstream tasks. The following subsections review these two approaches in detail.

2.3.1. Reconstruction Error-Based Methods

One of the most common applications of autoencoders and their variants is anomaly detection through reconstruction error. When trained on normal data, these models learn to compress and reconstruct typical patterns accurately. When an anomalous input is presented, its structure differs from what the model has learned to capture, and the reconstruction error is larger. This error can therefore be used directly as an anomaly score. This approach has been successfully applied across domains, including image data [15], network intrusion detection [16], and medical time series such as ECG signals [17].

In the context of financial crime detection, these methods have also shown promise, as highlighted in a recent review [18] that points to multiple supporting studies. On credit card datasets, Pumsirirat and Yan [19] trained autoencoders to learn normal transaction patterns and identify fraud through reconstruction error, reporting an AUC of 96.03%. Using the same data, Sweers et al. [20] explored stacked AEs and VAEs, while Renström and Holmsten [21] examined deeper AE architectures, both relying on reconstruction error to separate normal from fraudulent cases. More recently, Karkaba et al. [22] compared AEs with artificial neural networks (ANN) and convolutional neural networks (CNN) on imbalanced transaction data. They found the AE to be most effective, achieving an F1 score of 93% with no false positives. Alarfaj et al. [23] combined reconstruction-based anomaly detection with risk scoring methods, while Shende and Sontakke [24] demonstrated the practical feasibility of autoencoders for real-time fraud detection. Overall, these results confirm that reconstruction-based methods are a promising direction for detecting financial crime.

2.3.2. Latent Space-Based Methods

In recent years, research has increasingly explored the use of AEs and VAEs through their latent space. Several studies propose two-stage frameworks in which the autoencoder first compresses high-dimensional transaction data into a compact latent space, and the resulting representations are then used for clustering, classification, or sequence modeling.

For example, AEs have been combined with random forest [25], LightGBM [26], and XGBoost [27], demonstrating that latent features improve predictive performance compared to using raw attributes. Similarly, a semi supervised approach [28] embedded transactions into a lower dimensional space through an AE and then applied a linear classifier, showing the effectiveness of latent representations even with limited labels. Variational autoencoders have also been investigated in this context, with one study using a VAE to learn representations of normal transactions before applying a support vector data description model (SVDD) to identify fraud [29]. More complex pipelines combine autoencoders with deep classifiers, such as stacked AEs followed by convolutional neural networks for credit card fraud classification [30]. Student projects conducted with Deloitte and a partner bank have also explored the latent space of VAEs, focusing on outlier detection, classification and clustering [31, 32, 33, 34].

Across these approaches, the latent space produced by AEs and VAEs consistently emerges as an informative intermediate representation that captures complex patterns and improves the performance of downstream models. This highlights the central role of AEs and VAEs in the broader field of *representation learning*, where the objective is to extract meaningful features that make data more tractable. However, the interpretability of these features remains an open challenge, which we address in the next section.

2.4. Interpretability

Autoencoders and variational autoencoders learn to transform high-dimensional financial data into a smaller set of latent features. These latent variables act as a compact summary: instead of keeping every detail of the original data, the model encodes the information it considers most relevant for reconstruction.

However, while compact, these representations are not interpretable. Each latent variable is just a numerical dimension with no intuitive meaning. For example, one latent dimension might capture a complex combination of spending frequency, transaction timing, and geographic distribution of activities, but the model provides no label or explanation. As a result, when we use these latent representations for downstream tasks, it is unclear which aspects of the original data are driving the model's decisions.

When used to detect suspicious activity, this lack of transparency directly affects analysts who must investigate flagged accounts and transactions. Without knowing what triggered an alert, they have no clear lead on where to begin their review. By contrast, when alerts are based on rules, analysts know exactly which condition was breached, allowing them to focus on the relevant part of the data.

The limited interpretability is not only a practical challenge but also a regulatory problem. Financial institutions operate in one of the most tightly regulated environments, where every automated decision must be justified and auditable. These concerns have gained renewed importance with the adoption of the EU AI Act [35] in 2024, which sets out strict obligations for explainability, documentation, and human oversight. In practice, the difficulty of meeting these expectations has discouraged the adoption of machine learning and pushed many banks to continue relying on rule-based systems, where the rationale behind each decision is clear and can be easily explained.

In response to these pressures, a growing wave of research on explainable AI has emerged, emphasizing transparency, traceability, and accountability.

2.4.1. Approaches to Interpretability in AEs and VAEs

In the financial crime domain, research on interpretability in the use of autoencoders has so far been limited to anomaly detection, focusing on explaining why individual cases are flagged rather than on providing a global understanding of the latent space. For instance, Chaquet-Ulldemolins et al. [36] encoded transactions with an autoencoder and analyzed flagged cases by perturbing their latent codes and reconstructing them, allowing them to trace back which aspects of the input data were most influential in the model's decision. Similarly, Sattarov et al. [37] proposed a denoising autoencoder framework that detects anomalies and highlights the specific input fields within a transaction that caused the anomaly.

Beyond case-level explanations, there is also considerable value in explaining the latent space as a whole. Global interpretability makes it possible to go beyond saying *why* an individual account was flagged, toward understanding *how* the model organizes transaction behavior more broadly. If latent representations become interpretable, they can be used as descriptors of account behavior, turning the latent space from an abstract set of numerical encodings into a structured summary of transaction activity.

Making the latent space interpretable would create opportunities that go well beyond individual anomaly detection. One such opportunity is *typology discovery*. Accounts that are close in the latent space share similar transaction profiles, and grouping them can uncover new typologies of financial crime. Once a suspicious pattern has been identified, analysts can extend the investigation to other accounts within the same cluster, allowing the detection of linked suspicious behavior on a broader scale. A related use case is *risk profiling*. If regions of the latent space can be associated with established low-risk or high-risk behavioral patterns, the position of an account becomes an interpretable signal of its risk level.

The latent space also makes it possible to *track change in behavior*. By monitoring how an

account's position in the latent space changes over time, institutions can identify when its risk profile may need to be reassessed. In addition, latent representations can be *integrated into existing systems*. Because they summarize account behavior in a structured way, they can serve directly as input features, for example in outlier detection models. Their interpretability adds value by making the resulting predictions easier to explain and justify.

Despite the potential of these applications, a clear research gap remains. Little attention has been given to how autoencoders and variational autoencoders can be designed and evaluated to provide interpretable descriptors of account behavior. This thesis addresses this gap by working toward interpretable representations and, most importantly, by building a framework to systematically assess their interpretability.

3

Variational Autoencoders

In this chapter, we cover the fundamentals of Bayesian inference that are essential for understanding the methods presented in this thesis. In Section 3.1, we motivate and introduce latent-variable models in the context of probabilistic generative modeling of observed systems. Building upon this, in Section 3.2, we show how inference in latent-variable models can be framed as an optimization problem. Finally, in Section 3.3, we present variational inference within the framework of variational autoencoders, emphasizing the role of neural networks in approximating complex distributions.

3.1. Latent Variable Models

In probabilistic modeling, our goal is to approximate the true probability distribution $p^*(\mathbf{x})$ that governs an unknown but observable system. To achieve this, we introduce a model $p_\theta(\mathbf{x})$, parameterized by θ, and aim to make it closely match the true distribution:

$$p^*(\mathbf{x}) \approx p_\theta(\mathbf{x}).$$

Ideally, we want $p_\theta(\mathbf{x})$ to be expressive enough to capture the complexity of the real distribution. However, modeling high-dimensional distributions directly is notoriously difficult due to the curse of dimensionality and the need to represent complex dependencies among variables.

A common strategy to address this is to define $p_\theta(\mathbf{x})$ as a latent variable model. This means we introduce latent variables $\mathbf{z} \in \mathbb{R}^K$ that are not directly observed but are assumed to capture the underlying structure of the data. In the following, we assume that $\mathbf{z}$ are continuous random variables, but many of the ideas described below also apply to the discrete case.

A latent variable model defines two key components: a *prior distribution* $p(\mathbf{z})$, which reflects our assumptions about the latent variables and is typically chosen in advance, and a *likelihood distribution* $p_\theta(\mathbf{x} \mid \mathbf{z})$, which describes how an observation $\mathbf{x}$ is generated from a given latent variable $\mathbf{z}$.

This leads to a natural interpretation of the latent variable model as a two-step generative process: first, a latent variable is sampled from the prior, and then an observation is generated based on that latent representation:

$$\mathbf{z}' \sim p(\mathbf{z}) \quad \rightarrow \quad \mathbf{x}' \sim p_\theta(\mathbf{x} \mid \mathbf{z}').$$

The central challenge in latent variable modeling is determining how to learn the parameter θ from observed data. Assume we observe a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$, where each $\mathbf{x}_i$ is independently drawn from the true but unknown data distribution. The standard approach is maximum likelihood estimation, which aims to find the parameter that maximizes the likelihood of the observed data:

$$\theta^* = \arg\max_\theta \log p_\theta(\mathbf{x}_1, \ldots, \mathbf{x}_N) = \arg\max_\theta \sum_{i=1}^{N} \log p_\theta(\mathbf{x}_i).$$

In latent variable models, each marginal likelihood is computed by integrating over latent variables $\mathbf{z}$:

$$p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{x}, \mathbf{z})\, d\mathbf{z}.$$

Due to the complexity and high-dimensionality of this integral, it is typically impossible to evaluate exactly, making the marginal likelihood intractable. As a result, we cannot directly apply maximum likelihood estimation.

In addition to parameter estimation, we frequently need to infer which latent variables likely generated each observed data point. This is formalized through the posterior distribution:

$$p_\theta(\mathbf{z} \mid \mathbf{x}) = \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{x})}.$$

However, since the posterior depends on the marginal likelihood in the denominator, an intractable marginal likelihood also means an intractable posterior. This creates two linked problems: we cannot perform exact parameter learning or exact posterior inference.

To overcome these challenges, approximation methods are used to jointly estimate the posterior and enable parameter updates by approximating the marginal likelihood.

3.1.1. Approximate Inference

Approximation techniques generally fall into two main categories, either based on sampling or optimization. Sampling-based techniques, such as Markov Chain Monte Carlo (MCMC), are nonparametric methods that approximate the posterior by generating samples from it. They have the advantage of being asymptotically exact but are often computationally expensive and do not scale well to large datasets. Optimization-based methods, such as Variational Inference and Expectation Propagation, approximate complex distributions by selecting a member from a tractable family that is closest according to some divergence measure. They are generally faster and more scalable than sampling-based approaches, but not exact, even given unlimited computational resources.

Ultimately, the preferred approach is highly dependent on the available data and the intended application. This thesis focuses on large datasets of high-dimensional data, for which variational inference provides a good trade-off between quality of the approximation and scalability.

3.2. Variational Inference

Variational inference reframes the inference of the posterior distribution as an optimization problem. It introduces a variational family $\mathcal{Q}$ of distributions over the latent variables and selects, for each datapoint $\mathbf{x}$, a candidate $q(\mathbf{z}) \in \mathcal{Q}$ that approximates the true posterior:

$$p_\theta(\mathbf{z} \mid \mathbf{x}) \approx q(\mathbf{z}).$$

To find the best approximation, we minimize a measure of dissimilarity between $q(\mathbf{z})$ and $p_\theta(\mathbf{z} \mid \mathbf{x})$. While there are many different ways to measure how different two distributions are, variational inference uses the Kullback-Leibler (KL) divergence, which quantifies how much information is lost when using $q(\mathbf{z})$ in place of the true posterior. It is defined as:

$$\begin{aligned} \mathrm{KL}\left(q(\mathbf{z}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x})\right) &= \int q(\mathbf{z}) \log \frac{q(\mathbf{z})}{p_\theta(\mathbf{z} \mid \mathbf{x})}\, d\mathbf{z} \\ &= -\mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{z} \mid \mathbf{x})}{q(\mathbf{z})}\right]. \end{aligned} \tag{3.1}$$

Importantly, the KL divergence is a non-negative quantity, i.e., $\mathrm{KL}\left(q(\mathbf{z}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x})\right) \geq 0$, with equality if and only if $q(\mathbf{z}) = p_\theta(\mathbf{z} \mid \mathbf{x})$. The more similar $q(\mathbf{z})$ is to $p_\theta(\mathbf{z} \mid \mathbf{x})$, the smaller the KL divergence will be. Notice that this quantity is not a distance in the mathematical sense, as it is not symmetric if we swap the two distributions.

Our goal is to find a good variational approximation $q(\mathbf{z})$ that minimizes the KL divergence in Equation 3.1:

$$q^*(\mathbf{z}) = \arg\min_{q(\mathbf{z}) \in \mathcal{Q}} \mathrm{KL}\big(q(\mathbf{z}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x})\big).$$

However, this quantity is still not tractable, as the intractable posterior $p_\theta(\mathbf{z} \mid \mathbf{x})$ appears inside the logarithm. To address this, we turn to an equivalent and tractable objective.

In the following, we show that minimising the KL divergence between the variational distribution and the posterior is equivalent to maximizing a lower bound on the marginal likelihood, commonly referred to as the *Evidence Lower Bound* (ELBO). Typically, the ELBO is derived through Jensen's inequality. Here we will use an alternative derivation that avoids Jensen's inequality, building intuition into the origin and meaning of the bound.

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{z})}\left[\log p_\theta(\mathbf{x})\right] \\
&= \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{p_\theta(\mathbf{z} \mid \mathbf{x})}\right] \\
&= \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} \cdot \frac{q(\mathbf{z})}{p_\theta(\mathbf{z} \mid \mathbf{x})}\right] \\
&= \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})}\right] + \mathbb{E}_{q(\mathbf{z})}\left[\log \frac{q(\mathbf{z})}{p_\theta(\mathbf{z} \mid \mathbf{x})}\right].
\end{aligned} \tag{3.2}$$

The first term in Equation 3.2 is the ELBO, which bounds the log-likelihood from below due to the non-negativity of the KL divergence:

$$\begin{aligned}
\mathcal{F}(q(\mathbf{z}), \theta; \mathbf{x}) &= \mathbb{E}_{q(\mathbf{z})}\left[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q(\mathbf{z})\right] \\
&= \log p_\theta(\mathbf{x}) - \mathrm{KL}(q(\mathbf{z}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x})) \\
&\leq \log p_\theta(\mathbf{x}).
\end{aligned} \tag{3.3}$$

Observe that the log-evidence, $\log p_\theta(\mathbf{x})$, does not depend on $q(\mathbf{z})$, and therefore the variational distribution that maximises the ELBO concurrently minimises the KL divergence between the variational distribution and the posterior:

$$\arg\min_{q(\mathbf{z}) \in \mathcal{Q}} \mathrm{KL}\left(q(\mathbf{z}) \,\|\, p_\theta(\mathbf{z} \mid \mathbf{x})\right) = \arg\max_{q(\mathbf{z}) \in \mathcal{Q}} \mathcal{F}(q(\mathbf{z}), \theta; \mathbf{x}).$$

On the other hand, the ELBO is a lower bound on the marginal log-likelihood, and the KL divergence describes the tightness of the bound. Thus, maximizing the ELBO can be viewed as an alternative to the intractable maximum likelihood estimate:

$$\arg\max_\theta \log p_\theta(\mathbf{x}) \quad \rightarrow \quad \arg\max_\theta \mathcal{F}(q(\mathbf{z}), \theta; \mathbf{x}).$$

3.2.1. Traditional and Amortised Variational Inference

To make variational inference feasible in practice, the variational distribution is typically restricted to a simple and tractable family of distributions. This restriction allows the ELBO to be computed or approximated efficiently, and turns the optimization over $q(\mathbf{z})$ into a problem of tuning the parameters of the chosen family:

$$\max_{q(\mathbf{z}) \in \mathcal{Q}} \mathcal{F}(q(\mathbf{z}), \theta; \mathbf{x}) \quad \longrightarrow \quad \max_\phi \mathcal{F}(q_\phi(\mathbf{z}), \theta; \mathbf{x}).$$

In its conventional form, often referred to as *Factorised Variational Inference (F-VI)* or *Mean-Field Variational Inference*, the variational family is fully factorized, meaning that each latent dimension is modeled independently of the others. Let L denote the number of latent dimensions. Under this assumption, the approximate posterior factorizes across latent variables as:

$$q_\phi(\mathbf{z}) = \prod_{j=1}^{L} q_{\phi^{(j)}}(z_j).$$

Here, $\phi = \{\phi^{(1)}, \ldots, \phi^{(L)}\}$ denotes the set of variational parameters specific to datapoint $\mathbf{x}$.

Given a dataset $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^N$ of N observed samples, traditional variational inference assigns a distinct set of variational parameters ϕ_i to each sample $\mathbf{x}_i$. These parameters are optimized individually by maximizing the ELBO for each datapoint, typically using gradient-based methods.

While this approach is flexible, optimizing a separate set of variational parameters for each datapoint quickly becomes impractical for large datasets, as the number of parameters grows with the dataset size. Moreover, the process is memoryless: each observation is treated independently, so inference for one sample does not benefit from previous ones, and performing inference on a new datapoint requires re-running the entire optimization from scratch.

Amortised Variational Inference (A-VI) addresses these limitations by replacing per-datapoint optimization with a single shared function that maps any input $\mathbf{x}_i$ to the parameters of its approximate posterior. In this way, the cost of inference is *amortised* across the entire dataset, making training more scalable and allowing for fast inference on unseen data. With recent advances in deep learning, this amortised inference framework has become the basis for Variational Autoencoders.

3.3. Variational Autoencoders

A Variational Autoencoder (VAE) is a latent variable model designed to combine probabilistic inference with the representational power of neural networks. We assume that the reader is familiar with the basics of neural network architectures and refer to Goodfellow et al. [38] for a comprehensive introduction. In this thesis, we are mainly interested in the use of neural networks in probabilistic models, i.e., for modeling probability density functions, which is based on the idea of defining the distribution parameters as functions of the conditioning variables. This approach is central to the amortisation of the approximate posterior within VAEs.

To define a Variational Autoencoder, we need to describe its generative model (i.e. the latent variable model), the inference network (i.e. the variational approximation), and how to learn the parameters of the VAE.

Generative Model

The generative model describes how observed data $\mathbf{x}$ is generated from latent variables $\mathbf{z}$. The prior captures our initial belief about the distribution of these latent variables before seeing any data, and it is typically chosen as an isotropic multivariate Gaussian with zero mean and identity covariance matrix:

$$p_\theta(\mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I).$$

The likelihood $p_\theta(\mathbf{x} \mid \mathbf{z})$, also known as the decoder, is usually modeled as a Gaussian distribution for continuous data:

$$p_\theta(\mathbf{x} \mid \mathbf{z}) = \mathcal{N}\big(\mathbf{x}; \mu_\theta(\mathbf{z}), \Sigma_\theta(\mathbf{z})\big),$$

where $\mu_\theta(\mathbf{z})$ and $\Sigma_\theta(\mathbf{z})$ are functions of $\mathbf{z}$ implemented by a neural network with parameters θ. This defines the joint distribution

$$p_\theta(\mathbf{x}, \mathbf{z}) = p_\theta(\mathbf{x} \mid \mathbf{z})\, p_\theta(\mathbf{z}),$$

which represents the data generation process by first sampling $\mathbf{z}$ from the prior and then generating $\mathbf{x}$ from the conditional distribution.

Inference Network

Due to the non-linearities in the deep neural networks that parameterize $p_\theta(\mathbf{x} \mid \mathbf{z})$, computing the marginal likelihood $p_\theta(\mathbf{x})$ and the true posterior exactly is intractable. To address this, Variational Autoencoders maximize the Evidence Lower Bound (ELBO), as introduced in Section 3.2.

Directly optimizing separate variational parameters for each data point becomes computationally expensive because the number of parameters grows linearly with the dataset size. Instead, VAEs use amortized inference [39], where a single set of shared parameters ϕ is used across the entire dataset. These parameters define an *inference network* (or encoder) that maps each data point to the parameters of a variational posterior, typically modeled as a normal distribution with a diagonal covariance matrix:

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}\left(\mu_\phi(\mathbf{x}),\, \mathrm{diag}(\sigma^2_\phi(\mathbf{x}))\right).$$

Here, the shared parameters ϕ correspond to the weights and biases of the neural network.

Parameter Learning

To learn the parameters in the presence of latent variables, we use the ELBO introduced in Equation 3.3. Since the variational distribution $q_\phi(\mathbf{z} \mid \mathbf{x})$ belongs to a parametric family, the maximization over q becomes a maximization over the variational parameters ϕ. We therefore denote the ELBO as $\mathcal{F}(\theta, \phi)$:

$$\mathcal{F}(\theta, \phi; \mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z} \mid \mathbf{x})\right]. \tag{3.4}$$

Since both the generative model and the inference model are parameterized by neural networks, gradients of the ELBO with respect to both θ and ϕ can be computed via backpropagation [40]. This algorithm relies on the chain rule to break down the gradient computation of a complex function to a chain of gradients of simple functions.

The gradient with respect to the generative model parameters θ is given by:

$$\nabla_\theta \mathcal{F}(\theta, \phi; \mathbf{x}) = \nabla_\theta \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\nabla_\theta \log p_\theta(\mathbf{x} \mid \mathbf{z})\right].$$

Note that the second term in Equation 3.4 does not depend on θ, hence its gradient is zero. Moreover, the gradient can be moved inside the expectation because $q_\phi(\mathbf{z} \mid \mathbf{x})$ does not depend on θ. This allows us to estimate the gradient efficiently using Monte Carlo sampling.

However, computing the gradient with respect to the variational parameters ϕ is challenging because the expectation is taken over a distribution that depends on ϕ:

$$\nabla_\phi \mathcal{F}(\theta, \phi; \mathbf{x}) = \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right] \neq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\nabla_\phi \log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right].$$

This issue is addressed using the reparameterization trick [2, 41], which rewrites the random variable $\mathbf{z}$ as a deterministic transformation of a simpler and auxiliary random quantity $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon})$:

$$\mathbf{z} = g_\phi(\boldsymbol{\epsilon}),$$

where $\boldsymbol{\epsilon}$ follows some base distribution $p(\boldsymbol{\epsilon})$.

Equipped with this parameterization, we can move the gradient inside the expectation since the distribution $p(\boldsymbol{\epsilon})$ does not depend on ϕ. Specifically, the gradient of the ELBO becomes:

$$\begin{aligned}\nabla_\phi \mathcal{F}(\theta, \phi; \mathbf{x}) &= \nabla_\phi \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x}, \mathbf{z}) - \log q_\phi(\mathbf{z} \mid \mathbf{x})\right] \\ &= \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\nabla_\phi \log p_\theta(\mathbf{x}, g_\phi(\boldsymbol{\epsilon})) - \nabla_\phi \log q_\phi(g_\phi(\boldsymbol{\epsilon}) \mid \mathbf{x})\right],\end{aligned}$$

which can be efficiently estimated using Monte Carlo sampling from $p(\boldsymbol{\epsilon})$.

A common example where the reparameterization trick is applicable is the multivariate Gaussian. For instance, if $q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}(\mu_\phi(\mathbf{x}), \Sigma_\phi(\mathbf{x}))$ and $p(\boldsymbol{\epsilon}) = \mathcal{N}(0, I)$, we can write:

$$\mathbf{z} = \mu_\phi(\mathbf{x}) + L_\phi(\mathbf{x})\boldsymbol{\epsilon},$$

where $L_\phi(\mathbf{x})$ is a lower triangular matrix resulting from the Cholesky decomposition $\Sigma_\phi(\mathbf{x}) = L_\phi(\mathbf{x})L_\phi(\mathbf{x})^\top$.

Analysis of the ELBO

The ELBO can be rearranged into a more intuitive form that highlights the two primary objectives of a VAE. Starting from its definition in Equation 3.3, we can separate it into two parts:

$$\begin{aligned}
\mathcal{F}(\theta, \phi; \mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x}, \mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log \frac{p_\theta(\mathbf{x} \mid \mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right] \\
&= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{z}) - \log q_\phi(\mathbf{z} \mid \mathbf{x})\right] \\
&= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right]}_{\text{Reconstruction Likelihood}} - \underbrace{\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z}))}_{\text{KL Regularizer}}.
\end{aligned} \tag{3.5}$$

The first term, known as reconstruction likelihood, measures how well the decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ can reconstruct the input data $\mathbf{x}$ when given a latent code $\mathbf{z}$ sampled from the encoder's approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$. It encourages the model to learn latent representations $\mathbf{z}$ that retain sufficient information to accurately reconstruct the input. The second term measures the dissimilarity between the approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $p(\mathbf{z})$. It acts as a regularizer, helping structure the latent space and making it more continuous and less prone to holes or disjoint regions. Together, these terms balance faithful reconstruction of data and the enforcement of a structured, continuous latent space, which is central to the success of VAEs.

We will close this section by deriving the reparameterized estimate of the ELBO used for optimization, starting from the form in Equation 3.4. Recall that throughout this section, we express the ELBO with respect to a single datapoint $\mathbf{x}$. For i.i.d. data, the ELBO for the entire dataset is the sum of the ELBOs for each individual data point.

