A Unified Latent Space Disentanglement VAE Framework with Robust Disentanglement Effectiveness Evaluation

Anonymous authors
Paper under double-blind review

Abstract

Evaluating and interpreting latent representations, such as variational autoencoders (VAEs), remains a significant challenge for diverse data types, especially when ground-truth generative factors are unknown. To address this, we propose a general framework – bfVAE – that unifies several state-of-the-art disentangled VAE approaches and generates effective latent space disentanglement, especially for tabular data. To assess the effectiveness of a VAE disentanglement technique, we propose two procedures - Feature Variance Heterogeneity via Latent Traversal (FVH-LT) and Dirty Block Sparse Regression in Latent Space (DBSR-LS) for disentanglement assessment, along with the latent space disentanglement index (LSDI) which uses outputs of FVH-LT and DBSR-LS to summarize the overall effectiveness of a VAE disentanglement method without requiring access to or knowledge of the ground-truth generative factors. To the best of our knowledge, these are the first assessment tools to achieve this. FVH-LT and DBSR-LS also enhance latent space interpretability and provide guidance on more efficient content generation. To ensure robust and consistent disentanglement, we develop a greedy alignment strategy (GAS) that mitigates label switching and aligns latent dimensions across runs to obtain aggregated results. We assess the bfVAE framework and validate FVH-LT, DBSR-LS, and LSDI in extensive experiments on tabular and image data. The results suggest that bfVAE surpasses existing disentangled VAE frameworks in terms of disentanglement quality, robustness, achieving a near-zero false discovery rate for informative latent dimensions, that FVH-LT and DBSR-LS reliably uncover semantically meaningful and domain-relevant latent structures, and that LSDI makes an effective overall quantitative summary on disentanglement effectiveness.

1 Introduction

1.1 Background and motivation

Understanding and interpreting latent representations in deep generative models (DGMs) remains a fundamental challenge in the development of efficient and trustworthy AI systems, primarily due to their black-box nature. Disentangled representation learning offers a promising solution by aiming to uncover independent and interpretable generative factors underlying complex data, providing a structured and interpretable latent space (Wang et al., 2024; Shen et al., 2022; Higgins et al., 2018). Empirical evidence across different domains further underscores the practical benefits of disentangled representation learning, including medical imaging, single-cell transcriptomics, autonomous driving, and large language models (Liu et al., 2022; Baek et al., 2025; Qin et al., 2023; Pourkeshavarz et al., 2024), highlighting their positive impacts not only for interpretability but also for enhancing robustness in real-world AI systems.

We focus on variational autoencoders (VAEs) (Kingma & Welling, 2013), a widely popular DGM that utilizes latent spaces, in this work. In particular, disentangled VAEs aim to learn structured latent spaces in which distinct latent dimensions (LDs) capture independent and semantically meaningful factors of variation in the data, while retaining reconstruction quality. Disentangled VAEs have been employed in different domains and continue to drive methodological advances. For example, for image data, denoising VAE couples a multi-β VAE with a non-linear latent diffusion model to jointly achieve a disentangled representation and

high-resolution image reconstruction(Uppal et al., 2025); in signal processing, factor VAE enables efficient data compression and supports controlled data augmentation via latent-space interpolation, as applied to Global Navigation Satellite System interference classification.(Heublein et al., 2025).

Most existing work on disentangled VAE focuses on image data, with relatively little attention to tabular data. Our empirical studies suggest that disentanglement frameworks designed for image data may be ineffective for tabular data. Additionally, current methods typically rely on qualitative assessment by visualizing reconstructed images to assign semantic meaning and interpret disentangled latent dimensions (LD) – an approach not applicable to tabular or other data types that cannot be easily visualized. Some approaches assume knowing ground-truth generative factors and use supervised classifiers to enable quantitative evaluation (Higgins et al., 2017; Kim & Mnih, 2018), but such information is often unavailable or imprecise in the real world, greatly restricting their practical applicability (Locatello et al., 2018).

In summary, there remains a lack of a general framework for disentangled VAEs and effective techniques for quantifying the effect of disentanglement in various data modalities, especially when ground-truth generative factors can not be precisely defined – a scenario in practice. Furthermore, the inherent instability of training deep neural networks (e.g., random initialization, stochastic optimization, and local minima, etc) exacerbates the evaluation challenges. Some work in the literature ignores this instability by cherry-picking favorable runs, making the reported results hard to repeat and reproduce. Finally, latent spaces are subject to label-switching, where indexing of latent factors changes across independent runs, further complicating the interpretability and reproducibility of the learned representations (Poworoznek et al., 2021).

1.2 Our contributions

To address the lack of a general framework for disentangled VAEs, we propose bfVAE that unifies several state-of-the-art disentangled VAEs that are applicable to multiple data modalities for disentanglement purposes. bfVAE explicitly accounts for modality-dependent information capacity and latent dependence, providing a principled explanation for its effectiveness in identifying informative and non-informative LDs.

To address the challenges of interpreting and quantifying latent representations without precise knowledge on ground-truth generative factors, we propose Feature Variance Heterogeneity via Latent Traversal (FVH-LT) to quantify how each latent variable relates to inputs by computing variances of reconstructions through LT. In addition, we introduce Dirty Block Sparse Regression in Latent Space (DBSR-LS), a novel application of multi-task supervised learning (Jalali et al., 2010), to model associations between latent variables and input features. FVH-LT and DBSR-LS are applicable to all disentangled VAE formulations evaluated in this work.