In most VAEs, the approximate posterior $q_\phi(\mathbf{z} \mid \mathbf{x})$ and the prior $p(\mathbf{z})$ are chosen as multivariate normal distributions with diagonal covariance matrices:

$$q_\phi(\mathbf{z} \mid \mathbf{x}) = \mathcal{N}\big(\mu_\phi(\mathbf{x}),\, \mathrm{diag}(\sigma^2_\phi(\mathbf{x}))\big), \quad p(\mathbf{z}) = \mathcal{N}(0, I).$$

Thanks to this choice, the KL divergence term between these two Gaussians has a closed-form expression, which can be computed exactly and efficiently without sampling. Let L be the dimension of $\mathbf{z}$, then the KL divergence is given by:

$$D_{\mathrm{KL}}\big(q_\phi(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\big) = \frac{1}{2}\sum_{j=1}^{L}\left(\sigma^2_{\phi,j}(\mathbf{x}) + \mu^2_{\phi,j}(\mathbf{x}) - 1 - \log \sigma^2_{\phi,j}(\mathbf{x})\right).$$

The detailed derivation of this expression is provided in Appendix A.1.

On the other hand, the reconstruction term typically does not have a closed-form solution and is estimated via Monte Carlo sampling. Using the reparameterization trick, we express $\mathbf{z} = \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$, where $\odot$ denotes element-wise multiplication. This allows us to rewrite the expectation as

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_i)}\left[\log p_\theta(\mathbf{x} \mid \mathbf{z})\right] = \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,I)}\left[\log p_\theta\big(\mathbf{x} \mid \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}\big)\right],$$

which can be approximated by sampling $\boldsymbol{\epsilon} \sim \mathcal{N}(0, I)$.

The full estimate of the ELBO is then

$$\hat{\mathcal{F}}(\theta, \phi; \mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} \log p_\theta \left(\mathbf{x} \mid \mu_\phi(\mathbf{x}) + \sigma_\phi(\mathbf{x}) \odot \boldsymbol{\epsilon}^{(k)} \right) - \frac{1}{2} \sum_{j=1}^{L} \left(\sigma^2_{\phi,j}(\mathbf{x}) + \mu^2_{\phi,j}(\mathbf{x}) - 1 - \log \sigma^2_{\phi,j}(\mathbf{x}) \right),$$

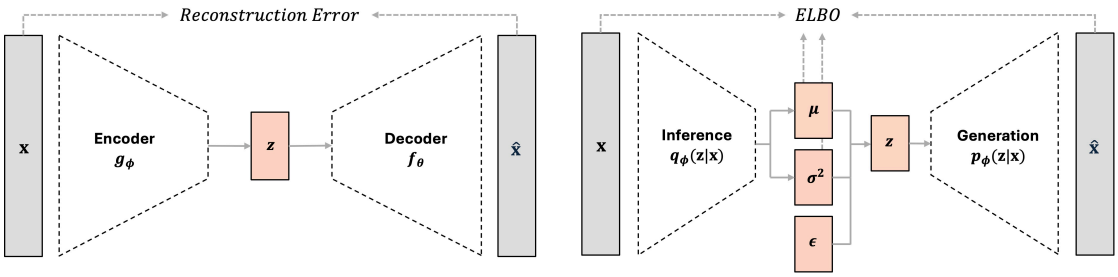
where K is the number of Monte Carlo samples used to estimate the expectation. This expression is then optimized jointly over θ and ϕ using stochastic optimization, where an average is taken over a mini-batch of data.

Connection to Autoencoders

So far, we have described VAEs in the context of probabilistic modeling. We now examine how this probabilistic framework relates to the standard autoencoder architecture.

Autoencoders are a class of unsupervised neural network models designed to learn efficient data representations through reconstruction. The architecture consists of two neural networks: an encoder g_ϕ that maps input data $\mathbf{x}$ to a lower-dimensional latent code $\mathbf{z} = g_\phi(\mathbf{x})$, and a decoder f_θ that reconstructs the original input as $\mathbf{x}^* = f_\theta(\mathbf{z})$. The model is trained by minimizing a reconstruction loss $\mathcal{L}(\mathbf{x}, \mathbf{x}^*)$ that measures the difference between input and reconstruction. The idea behind this design is that by forcing the data through a compressed intermediate representation, the model learns to capture the most essential features of the input.

The connection between VAEs and AEs becomes immediately apparent when comparing their architectures, as illustrated in Figure 3.1.



(a) AE architecture **(b)** VAE architecture

Figure 3.1: Architectures of a standard Autoencoder (AE) and a Variational Autoencoder (VAE)

In VAEs, the inference network $q_\phi(\mathbf{z} \mid \mathbf{x})$ serves as the encoder, mapping inputs to latent space, while the generative network $p_\theta(\mathbf{x} \mid \mathbf{z})$ acts as the decoder, reconstructing data from latent representations. The key distinction between VAEs and AEs lies in how they handle these mappings. Traditional autoencoders are deterministic: given the same input, they always produce identical latent representations and reconstructions. VAEs, in contrast, are probabilistic models in which the encoder outputs the parameters of a distribution rather than a fixed point. This allows VAEs to model uncertainty and encourages the learning of a smooth and continuous latent space. As a result, VAEs tend to generalize better: nearby points in the latent space correspond to similar outputs. This allows the model to generate new samples by drawing from the latent space, producing outputs it has never seen before but that are still consistent with the training distribution.

4

Disentangled Representation Learning

In this chapter, we explore the principles, motivations, and methods behind learning useful representations of data. In Section 4.1, we introduce the core challenges of representation learning and outline the desirable properties a good representation should have, both from an intuitive and information-theoretic perspective. In Section 4.2, we narrow our focus to disentangled representations, which aim to separate the underlying factors of data into distinct features. To encourage such representations, we examine the β-VAE, an extension of the variational autoencoder. Finally, in Section 4.3, we address the challenge of evaluating disentanglement and describe a taxonomy of existing metrics. Particular attention is given to the DCI metrics, which we adopt as the main evaluation tool in this thesis.

4.1. Representation Learning

The performance of machine learning models largely depends on how data is represented. In many applications, data is high-dimensional, yet only a few underlying patterns or structures are truly relevant to the task at hand. The rest often consists of noise or redundant information, which can hinder learning and increase the risk of overfitting. To address this, significant effort goes into transforming raw data into meaningful representations, a process known as feature engineering. Traditionally, this relies heavily on expert knowledge and domain expertise. However, manual feature design not only demands time and resources, but also limits the flexibility of models, as it ties the learning process to human intuition about which aspects of the data are important and which are not.

Representation learning offers a promising solution to these challenges. Instead of relying on manual feature engineering, it aims to automatically learn representations that capture the essential information in the data. A clear example is the latent space of a variational autoencoder (VAE), which compresses high-dimensional input into a structured, lower-dimensional representation that preserves the most relevant patterns.

4.1.1. Desiderata for Representations

One of the key challenges in representation learning is the lack of a clear and explicit objective. Rather than optimizing for a specific outcome, the goal is to learn features that are general, transferable, and useful across a wide variety of tasks. This makes both the learning process and the evaluation of representations particularly difficult, as success is not tied to a single performance metric.

To provide a principled framework for what constitutes a "good" representation, recent work has turned to information theory. A foundational concept in this line of research is the *Information*

Bottleneck (IB) principle [42], which formalizes the problem as an optimization trade-off: given input data X and a task variable Y, the goal is to learn a representation Z that retains as much information about Y as possible while discarding irrelevant information from X. This is captured by maximizing the mutual information $I(Z;Y)$ and minimizing $I(Z;X)$.

This formulation has been extended to deep learning through methods like the *Variational Information Bottleneck* [43], and further analyzed by Achille and Soatto [44], who identify three desirable properties that naturally emerge from this perspective:

- **Sufficiency:** The representation contains all information relevant to the task, i.e., $I(Z;Y) = I(X;Y)$.
- **Minimality:** The representation retains as little information about the input as possible, i.e., $I(Z;X)$ is minimized subject to sufficiency.
- **Invariance:** The representation is independent of nuisance factors N unrelated to the task, i.e., $I(Z;N) = 0$.

These formal properties are consistent with intuitive criteria frequently discussed in the literature. For instance, Ridgeway [45] summarizes four desirable attributes of learned representations that reflect practical needs in machine learning. A good representation should be *compact*, meaning it reduces redundancy and encodes information in a low-dimensional form. It should be *faithful* to the input, preserving the essential information necessary for downstream tasks. Additionally, representations should *explicitly* encode task-relevant attributes, making important factors easily accessible and robust to noise. Finally, the features within this learned representation should be *interpretable*.

These qualities align closely with the formal criteria introduced earlier. Compactness and faithfulness reflect the balance between minimality and sufficiency; explicitness relates to invariance, as it requires isolating and preserving only the relevant factors; and interpretability, though less formally defined, can emerge when a representation is both compressed and well-structured, such that the retained information corresponds to meaningful, high-level aspects of the input.

4.2. Disentanglement

While interpretability is often considered a desirable property in representation learning, it remains loosely defined and difficult to quantify. One way to formalize interpretability, and to encourage structured, meaningful representations, is through the notion of disentanglement. The core idea is that the data we observe is generated by a set of underlying, unobserved variables, called factors of variation. Disentangled Representation Learning (DRL) aims to map each factor to a distinct feature in the learned representation.

An example of DRL is presented in Figure 4.1.

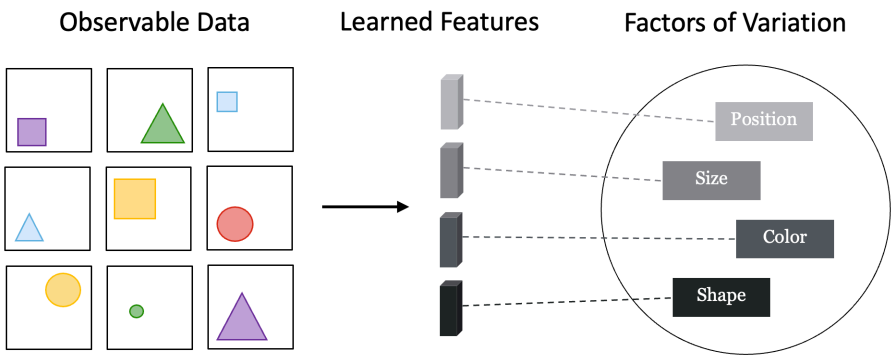


Figure 4.1: Illustration of DRL, where independent factors of variation are separated into distinct features in the representation.

The high-dimensional image data can be captured by four simple dimensions: shape, size, color, and position. These dimensions are the generative factors behind the dataset. DRL aims to separate these factors and encode them into distinct features in the representation space.

From an information-theoretic perspective, disentanglement can be seen as a structural constraint on top of minimality and sufficiency. While a minimal and sufficient representation captures task-relevant information in a compressed form, it may still entangle multiple generative factors in the same feature. Disentanglement introduces an additional preference for feature spaces where generative factors are cleanly separated.

4.2.1. Formal Definition

Despite being a topic of great interest, there is no general consensus on the definition of a disentangled representation. One of the earliest and most widely cited definitions was given by Bengio et al. [3], and has since been adopted in several works [4, 46].

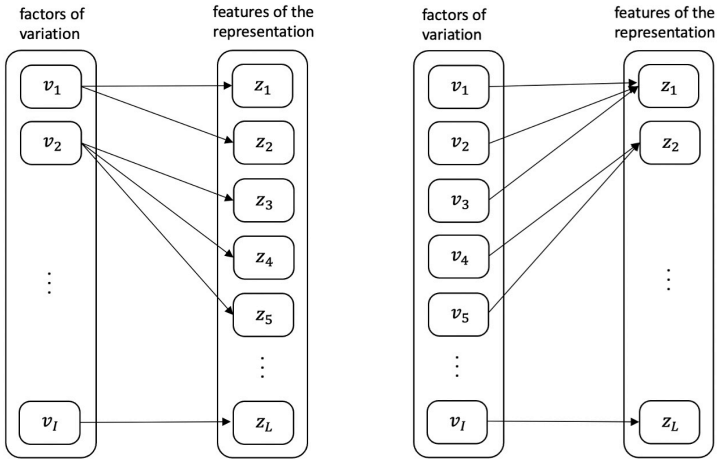
Definition 1 *A disentangled representation is one in which a change in a factor of the representation corresponds to a change in a factor of variation, while being relatively invariant to changes in other factors.*

Other authors shift the focus: instead of looking at how changes in the representation affect the generative factors, they consider how changes in the factors are reflected in the learned features [47, 48].

Definition 2 *A disentangled representation is one in which a change in a factor of variation translates into a change in a factor of the representation.*

The key distinction between these two definitions lies in how uniquely the generative factors and representation features are aligned.

Figure 4.2(a) shows a representation that satisfies Definition 1 but not Definition 2: while each feature of the representation captures at most one factor, multiple features redundantly capture the same factor. Conversely, Figure 4.2(b) shows a representation that satisfies Definition 2 but not Definition 1: each generative factor is captured by exactly one feature, yet some features capture multiple generative factors simultaneously.



(a) A representation that satisfies Definition 1 but not Definition 2.

(b) A representation that satisfies Definition 2 but not Definition 1.

Figure 4.2: Representations that satisfy only one of the two definitions of disentanglement.

Given the existence of multiple definitions, several papers propose to define disentangled representations based on a list of properties that they must satisfy. Among them, Eastwood and Williams [49] and Kumar et al. [48] each identified three criteria that a representation must meet in order to have all the advantages traditionally attributed to disentangled representations.

Although they refer to the same ideas, the two papers refer to these properties by different names. In this thesis, we use those of Kumar et al. [48]. The properties are:

- **Modularity** (Disentanglement in [49]): Each variable in the representation captures at most one factor of variation.
- **Compactness** (Completeness in [49]): Each factor of variation is captured by only one variable of the representation.
- **Explicitness** (Informativeness in [49]): The representation captures all information necessary to accurately reconstruct the factors of variation.

These three properties will serve as the basis of the framework used throughout the thesis to evaluate the quality of learned representations in terms of disentanglement. In the following, we focus on one specific approach designed to encourage such representations: the β-Variational Autoencoder (β-VAE).

4.2.2. β-VAEs

The β-VAE [4] is an extension of the standard variational autoencoder designed to promote disentangled representations. It builds on the VAE framework by introducing a hyperparameter β in the ELBO:

$$\mathcal{F}_\beta(\phi, \theta) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[\log(p_\theta(\mathbf{x} \mid \mathbf{z}))\right] - \beta D_{\mathrm{KL}}(q_\phi(\mathbf{z} \mid \mathbf{x}) \,\|\, p(\mathbf{z})). \tag{4.1}$$

As we have discussed in Section 3.2, the first term in the ELBO encourages reconstruction of the input, while the second term can be interpreted as a regularizer of the approximate posterior with respect to the prior. When $\beta = 1$ we recover the original ELBO, and larger values of β are believed to promote disentanglement [4].

The behavior of β-VAE can be better understood by drawing a connection to the Information Bottleneck (IB) principle [42]. Recall from Section 4.1.1 that the Information Bottleneck describes an objective that aims to maximize the mutual information between a representation Z and a task Y while constraining information with regards to the input data X:

$$\max_{q(Z|X)} \; I(Z;Y) - \beta I(Z;X).$$

In the unsupervised setting of autoencoders, there is no external target variable Y, and the reconstruction of the input itself becomes the task. Under this view, the IB objective translates naturally to the VAE setting: the goal is to find a latent representation Z that captures the most important information for reconstructing X, while being as compressed as possible. The β-VAE objective can be viewed as a practical instance of this principle, where the KL divergence term acts as an upper bound on the mutual information $I(Z;X)$, and the reconstruction term approximates $I(Z;X)$ from below via the data log-likelihood [43, 50].

This interpretation provides insight into why β-VAEs tend to produce disentangled representations. As β increases, the KL divergence term is given more weight, tightening the information bottleneck and limiting the total capacity of the latent representation. Under this constraint, the model is forced to be selective about what information it retains. Burgess et al. [50] argue that this pressure encourages the model to encode the most informative factors: those that are most critical for accurate reconstruction.

To manage the restricted capacity, the model assigns different factors to separate latent dimensions, enabling it to capture the data structure efficiently without redundancy. This effect is reinforced by the choice of a factorized prior, such as the isotropic Gaussian $p(\mathbf{z}) = \prod_j p(z_j)$. The KL divergence term in the β-VAE loss encourages the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})$ to

match this prior. When averaged over the data distribution, this pressure shapes the aggregate posterior $q(\mathbf{z}) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x})]$ to also be approximately factorized. As a result, each latent dimension tends to specialize in capturing a single factor.

Importantly, the authors highlight that disentanglement does not increase monotonically with β. While increasing β can promote more independent and interpretable representations, setting it too high can lead to overly compressed latent codes. In this regime, the model may begin to entangle multiple generative factors within the same latent dimension, as it lacks the capacity to represent them separately.

β-VAE Applications Across Domains

Beyond the theory, successful applications of the β-VAE in different domains demonstrate its potential. In cardiology, for example, latent dimensions learned from electrocardiograms were found to align with clinically meaningful ECG features, making the model more transparent for medical use [51]. In fluid dynamics, β-VAEs have been used to capture coherent structures in turbulent flows, offering compact and interpretable representations [52]. In graph analysis, disentangled latent variables could be directly linked to generative parameters such as the number of nodes or connection probability [53]. Finally, in the analysis of eye imaging data, β-VAEs disentangled anatomical features such as chamber depth, pupil size, and corneal shape, each captured by a separate latent variable [54].

These findings from diverse fields motivate this thesis to explore the β-VAE in financial crime detection, where disentangled representations could serve as interpretable descriptors of account behavior and provide a foundation for detecting and understanding financial crime.

4.3. Measuring Disentanglement

Since there is no consensus on the definition of disentanglement, there is also no consensus on how to measure it. In [55], the various evaluation methods are categorized into three main groups based on their underlying principles. Inspired by [56], they further divide each family into subgroups according to the specific disentanglement property each metric is designed to measure. Some methods, referred to as holistic, attempt to capture multiple properties within a single score. Figure 4.3 presents all metrics organized according to this expanded taxonomy.

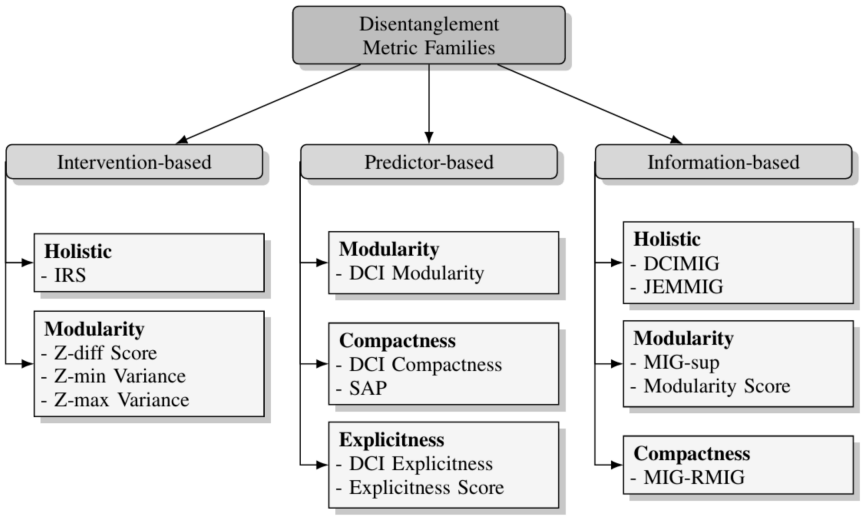


Figure 4.3: Taxonomy of disentanglement metrics, grouped by evaluation principle and the specific property each metric is designed to measure. Reproduced from [55].

Below, we provide an overview of the three metric families.

Intervention-based Metrics. These methods assess disentanglement by isolating one factor of variation at a time and creating subsets of data in which only that factor remains constant. To obtain these subsets, the factor space is typically discretized, and a fixed number of samples per bin are drawn. The latent representations of these subsets are then compared in different ways to derive a score. Examples include the Z-diff [4], Z-min Variance [46], and Z-max Variance metrics [57]. The main advantage of these methods is that they make no assumptions about the relationship between latent codes and generative factors. However, they come with several limitations. Most notably, their performance is sensitive to a number of hyperparameters, such as the number and size of data subsets, the level of discretization, and the choice of distance metric, all of which can significantly affect the resulting scores. Moreover, prior studies [46, 56] have highlighted several failure modes in which these metrics assign high scores to clearly entangled representations, or low scores to disentangled ones.

Information-based Metrics. These metrics estimate mutual information between latent variables and generative factors to assess disentanglement. Common examples include MIG [58], RMIG [59], and DCIMIG [56]. A key limitation of this family is that they consider only the difference in mutual information between the two most informative latent dimensions for each factor. As a result, they fail to capture situations where a single factor is distributed across multiple latent dimensions. Carbonneau et al. [55] showed that MIG, for instance, assigns identical scores to representations in which a factor is encoded by either two or four latent dimensions. Furthermore, mutual information estimation is often sensitive to discretization and binning choices, which can introduce additional variability in the results.

Predictor-based Metrics. These methods rely on training supervised models to predict factors from codes. The trained models are then analyzed to obtain a score. Metrics such as SAP [48], DCI [49], and the Explicitness Score [60] fall into this category. These approaches offer a more detailed view of disentanglement properties and have proven effective in realistic settings [55]. While they do involve some modeling choices and hyperparameter tuning, this added complexity can also enable more interpretable and fine-grained analysis.

For this thesis, we chose to use DCI [49]. Unlike intervention-based and information-based metrics, DCI does not require discretization of the factor or latent space, which makes it more robust to non-linear relationships and noise. It evaluates the three disentanglement properties separately but using a consistent approach, and returns a distinct score for each. In addition, DCI provides per-factor and per-dimension scores, which makes it easier to analyze which parts of the representation contribute most to disentanglement. While it does require training a predictive model and tuning its hyperparameters, we found this trade-off acceptable given its flexibility and interpretability. In the following section, we will first explain the intuition behind DCI, and then present its mathematical formulation.

4.3.1. DCI

The central idea behind DCI is to assess disentanglement by studying how the latent representation relates to a set of factors of variation. To do so, supervised models are trained to predict each factor from the latent variables. The goal is not accurate prediction itself, but to reveal how factor information is distributed across the latent space.

In practice, DCI requires three modeling choices: a predictive model used to map latent variables to factors, a method for estimating relative importances across latent dimensions, and an error function used to quantify prediction quality. Once the models are trained, these choices yield two outputs. The first is the *relative importance matrix*, which indicates how much each latent variable contributes to each factor prediction. By examining its structure, we can assess whether latent dimensions specialize in individual factors (modularity) and whether each factor relies on only a few dimensions (compactness). The second output is the set of *prediction errors*, computed by comparing the predicted and true values of each factor. These errors indicate how well the overall representation explains the factors, independent of how the information is distributed, and they form the basis for the explicitness score.

We now present the mathematical formulation of the three metrics. We assume access to a

dataset of samples $\mathbf{x}$, their corresponding factors of variation $\mathbf{v} \in \mathbb{R}^I$, and an encoder En that encodes each sample into a latent representation $\mathbf{z} = En(\mathbf{x}) \in \mathbb{R}^L$. A supervised model is trained for each generative factor v_i to approximate the mapping:

$$\hat{v}_i = f_i(\mathbf{z}),$$

where $f_i : \mathbb{R}^L \to \mathbb{R}$ is the predictive model for factor v_i.

From the trained models we obtain the importance matrix $R \in \mathbb{R}^{I \times L}$ and the set of prediction errors $\{Err(v_i, \hat{v}_i)\}_{i=1}^I$.

DCI Modularity. Modularity is computed by analyzing how concentrated the importance of each latent variable z_j is across all generative factors. The modularity score for z_j is defined as:

$$M_j = 1 - H_I(P_{\cdot j}), \quad \text{with} \quad P_{ij} = \frac{R_{ij}}{\sum_{k=0}^{I-1} R_{kj}}.$$

The term H_I corresponds to the entropy normalized over the I factors, given by:

$$H_I(P_{\cdot j}) = \frac{H(P_{\cdot j})}{H_{\max}} = \frac{-\sum_{k=0}^{I-1} P_{kj} \log P_{kj}}{-\sum_{k=0}^{I-1} \frac{1}{I} \log \frac{1}{I}} = \frac{-\sum_{k=0}^{I-1} P_{kj} \log P_{kj}}{\log I} = -\sum_{k=0}^{I-1} P_{kj} \log_I P_{kj}.$$

A latent variable that contributes mainly to a single factor will have low entropy and thus a high modularity score.

The overall modularity score is computed as a weighted average of the individual scores M_j, with weights ρ_j reflecting the total importance of each latent dimension across all factors. These weights reduce the influence of latent dimensions that are not used to predict any factor and ensure the score focuses on the meaningful parts of the representation:

$$M = \sum_{j=0}^{L-1} \rho_j M_j, \quad \text{where} \quad \rho_j = \frac{\sum_{i=0}^{I-1} R_{ij}}{\sum_{i=0}^{I-1} \sum_{j=0}^{L-1} R_{ij}}.$$

DCI Compactness. Compactness measures whether each factor v_i is captured primarily by a single latent variable. It is computed as:

$$C_i = 1 - H_L(\tilde{P}_{i\cdot}), \quad \text{with} \quad \tilde{P}_{ij} = \frac{R_{ij}}{\sum_{k=0}^{L-1} R_{ik}}.$$

The normalized entropy over the L latent variables is given by:

$$H_L(\tilde{P}_{i\cdot}) = \frac{H(\tilde{P}_{i\cdot})}{H_{\max}} = \frac{-\sum_{k=0}^{L-1} \tilde{P}_{ik} \log \tilde{P}_{ik}}{-\sum_{k=0}^{L-1} \frac{1}{L} \log \frac{1}{L}} = \frac{-\sum_{k=0}^{L-1} \tilde{P}_{ik} \log \tilde{P}_{ik}}{\log L} = -\sum_{k=0}^{L-1} \tilde{P}_{ik} \log_L \tilde{P}_{ik}.$$

A high compactness score indicates that the prediction of v_i depends mostly on one latent dimension. The overall compactness score is computed as the average of all C_i:

$$C = \frac{1}{I} \sum_{i=0}^{I-1} C_i.$$

DCI Explicitness. Explicitness quantifies how well the latent representation captures the generative factors in a way that enables accurate prediction. It is defined directly from the prediction errors, with the score for each factor v_i given by:

$$E_i = Err(v_i, \hat{v}_i).$$

The overall explicitness is computed as the average over all E_i:

$$E = \frac{1}{I} \sum_{i=0}^{I-1} E_i.$$

In contrast to modularity and compactness, lower values of explicitness correspond to better performance, since the metric is measured in terms of prediction error.