To further quantify the aggregated latent-feature association matrices obtained from FVH-LT and DBSR-LS, we introduce the latent space disentanglement index (LSDI). By measuring the structural separation of latent-feature association patterns across LDs, LSDI provides a quantitative overall assessment of disentanglement effectiveness, without knowing ground-truth generative factors.

To address the inherent instability during training and ensure robustness of FVH-LT and DBSR-LS, we recommend training VAEs multiple times to aggregate the results. To resolve the intrinsic label-switching issue in latent spaces across multiple runs, we develop a greedy alignment strategy (GAS). GAS ensures consistent indexing and supports stable and reproducible interpretation of learned disentanglement.

We run extensive experiments on data of various types and sizes to evaluate our methods. The results show that bfVAE is more effective than existing disentangled VAEs in disentangling latent space with tabular input and has a near-zero false discovery rate for informative LDs. The LSDI results further reinforce this finding, with bfVAE attaining substantially higher scores than other objectives. Moreover, both FVH-LT and DBSR-LS consistently and reliably uncover disentangled, semantically meaningful, and domain-relevant LDs. To our knowledge, FVH-LT and DBSR-LS are the first techniques, coupled with the GAS procedure, that evaluate the latent space disentanglement effectiveness in VAEs for multiple data types without precise knowledge of ground-truth generative factors, providing a robust and general tool for interpretable representation learning.

1.3 Related work

Building on the classical VAE formulation, various extensions have been proposed to promote disentanglement of latent space for enhanced interpretability, while balancing reconstruction fidelity. β-VAE (Higgins et al., 2017) introduces a hyperparameter β that scales the Kullback–Leibler (KL) divergence in the classical VAE objective with $beta = 1$. Larger β would correspond to a higher degree of disentanglement. Disentangled β-VAE (Burgess et al., 2018) introduces an additional capacity parameter C to β-VAE that controls the information transmitted from input space to latent space. Factor-VAE (Kim & Mnih, 2018) incorporates a total correlation penalty for the average posterior of latent variables over data, reducing dependencies among the LDs while preserving reconstruction quality. β-TCVAE (Chen et al., 2018) decomposes the KL divergence term in the evidence lower bound into index-code mutual information, total correlation, and dimension-wise KL, and promotes disentanglement by imposing a stronger penalty on the total correlation term. DIP-VAE (Kumar et al., 2017) penalizes the off-diagonal elements in the covariance matrix of the expected posterior means over the marginal distribution of data (DIP-VAE-I) or the posterior covariance matrix over its marginal variational distribution (DIP-VAE-II). Li et al. (2025) generalizes the total correlation penalty to a partial correlation penalty, enabling flexible group-wise independence while improving disentanglement with a low-variance batch estimator. Disentanglement is also explored in settings of conditional VAE (cVAE) (Sohn et al., 2015; Harvey et al., 2021; Zhang et al., 2020) and an integrated structural causal model with a bidirectional generative process to achieve causal disentanglement (Shen et al., 2022).

Prior works have also introduced metrics to measure the effectiveness of disentangled VAEs, which all rely on knowing ground-truth generative factors. Higgins et al. (2017) proposed a disentanglement score by fixing one generative factor at a time, varying the others to generate sample pairs, computing and averaging the absolute differences in the pairs on their corresponding latent representations. A linear classifier is then trained on the averaged latent differences to predict which generative factor was fixed, the prediction accuracy from which is defined as the disentanglement score. The disentanglement metric in factor VAE (Kim & Mnih, 2018) adopts a similar strategy as in β-VAE; but instead of using latent differences, it identifies the LD with smallest empirical variance and uses a majority-vote classifier to compute the disentanglement score. The Separated Attribute Predictability score (Kumar et al., 2017) constructs a latent-factor matrix, where each entry measures how well a single LD predicts a ground-truth factor using a linear classifier or regression, and then averages the gap between the top two scores by factor in the matrix. (Chen et al., 2018) measures the mutual information between top LDs and each ground-truth factor. The disentanglement completeness informativeness (Eastwood & Williams, 2018) is based on training regressors to quantify the importance of each LD in predicting generative factors.

2 A Unified Disentangled VAE framework

In this section, we first introduce bfVAE as a unified framework for disentangled VAE, and then provide an information bottleneck perspective on the bfVAE formulation.

2.1 bfVAE

We propose a general disentanglement framework, referred to as bfVAE, extending the existing β-VAE and factor-VAE. Denote the input data by $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^n$, where $\mathbf{x}_i$ denotes the i-th observed data point. Consider a standard VAE setting with a standard Gaussian prior for the latent space $\mathbf{Z}$ of dimension K, $p(\mathbf{z}_i) = \mathcal{N}_K(0, I)$. Denote the variational posterior of $\mathbf{Z}$ by $q(\mathbf{z}_i|\mathbf{x})$. Define $q(\mathbf{z}) := n^{-1}\sum_i q(\mathbf{z}_i|\mathbf{x})$ as the sample estimate of $q(\mathbf{z}_i|\mathbf{x})$ and $\bar{q}(\mathbf{z}) := \prod_{j=1}^K q(z_j)$ be the marginalized posterior distribution of $\mathbf{Z}$ if all K LDs are completely disentangled (mutually independent). The bfVAE loss is defined as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\omega}) =& \frac{1}{n}\sum_{i=1}^{n}\Bigg\{\underbrace{-\mathbb{E}_{q(\mathbf{z}_i|\mathbf{x}_i,\boldsymbol{\phi})}\log p(\mathbf{x}_i \mid \mathbf{z}_i, \boldsymbol{\omega})}_{\text{T1: reconstruction error}} + \underbrace{\beta \cdot |\mathrm{KL}\left(q(\mathbf{z}_i \mid \mathbf{x}_i, \boldsymbol{\phi})\|p(\mathbf{z}_i)\right) - C|}_{\text{T2: capacity constraint}}\Bigg\} \\ &+ \gamma \cdot \underbrace{\mathrm{KL}\left(q(\mathbf{z})\|\bar{q}(\mathbf{z})\right)}_{\text{T3: disentanglement regularization}}, \end{aligned} \tag{1}$$