5

Experimental Setup

In this chapter, we describe the experimental setup used in this thesis. Section 5.1 introduces the datasets used in our study. Section 5.2 outlines the preprocessing steps that transform raw transactions into structured account-level samples. Section 5.3 presents the β-VAE architecture, training configuration, and objective function. Finally, Section 5.4 introduces the interpretability framework based on the DCI metrics, which we use to evaluate disentanglement in the learned representations.

5.1. Dataset

For this research, two datasets have been prepared: one representing predominantly normal business behavior, and a second consisting of accounts linked to financial economic crime, either through confirmed cases or based on behavioral patterns associated with known FEC typologies. It is important to note that all analysis is conducted at the account level rather than the client level, meaning each sample corresponds to an individual account rather than a client entity. Throughout this thesis, we refer to the dataset representing typical behavior as the Non-TP set, and the one containing confirmed or likely fraud cases as the TP set. The remainder of this section describes these two datasets in detail.

Non-TP set

The Non-TP set is derived from a large dataset provided by the bank, covering transactions from 1 January 2022 to 1 January 2025. It includes hundreds of millions of individual transactions, with associated metadata such as timestamps, transaction amounts, counterparty information, and categorical labels relevant to transaction type and context.

While the raw dataset is at the transaction level, the goal at this stage is to generate account-level samples suitable for modeling. The process begins by filtering out transactions involving natural persons and excluding accounts with unacceptable risk scores, as the focus of this study is on business behavior and predominantly normal activity. While some undetected criminal activity may still be present, domain experts estimate its proportion to be below 1%, which is considered acceptable for modeling purposes. After these filters, approximately 300 000 business accounts and their related transactions remained. From this pool, we generated 12-month samples by sliding a one-year window over each account's transaction history. Each sample begins on the first day of a calendar month, with start dates ranging from January 2022 to January 2024, resulting in 25 overlapping samples per account. To focus on active accounts, we applied further filters based on transaction count and total volume. This process resulted in approximately 6 million samples, each corresponding to a unique combination of account and 12-month transaction history.

TP set

The second dataset consists of transaction-level data from accounts linked to different types of financial economic crime (FEC). Some accounts are confirmed cases, while others were selected because their behavior matches known criminal patterns. The selection was carried out by a previous intern [34] using internal records, expert input, and targeted filtering. The dataset includes the following categories:

- **VAT Carousel Fraud**: a scheme in which companies exploit VAT rules on cross-border trade within the EU. Goods are repeatedly moved between entities, often across countries, and one or more companies in the chain retain VAT that should have been transferred to tax authorities.
- **Cash Compensation Model (CCM)**: a typology where employees are paid in cash rather than through formal payroll systems. This practice is more common in labor-intensive sectors and typically leads to high volumes of cash withdrawals.
- **Cash-Intensive**: characterized by unusually high levels of cash transactions, which may indicate unregistered business activity or cash-based money laundering.
- **High-Risk Geography (HRG)**: involves financial activity linked to countries identified as high-risk due to money laundering or sanctions concerns. These patterns may suggest involvement in cross-border money laundering or attempts to evade international sanctions.
- **Suspicious Activity Reports (SARs)**: based on internal compliance filings. SARs do not point to a single FEC typology but cover a broader range of unusual behaviors identified by compliance teams.

Table 5.1 summarizes the number of samples associated with each FEC typology included in the dataset.

Table 5.1: Number of samples per fraud typology

Typology	Number of Samples
VAT	132
CCM	138
CASH	168
HRG	181
SAR	192
Total	811

5.2. Preprocessing

Raw transaction data is not directly suitable for models that require structured, fixed-length input, because transactions occur at irregular intervals, and activity levels vary widely between accounts. To create a consistent structure, we followed a methodology developed by previous interns [31, 32, 33, 34], aggregating transactions into seven daily time slots (9:00, 10:00, 11:00, 12:00, 13:00, 14:00, and 16:00) aligned with business hours. For each measuring point x, all transactions that occurred between the previous point $x - 1$ and x were grouped together for feature extraction. Each 12-month sample is thus represented as a sequence of 2555 fixed time points (365 days $\times$ 7 slots).

At each of these time points, we compute a set of 20 features that capture different aspects of account behavior. The feature set was selected to capture general transaction patterns rather than hand-crafted FEC indicators. It includes account balance, credit and debit volumes by transaction type (cash, crypto, salary, tax) and geography (EU, non-EU, high-risk), as well as volumes linked to the top three counterparties and total credit and debit volumes. The full list is shown in Table 5.2.

Table 5.2: Feature set used at each time point. C = Credit, D = Debit.

Feature	C / D	Window size
Balance	–	–
Total	C & D	7 days
Cash	C & D	7 days
HRG	C & D	7 days
Crypto	C & D	7 days
Tax	C & D	31 days
Invoice	C & D	31 days
Salary	D	31 days
EU	C & D	31 days
Non-EU	C & D	31 days
TopCounterparties	C & D	31 days

A key challenge in working with transaction data is that many transaction types follow regular periodic schedules, yet appear sparse when viewed at individual time points. For example, salary payments and tax transactions typically happen only once per month. To address the sparsity problem and capture these periodic patterns, we used a sliding window approach. Each fixed-length window moves step by step along the time series and, at each position, it aggregates all values within that window and assigns the sum to the window's center point. Based on their temporal patterns, some features were aggregated using 31-day windows, while others were aggregated over 7-day windows. The specific window sizes used for each feature are listed in Table 5.2.

Following this temporal aggregation, we applied log normalization to the extracted features. This transformation reduces skewness in the data and balances the influence of large values, making the inputs more suitable for machine learning algorithms that assume approximately normal distributions.

As a result of the full preprocessing pipeline, we obtained approximately 6 million samples. Each sample summarizes a 12-month period as a sequence of 2 555 time points, with 20 features per time point, resulting in an input vector of 51 100 values. An example is provided in Figure 5.1.

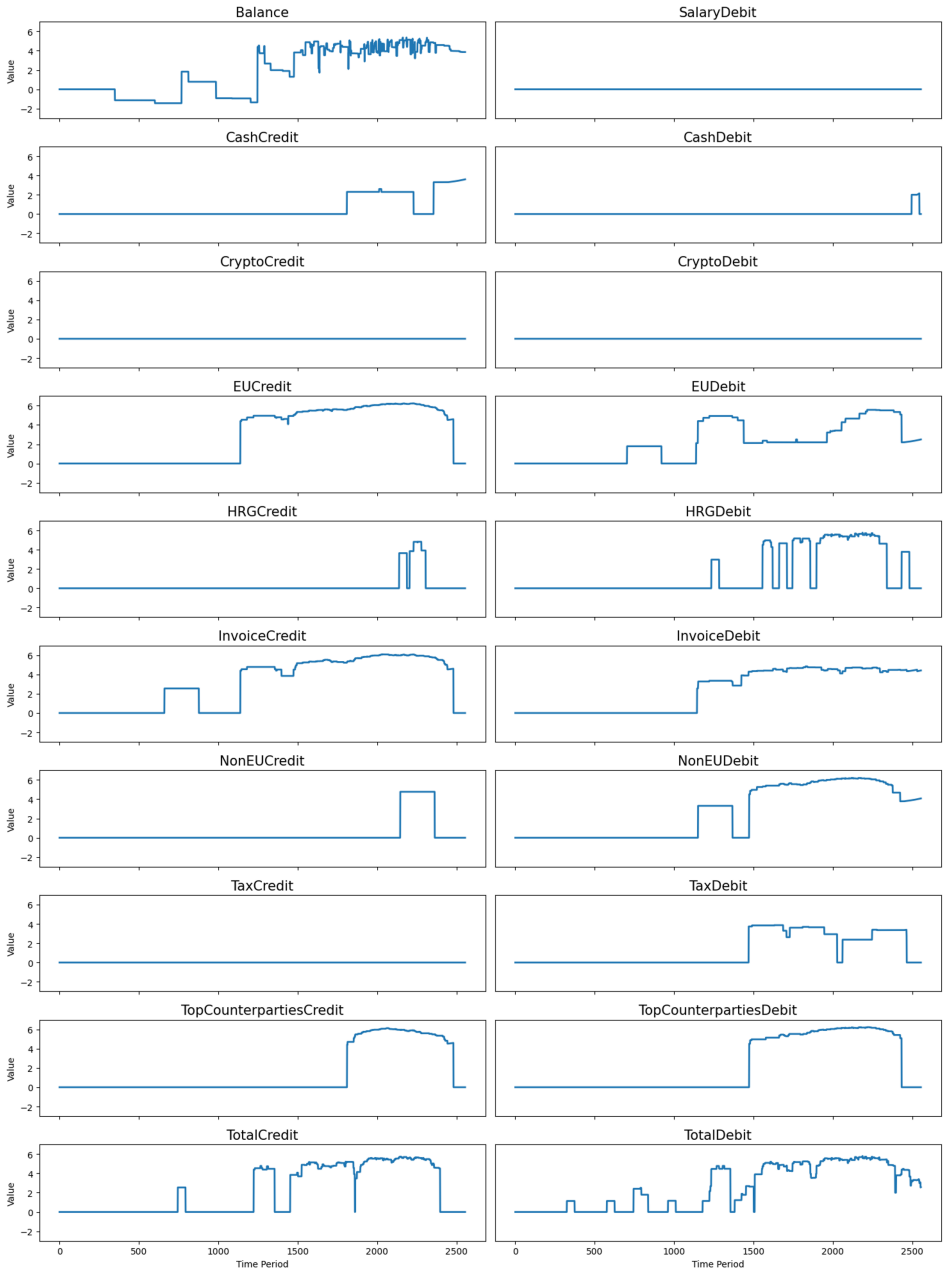


Figure 5.1: Example of one sample in the dataset. The figure shows the 20 features across the full 12-month period, with the x-axis indicating time (2,555 points) and the y-axis the corresponding feature values.

5.3. β-VAE

Our goal is to learn a disentangled representation of business account behavior using a β-VAE, and to study how disentanglement is affected by the value of β and the size of the latent space. To isolate these effects, all other aspects of the model are held constant across experiments, including the encoder and decoder architecture, the optimizer, and all training hyperparameters.

5.3.1. Architecture

The model configuration builds on work previously carried out by interns as part of a collaboration between Deloitte and the bank [31, 32, 33, 34]. This earlier setup demonstrated strong performance, and for this thesis, we adopt the same core architecture, which is illustrated in

Figure 5.2.

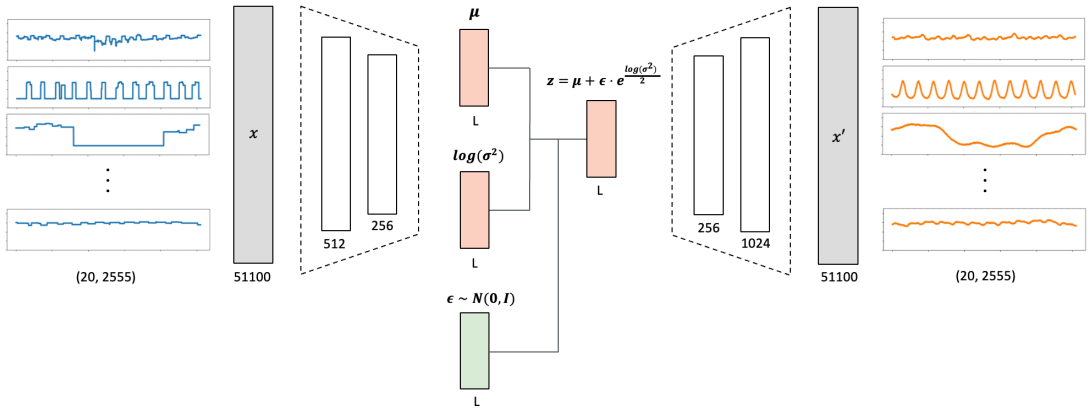


Figure 5.2: Overview of the VAE architecture. From left to right: input sample $\mathbf{x}$, encoder, latent sampling, decoder, and reconstruction $\mathbf{x}'$. The small plots illustrate one concrete sample and the output that the model reconstructs from it.

The encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$ and decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$ are implemented as feedforward neural networks. Each input sample, originally a matrix of 2555 time steps by 20 features, is flattened into a 51100-dimensional vector before being passed through the encoder. The encoder processes this input through two hidden layers with ReLU activations and outputs the parameters of a diagonal Gaussian distribution: a mean vector $\mu(\mathbf{x})$ and a log-variance vector $\log \sigma^2(\mathbf{x})$. The variance is modeled in log-space for numerical stability. Latent variables are then sampled using the reparameterization trick:

$$\mathbf{z} = \mu(\mathbf{x}) + \exp\left(\tfrac{1}{2} \log \sigma^2(\mathbf{x})\right) \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I).$$

The decoder takes the sampled latent variable $\mathbf{z}$ and reconstructs the input through two hidden layers followed by a linear output layer of size 51100. The output is then reshaped back into the original 2D form. A detailed summary of the architecture is provided in Table 5.3.

Table 5.3: Architecture of the encoder and decoder networks.

Component	Layer Type	Output Size	Activation
Encoder $q_\phi(\mathbf{z} \mid \mathbf{x})$	Input	(2555, 20)	–
	Flatten	51100	–
	Hidden Layer 1	512	ReLU
	Hidden Layer 2	256	ReLU
	Output Layer	$2 \times L$	–
Decoder $p_\theta(\mathbf{x} \mid \mathbf{z})$	Input Layer	L	–
	Hidden Layer 1	256	ReLU
	Hidden Layer 2	1024	ReLU
	Output Layer	51100	Linear
	Unflatten	(2555, 20)	–

5.3.2. Training Setup

Table 5.4 provides an overview of the training configuration.

All models are trained using the Adam optimizer [61], with a learning rate of 10^{-4}, a batch size of 64 and for 3 epochs. The dataset used for training contains approximately 600 000 samples, resulting in roughly 9400 updates per epoch and a total of about 28 000 training iterations. The only parameters that vary across experiments are the value of β and the dimensionality of the latent space L.

Table 5.4: Training configuration

Parameter	Value
Optimizer	Adam
Learning rate	10^{-4}
Batch size	64
Epochs	3
L	{25, 50, 75, 100}
β	$\{10^0, 10^1, 10^2, 10^3, 10^4\}$

5.3.3. Training Objective

Given a mini-batch of N samples $\{\mathbf{x}^{(n)}\}_{n=1}^{N}$, the objective function minimized during training is the expected negative Evidence Lower Bound (ELBO):

$$\mathcal{L}_{\beta\text{-VAE}} = \frac{1}{N}\sum_{n=1}^{N}\left[\|\mathbf{x}^{(n)} - \hat{\mathbf{x}}^{(n)}\|_2^2 + \beta \cdot \frac{1}{2}\sum_{j=1}^{L}\left(\mu_j^{(n)2} + \sigma_j^{(n)2} - \log\sigma_j^{(n)2} - 1\right)\right]. \tag{5.1}$$

The first term corresponds to the reconstruction error, while the second is the closed-form KL divergence derived in Section 3.3. Here, $\mathbf{x}^{(n)} \in \mathbb{R}^K$ and $\hat{\mathbf{x}}^{(n)} \in \mathbb{R}^K$ denote the input and its reconstruction, $K = 51100$ is the input dimensionality in our dataset, $\mu_j^{(n)}$ and $\sigma_j^{(n)2}$ are the mean and variance of the approximate posterior $q_\phi(z_j \mid \mathbf{x}^{(n)})$, L is the latent dimensionality, N is the batch size, and β controls the strength of the KL regularization.

5.4. Interpretability Framework

We develop an interpretability framework to quantify the relationship between latent variables and meaningful aspects of transaction behavior, building on the DCI metrics introduced in Section 4.3.1. Its overall structure is summarized in Figure 5.3. After encoding the dataset into latent representations with the VAE's encoder, the framework follows four stages: (i) training predictive models to map latent variables to factors, (ii) attributing relative importance scores to latent dimensions, (iii) evaluating predictive accuracy, and (iv) computing the DCI metrics.

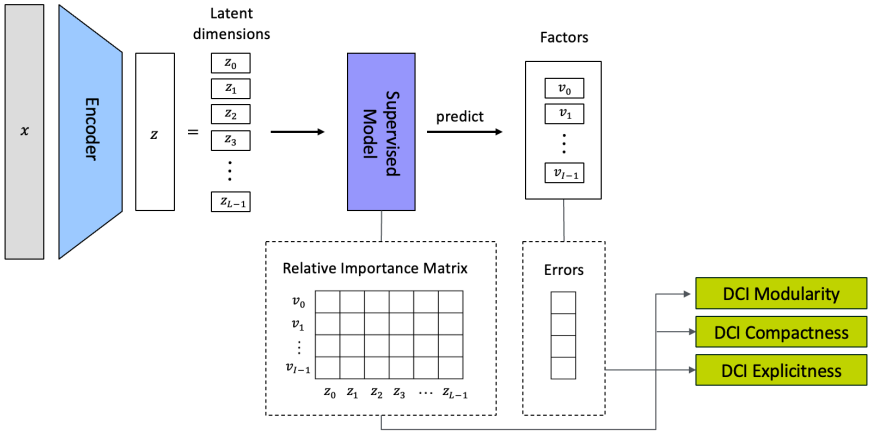


Figure 5.3: Overview of the DCI framework. The encoder maps inputs $\mathbf{x}$ to latent representations $\mathbf{z}$. Supervised models are then trained to predict the generative factors $\mathbf{v}$, from which the relative importance matrix and prediction errors are extracted to compute the DCI scores.

In what follows, we introduce the dataset and behavioral factors before describing the main

design choices of the framework.

5.4.1. Data

The evaluation relies on the subset summarized in Table 5.5, which is kept separate from the VAE training set to ensure evaluation on unseen data. Its latent representations $\mathbf{z}$ are obtained via the encoder and then divided into training and test subsets using an 80–20 split. The training split is used to fit the predictive models, and the test split is reserved for computing feature importance and prediction errors.

Table 5.5: Composition of the dataset used in the interpretability framework.

Typology	Number of Samples
VAT	132
CCM	138
CASH	168
HRG	181
SAR	192
Non-TP	28695
Total	29506

Factors

For the experiments in this thesis, we defined a set of factors that capture some relevant aspects of transaction behavior. These factors are deliberately simple and are not meant to be exhaustive, but they provide a practical way to connect the learned representation to interpretable patterns in the data.

The factors are computed as the average values of the input features over the observation period. Some of them are sparse, with distributions dominated by zeros. Because sparsity creates specific challenges for modeling, such factors are handled with a dedicated approach described in Section 5.4.2. In our experiments, however, the framework became unreliable in cases of extreme sparsity, since the factors showed too little variation across accounts to be predicted effectively. For this reason, we retained only factors that were nonzero in at least 10% of accounts. HRG and Crypto credit and debit did not meet this condition and were excluded, leaving 16 factors in total, shown in Table 5.6.

Table 5.6: Factors used in the experiments. Each factor is defined as the average value of the corresponding input feature over the observation period. The table also reports the proportion of nonzero accounts.

Factors	C/D	Nonzero (%)
Average Balance	-	100
Average Cash	C	16
	D	32
Average EU	C	40
	D	62
Average Invoice	C	79
	D	90
Average NonEU	C	10
	D	12
Average Salary	D	45
Average Tax	C	52
	D	84
Average TopCounterparties	C	100
	D	100
Average Total	C	100
	D	100

5.4.2. Predictive Model

The relationship between latent representations and factors is modeled using *gradient boosting*, an ensemble method that builds decision trees sequentially. At each iteration, a new tree is trained on the residuals of the current model, and its predictions are added to the ensemble. This iterative refinement yields models that are flexible enough to capture complex non-linear relationships between latent variables and factors. Such flexibility is important because, as Eastwood and Williams note [49], with generic priors such as the standard normal, factors may be encoded in distributed and non-linear ways across multiple latent dimensions. Linear models would underestimate disentanglement in this setting, while gradient boosting provides a more reliable mapping.

Factors are modeled in two different ways depending on their distribution. If more than 90% of values are nonzero, a standard gradient boosting regressor is used; otherwise, we use the two-stage approach described below.

Sparse Factors

Standard regression models struggle with sparse targets because the dominance of zeros drives the model toward predicting near-zero values across all samples. Although this reduces error on average, it fails to capture the variation in the nonzero cases that carry meaningful information.

To address this, the framework includes a *two-stage approach* inspired by hurdle models. The first stage is a binary gradient boosting classifier that predicts whether the factor is zero or nonzero. The second stage is a gradient boosting regressor trained only on the nonzero cases to estimate the magnitude of the factor. This separation isolates the zero inflation from the distribution of positive values, allowing each to be modeled more effectively.

When making predictions, the process works sequentially, as shown in Figure 5.4. Given an input x, the classifier C outputs a decision $C(x)$ indicating whether the factor is zero or nonzero. If $C(x) = 0$, the prediction is set to zero. If $C(x) = 1$, the regressor R is applied to the same input x, and the prediction is given by $R(x)$.

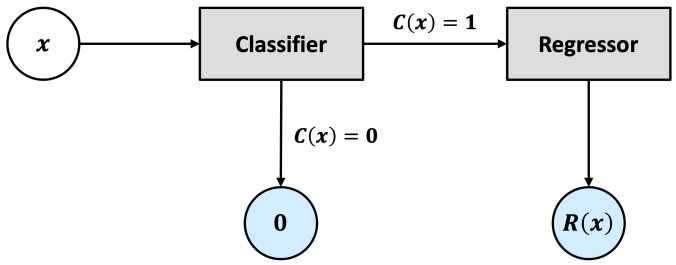


Figure 5.4: Sequential prediction process of the two-stage model: the classifier determines whether the factor is zero or nonzero, and if nonzero the regressor provides the estimate.

Hyperparameter Tuning

The performance of gradient boosting depends strongly on its hyperparameters. The most relevant ones are the number of trees, the maximum depth of each tree, and the learning rate, which together control the balance between model complexity and generalization. If the trees are too shallow or too few, the regressor may underfit, while an excessive depth or too many trees risk overfitting by memorizing the training data. The learning rate determines the size of each step when updating the model: a higher learning rate speeds up training, while a lower one slows it down but may improve stability. In our setup, we selected 200 trees with a maximum depth of 3 and a learning rate of 0.15, based on grid search with cross-validation.

5.4.3. Feature Importance

The framework quantifies interpretability by analyzing how information about each factor is distributed across the latent dimensions. This is achieved through feature attribution methods that assign an importance score to each latent variable in the predictive model. While several approaches exist, we adopt SHAP (SHapley Additive exPlanations) because it is both theoretically grounded and computationally efficient for tree-based models such as gradient boosting [62, 63].

SHAP explains each prediction by decomposing it into a baseline and a set of contributions from the latent variables. The baseline corresponds to the average prediction across the dataset, while each SHAP value ϕ_j represents how much latent variable j shifts the prediction up or down relative to that baseline. Formally, for a model f and input x, the prediction can be expressed as

$$f(x) = \phi_0 + \sum_{j=1}^{p} \phi_j,$$

where ϕ_0 is the baseline prediction and the ϕ_j are the contributions of the latent variables. By construction, these contributions always sum exactly to the prediction (*local accuracy*), and their values increase when a variable has a stronger effect on the model (*consistency*), ensuring that the importance scores faithfully capture how the model uses each latent dimension.

Sparse Factors

For sparse factors, predictions are produced by a two-stage model: a classifier $C(x)$ first determines whether the factor takes a zero or nonzero value, and a regressor $R(x)$ then estimates its magnitude in the nonzero case. The combined prediction can be written as

$$h(x) = C(x) \cdot R(x).$$

Interpreting such models requires more than analyzing the components separately, since the overall prediction depends on their interaction. To obtain meaningful importance scores, we use multiplicative SHAP (mSHAP) [64], which extends the SHAP framework to product models. In this setting, the prediction is decomposed into a baseline term and contributions from the latent variables,

$$h(x) = \mu_h + \sum_{j=1}^{p} \phi_j^h,$$

where $\mu_h = E[h(x)]$ is the expected output of the two-stage model. The contribution of latent variable j is given by:

$$\phi_j^h = \mu_C \phi_j^R + \mu_R \phi_j^C + \tfrac{1}{2} \sum_{a=1}^{p} \left(\phi_j^C \phi_a^R + \phi_j^R \phi_a^C \right) + \frac{|\phi_j^{h\,\prime}|}{\sum_{k=1}^{p} |\phi_k^{h\,\prime}|}\, \alpha,$$

where $\mu_C = E[C(x)]$, $\mu_R = E[R(x)]$, and ϕ_j^C, ϕ_j^R are the SHAP values from the classifier and regressor. The first two terms capture the direct effects of the classifier and regressor, while the cross-terms model their interactions. The final term distributes any residual α proportionally across variables, ensuring that the local accuracy property is preserved.