where $\boldsymbol{\omega}$ and $\boldsymbol{\phi}$ are the parameters in the corresponding distributions and $\beta \geq 0, \gamma \geq 0$, and $C \geq 0$ are hyperparameters. C is the capacity parameter that specifies a target value for the KL divergence. C

constrains the model's capacity – large C allows the posterior to extract more information from the input data and thus drift from the prior. During training, C is often gradually increased, providing a controlled expansion of the information bottleneck. Different terms in Eq. 1 serve different purposes. T1 ensures reconstruction fidelity by maximizing the expected log-likelihood; T2 penalizes the distance between the prior-posterior KL divergence and the target C: if $\mathrm{KL} = C$, T2 disappears, if $\mathrm{KL} > C$ ($< C$), T2 pushes the KL down (up) toward C; T3 encourages statistical independence among the LDs in the latent space.

Some existing disentangled VAEs are special cases of bfVAE by setting β, γ, and C at specific values. Specifically, when $\gamma = 0, C = 0$, bfVAE in Eq. 1 reduces to β-VAE; when $\beta = 1, C = 0$, it reduces to factor-VAE – the reason behind the name bfVAE (when $\gamma = 0, C = 0, \beta = 1$, it becomes the vanilla VAE). For tabular data, our empirical studies suggest $C = 0$ works well but setting $\beta > 1$ often leads to posterior collapse during training; we thus suggest $\beta < 1$ in Eq. 1 to mitigate the issue. For image data, our empirical studies suggest that $C > 0$ seems to be more effective than $C = 0$ for balancing disentanglement and reconstruction fidelity.

2.2 An Information Bottleneck Perspective on bfVAE

We can interpret the bfVAE formulation in Eq. 1 via the variational information bottleneck framework (Alemi et al., 2016; Burgess et al., 2018). The original information bottleneck formulation as proposed by Tishby et al. (2000) is

$$\max \left\{ I(\mathbf{Z}; \mathbf{Y}) - \beta I(\mathbf{X}; \mathbf{Z}) \right\}, \tag{2}$$

where $\mathbf{X}$ is the input data, $\mathbf{Y}$ is the target, and $\mathbf{Z}$ is a representation of $\mathbf{X}$ that is maximally informative about $\mathbf{Y}$ while being as compressed as possible with respect to $\mathbf{X}$. β acts as the Lagrange multiplier that regulates information of $\mathbf{X}$ preserved in $\mathbf{Z}$.

The bfVAE formulation in Eq. 1 is not a classical information bottleneck objective in the form of Eq. 2, but it admits a natural interpretation inspired by the information bottleneck. Specifically, T1 plays the role of a relevance term and rewards representations that remain predictive of the target of interest. In the unsupervised setting like VAE, the target is the input itself, so the model seeks a compressed representation that still preserves the information necessary to reproduce $\mathbf{x}$. If this term were optimized alone, the model would favor highly informative $\mathbf{z}$, approaching the behavior of a standard autoencoder, with little incentive to compress or organize the representation.

T2 controls the amount of information transmitted from $\mathbf{x}$ into $\mathbf{z}$, where the posterior-prior KL divergence is a commonly used as proxy for latent information rate in VAEs. Averaged over the data distribution, it satisfies the decomposition

$$\mathbb{E}_{p(\mathbf{x})} \Big[\mathrm{KL}(q_{\boldsymbol{\phi}}(\mathbf{z} \mid \mathbf{x}) \| p(\mathbf{z})) \Big] = I_q(\mathbf{x}; \mathbf{z}) + \mathrm{KL}(q(\mathbf{z}) \| p(\mathbf{z})) \,, \tag{3}$$

implying that T2 simultaneously penalizes the mutual information between $\mathbf{x}$ and $\mathbf{z}$ and encourages the aggregated posterior $q(\mathbf{z})$ to match the prior. The two hyperparameters in T2 have distinct roles. C specifies the target capacity of the bottleneck or the desired amount of information that $\mathbf{z}$ is allowed to carry – a small C imposes a narrow bottleneck and forces aggressive compression, whereas a large C permits more information to pass through. β, on the other hand, controls the strength of enforcement of this target capacity, thus determining the tradeoff between T1 and T2. This distinction is important. In β-VAE, the KL term is simply controlled by β, which indirectly influences the achievable information rate. By introducing an explicit target C, one can control the bottleneck more directly in the sense that the model is not merely encouraged to reduce KL, but rather to allocate approximately C units of latent capacity. This often yields a more stable tradeoff between reconstruction quality and compression.

T3 is the total correlation of the aggregated posterior and measures the extent to which the dimensions in $\mathbf{z}$ are statistically dependent, penalizing redundancy and dependence and encouraging the retained information to be organized into approximately independent components. From an IB perspective, T2 primarily controls how much information is transmitted, and T3 determines how the transmitted information is structured after passing through the bottleneck and encourages that the information to be distributed across the LDs in an orthogonal non-redundant manner.

The three terms in Eq. 1 play complementary roles in disentangling the latent space in VAEs. T1 together with T2 yields a compressed latent representation, but compression alone does not guarantee disentanglement. Even under a restricted information budget, the model may encode the retained information in correlated latent dimensions. Though the KL term in T2 already imposes some weak pressure toward independence when the prior $p(\mathbf{z})$ is factorized. However, this effect is indirect, since the KL term mixes multiple objectives together – information-rate control and prior matching. In contrast, T3 directly targets dependence in $q(\mathbf{z})$, making it a more explicit and effective regularizer for disentanglement. However, T1 together with T3 alone would also be inadequate, because encouraging independence without controlling capacity may still allow the latent space to preserve excessive, potentially noisy or sample-specific information. In summary, the bfVAE formulation extends the standard variational bottleneck from controlling only the quantity of latent information to also shaping its structure across dimensions. The resulting latent space is expected to be not only informative and compressed, but also more interpretable and disentangled.

The above interpretation is stated at the level of the overall latent representation; the effect of Eq. 1 can be expressed unevenly across individual LDs. Under the joint action of T1, T2, and T3, some LDs actively encodes nontrivial information in $\mathbf{x}$, whereas others contribute little. This observation motivates a dimension-wise characterization of the latent space, as formally stated in Defs. 1 and 2 below. This distinction provides a more refined view of how the bottleneck allocates information across coordinates of the latent space.

Definition 1 (informative latent dimension) *A latent dimension j in a VAE is informative if the expected KL divergence between its posterior and prior distributions exceeds a information threshold ϵ:*

$$\mathbb{E}_{p(\mathbf{x})}\big[\mathrm{KL}(q(z_j|\mathbf{x},\boldsymbol{\phi})\|p(z_j))\big] > \epsilon, \quad \textit{where } \epsilon \gg 0. \tag{4}$$

While an informative LD encodes meaningful information in the input data $\mathbf{x}$, it does not offer semantic interpretation or explanation of how an informative dimension relates to the inputs – thus the need for techniques like FVH-LT and DBSR-LS developed in our work. When $p = \mathcal{N}(0,1)$ and $q(z_j|\mathbf{x}) = \mathcal{N}(\mu_j(\mathbf{x}), \sigma_j^2(\mathbf{x}))$ – commonly choices for $\mathbf{Z}$ in VAEs, Definition 1 implies $q(z_j|\mathbf{x})$ has successfully "escaped" the information bottleneck by significantly shifting its posterior mean $\mu_j(\mathbf{x})$ away from 0 or its posterior variance $\sigma_j^2(\mathbf{x})$ away from 1.

Definition 2 (non-informative latent dimension) *A latent dimension j is non-informative (or collapsed) if its posterior conveys negligible information about the input $\mathbf{x}$:*

$$\mathbb{E}_{p(\mathbf{x})}\big[\mathrm{KL}(q(z_j|\mathbf{x},\boldsymbol{\phi})\|p(z_j))\big] < \delta, \tag{5}$$

for a vanishingly small $\delta \approx 0$. When $q(z_j|\mathbf{x},\boldsymbol{\phi}) \approx p(z_j)$, the j-th LD carries little to no information about $\mathbf{x}$ and thus behaves essentially as random noise.

We hypothesize that the relative prevalence of informative and non-informative LDs is jointly governed by the capacity target C and the enforcement parameter β of T2, but also relates to input data type, along with the characteristics of input data $\mathbf{x}$.

As stated above, C sets the effective information allowance in the latent space,while β determines how tightly the model adheres to that. Larger C values generally favor more informative LDs, particularly when accompanied by large β, which strongly enforces the target. In contrast, smaller C values impose a tighter bottleneck and therefore tend to produce more non-informative LDs, with this effect becoming stronger as β increases. For fixed large C, a small β may yield an intermediate regime in which the separation between informative and non-informative LDs is less distinct.

Previous work on β-VAE for image data often adopts $\beta > 1$ (with $C = 0$), where the bottleneck is relatively restrictive, forcing the model to be selective and to prioritize the aspects of $\mathbf{X}$ that most improve reconstruction. The transmitted information typically aligns with the dominant generative factors underlying $\mathbf{X}$, since these factors explain systematic and recurring patterns in the data. This combination of relatively large β and small C is often effective for image data because images, although high-dimensional, usually contain substantial redundancy across input features. As a result, meaningful structure can often still be preserved under strong compression, and a relatively small latent capacity may suffice to capture the dominant generating factors. In contrast, for lower-dimensional datasets or data whose semantics are more dispersed across

input features, as is often the case for low-dimensional tabular data, there is typically less redundancy and therefore less compressible information. Applying the same bottleneck formulation used for images may then be overly restrictive, causing substantial signal to be discarded, leading to an unduly low number of informative LDs, and degrading reconstruction quality. Instead, a larger capacity target C together with a weaker penalty, such as $\beta < 1$, may be more appropriate. Such a setting allows the latent space to retain more nontrivial information from $\mathbf{X}$, thereby increasing the likelihood of recovering a subset of informative and disentangled LDs. Our experiments in Sec. 4 provide empirical evidence in support of this claim.