For both standard and two-stage models, global importance scores are computed by averaging the absolute local contributions across the dataset.

5.4.4. Error Function

To assess how accurately the factors can be predicted from the latent representations, we use the normalized root mean squared error (NRMSE). Given true values $\{y^{(n)}\}_{n=1}^{N}$ and predictions $\{\hat{y}^{(n)}\}_{n=1}^{N}$, it is defined as

$$\mathrm{NRMSE}(y, \hat{y}) = \frac{\sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(y^{(n)} - \hat{y}^{(n)} \right)^2}}{\sigma_y},$$

where σ_y is the standard deviation of the true values $\{y^{(n)}\}_{n=1}^{N}$. The normalization by σ_y makes the error scale-independent, which allows scores to be compared consistently across different factors.

6

Results

In this chapter, we present the experimental findings of this thesis. Section 6.1 explores the trade-off between reconstruction and regularization under different values of β and L. Section 6.2 investigates how information is distributed across dimensions through dimension-wise KL divergence. Finally, Section 6.3 uses the interpretability framework to evaluate the learned representations, with a focus on how β and L shape their structure.

6.1. Balancing Reconstruction and Regularization

The loss of a variational autoencoder combines two competing objectives: accurate reconstruction of the input data and regularization of the latent space. In this section, we study how these objectives play out in practice by tracking reconstruction error and KL loss during training and by comparing the final reconstruction error across different values of β and L. This is a first step toward understanding how the latent space encodes information and how design choices influence this encoding.

6.1.1. Training Dynamics

We tracked reconstruction error and KL divergence during training for all tested values of β and L. Since the qualitative trends were consistent across latent dimensionalities, Figure 6.1 reports the representative case of $L = 50$. All losses are computed on a separate validation set to avoid measuring performance on data the model was directly trained on.

For small values of β, reconstruction error remains low while KL loss is relatively high. As β increases, the two curves move in opposite directions: reconstruction error gradually rises, whereas KL loss steadily decreases. At very large values such as $\beta = 10^3$ and $\beta = 10^4$, this trend becomes extreme, with reconstruction error spiking sharply as KL loss approaches zero.

These trends can be explained by the relative weight of the two terms in the loss function. With small β, the reconstruction term dominates, so the model is encouraged to encode detailed information about the input in order to minimize reconstruction error. As β increases, the KL term carries more weight, and the model can reduce the loss more effectively by minimizing the difference between the posterior and the prior. This pressure increases the overlap across different inputs, which reduces the ability of the latent variables to encode distinct information and leads to higher reconstruction error. In the extreme case of very large β, the KL penalty overwhelms the reconstruction term, forcing the posteriors to collapse toward the prior and leaving the decoder with very limited information, which produces poor reconstructions.

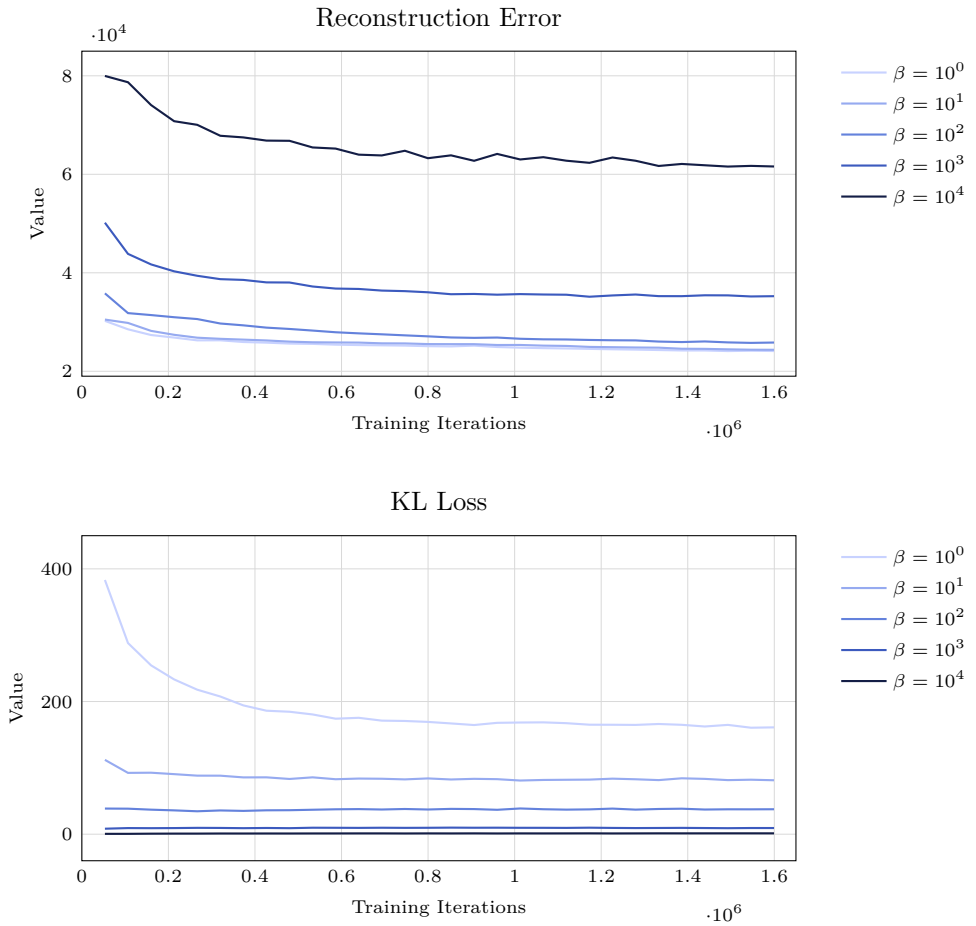


Figure 6.1: Training curves for models with latent dimensionality $L = 50$ and different values of β. The top panel shows reconstruction error, the bottom panel shows KL loss.

6.1.2. Reconstruction Quality After Training

To evaluate reconstruction quality after convergence, we compute the average reconstruction error on the validation set, defined as

$$\mathcal{L}_{\text{rec}} = \frac{1}{NK} \sum_{n=1}^{N} \|\mathbf{x}^{(n)} - \hat{\mathbf{x}}^{(n)}\|_2^2, \tag{6.1}$$

where $\mathbf{x}^{(i)} \in \mathbb{R}^K$ is the flattened input vector of dimensionality $K = 51\,100$, $\hat{\mathbf{x}}^{(i)}$ is its reconstruction, and N is the number of validation samples.

To study the effect of the hyperparameters, we evaluate this error across models trained with varying β and latent dimensionality L. Figure 6.2 summarizes these results in a heatmap over β (x-axis) and L (y-axis).

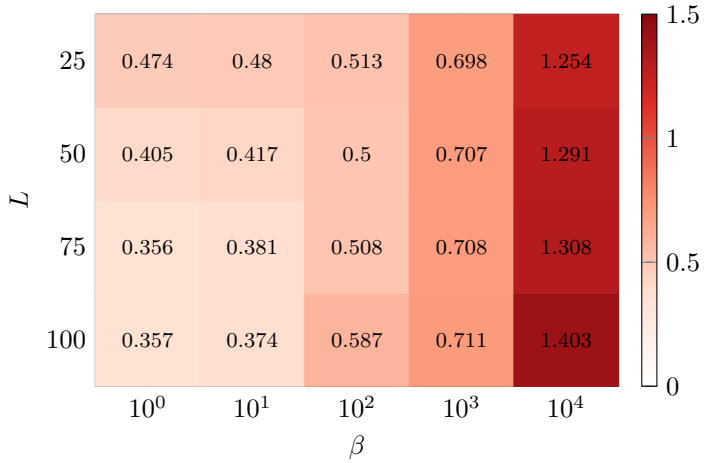


Figure 6.2: Average reconstruction error after training for models with latent dimensionality $L \in \{25, 50, 75, 100\}$ across $\beta \in \{10^0, 10^1, 10^2, 10^3, 10^4\}$.

We first examine the effect of β, corresponding to moving from left to right in the heatmap. For all latent dimensionalities, reconstruction error increases as β grows. The transition from $\beta = 10^0$ to $\beta = 10^1$ leads to only a modest rise in error, whereas for $\beta = 10^3$ and especially $\beta = 10^4$ the increase becomes substantial. This confirms the trend observed during training: stronger regularization limits the amount of information that can be encoded in the latent representation, reducing reconstruction quality.

Next, we examine the effect of latent dimensionality L, which increases from top to bottom in the heatmap. At low values of β, reconstruction error decreases as L grows. This reflects the additional capacity that a larger latent space provides: with more dimensions available and no strong constraint on the amount of information that can be encoded, the encoder can distribute the information more effectively and preserve finer details of the input, which leads to lower reconstruction error. However, this trend does not hold at higher values of β, where larger latent spaces instead lead to higher reconstruction error. A possible explanation is that once strong regularization forces the total KL to remain very low, the extra dimensions cannot be used effectively and may even amplify the penalty introduced by the KL term. We will explore this effect directly in the next section, by analyzing how activity is distributed across latent dimensions.

6.2. Activity of Latent Dimensions

In the previous section we noted that stronger regularization limits the overall KL divergence, reducing the total amount of information that can be encoded. To better understand how this reduction is achieved, we now examine KL divergence at the level of individual latent dimensions.

The KL term in the loss for a single datapoint $\mathbf{x}^{(i)}$ decomposes as:

$$\mathrm{KL}(q_\phi(\mathbf{z} \mid \mathbf{x}^{(i)}) \,\|\, p(\mathbf{z})) = \sum_{j=1}^{L} \mathrm{KL}(q_\phi(z_j \mid \mathbf{x}^{(i)}) \,\|\, p(z_j)) = \frac{1}{2} \sum_{j=1}^{L} \left(\mu_j^{(i)2} + \sigma_j^{(i)2} - \log \sigma_j^{(i)2} - 1 \right).$$

Since the training objective is defined as an average loss over datapoints, we follow the same principle here and compute the average per-dimension contribution across the dataset:

$$\bar{K}_j = \frac{1}{N} \sum_{i=1}^{N} \mathrm{KL}(q_\phi(z_j \mid \mathbf{x}^{(i)}) \,\|\, p(z_j)) = \frac{1}{2N} \sum_{i=1}^{N} \left(\mu_j^{(i)2} + \sigma_j^{(i)2} - \log \sigma_j^{(i)2} - 1 \right), \quad j = 1, \ldots, L,$$

which provides a dimension-wise measure of deviation from the prior.

The analysis was carried out for all combinations of $\beta \in \{10^0, 10^1, 10^2, 10^3, 10^4\}$ and latent dimensionalities $L \in \{25, 50, 75, 100\}$. Since the qualitative trends were consistent across settings, we report the case of $L = 50$ in Figure 6.3, while the results for other values of L are provided in Appendix B.1.

The results show that the total KL does not decrease uniformly across dimensions. At $\beta = 1$, all coordinates contribute, but as β increases, KL mass becomes increasingly concentrated in a few dimensions, while most approach zero. Quantitatively, the top five dimensions account for 13% of the total KL at $\beta = 10^0$, rising to 67% at $\beta = 10^3$ and 96% at $\beta = 10^4$.

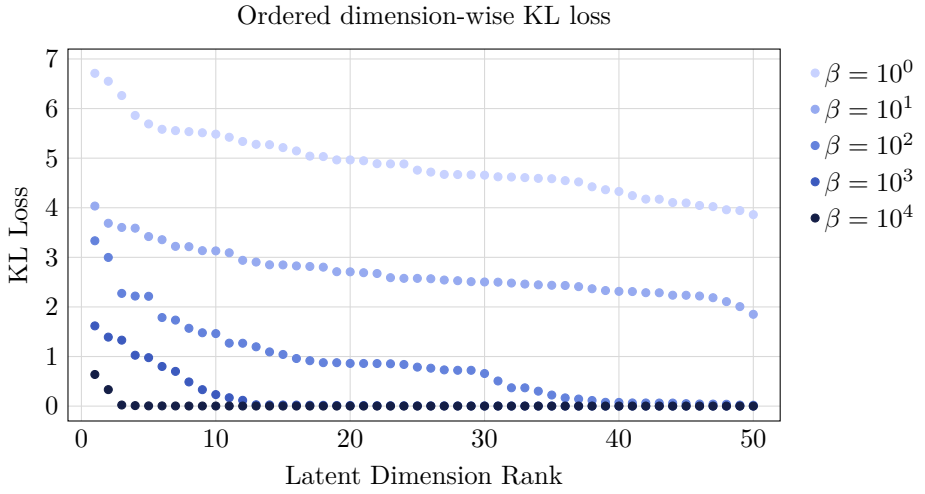


Figure 6.3: Average KL divergence per latent dimension for models trained with different values of β at latent size $L = 50$. Dimensions are ordered in descending KL.

Dimensions with higher values of $\bar{K}_j$ correspond to posteriors that deviate more strongly from the prior and remain distinct across datapoints. These dimensions preserve input-dependent variation and therefore provide discriminative power to the latent space. By contrast, dimensions with very small $\bar{K}_j$ map different inputs to overlapping posteriors, so they carry little information. In the limiting case of $\bar{K}_j = 0$, the posterior exactly matches the prior, meaning the encoder outputs the same distribution across all datapoints. This phenomenon is known as *posterior collapse* [65] and results in dimensions that are ignored by the decoder.

To quantify this effect, we adopt the (ϵ, δ)-criterion introduced by Lucas et al. [65], where a dimension is considered active if the KL divergence exceeds a threshold ϵ for at least a fraction

$1-\delta$ of datapoints. Following their approach, we set $\delta = 0.01$ and $\epsilon = 0.2$. Figure 6.4 shows how the number of active dimensions changes with β for different latent sizes L.

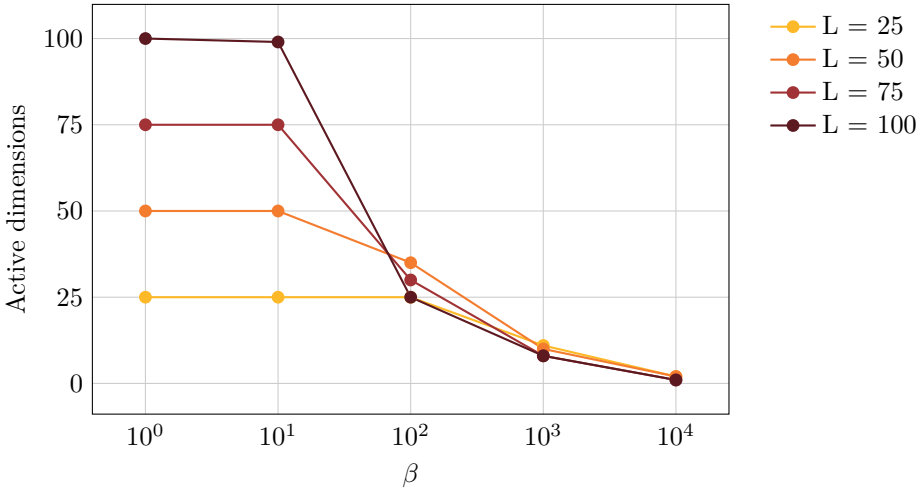


Figure 6.4: Number of active dimensions across β values for different latent sizes.

At $\beta = 10^0$ and $\beta = 10^1$, all available dimensions are active, indicating that the encoder distributes information broadly across the latent space when regularization is weak. Between $\beta = 10^1$ and $\beta = 10^2$, the number of active dimensions begins to drop, with larger latent spaces ($L = 75$ and $L = 100$) collapsing more sharply. From $\beta = 10^2$, the curves start to converge: regardless of the total latent size, only a small fraction of dimensions remain active. This shows that strong regularization limits the effective capacity of the latent space, and beyond a certain point the latent size L no longer influences how many dimensions are used. Interestingly, the effect can even reverse: at $\beta = 10^4$, models starting with $L = 25$ or $L = 50$ retain two active dimensions, whereas $L = 75$ and $L = 100$ collapse to only one.

Qualitative Analysis

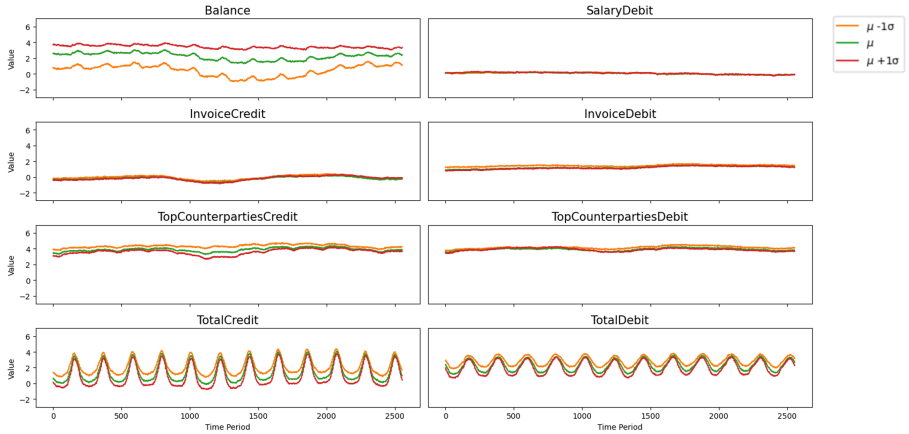
To illustrate the distinction between active and inactive dimensions, we perform latent traversals. A traversal consists of taking an input $\mathbf{x}$, encoding it to obtain the latent mean $\mu(\mathbf{x})$, and then varying a single coordinate z_j by a few multiples of its standard deviation,

$$z_j = \mu_j(\mathbf{x}) \pm k \cdot \sigma_j(\mathbf{x}),$$

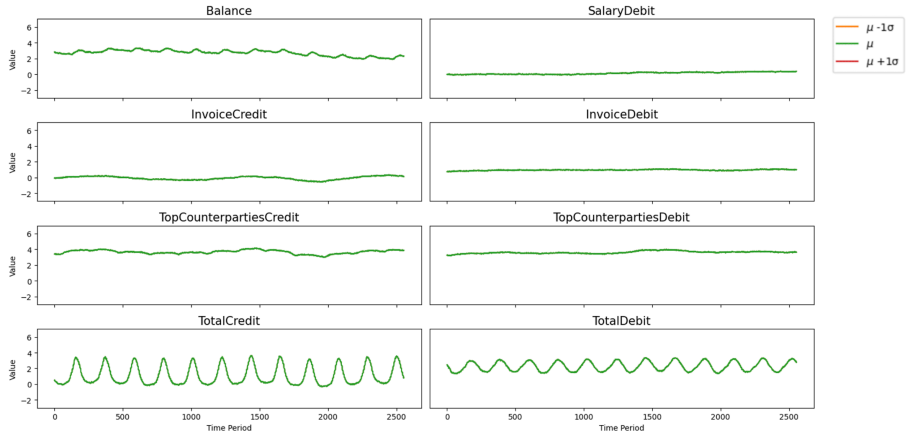
while keeping all other coordinates fixed. The modified latent vectors are then passed through the decoder to generate reconstructions. If the dimension is active, these reconstructions change in a systematic way that reflects the information encoded in z_j. If the dimension is inactive, varying its value has no effect and the reconstructions remain unchanged.

As an example, Figure 6.5 shows traversals from a model with $L = 50$ and $\beta = 100$. In Figure 6.5(a), varying the dimension with the highest KL divergence produces clear changes in the reconstructions: the *Balance* shifts upward or downward in magnitude, while *TotalCredit* and *TotalDebit* vary in amplitude. In contrast, Figure 6.5(b) shows the effect of traversing the dimension with the lowest KL divergence. In this case, the reconstructions overlap, suggesting that the dimension does not influence the output. For readability, only eight of the twenty features of a single sample are displayed; complete traversal results are provided in Appendix B.2.

This example reflects the general pattern observed across our experiments: active dimensions induce systematic changes in the reconstruction, while collapsed ones are effectively ignored by the decoder.



(a) Traversals along the highest KL dimension, producing visible changes in reconstruction.



(b) Traversals along the lowest KL dimension, with reconstructions perfectly overlapping.

Figure 6.5: Latent traversals from a model with 50 latent dimensions trained with $\beta = 100$.

6.3. Interpretability

The results on reconstruction and KL divergence showed the effect of regularization on the total amount of information the latent space can encode, while the study of latent activity revealed how this information is distributed across dimensions. We now turn to interpretability, examining how the latent space relates to the chosen factors and how β and L shape this relationship.

6.3.1. Effect of Regularization Strength β

Explicitness

Explicitness measures how accurately factors can be predicted from the latent variables. Lower values correspond to better predictions, with 0 indicating perfect accuracy and 1 matching the error of a constant mean predictor. The overall score E averages across all factors.

Figure 6.6 shows E as a function of β for latent dimensionalities $L \in \{25, 50, 75, 100\}$.

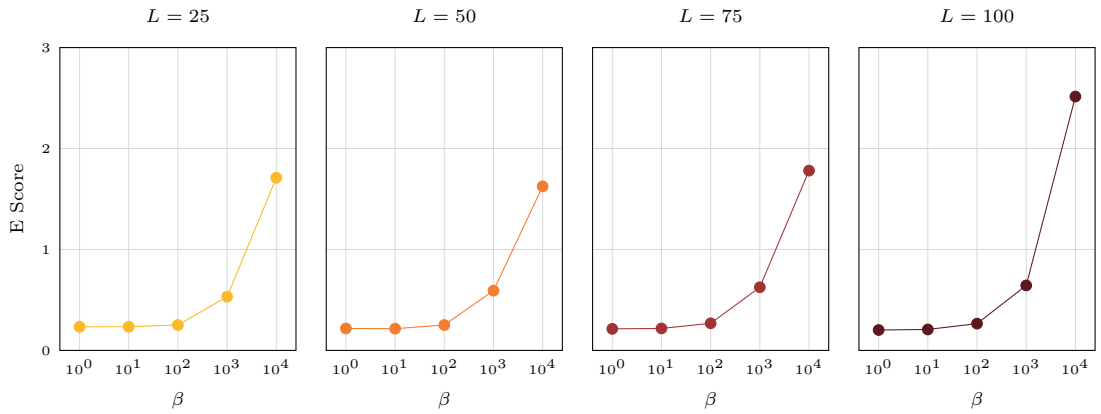


Figure 6.6: E Score versus β for $L \in \{25, 50, 75, 100\}$.

For each fixed latent dimensionality, we observe the same trend in explicitness as β increases. For small and moderate β values, the scores remain consistently low, indicating that the latent space retains enough information about the factors to keep prediction error low. This aligns with our earlier observation that weak regularization allows the model to preserve details in the representation. Up to a certain extent, β can be increased without causing a loss of factor information and harming explicitness. Beyond this range, however, the score rises sharply, particularly for larger latent dimensionalities.

To examine how this relates to reconstruction, Figure 6.7 plots the E score against the average reconstruction error reported in Section 6.1.2. Each panel corresponds to a latent dimensionality $L \in \{25, 50, 75, 100\}$, with points marking $\beta \in \{10^0, 10^1, 10^2, 10^3, 10^4\}$ (lighter to darker shades for increasing β).

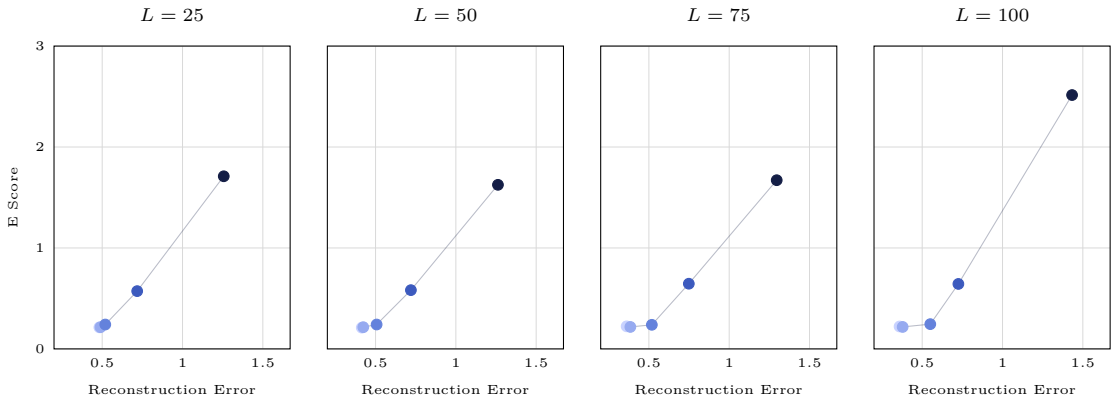


Figure 6.7: Relationship between reconstruction error and E score for different latent dimensionalities. Each point corresponds to a β value, from 10^0 (light) to 10^4 (dark).

The results for $\beta = 10^0$ and $\beta = 10^1$ nearly overlap, indicating that the latent space encodes a similar amount of information and weak regularization has little effect on either reconstruction or factor recovery. At $\beta = 10^2$, reconstruction error begins to increase, especially for larger latent dimensionalities, while explicitness stays nearly constant. This suggests that moderate regularization reduces detail in reconstructions but does not yet cause a loss of factor-related information. At very large β, both measures rise sharply, especially for larger L, reflecting that strong regularization forces the latent space to store less information overall and harms both reconstruction quality and factor recovery.

Modularity and Compactness

We now turn to modularity and compactness, two complementary metrics that describe how factor information is organized in the latent space.