3 Latent Space Interpretability Techniques

While bfVAE promotes and learns disentangled LDs through capacity and dependence regulation, it itself does not explicitly identify feature–latent associations, motivating the need for dedicated latent space interpretability techniques to quantify, visualize, and interpret the disentangled LDs. We propose two techniques – FVH-LT and DBSR-LS – below. To enable stable aggregation of FVH-LT and DBSR-LS results across multiple training runs, we introduce the GAS procedure to resolve LD misalignment. Finally, we proposed a Latent Space Disentanglement Index (LSDI) to quantitatively summarize the degree of structural separation captured by the latent–feature association matrix output from FVH-LT and DBSR-LS.

3.1 FVH-LT

FVH-LT is an evaluation method that quantifies the associations between LDs and input features through latent traversal (LT) that systematically varies one LD at a time while holding the remaining dimensions fixed and measures the resulting changes in reconstructed outputs. The output of the FVH-LT procedure is a quantitative matrix measuring latent–feature associations that can be conveniently visualized via a heatmap, providing an effective interpretable summary of the individual LDs. Unlike LT, which is primarily qualitative, FVH-LT provides a quantitative and comparable summary of latent effects across features, LDs, and training runs. Importantly, FVH-LT does not require access to ground-truth generative factors or external supervision, making it applicable to real-world datasets where such information is unavailable.

Fig. 1 visually illustrates the FVH-LT procedure. For each instance in the input data, the trained VAE encoder is used to obtain its latent representation. LT is then performed, followed by reconstruction through

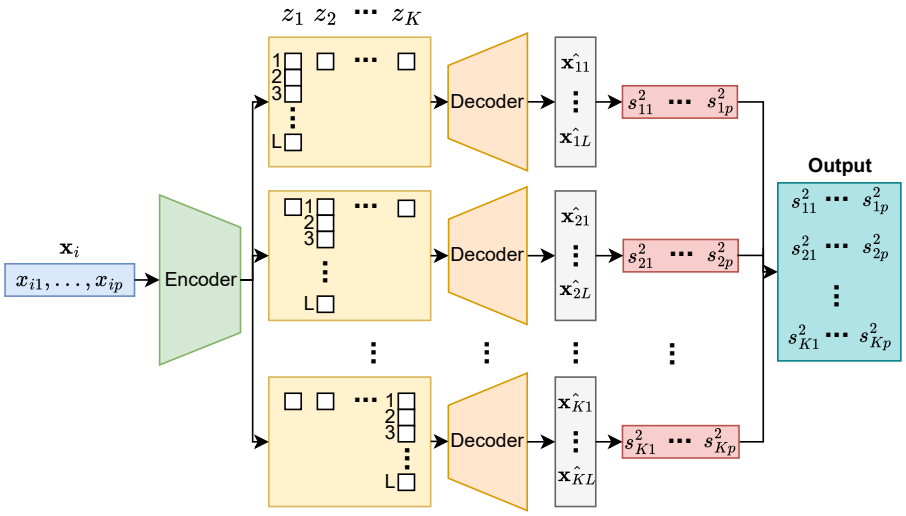


Figure 1: Flowchart for FVH-LT. $\mathbf{x}_i$ contains input features for the i-th observation; z_k is the k-th LD and $\tilde{\mathbf{x}}_{kl}$ for $k = 1, \ldots, K$ and $l = 1, \ldots, L$ are generated features from LT of z_k; s_{kj} is the sample variance of generated data for the j-th feature in the k-th LT.

the decoder. The resulting changes in the reconstructed inputs are quantified on a per-feature basis, yielding a measure of how strongly each feature responds to variations in a given LD. Repeating this process across LDs and data samples produces a structured summary of latent–feature associations.

Alg. 1 presents the corresponding pseudo-code. The LT ranges and some hyperparameters in Alg. 1 play an important role in the effectiveness of FVH-LT. First, there are three common strategies to define the LT ranges in the latent space. The first uses a fixed range across all n observations and K LDs; The second defines a data- and dimension-dependent range, such as $[\mu_{ik} - c\sigma_{ik}, \mu_{ik} + c\sigma_{ik}]$, where μ_{ik} and σ^2_{ik} are the posterior mean and variance of the k-th LD for observation i; the third approach combines the previous two, centering the range at μ_{ik} and applying a global interval half-width. In our experiments, we find the latter two strategies are more effective in identifying informative LDs than the first, because fixed LT ranges may fail to probe high-density regions of the latent posterior when posterior mean μ_{ik} deviates from zero, reducing sensitivity to LDs. As for the number of traversal steps L, a large L creates a fine grid for LT to better detect pattern changes across the LD; a small L would lead to a coarse grid, potentially obscuring meaningful signals, but at a lower computational cost. As for the dimensionality of latent space K, there is no need to match the ground-truth value K_0. Both FVT-LT and DBSR-LS exhibit robustness to K, maintaining a low false discovery rate in identifying informative LDs even when $K > K_0$. This robustness is particularly advantageous in real-world applications where K_0 is typically unknown. Consequently, selecting a slightly larger K is generally preferable to an overly restrictive one.

Algorithm 1 The FVH-LT procedure

Input: training data $\mathbf{x}_{n\times p}$, prior $\mathbf{z} = (\mathbf{z}_1 \ldots, \mathbf{z}_K) \sim f(\mathbf{z})$, LT range $[-a_{ik}, a_{ik}]$ for $k = 1, \ldots, K$ and $i=1, \ldots, n$, LT steps L, run number R, hyper-parameter initialization
Output: Variance matrix $\mathbf{S}$ of reconstructed $\hat{\mathbf{x}}$ given LT in $\mathbf{z}$; posterior-prior KL divergence of $\mathbf{z}$
Pre-processing: standardize $\mathbf{x}$ as needed
for $r = 1$ to R **do**
  Train VAE$_r$ on $\mathbf{x}$ with the loss functions in Eq. 1
  for $i = 1$ to n **do**
    for $k = 1$ to K **do**
      Calculate $d_{i,rk} = D_{\mathrm{KL}}(q_k(z_{ik}|\mathbf{x}_i)||p(z_{ik}))$
      Generate a grid of L values $\mathbf{z}_{ik} = \{z_{ikl}\}$ over its range $[-a_{ik}, a_{ik}]$, fixing all the other LDs at randomly sampled points from $q_{-k}(z_{i,-k}|\mathbf{x}_i)$
      for $l = 1$ to L **do**
        Run $\{z_{ikl}, \mathbf{z}_{i,-k}\}$ through the decoder of VAE$_r$ to reconstruct $\{\hat{x}_{irjl,k}\}_{j=1,\ldots,p}$
      end for
      Calculate sample variance $s^2_{irj,k}$ of $(\hat{x}_{irj1}, \ldots, \hat{x}_{irjL})$ for $j = 1$ to p
    end for
  end for
  Calculate $\bar{s}^2_{rjk} = n^{-1} \sum_{i=1}^{n} s^2_{irj,k}$
end for
Apply GAS (Alg. 3) to align $\bar{s}^2_{rj,k}$ across $r = 1, \ldots, R$
Compute $s^2_{jk} = \frac{1}{R}\sum_{r=1}^{R} \bar{s}^2_{rj,k}$ for $j=1, \ldots, p$ and $d_{ik} = \frac{1}{R}\sum_{r=1}^{R} d_{i,rk}$ for $k=1, \ldots, K$; $i=1, \ldots, n$
Return variance matrix $S = [s^2_{jk}]_{j=1,\ldots,p;k=1,\ldots,K}$ and KL divergence $\mathbf{d} = \{d_{ik}\}_{i=1,\ldots,n;k=1,\ldots,K}$

3.2 DBSR-LS

DBSR promotes shared-sparsity pattern and task-specific sparsity pattern in regression coefficient matrices in a multi-task supervised learning setting (Jalali et al., 2010). We make a novel use of DBSR towards LS disentanglement by treating each LD as a separate regression task and leveraging shared and task-specific sparsity to identify feature–latent associations. Intuitively, this formulation decomposes latent–feature relationships into components that are unique to individual LDs and components that are shared across multiple LDs, providing a structured view of how information is distributed in the latent space. DBSR-LS estimates two regression coefficient matrices $\mathbf{D}$ and B via minimizing the following loss function,

$$\mathcal{L}(D, B) = (2n)^{-1} \sum_{k=1}^{K} \left\| \boldsymbol{\mu}_Z^{(k)} - X^{(k)} \left(D[,k] + B[,k]\right) \right\|_2^2 + \lambda_D \|D\|_{1,1} + \lambda_B \|B\|_{1,\infty}. \tag{6}$$

$\boldsymbol{\mu}^{(k)}_{Z,n\times 1}$ contains the posterior means of the k-th latent variables of the n inputs and $X^{(k)}_{n\times p}$ is the corresponding design matrix comprising the observed features. $\mathbf{D}$ and B are both of dimension $p \times K$, and $D[,k]$ and $B[,k]$

are the respective k-th column. Both $\|D\|_{1,1} = \sum_{i=1}^{p}\sum_{k=1}^{K}|D_{ik}|$ and $\|B\|_{1,\infty} = \sum_{i=1}^{p}\max_k|B_{ik}|$ promotes sparsity, but the former captures the latent-dimension-specific sparsity whereas the latter represents shared sparse structures across the LDs. As a result, nonzero entries in $\mathbf{D}$ indicate features associated distinct LDs. Eq. 6 is formulated using the posterior means $\boldsymbol{\mu}_Z$ as the response variable. An alternative is to use random samples $\mathbf{z}$ from its posterior distributions; but we expect $\boldsymbol{\mu}_Z$ would lead to more stable estimates.

Fig. 2 provides a conceptual illustration of DBSR-LS, highlighting the construction of the multi-task regression problem in latent space. Alg. 2 lists the steps for DBSR-LS. For hyperparameters shared with Alg. 1, their specifications in Alg. 2 are similar. For those unique to Alg. 2 (i.e., $\lambda_B \geq 0, \lambda_D \geq 0$), they can be tuned to balance sparsity in the regression coefficients with prediction accuracy – large values may lead to overly sparse $\hat{\mathbf{D}}$ while small values may have limited power to identify the learned disentanglement. The default algorithm output is $|\hat{\mathbf{D}}|$ ($\hat{\mathbf{D}}$ may provide additional insights in some cases (see App. P.2 for an example) and identifies features relevant for each LD to aid the interpretation of learned disentanglement.