Modularity measures the extent to which individual latent dimensions specialize in a limited number of factors. Figure 6.8 shows how modularity varies with β for different latent dimensionalities. The curves display a slight increase as β moves from low to intermediate values, followed by a clear decline when β becomes large. In other words, modularity peaks at a particular value of β, and this peak occurs at different positions depending on the latent dimensionality L.

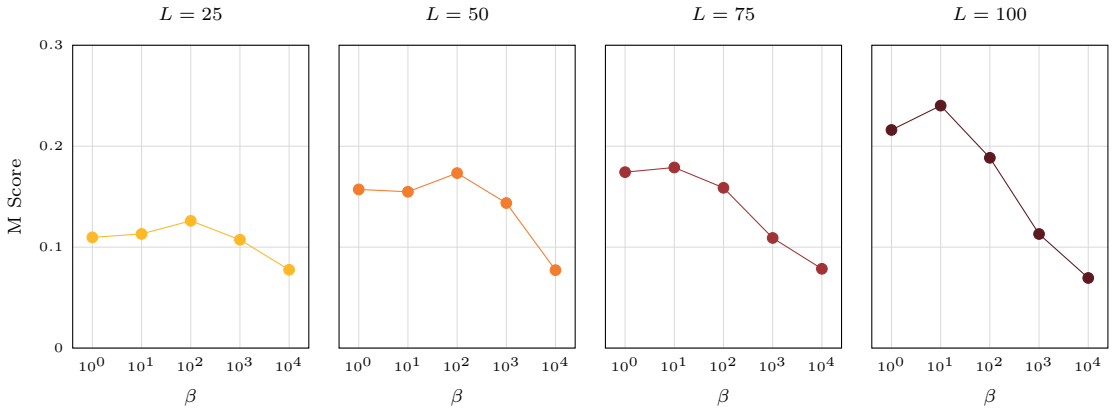


Figure 6.8: M Score versus β for $L \in \{25, 50, 75, 100\}$.

Compactness, in contrast, captures how concentrated or spread the factor information is across latent dimensions. Figure 6.9 illustrates that compactness rises more strongly when moving from low to intermediate β, before it too decreases at very large values. This indicates that compactness benefits more from moderate regularization than modularity does, although it is still reduced when regularization is very strong. As with modularity, the value of β that maximizes compactness depends on the latent dimensionality L.

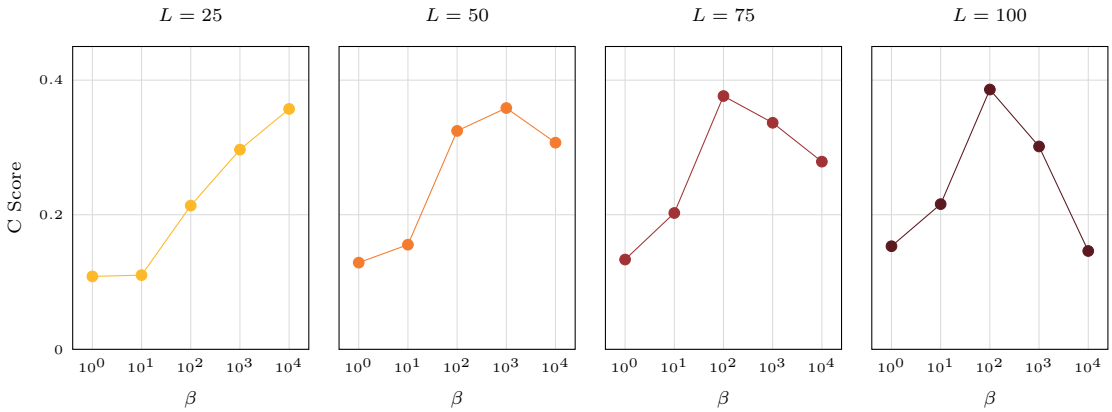


Figure 6.9: C Score versus β for $L \in \{25, 50, 75, 100\}$.

To better understand the effect of β on the two scores, we examine the full output of the DCI framework. While the detailed values depend on the latent dimensionality L, the qualitative effect of increasing β is consistent across settings. For clarity, we focus here on a subset of representative cases, while the complete set of results is provided in Appendix C.1. Figure 6.10 presents two examples for a latent dimensionality of $L = 50$, comparing $\beta = 10^0$ (top) with $\beta = 10^2$ (bottom).

At the center of each panel, the relative importance matrix indicates how much each latent dimension contributes to predicting each factor, with rows corresponding to factors and columns to latent variables. Alongside the matrix, the figures display the per-dimension and per-factor scores together with the aggregate values. At the bottom of each panel, we also report the per-dimension KL divergence, which provides a complementary view of latent activity.

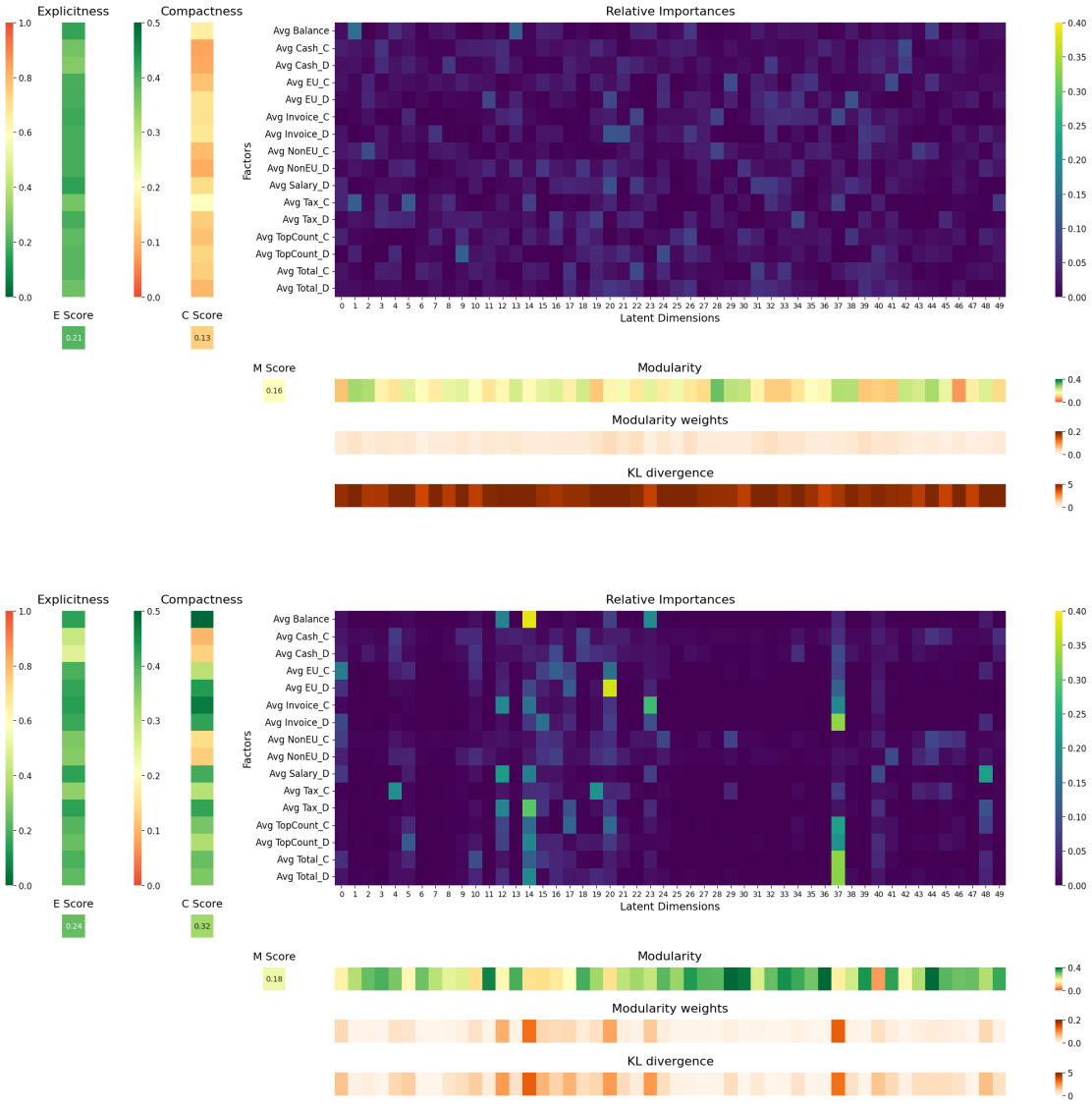


Figure 6.10: DCI outputs for a latent dimensionality of $L = 50$: $\beta = 1$ (top) and $\beta = 100$ (bottom). Each output includes the relative importance matrix and per-dimension scores for modularity, compactness and explicitness.

At $\beta = 10^0$, columns of the relative importance matrix show similar patterns, with contributions spread across several factors. Consequently, the corresponding modularity scores are relatively uniform across dimensions, and mostly between 0.10 and 0.20. The weights are also distributed evenly, indicating that all dimensions are used to a similar extent in prediction. At $\beta = 10^2$, some columns show much stronger contributions than others; for example, dimension 37 plays a clear role, while dimension 38 contributes very little. The weights are therefore more concentrated, with prediction relying more on a smaller subset of dimensions. The modularity scores are more spread out than at $\beta = 10^0$: some dimensions reach higher values and appear more specialized, while the most heavily used ones often encode several factors simultaneously and show less

improvement.

Turning to rows, at $\beta = 10^0$, the contributions are spread across many dimensions, leading to compactness scores that are both low and similar across factors. At $\beta = 10^2$, the rows become more concentrated, with most factors relying on fewer dimensions, and compactness scores rise accordingly. A small subset of factors, however, remains distributed across several coordinates, and since these deviations persist across values of β, we return to them in more detail at the end of this section.

Overall, moving from very low to moderate regularization produces clear changes in how information is distributed in the latent space. At very low β, the low and uniform modularity and compactness scores reflect a diffuse use of the latent space, where all dimensions contribute but none are clearly specialized. This is consistent with the dimension-wise KL divergence, which is high and broadly distributed across dimensions. As regularization increases, modularity rises modestly as some dimensions start to specialize, while compactness improves more substantially as factors become concentrated in fewer coordinates. In line with this, the KL divergence also shifts from being broadly distributed to concentrated in the same subset of dimensions that carry most of the predictive weight.

Finally, we explore the limit case of extreme regularization. For consistency, we fix the latent dimensionality at $L = 50$ and present the results for $\beta = 10^4$ in Figure 6.11.

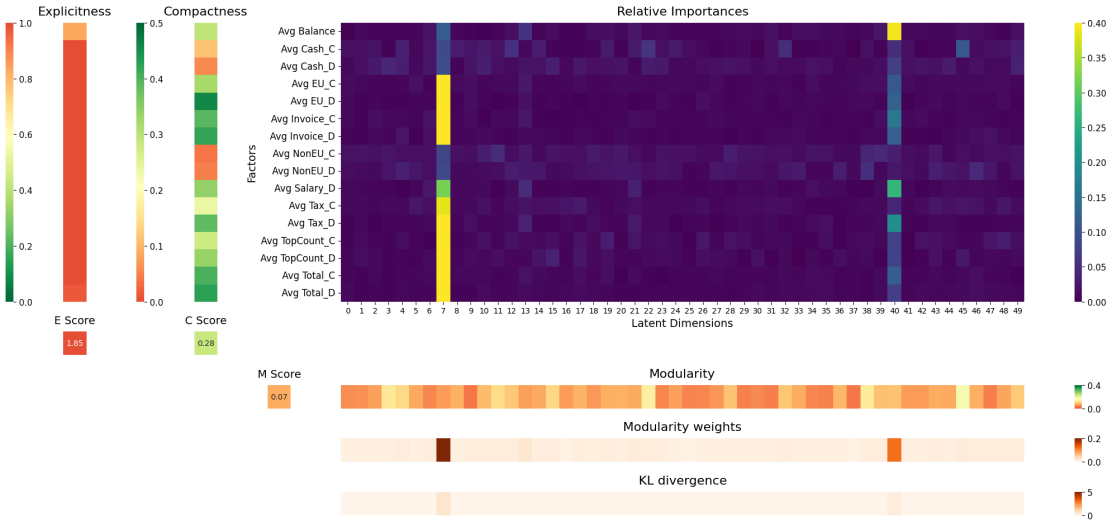


Figure 6.11: DCI output for a latent dimensionality of $L = 50$ with $\beta = 10^4$.

The relative importance matrix shows that nearly all information is concentrated in two latent dimensions (7 and 40). This is confirmed by the Modularity weights and consistent with the KL divergence, which shows non-negligible values only for these two coordinates. Compactness decreases, especially for a subset of factors, and modularity drops sharply. Inactive dimensions contribute small amounts to many factors without aligning clearly with any, while active dimensions become overloaded, encoding several factors at once.

Thus, extreme regularization concentrates all information into a few active dimensions, limiting the extent to which these can be tied to specific aspects of behavior.

6.3.2. Effect of Latent Dimensionality L

Explicitness

Figure 6.12 compares E scores across latent dimensionalities by plotting them together as a function of β.

At low β values, explicitness scores are similar across latent sizes. This is somewhat counterintu-

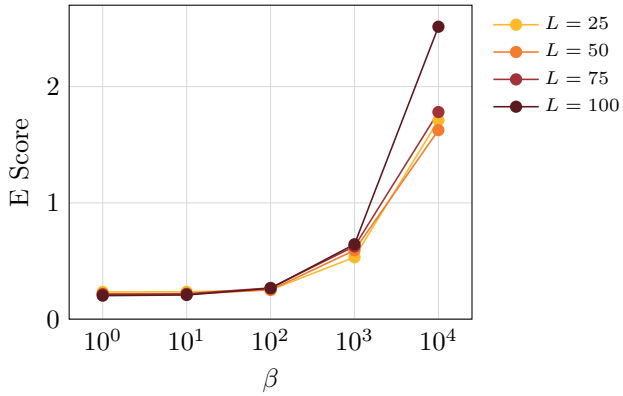


Figure 6.12: E Score versus β for $L \in \{25, 50, 75, 100\}$.

itive, since one might expect that larger latent spaces combined with weak regularization would allow the model to encode more information about the factors, leading to lower prediction error. The curves, however, show that this additional capacity does not translate into better scores. Reconstruction error, on the other hand, does change, as also seen in Figure 6.2. For example, at $\beta = 10^0$, the error is 0.474 with $L = 25$ compared to 0.357 with $L = 100$, showing that larger latent spaces are indeed used to encode more information. One possible explanation, linked to the framework, is that the predictor may struggle when the latent space has many dimensions, making it harder to use the extra information effectively. Another, related to the latent space itself, is that the additional capacity may be used to encode details not directly tied to the chosen factors, which improves reconstruction but not prediction.

At high β values, the difference between small and large latent spaces becomes evident, with explicitness rising more sharply in the larger ones. This trend is consistent with the reconstruction error, which is also higher for larger latent spaces under strong regularization. As seen in Section 6.2, with high β only a few dimensions remain active, regardless of the total latent size. At $\beta = 10^4$, the effect even reverses: larger latent spaces ($L = 75$ and $L = 100$) collapse to a single active dimension, while smaller ones ($L = 25$ or $L = 50$) retain two. A possible explanation is that inactive dimensions may still contribute a small cost in the KL term, and when the latent space is large these contributions accumulate. This could reduce the effective amount of information that can be allocated to the active dimensions, leading to less efficient use of the latent space.

Modularity and Compactness

Figure 6.13 present modularity and compactness as functions of β for latent dimensionalities $L \in \{25, 50, 75, 100\}$.

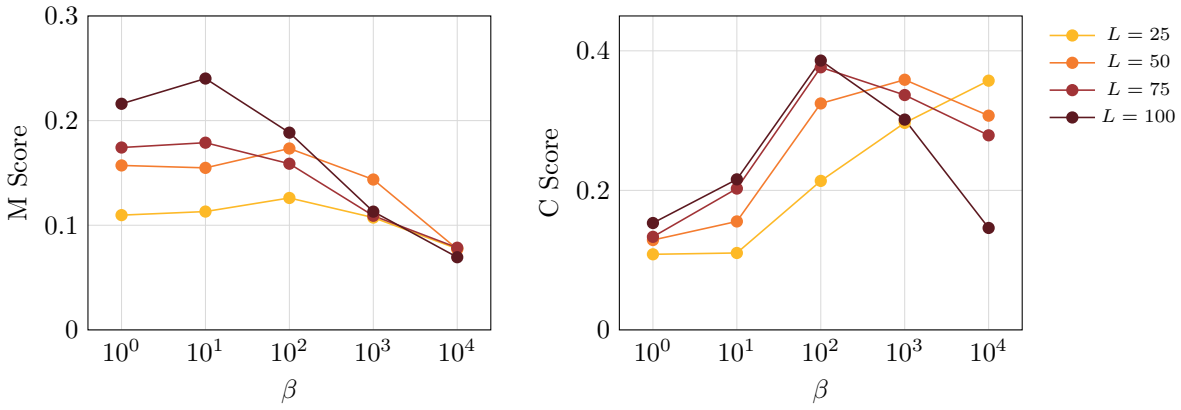


Figure 6.13: M Score and C Score versus β for $L \in \{25, 50, 75, 100\}$.

For modularity, larger latent spaces ($L = 75, 100$) reach higher scores than smaller ones ($L = 25, 50$) at low and intermediate values of β, indicating that having more dimensions enables individual latent units to specialize more effectively. The curves also reveal that the position of the peak shifts with latent size: larger spaces achieve their maximum at lower β, while smaller ones peak later and at lower levels. This suggests that when more dimensions are available, only mild regularization is needed for them to specialize, whereas smaller spaces require stronger pressure. At very high β, however, modularity declines for all models and the differences between latent sizes largely disappear. A possible explanation is that at $\beta = 10^3$ and $\beta = 10^4$ the number of active dimensions becomes similar across latent sizes, so the few surviving coordinates are the only ones contributing and all end up encoding many aspects of behavior at once. As a result, differences in nominal capacity no longer matter, and modularity converges to similar values across models.

Compactness shows broadly similar peak values across latent sizes, but again the peak shifts with L: larger latent spaces reach their maximum at lower β, whereas smaller ones peak later. Beyond the peak, compactness declines for most latent sizes, with the drop being sharper for $L = 100$. In contrast, for $L = 25$ compactness continues to rise across the explored range, suggesting that its maximum may lie beyond the values shown.

To better understand how modularity and compactness relate to one another, Figure 6.14 plots both metrics against the total KL divergence. Colors represent increasing β from light to dark, so the figure should be read from right (high KL, corresponding to low β) to left (low KL, corresponding to high β).

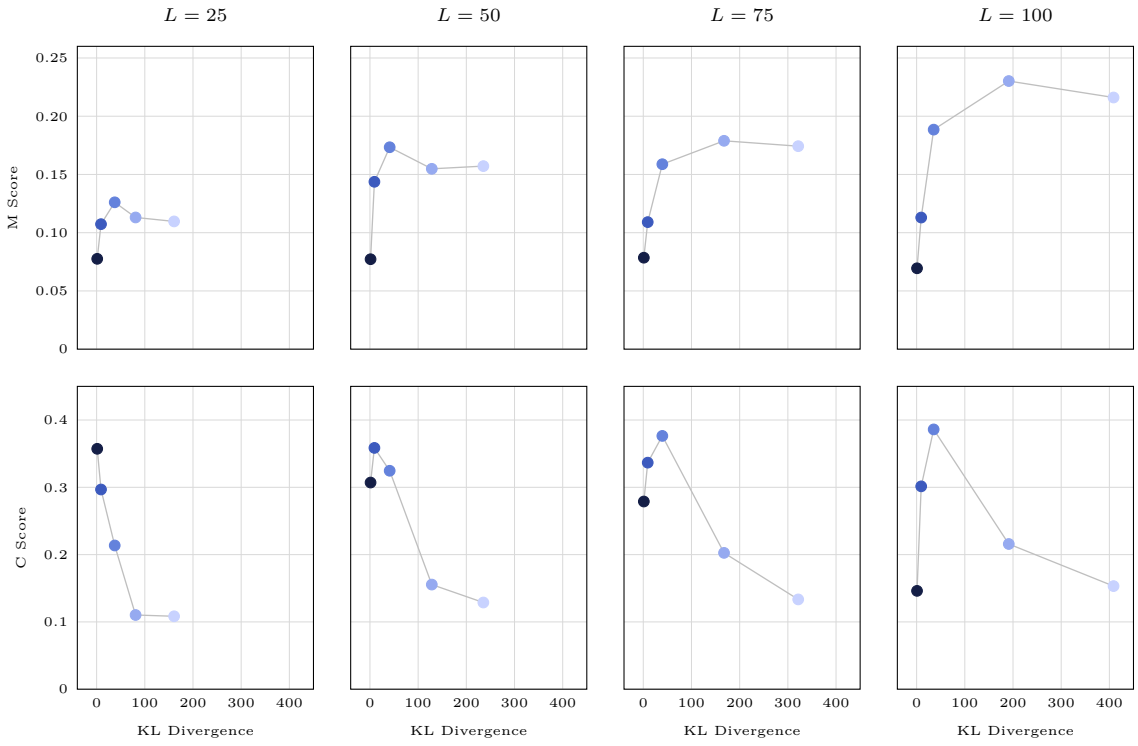


Figure 6.14: Modularity (top) and compactness (bottom) across latent sizes $L \in \{25, 50, 75, 100\}$, plotted against total KL divergence. Colors indicate increasing β from light to dark.

A consistent offset is visible across all latent sizes: compactness continues to rise as KL decreases, beyond the point where modularity reaches its maximum and begins to decline. This offset is most pronounced for larger latent spaces, where compactness peaks at distinctly lower KL than modularity, while for smaller spaces the two maxima lie closer together. These results suggest that compactness and modularity are driven by slightly different regimes of regularization. The region that maximizes compactness may already involve a trade-off in modularity, and vice versa.

Factor-specific behavior

Most factors follow the same qualitative behavior across different levels of regularization. However, a subset deviates from this pattern: they remain distributed across several dimensions even when compactness increases for most others. To illustrate this effect, Figure 6.15 reports the compactness scores of individual factors as a function of β for the case $L = 50$. The results for other latent dimensionalities are consistent and provided in Appendix C.2.

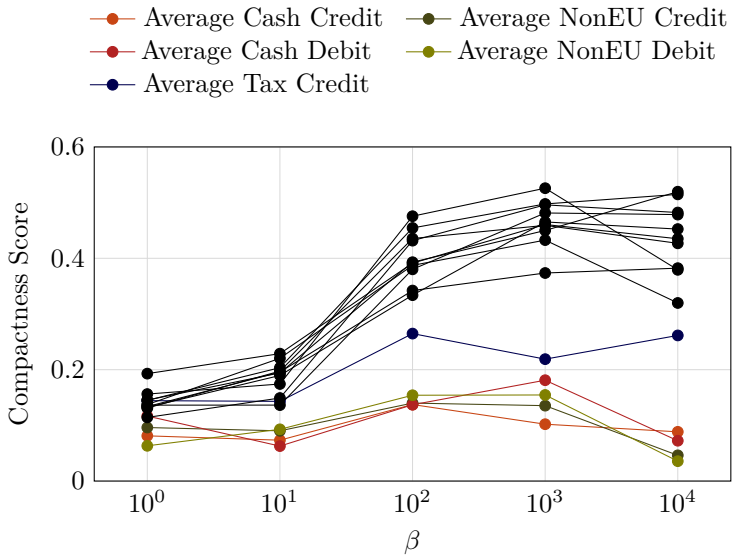


Figure 6.15: Compactness scores of individual factors as a function of β for latent dimensionality $L = 50$. The majority of factors follow a similar trend and are shown in black, while the five factors with distinct behavior are highlighted in color and listed in the legend.

The figure shows that most factors increase in compactness from low to intermediate values of β and slightly decrease again under strong regularization, consistent with the aggregate trend. In contrast, five factors display consistently lower scores across all settings.

Interestingly, these factors are also very sparse, which may help explain why their information is more difficult to concentrate. Figure 6.16 shows the relationship between sparsity (measured as the proportion of zero entries) and factor-wise compactness at $\beta = 10^4$ for $L = 50$.

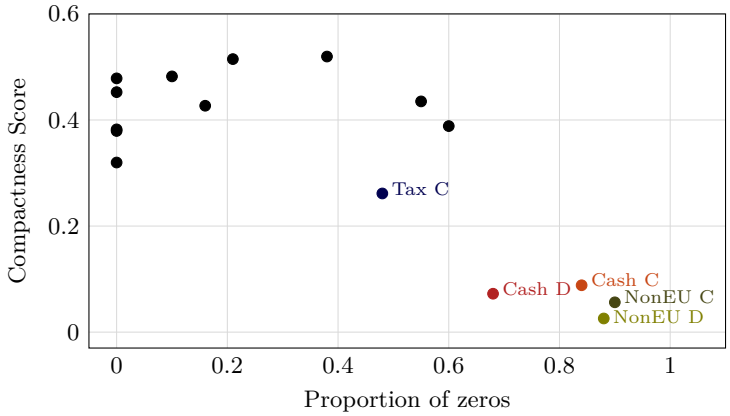


Figure 6.16: Relationship between factor sparsity and compactness at $\beta = 10^4$ for latent dimensionality $L = 50$. The x-axis shows the proportion of zero entries in each factor, and the y-axis shows the corresponding compactness score.