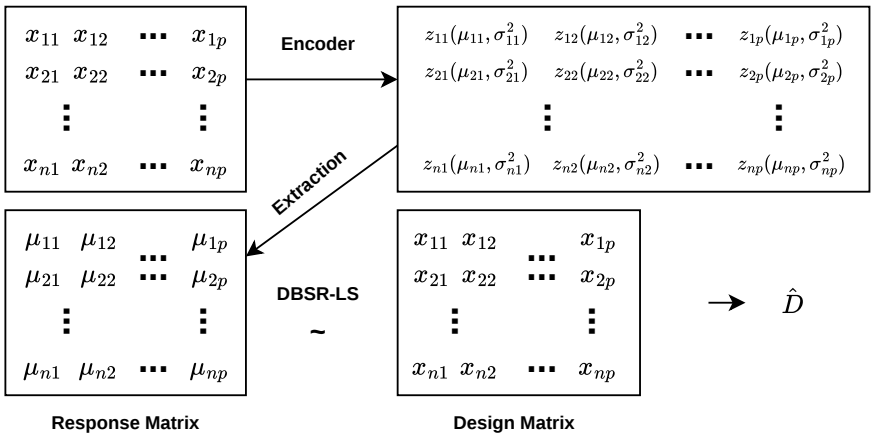


Figure 2: Conceptual illustration of DBSR-LS. Posterior means $\boldsymbol{\mu}_Z$ in the latent space are multi-task regression responses; $\mathbf{x}$ are predictors; sparse regression coefficient matrix $\hat{\mathbf{D}}$ summarizes latent-feature association.

Algorithm 2 The DBSR-LS Procedure

1: **Input**: training data $\mathbf{x}_{n\times p}$, prior $\mathbf{z} = (z_1 \ldots, z_r) \sim f(\mathbf{z})$, initialized hyper-parameters (λ_B, λ_D, those in VAE), run number R
2: **Output**: latent-dimension-specific estimated coefficient matrix $\hat{\mathbf{D}}$ and its absolute value $|\hat{\mathbf{D}}|$
3: Pre-processing: standardize $\mathbf{x}$ as needed
4: **for** $r = 1$ to R **do**
5: Train VAE$_r$ on $\mathbf{x}$ with the loss in Eq. 1 and extract posterior means $\boldsymbol{\mu}_{n\times K,r}$ of $q(\mathbf{z}|\mathbf{x})$
6: Compute $d_{r,ik} = D_{\mathrm{KL}}(q_k(z_{ik}|\mathbf{x})||p(z_{ik}))$ for $k = 1, \ldots, K$ and $i = 1, \ldots, n$
7: Run DBSR-LS with $\mathbf{x}$ as the design matrix and $\boldsymbol{\mu}_r[,k]$ as $Y^{(k)}$ to estimate D_r and B_r
8: **end for**
9: Apply GAS (Alg. 3) to align D_r and $|D_r|$, respectively, across $r = 1, \ldots, R$
10: **Return** averaged matrices $|\hat{D}| = R^{-1}\sum_{r=1}^{R}|D_r|$ and $\hat{D} = R^{-1}\sum_{r=1}^{R}D_r$

3.3 GAS for latent space alignment

Aggregating the final variance matrix $\mathbf{S}$ from FVH-LT and the coefficient matrix $|\hat{\mathbf{D}}|$ from DBSR-LS over repeated runs in Algs. 1 and 2 ($R > 1$) helps mitigate the variability associated with any single realization and provides more reliable characterizations of learned LDs. However, the aggregation is not straightforward because the latent representations obtained across different runs are not necessarily aligned, even when the dimensionality of the latent space is fixed. For example, the first LD in one run may align with the third LD in a different run. Consequently, the latent representations learned across runs are not inherently comparable on a dimension-by-dimension basis. As a result, direct aggregation across runs may be misleading unless an appropriate alignment procedure is applied beforehand.

To address this, we introduce GAS in Alg. 3. GAS clusters LDs based on their informativeness as defined in Defs 1 and 2 to address label switching in the latent space across multiple runs Fig. 3 illustrates the GAS procedure. GAS selects a reference run based on the number of informative LDs, then aligns LDs from other runs to this reference by maximizing the correlation between FVH-LT variance vectors or DBSR-LS coefficient vectors across runs, thereby establishing latent-dimension correspondence and ensuring consistent latent indexing prior to aggregation.