One possibility is that the framework becomes less reliable when applied to very sparse factors,

since limited variation makes it harder for predictive models and attribution methods to capture their contribution consistently. Another possibility is that the deviation reflects a property of the latent space itself. Latent dimensions may be more likely to specialize in signals that explain a large share of the overall variation. Sparse factors may be too weak to drive such specialization, leaving their information dispersed and less responsive to regularization. Whether this pattern reflects properties of the VAE or limits of the framework remains uncertain and would require further investigation.

7

Conclusion

This thesis addressed the problem of interpretability in variational autoencoders (VAEs) applied to transaction data. The latent space of a VAE offers a compact representation of account behavior that can support tasks such as anomaly detection, clustering, and risk profiling, which are particularly relevant for financial crime detection. For such representations to be useful in practice, however, they must also be interpretable. Disentanglement offers a promising path, encouraging latent dimensions to capture distinct and meaningful aspects of the data, but systematic ways to measure it in this domain are still missing.

Building on this motivation, the thesis had two objectives. The first was to design a framework that quantifies how latent dimensions of a variational autoencoder relate to distinct aspects of behavior. The second was to apply this framework to study how model parameters influence disentanglement and, in turn, interpretability. To address the first objective, we introduced an interpretability framework based on the DCI metrics. It uses behavioral factors as reference points to compute explicitness, modularity, and compactness scores. These scores capture how information about the factors is represented in the latent space, both in aggregate and at a more detailed level. To address the second objective, we evaluated disentanglement against a chosen set of behavioral factors for different values of β and L.

The experiments showed clear trade-offs. Weak regularization preserved reconstruction accuracy and explicitness, but produced diffuse representations without clear specialization. Intermediate values of β balanced reconstruction quality with a more structured and specialized latent space, although modularity and compactness peaked at different points. Very high values of β, however, caused the latent space to collapse, leaving only a few active dimensions and reducing interpretability overall. Latent dimensionality played a complementary role: larger spaces improved reconstruction at low β but did not enhance factor recovery, and under strong regularization they collapsed more sharply than smaller spaces. Moreover, the position of the peak in modularity and compactness depended on L: larger spaces peaked at lower values of β, whereas smaller spaces needed stronger regularization.

Within the scope of the behavioral factors considered, this study shows that there is no single optimal setting for β and L. Instead, the framework highlights parameter regimes that depend on the goal. If the priority is to capture fine detail, lower values of β combined with larger latent spaces allow subtle variations in the data to be retained, resulting in higher reconstruction quality. The trade-off is that the latent representation lacks clear organization, so relationships with interpretable aspects of behavior are difficult to trace. If the priority is interpretability, intermediate values of β offer a better balance: several dimensions stay active, explicitness is stable and the latent space becomes more structured, with clear gains in modularity and compactness. In this case, moderate latent dimensionalities are most practical, since they support specialization without creating an excessive number of dimensions to explain. Even within this

range, however, the most suitable choice depends on which aspect is most important, since modularity and compactness peak at different values. The framework makes these trade-offs explicit, allowing design choices to be aligned with the priorities of the task rather than aiming for a single universal optimum.

The findings are tied to the behavioral factors used in this study, which are not meant to be exhaustive or universal. What is general is the framework itself, which can be applied with any chosen set of factors. By combining complementary metrics, it captures different aspects of disentanglement and provides a systematic way to connect latent dimensions with interpretable aspects of behavior. The contribution is therefore twofold: the empirical insights gained here, and the framework as a tool to guide the design of VAEs. Beyond this work, it may serve as a step toward more interpretable latent spaces, supporting the adoption of VAEs in financial crime detection.

7.1. Limitations

The limitations of this thesis concern both the interpretability framework and the empirical findings, and they should be considered when evaluating the contributions of this work.

First, the framework builds on the DCI metrics, which are sensitive to design choices, as discussed in Section 4.3. These metrics rely on a supervised predictive model, and both the choice of model (for example, gradient boosting versus linear models) and its hyperparameters influence how well factor-latent relationships are captured. The choice of feature importance method and error function further shape the reported values. The scores should therefore be interpreted as relative comparisons across settings, in line with the framework's goal of highlighting trade-offs and guiding modeling choices, rather than as absolute measures of disentanglement.

Second, the interpretation of the results is limited by the choice of behavioral factors used as reference points. Unlike the independent ground-truth factors often assumed in disentanglement research, the descriptors used here reflect selected aspects of account behavior. They are not exhaustive, not independent, and unlikely to represent the full set of underlying processes that shape transaction activity. Their non-independence is particularly relevant: when two factors share information, a latent dimension that aligns with one may also appear linked to the other, which can blur the interpretation of disentanglement and reduce modularity scores. Sparsity adds another challenge, since factors with few nonzero values appeared less compactly represented and were harder to associate with individual dimensions. This may reflect limits of the framework, whose predictive model and metrics may become less reliable when factor variation is scarce, or it may suggest that such sparse factors are simply not suitable candidates for representation in the latent space. Overall, the findings should be read as showing how latent dimensions align with this particular set of descriptors, without claiming a complete explanation of the latent structure.

More broadly, this reflects a central challenge for interpretability. Fully explaining a latent space requires first defining candidate behavioral factors and then examining how they relate to individual dimensions. With complex data, however, identifying suitable descriptors is itself difficult, and some dimensions may capture patterns that cannot be reduced to simple human-understandable explanations. The framework provides a structured way to investigate these links, while recognizing that parts of the latent space may never map cleanly to interpretable concepts.

7.2. Future Work

Several directions for future work follow from these limitations.

A first line of research is to explicitly account for statistical dependence between factors. Recent work proposes to redefine core disentanglement properties from an information-theoretic perspec-

tive, adapting them to cases where factors are not independent [66]. This would be particularly useful in the present setting, where dependencies among descriptors are expected and cannot be fully avoided.

A second direction concerns the choice of latent dimensionality. The experiments showed that larger latent spaces improved reconstruction in some regimes but collapsed more sharply under strong regularization. Smaller spaces were more stable, yet they required stronger regularization to reach their peak in modularity and compactness, whereas larger ones peaked earlier. The underlying causes of these patterns are not fully understood, and future research should investigate them to establish clearer guidelines for selecting latent dimensionality in practice. Beyond manual tuning, systematic methods that adapt dimensionality during training, such as ARD-VAEs [67], represent a promising alternative.

A further direction is to broaden the evaluation by considering additional factors. The descriptors used in this study provided useful reference points but remain simple and limited in scope. Introducing richer descriptors could help validate whether the observed patterns hold more generally and reveal stronger alignments with latent dimensions. In addition, the experiments showed that sparse factors were represented less compactly, raising the question of whether this reflects properties of the latent space itself or of the framework used to measure factor-latent relationships. Future work could examine the impact of factor distributions by systematically varying sparsity, comparing dense and sparse descriptors, or analyzing how distributional properties such as skewness affect the scores.

Finally, it remains important to clarify which aspects of interpretability are most relevant in practice. The results showed that modularity and compactness peaked at different values of β, indicating that the two properties do not always move together. Future research could examine how these trade-offs align with the needs of specific applications in financial crime detection, since some tasks may benefit from stronger modularity while others may require more compact representations.

These open questions highlight both the complexity of the problem and the opportunities for further research in this area.

References

[1] United Nations Office on Drugs and Crime. *Estimating Illicit Financial Flows Resulting from Drug Trafficking and Other Transnational Organized Crimes*. 2011. URL: https://www.unodc.org/.

[2] Diederik P Kingma and Max Welling. "Auto-Encoding Variational Bayes". In: *arXiv preprint arXiv:1312.6114* (2014). URL: https://arxiv.org/abs/1312.6114.

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives". In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[4] Irina Higgins et al. "beta-VAE: Learning basic visual concepts with a constrained variational framework". In: *International Conference on Learning Representations (ICLR)* (2017).

[5] Financial Conduct Authority. *Financial Crime: A Guide for Firms*. 2021. URL: https://www.fca.org.uk/.

[6] Financial Action Task Force. *Money Laundering and Terrorist Financing: FATF Report*. 2017. URL: https://www.fatf-gafi.org/.

[7] Management Solutions. *Financial Crime: Challenges and Trends in the Digital Era*. 2023. URL: https://www.managementsolutions.com/.

[8] Fiserv. *Gain a Strategic Advantage in the Fight Against Financial Crime*. 2021. URL: https://www.unodc.org/unodc/en/money-laundering/overview.html.

[9] *Wet ter voorkoming van witwassen en financieren van terrorisme (Wwft)*. Accessed: 2025-08-30. 2018. URL: https://wetten.overheid.nl/BWBR0024282/.

[10] De Nederlandsche Bank. *From Recovery to Balance: A look ahead to a more risk-based approach to preventing and combating money laundering and terrorist financing*. Amsterdam, The Netherlands: De Nederlandsche Bank, 2022.

[11] Deloitte. *NextGen AML: Transforming Financial Crime Detection*. Deloitte Insights, 2020.

[12] Alireza Makhzani and Brendan J Frey. "k-Sparse autoencoders". In: *International Conference on Learning Representations (ICLR) Workshop*. 2014.

[13] Pascal Vincent et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion". In: *Journal of Machine Learning Research* 11 (2010), pp. 3371–3408.

[14] Salah Rifai et al. "Contractive auto-encoders: Explicit invariance during feature extraction". In: *Proceedings of the 28th International Conference on Machine Learning (ICML)*. 2011, pp. 833–840.

[15] Jinwon An and Sungzoon Cho. "Variational autoencoder based anomaly detection using reconstruction probability". In: *Proceedings of the Special Lecture on IE*. 2015, pp. 1–18.

[16] Toan Van Nguyen et al. "Variational autoencoder based anomaly detection for novel attack detection in network intrusion detection systems". In: *Information Sciences* 511 (2019), pp. 174–190.

[17] Rafael Pereira et al. "Anomaly detection in ECG time signals via variational autoencoder latent space analysis". In: *Computers in Biology and Medicine* 104 (2019), pp. 64–74.

[18] Hilal Hilal et al. "Financial Fraud: A Review of Anomaly Detection Techniques and Recent Advances". In: *Applied Sciences* 12.5 (2022), p. 2345.

[19] Attar Pumsirirat and Liu Yan. "Credit Card Fraud Detection using Deep Learning based on Auto-Encoder and Restricted Boltzmann Machine". In: *2018 11th International Conference on Advanced Computational Intelligence (ICACI)*. IEEE. 2018, pp. 128–132.

[20] Erik Sweers, Tom Heskes, and Jesse Krijthe. "Comparison of Variational and Classical Autoencoders for Credit Card Fraud Detection". MA thesis. Radboud University, 2018.

[21] Emil Renström and Axel Holmsten. "Detecting Credit Card Fraud using Autoencoders". MA thesis. Lund University, 2018.

[22] IMANE Karkaba, EM Adnani, and M Erritali. "Deep Learning Detecting Fraud in Credit Card Transactions". In: *Journal of Theoretical and Applied Information Technology* 101.9 (2023).

[23] Fawaz Alarfaj, Abdullah Alshamrani, and Faisal Abdu. "A Suspicious Financial Transaction Detection Model Using Auto Encoder and Risk Based Approach". In: *Journal of Information Security and Applications* 54 (2020), pp. 102–116.

[24] Amol Shende and Sandhya Sontakke. "Real-Time Fraud Detection in Financial Transactions Using Autoencoders". In: *2021 International Conference on Machine Learning and Data Science (ICMLDS)*. IEEE. 2021, pp. 45–51.

[25] Tzu-Hsuan Lin and Jehn-Ruey Jiang. "Credit card fraud detection with autoencoder and probabilistic random forest". In: *Mathematics* 9.21 (2021), p. 2683.

[26] Haichao Du et al. "AutoEncoder and LightGBM for credit card fraud detection problems". In: *Symmetry* 15.4 (2023), p. 870.

[27] HaiChao Du et al. "A novel method for detecting credit card fraud problems". In: *PloS one* 19.3 (2024), e0294537.

[28] Nur Rachman Dzakiyullah, Andri Pramuntadi, and Anni Karimatul Fauziyyah. "Semi-supervised classification on credit card fraud detection using autoencoders". In: *Journal of Applied Data Sciences* 2.1 (2021), pp. 01–07.

[29] Abdoul-Fatao Ouedraogo et al. "Data-driven approach for credit card fraud detection with autoencoder and one-class classification techniques". In: *IFIP International Conference on Advances in Production Management Systems*. Springer. 2021, pp. 31–38.

[30] Georgios Zioviris, Kostas Kolomvatsos, and George Stamoulis. "Credit card fraud detection using a deep learning multistage model". In: *The Journal of Supercomputing* 78.12 (2022), pp. 14571–14596. DOI: 10.1007/s11227-022-04465-9.

[31] Francine Verbeek. "Anomaly Detection in Financial Transaction Data Using Variational Autoencoders". Master's Thesis. Vrije Universiteit Amsterdam, 2022.

[32] Luuk Jacobs. "Detecting Suspicious Transaction Behavior Using Variational Autoencoders". Master's Thesis. Vrije Universiteit Amsterdam, 2023.

[33] Ruben Laan. "Time Series Anomaly Detection in the Latent Space of a Variational Autoencoder on Financial Data, a Qualitative Research". Master's Thesis. Vrije Universiteit Amsterdam, 2021.

[34] Young Kon Yong. "Using a Variational Autoencoder for the Detection of Financial Crime by Encoding Account Behavior". Master's Thesis. Vrije Universiteit Amsterdam, 2025.

[35] European Parliament and Council. *Artificial Intelligence Act (Regulation (EU) 2024/1689)*. Regulation (EU) 2024/1689 of 13 June 2024, Official Journal of the European Union, L 1689, 12 July 2024, entered into force 1 August 2024.

[36] Jacobo Chaquet-Ulldemolins et al. "On the black-box challenge for fraud detection using machine learning (ii): nonlinear analysis through interpretable autoencoders". In: *Applied Sciences* 12.8 (2022), p. 3856.

[37] Timur Sattarov, Dayananda Herurkar, and Jörn Hees. "Explaining anomalies using denoising autoencoders for financial tabular data". In: *arXiv preprint arXiv:2209.10658* (2022).

[38] Ian Goodfellow et al. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[39] Samuel J Gershman and Noah D Goodman. "Amortized inference in probabilistic reasoning". In: *Proceedings of the 36th Annual Conference of the Cognitive Science Society*. 2014, pp. 517–522.

[40] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0.

[41] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". In: *Proceedings of the 31st International Conference on Machine Learning (ICML)*. 2014, pp. 1278–1286. URL: https://proceedings.mlr.press/v32/rezende14.html.

[42] Naftali Tishby, Fernando Pereira, and William Bialek. "The information bottleneck method". In: *arXiv preprint physics/0004057* (2000).

[43] Alexander A Alemi et al. "Deep Variational Information Bottleneck". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2017.

[44] Alessandro Achille and Stefano Soatto. "Emergence of invariance and disentanglement in deep representations". In: *Journal of Machine Learning Research* 19.50 (2018), pp. 1–34.

[45] Karl Ridgeway. "A survey of inductive biases for factorial representation-learning". In: *arXiv preprint arXiv:1612.05299* (2016).

[46] Hyunjik Kim and Andriy Mnih. "Disentangling by Factorising". In: *International Conference on Machine Learning*. 2018.

[47] Francesco Locatello et al. "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations". In: *International Conference on Machine Learning*. 2019.

[48] Abhishek Kumar, Prasanna Sattigeri, and Avinash Balakrishnan. "Variational Inference of Disentangled Latent Concepts from Unlabeled Observations". In: *International Conference on Learning Representations*. 2018.

[49] Cian Eastwood and Christopher KI Williams. "A framework for the quantitative evaluation of disentangled representations". In: *International Conference on Learning Representations (ICLR)*. 2018.

[50] Christopher Burgess et al. "Understanding disentangling in β-VAE". In: *arXiv preprint arXiv:1804.03599* (2018).

[51] Rutger R. van de Leur et al. "Improving explainability of deep neural network-based electrocardiogram interpretation using variational auto-encoders". In: *European Heart Journal - Digital Health* 3.3 (2022), pp. 390–404. DOI: 10.1093/ehjdh/ztac038.

[52] Alberto Solera-Rico et al. "β-Variational autoencoders and transformers for reduced-order modelling of fluid flows". In: *Nature Communications* 15 (2024), p. 1361. DOI: 10.1038/s41467-024-45578-4.

[53] Niklas Stoehr et al. "Disentangling Interpretable Generative Parameters of Random and Real-World Graphs". In: *Advances in Neural Information Processing Systems (NeurIPS)*. 2019.

[54] Kilhwan Shon et al. "Development of a β-variational autoencoder for disentangled latent space representation of anterior segment optical coherence tomography images". In: *Translational Vision Science Technology* 11.2 (2022), p. 11. DOI: 10.1167/tvst.11.2.11.

[55] Marc-André Carbonneau et al. "Measuring Disentanglement: A Review of Metrics". In: *arXiv preprint arXiv:2012.09276* (2022).

[56] Anna Sepliarskaia, Julia Kiseleva, and Maarten de Rijke. "Evaluating Disentangled Representations". In: *arXiv preprint arXiv:1910.05587*. 2020.

[57] Minsuk Kim et al. "Relevance Factor VAE: Learning and Identifying Disentangled Factors". In: *arXiv preprint arXiv:1902.01568* (2019).

[58] Ricky T. Q. Chen et al. “Isolating Sources of Disentanglement in Variational Autoencoders”. In: *Advances in Neural Information Processing Systems*. 2018.

[59] Kilian Do and Truyen Tran. “Theory and Evaluation Metrics for Learning Disentangled Representations”. In: *International Conference on Learning Representations*. 2020.

[60] Karl Ridgeway and Michael C. Mozer. “Learning Deep Disentangled Embeddings with the F-Statistic Loss”. In: *Advances in Neural Information Processing Systems*. 2018.

[61] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *International Conference on Learning Representations (ICLR)* (2015). arXiv:1412.6980.

[62] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30 (2017).

[63] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. “Consistent individualized feature attribution for tree ensembles”. In: *arXiv preprint arXiv:1802.03888* (2018).

[64] Spencer Matthews and Brian Hartman. “mshap: Shap values for two-part models”. In: *Risks* 10.1 (2021), p. 3.

[65] James Lucas et al. “Don’t blame the elbo! a linear vae perspective on posterior collapse”. In: *Advances in Neural Information Processing Systems* 32 (2019).

[66] Antonio Almudévar et al. “Defining and measuring disentanglement for non-independent factors of variation”. In: *arXiv preprint arXiv:2408.07016* (2024).

[67] Surojit Saha, Sarang Joshi, and Ross Whitaker. “Ard-vae: A statistical formulation to find the relevant latent dimensions of variational autoencoders”. In: *2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2025, pp. 889–898.

A

Proofs

A.1. Closed-Form of the KL Divergence

We derive the closed form expression of the KL divergence term used in the β-VAE loss when the approximate posterior has diagonal covariance.

Let $q(z \mid x) = \mathcal{N}\big(\mu(x), \mathrm{diag}(\sigma^2(x))\big)$ and let the prior be $p(z) = \mathcal{N}(0, I)$, with latent dimension L. Write $\mu = \mu_\phi(x)$ and $\sigma^2 = \sigma^2_\phi(x)$. By definition,

$$\mathrm{KL}(q\|p) = \mathbb{E}_q\big[\log q(z \mid x) - \log p(z)\big].$$

For q we have

$$\log q(z \mid x) = -\frac{L}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{L}\log\sigma_j^2 - \frac{1}{2}\sum_{j=1}^{L}\frac{(z_j-\mu_j)^2}{\sigma_j^2}.$$

Taking expectation under q and using $\mathbb{E}_q\big[(z_j-\mu_j)^2\big] = \sigma_j^2$ gives

$$\mathbb{E}_q[\log q(z \mid x)] = -\frac{L}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{L}\log\sigma_j^2 - \frac{L}{2}.$$

For p we have

$$\log p(z) = -\frac{L}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{L} z_j^2,$$

hence

$$\mathbb{E}_q[\log p(z)] = -\frac{L}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{L}\mathbb{E}_q[z_j^2] = -\frac{L}{2}\log(2\pi) - \frac{1}{2}\sum_{j=1}^{L}\big(\mu_j^2+\sigma_j^2\big),$$

since $\mathbb{E}_q[z_j^2] = \mu_j^2 + \sigma_j^2$.

Subtracting, we obtain

$$\mathrm{KL}(q\|p) = \frac{1}{2}\sum_{j=1}^{L}\Big(\mu_j^2 + \sigma_j^2 - \log\sigma_j^2 - 1\Big).$$

Equivalently, if the network outputs $\log\sigma_j^2$, the same expression applies directly.

B

Additional Results on Latent Activity

B.1. Average Dimension-Wise KL Divergence Across β and L

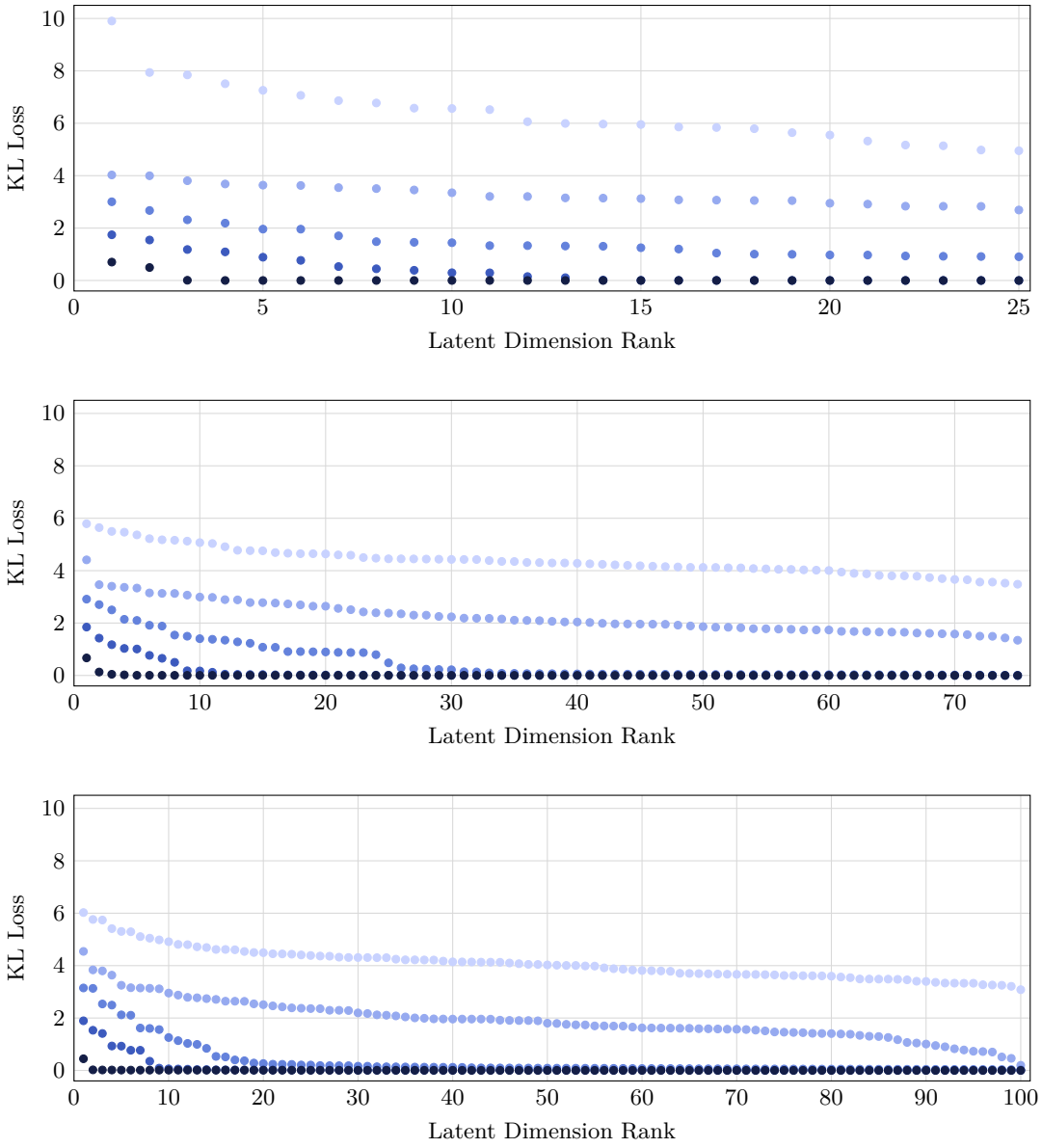


Figure B.1: Average KL divergence per latent dimension for models trained with different values of β at latent sizes $L \in \{25, 75, 100\}$. From top to bottom: $L = 25$, $L = 75$, and $L = 100$. Color intensity reflects β: light blue corresponds to $\beta = 10^0$, and progressively darker shades indicate higher values up to $\beta = 10^4$.

B.2. Latent Traversal Examples

We include two further traversal examples from a model with 50 latent dimensions trained with $\beta = 100$. Figures B.2 and B.3 correspond to the first sample, where the traversals are performed along the latent dimension with the highest and lowest KL divergence respectively. Figures B.4 and B.5 show the same procedure for the second sample. All twenty reconstructed time series are displayed.

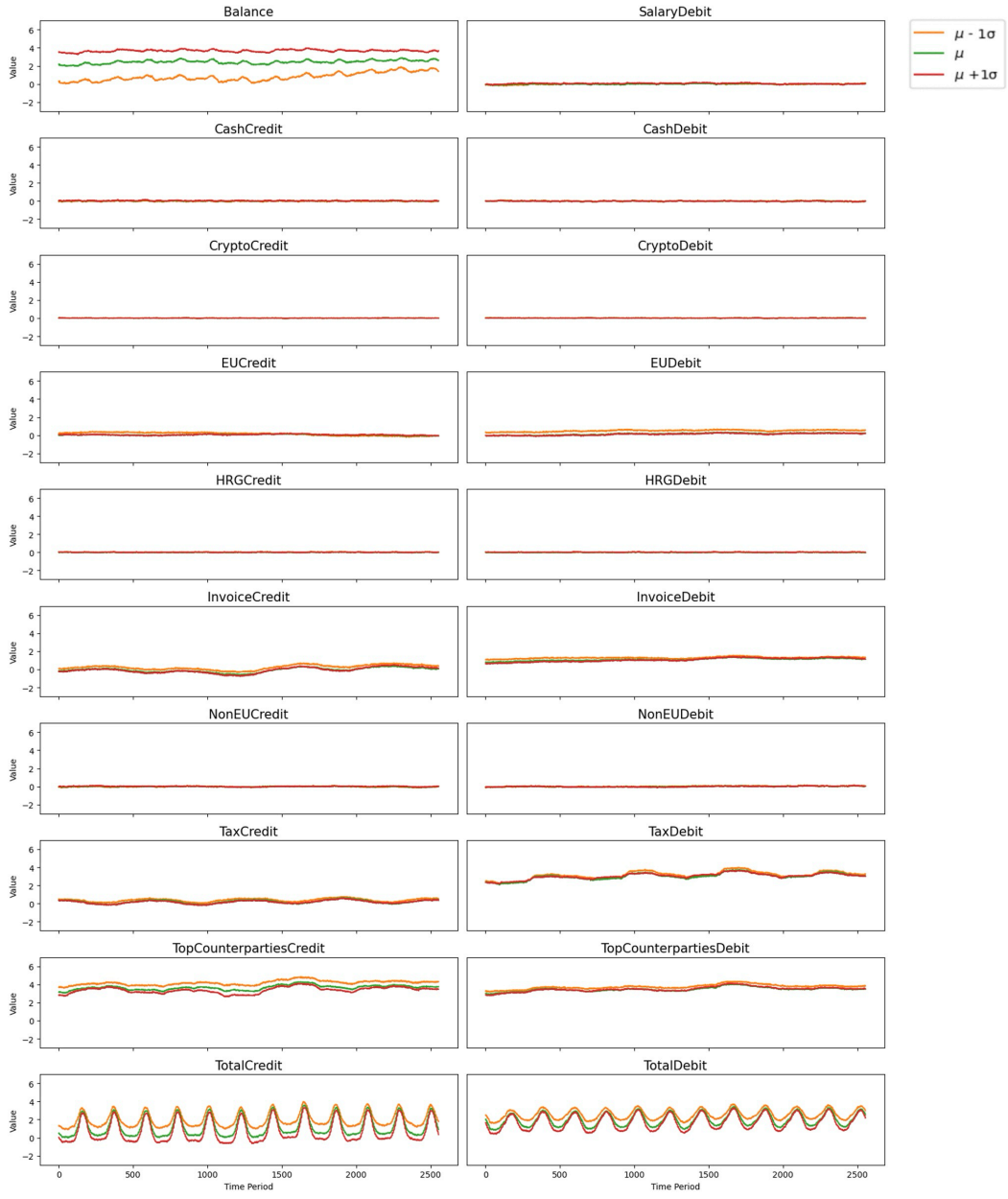


Figure B.2: Latent traversal for Sample 1 along the dimension with the highest KL divergence.

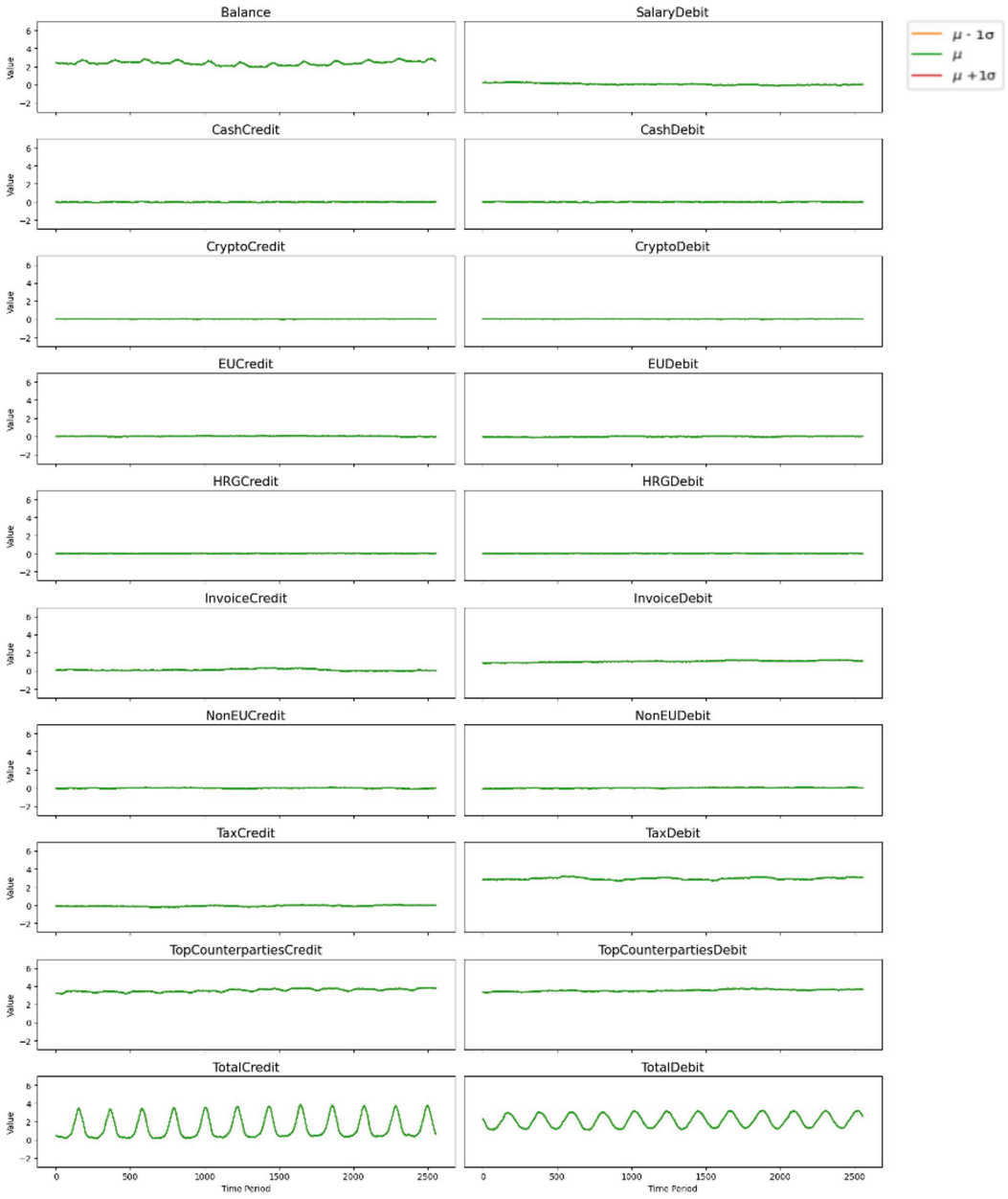


Figure B.3: Latent traversal for Sample 1 along the dimension with the lowest KL divergence.

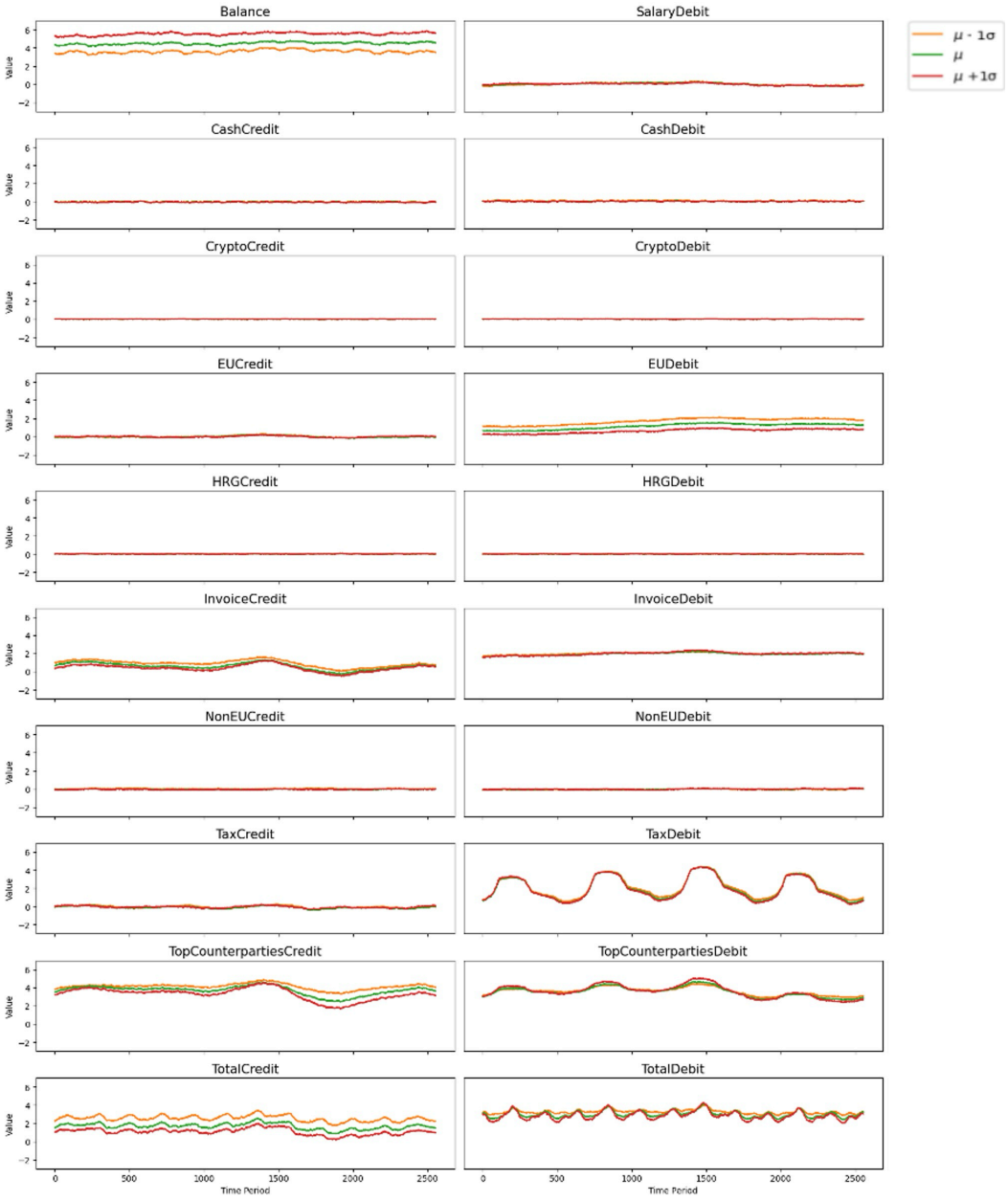


Figure B.4: Latent traversal for Sample 2 along the dimension with the highest KL divergence.

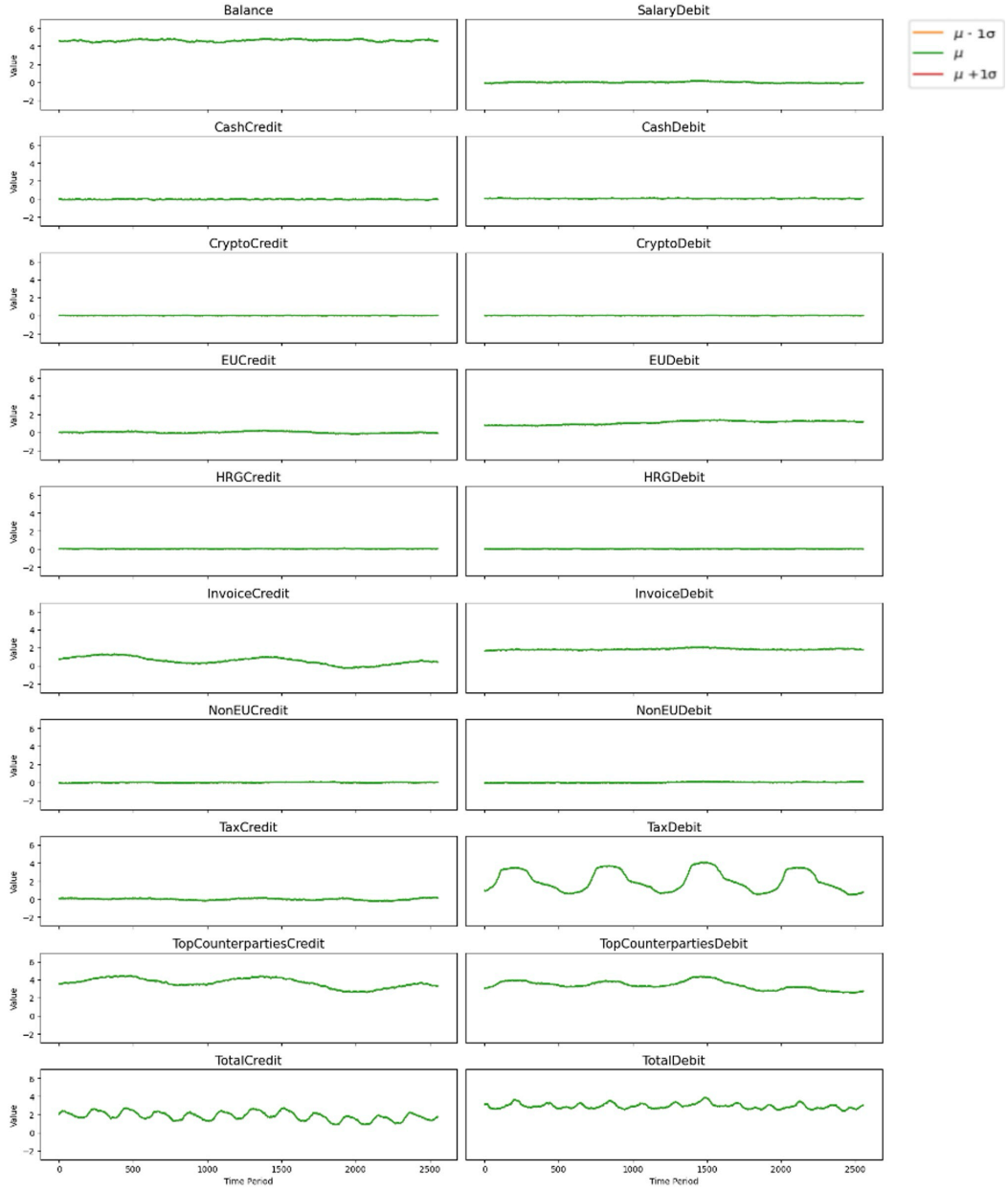


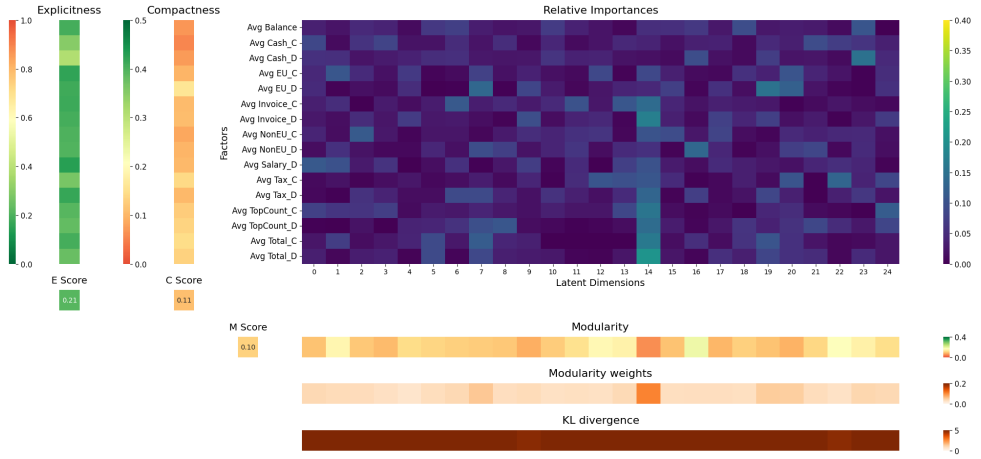
Figure B.5: Latent traversal for Sample 2 along the dimension with the lowest KL divergence.

C

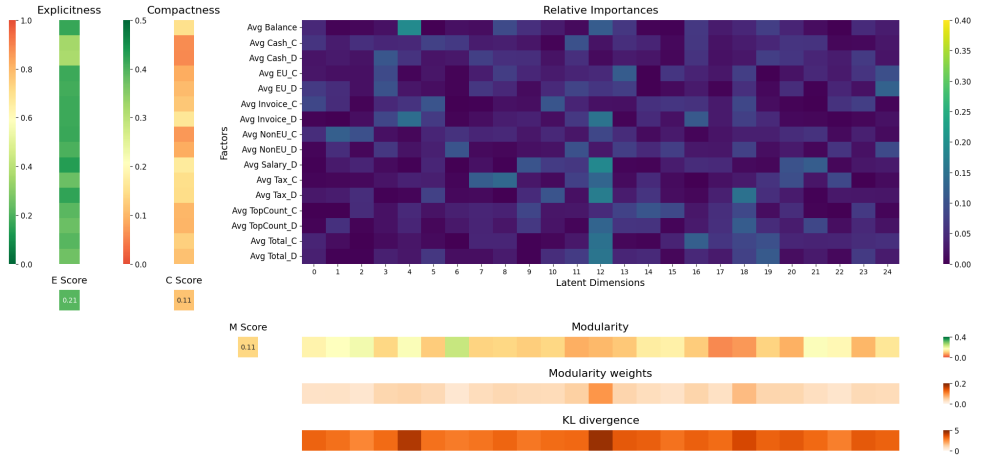
Additional Results of the Interpretability Framework

C.1. DCI Outputs

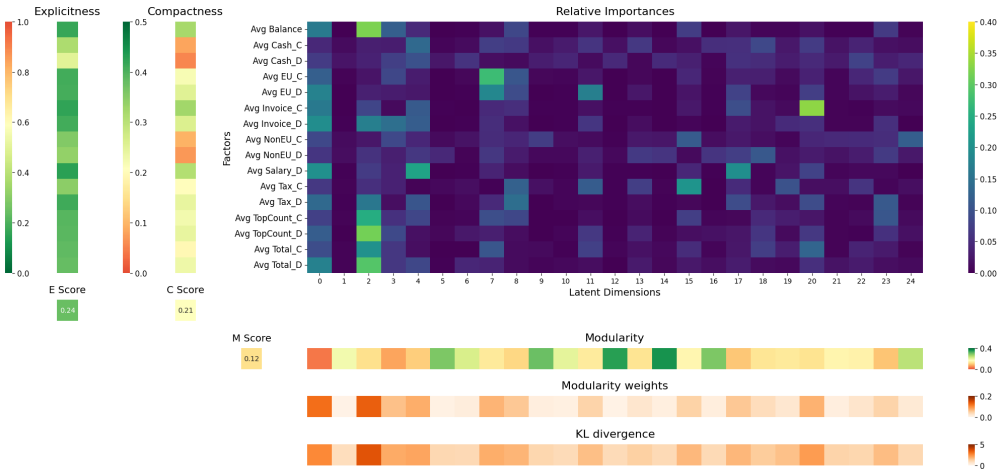
All DCI outputs in this appendix follow the same format. Each figure shows, for a given latent dimensionality and β: (i) the relative importance matrix, (ii) the per-dimension and per-factor scores with the corresponding aggregate values, and (iii) the per-dimension KL divergence.



(a) $L = 25$, $\beta = 10^0$.

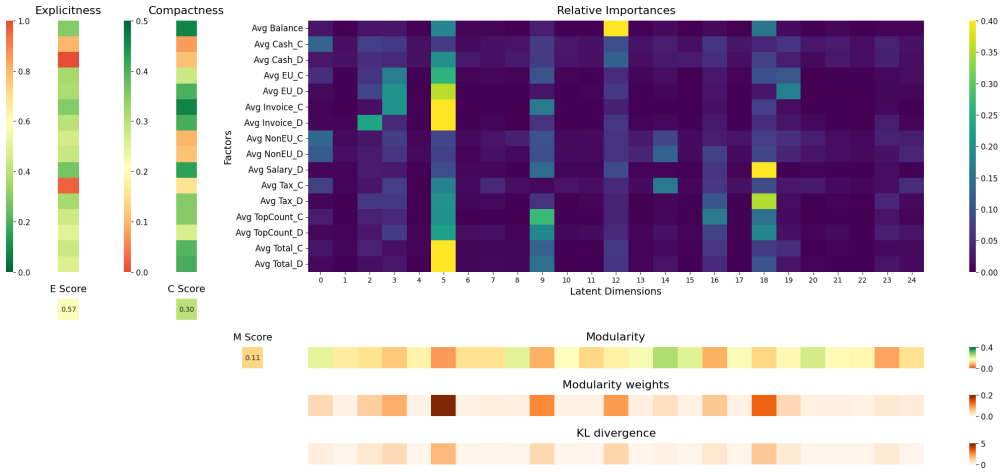


(b) $L = 25$, $\beta = 10^1$.

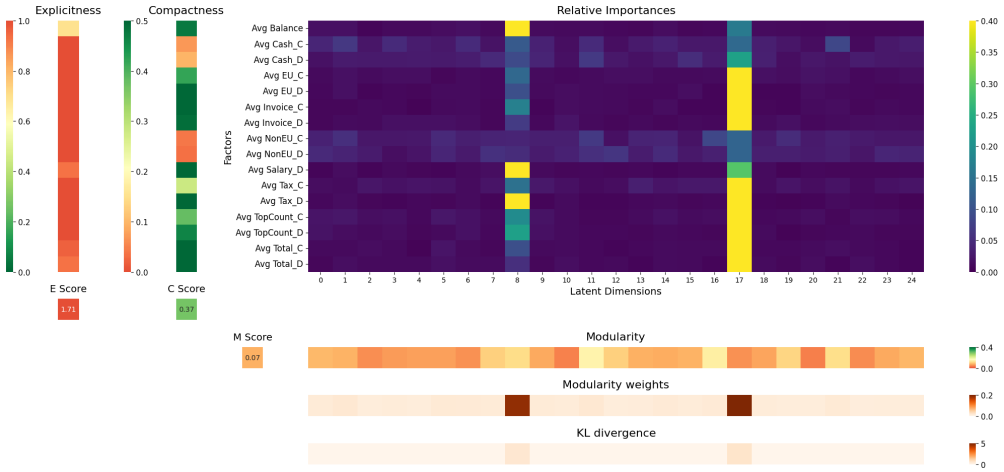


(c) $L = 25$, $\beta = 10^2$.

Figure C.1: DCI outputs for $L = 25$ at different values of β.

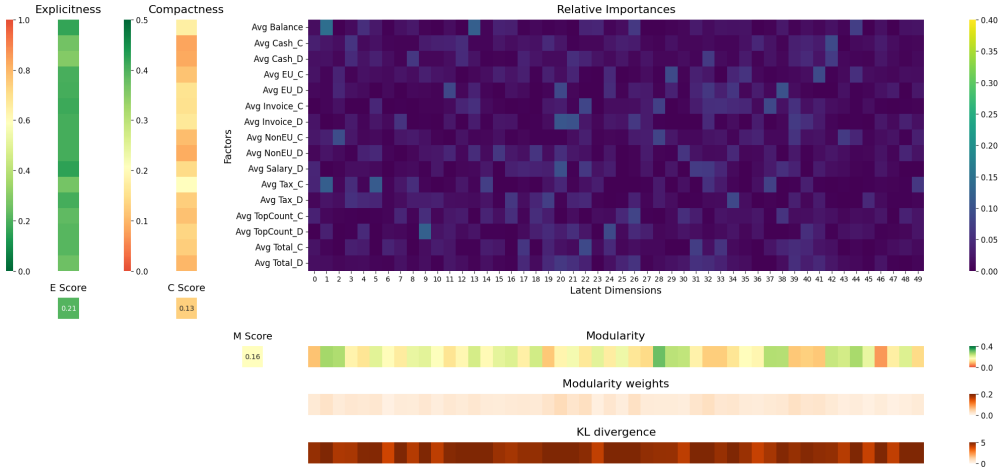


(a) $L = 25$, $\beta = 10^3$.

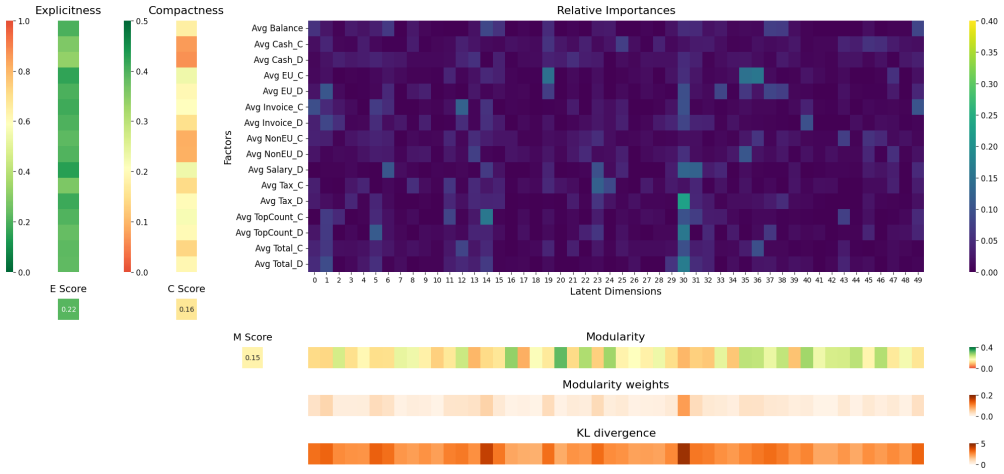


(b) $L = 25$, $\beta = 10^4$.

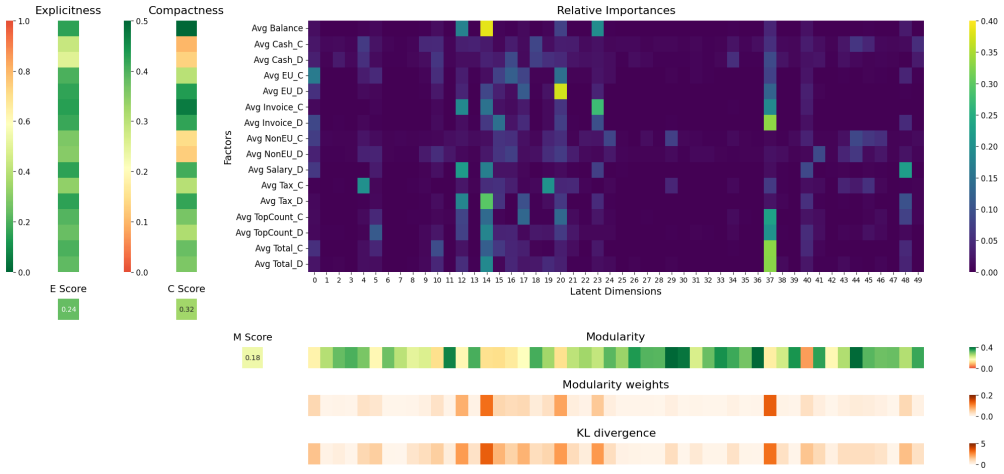
Figure C.2: DCI outputs for $L = 25$ at different values of β.



(a) $L = 50$, $\beta = 10^0$.

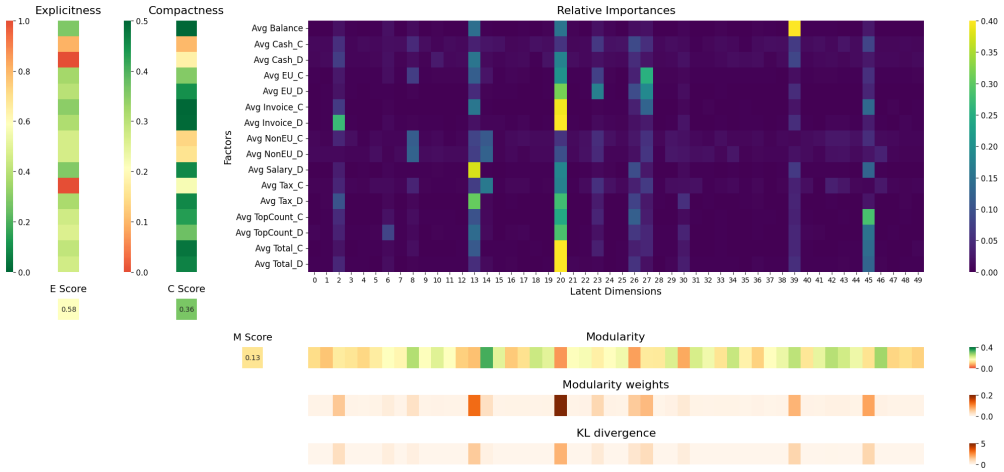


(b) $L = 50$, $\beta = 10^1$.

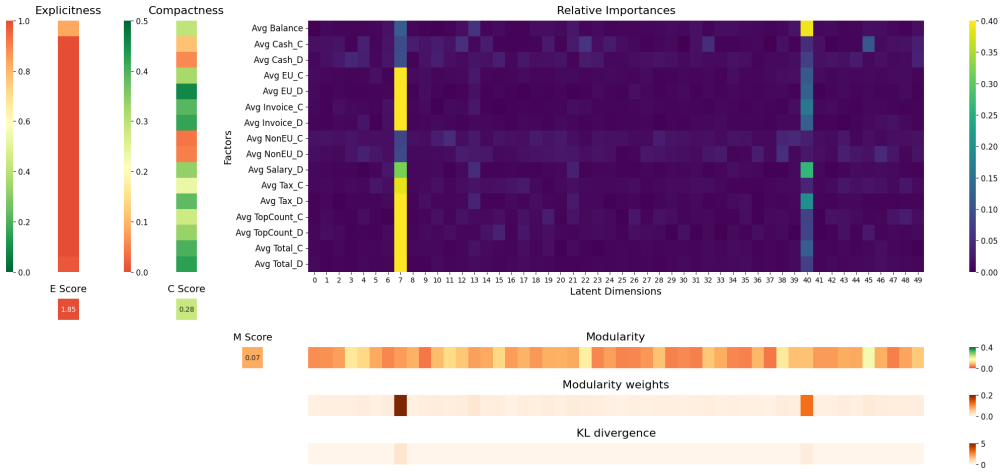


(c) $L = 50$, $\beta = 10^2$.

Figure C.3: DCI outputs for $L = 50$ at different values of β.

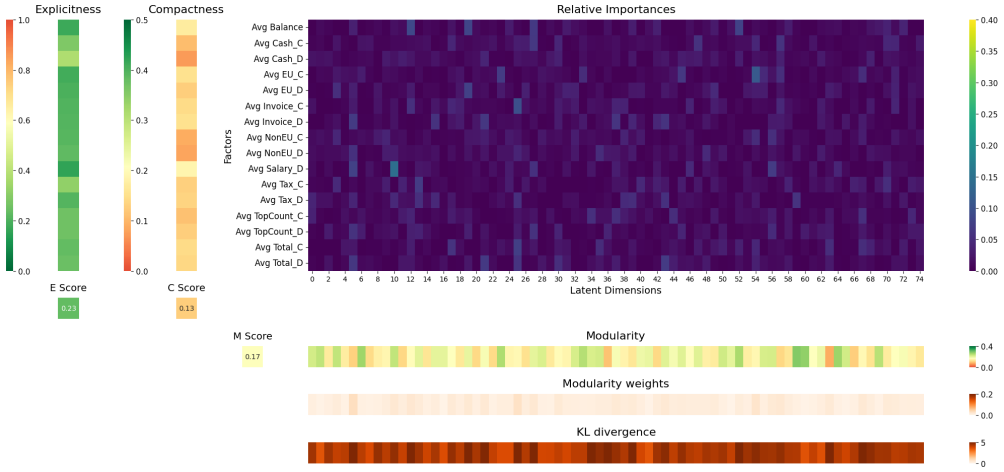


(a) $L = 50$, $\beta = 10^3$.

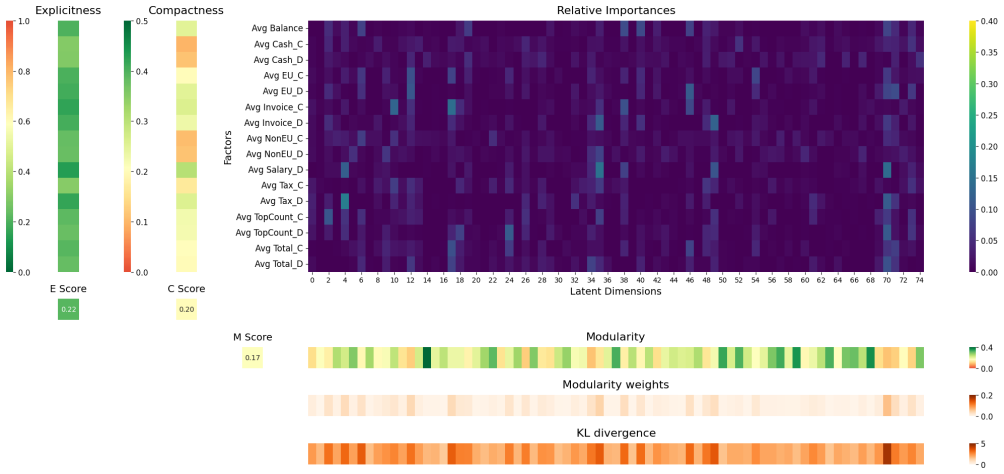


(b) $L = 50$, $\beta = 10^4$.

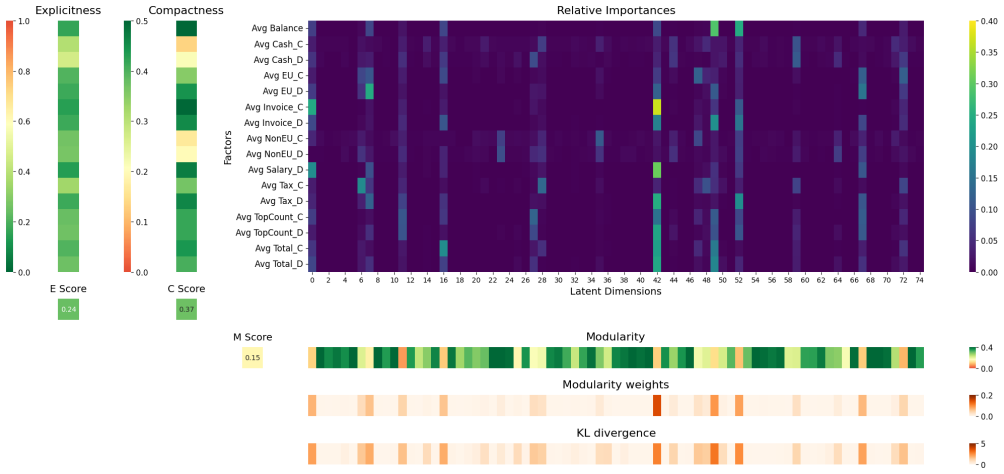
Figure C.4: DCI outputs for $L = 50$ at different values of β.



(a) $L = 75$, $\beta = 10^0$.

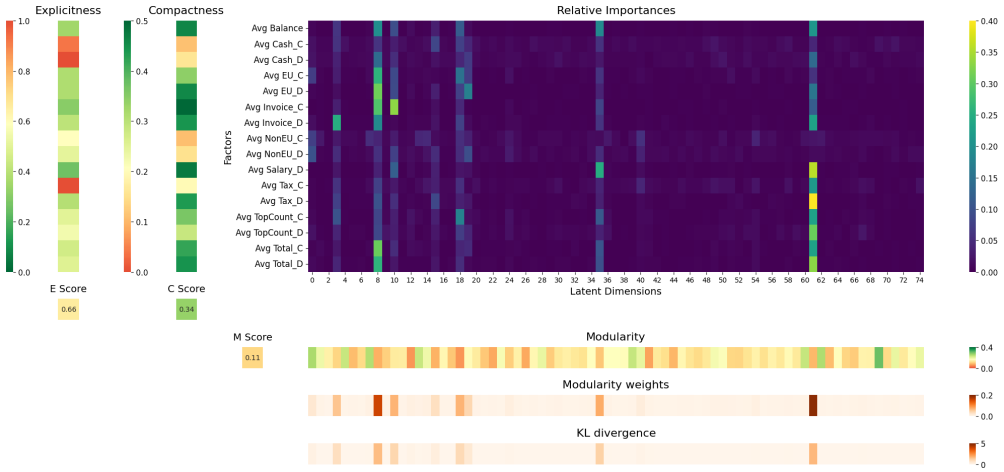


(b) $L = 75$, $\beta = 10^1$.

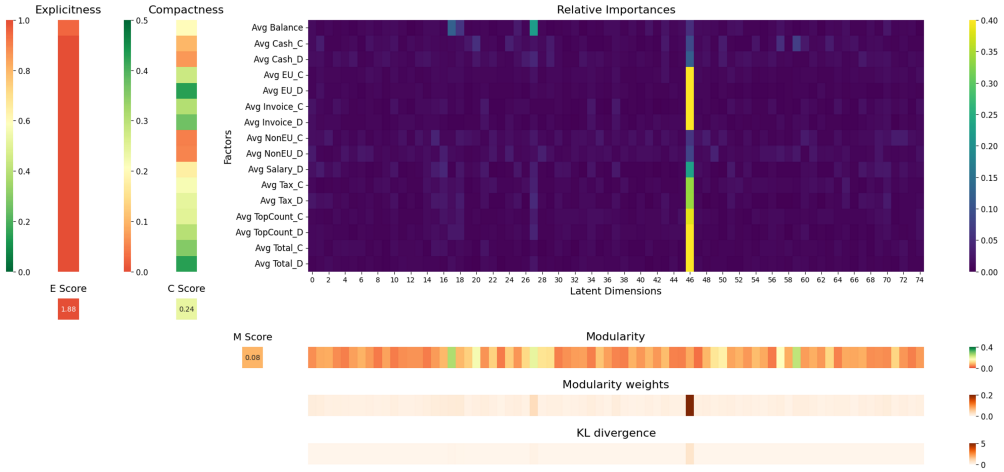


(c) $L = 75$, $\beta = 10^2$.

Figure C.5: DCI outputs for $L = 75$ at different values of β.

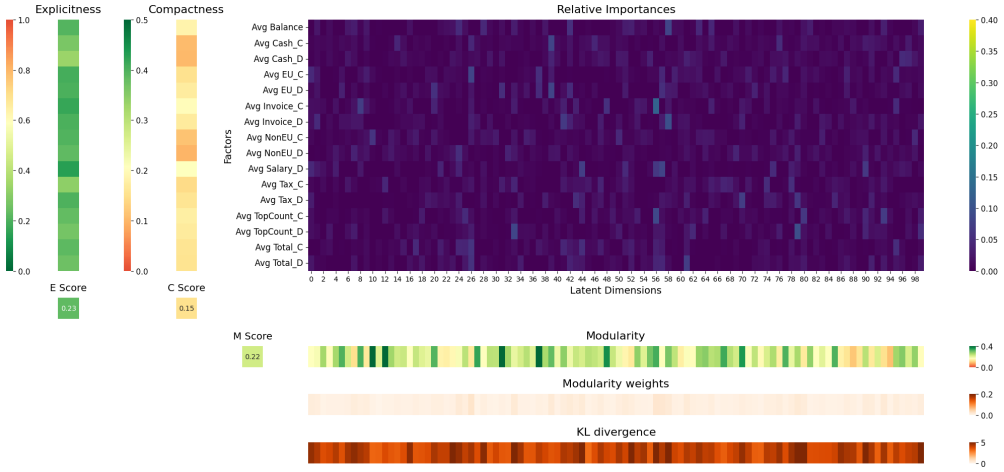


(a) $L = 75$, $\beta = 10^3$.

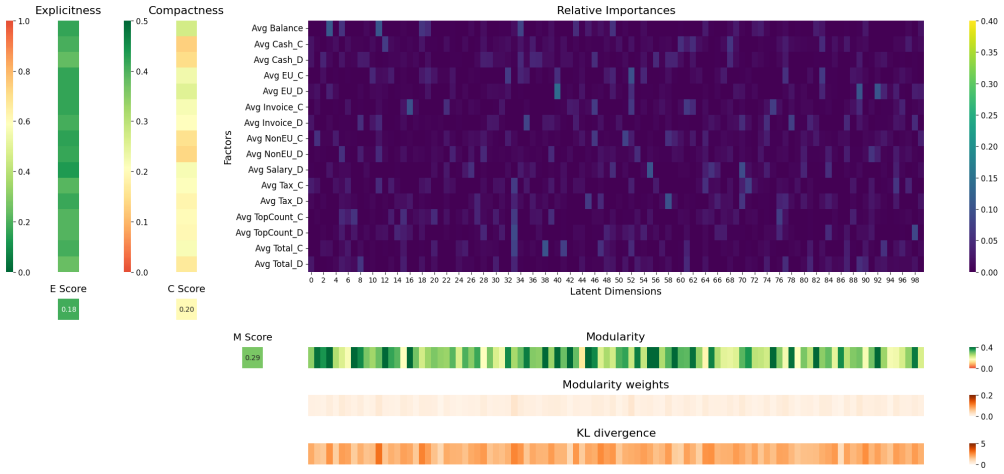


(b) $L = 75$, $\beta = 10^4$.

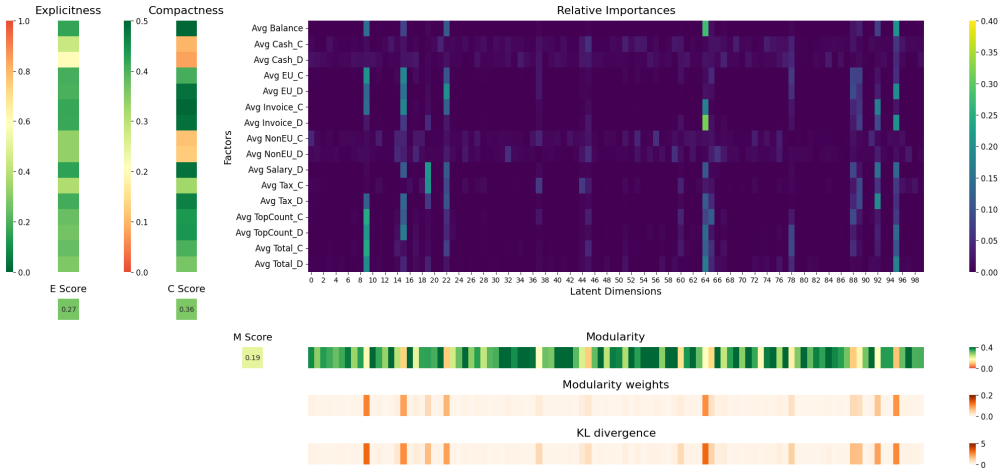
Figure C.6: DCI outputs for $L = 75$ at different values of β.



(a) $L = 100$, $\beta = 10^0$.

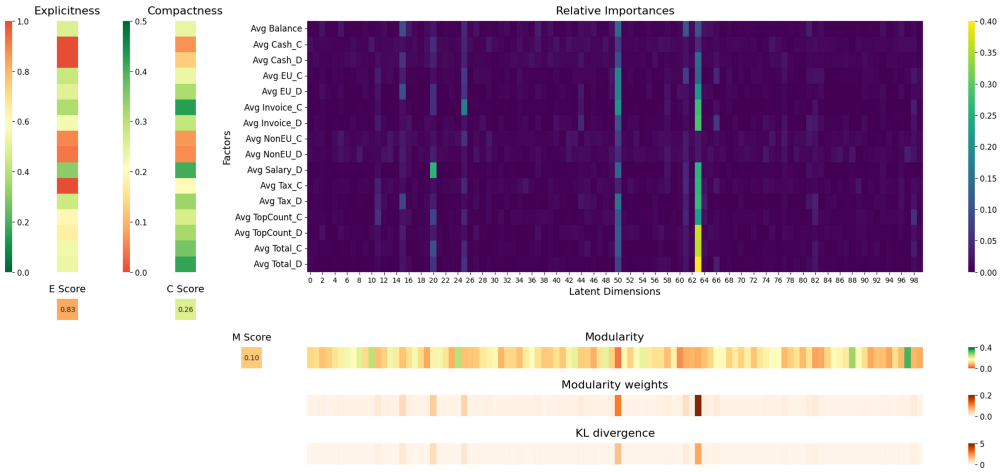


(b) $L = 100$, $\beta = 10^1$.

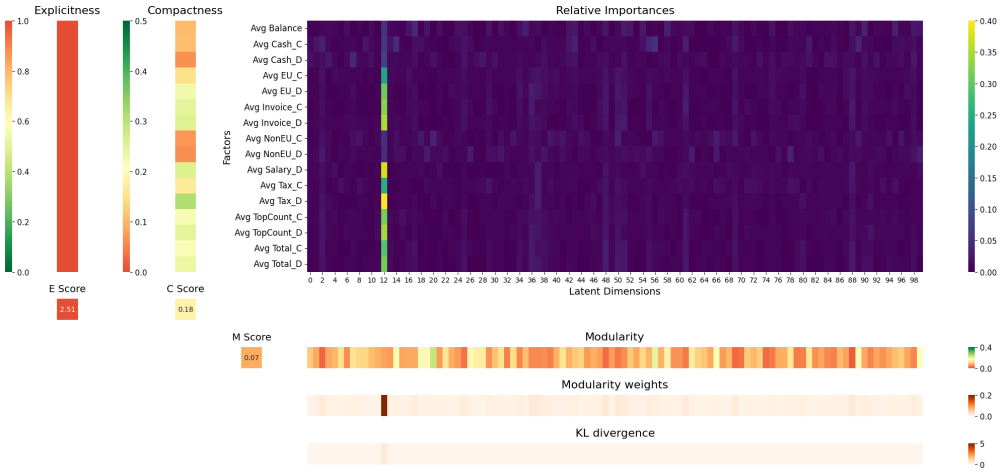


(c) $L = 100$, $\beta = 10^2$.

Figure C.7: DCI outputs for $L = 100$ at different values of β.



(a) $L = 100$, $\beta = 10^3$.



(b) $L = 100$, $\beta = 10^4$.

Figure C.8: DCI outputs for $L = 100$ at different values of β.

C.2. Factor-Wise Compactness Scores Across β and L

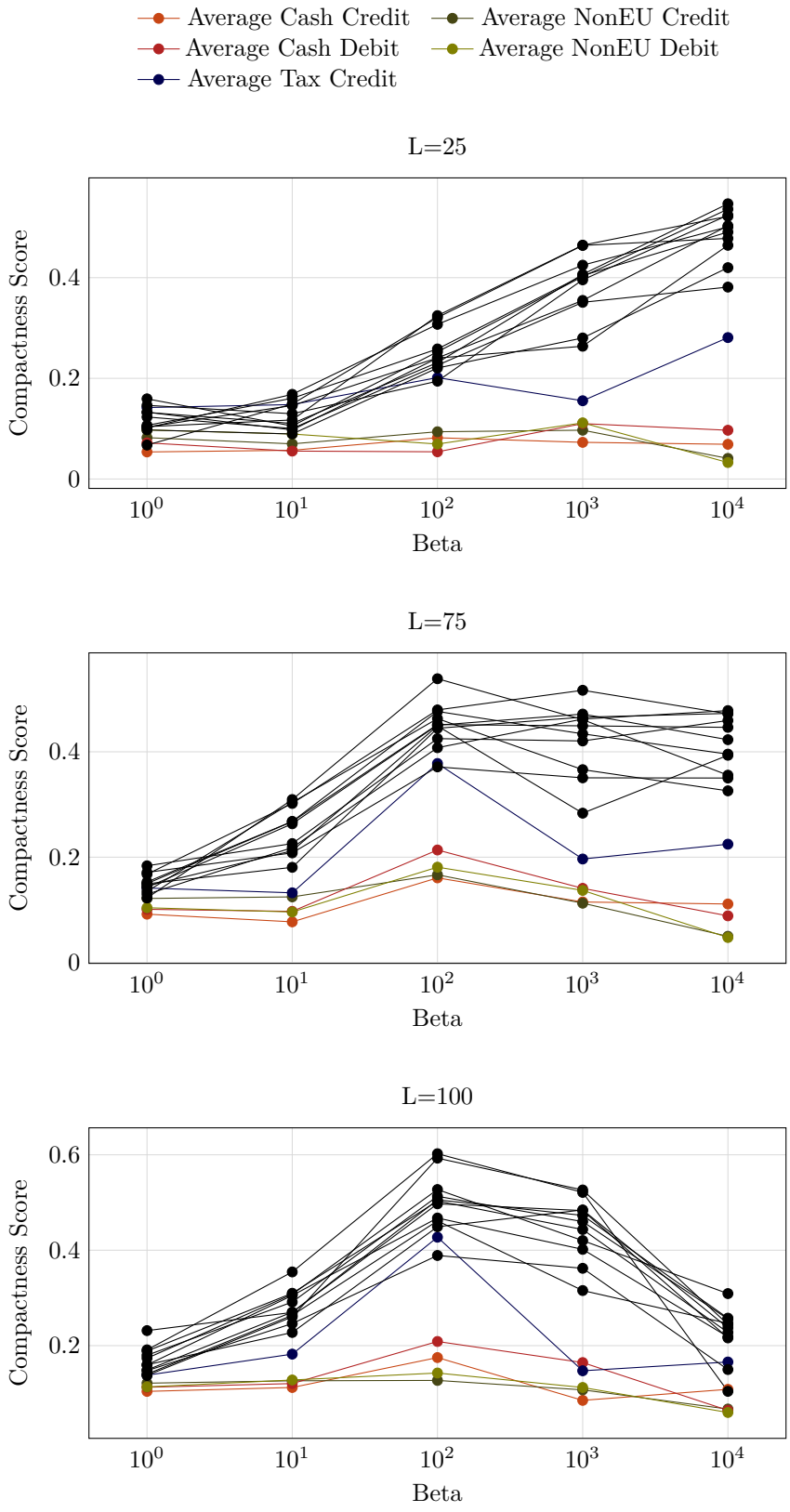


Figure C.9: Compactness scores of individual factors as a function of β for latent dimensionalities $L = 25, 75, 100$. The majority of factors follow a similar trend and are shown in black, while the five factors with distinct behavior are highlighted in color.