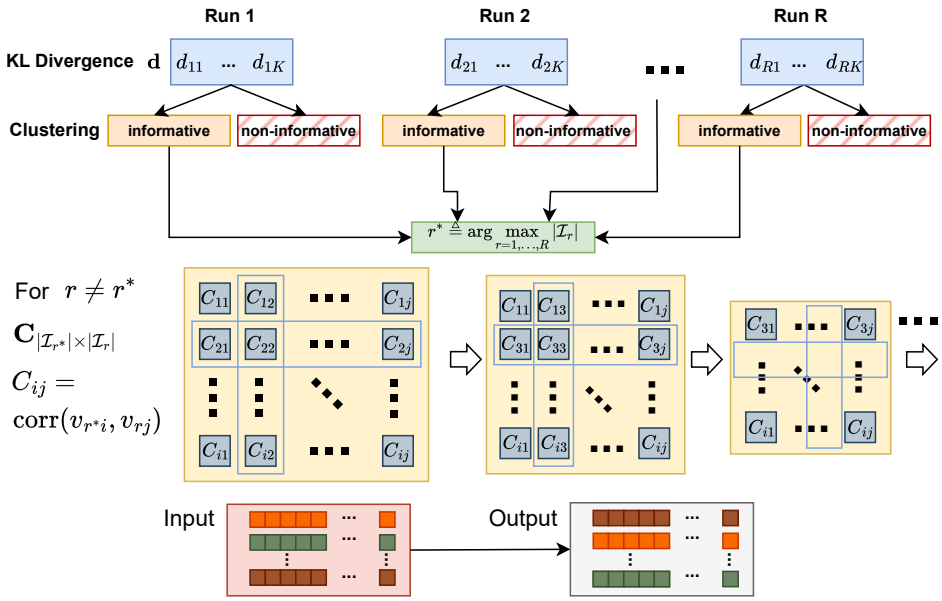


Figure 3: The GAS procedure. d_{rk} denotes the prior-posterior KL divergence of k-th LD in run r; C_{ij} represents the correlation along the i-th LD between the reference run r^* and the j-th LD in current run r, computed from FVH-LT (variance matrix $\mathbf{S}$) or DBSR-LS (coefficient matrix $|\mathbf{D}|$).

Algorithm 3 The GAS

Input: KL divergence $\mathbf{d} = \{d_{rk}\}_{r=1,\ldots,R;\ k=1,\ldots,K}$, matrix $\{\mathbf{A}_r\}_{r=1,\ldots,R}$ to be aligned (variance matrix $\mathbf{S}$ from FVH-LT, coefficient matrix $|\mathbf{D}|$ from DBSR-LS), correlation threshold ρ
Output: Latent-space-aligned $\mathbf{A}$
for $r = 1$ to R **do**
    Cluster the LDs into two groups based on $\{d_{rk}\}_{k=1,\ldots,K}$; the cluster with the higher average KL is defined as the informative set $\mathcal{I}_r$
end for
Define $r^* \triangleq \arg\max_{r=1,\ldots,R} |\mathcal{I}_r|$
for $r \neq r^*$ **do**
    Define $\mathbf{C}_{|\mathcal{I}_{r^*}|\times|\mathcal{I}_r|}$, where $C_{ij} = \mathrm{corr}(v_{r^*i}, v_{rj})$ for $i \in \mathcal{I}_{r^*}, j \in \mathcal{I}_r$
    Initialize position mapping $\mathcal{F}_r = \emptyset$
    while unmatched indices remain in both $\mathcal{I}_{r^*}$ and $\mathcal{I}_r$ & $\max(\mathbf{C}) > \rho$ **do**
        $(i, j) \triangleq \arg\max_{i',j'} C_{i'j'}$
        $\mathcal{F}_r \leftarrow \{\mathcal{F}_r, j \to i\}$; $\mathbf{C} \leftarrow \mathbf{C}_{-i,-j}$
    end while
    Randomly match remaining positions in (A_{r^*}, A_r)
    Re-align by LDs: $A_r \leftarrow \mathcal{F}_r(A_r)$
end for
Return aligned $\{A_r\}_{r=1,\ldots,R}$

3.4 Latent Space Disentanglement Index (LSDI)

From FVH-LT and DBSR-LS, we obtain quantitative latent–feature association matrix $\mathbf{A}$ of dimension $K \times p$ ($\mathbf{S}$ in FVH-LT and $\hat{\mathbf{D}}$ in DBSR-LS). $\mathbf{A}$ characterize how each LD relates to input features and can be visualized as heatmaps, providing a straightforward and intuitive visualization of the structural relationship between LDs and $\mathbf{X}$. To further quantify the disentanglement, we introduce a metric that can be computed from $\mathbf{A}$ – Latent Space Disentanglement index (LSDI). Since the generation of $\mathbf{A}$ does not rely on knowledge of ground-truth generative factors, neither does LSD, offering an advantage over existing disentanglement metrics that require such information. Before we present LSDI, we first define what a perfectly disentangled latent space is and what a completely entangled latent space is in Defs 3 and 4, respectively.

Definition 3 (perfectly disentangled latent space) *A latent space is perfectly disentangled if each informative LD encodes a unique and non-overlapping aspect of the underlying generative factors, while non-informative latent dimensions contain no meaningful information.*

Definition 4 (completely entangled latent space) *A latent space is completely entangled if no separation exists between informative and non-informative LDs, and all LDs channel identical and overlapping information transmitted from* $\mathbf{X}$ *to* $\mathbf{Z}$*.*

Per Definitions 3, $\mathbf{A}$ exhibits a block-separable structure: each informative LD would have non-zero entries only on features corresponding to a single generative factor, with all other entries equal or close to zero, whereas non-informative LDs would contain only zero or close to zero entries. In such a fully separated configuration, a disentanglement evaluation metric should reach its maximum value. Per Definitions 4, the rows in $\mathbf{A}$ are indistinguishable rows, leading to the minimum value of the disentanglement metric. The proposed LSDI below is designed to reflect this continuum, assigning higher values to more clearly separated latent–feature association structures and lower values to less distinguishable ones.

Definition 5 (LSDI) *Let* $A \in \mathbb{R}^{K \times p}$ *denote the output matrix from FVH-LT or DBSR-LS. and* $A_i \in \mathbb{R}^p$ *denote the i-th row of* $\mathbf{A}$*, the* ℓ_1*-norm of which is* $||A_i||_1 = \sum_{k=1}^{p} |A_{i,k}|$*. Let* $\mathcal{P} = \{(i,j) : 1 \leq i < j \leq K\}$ *denote the set of all unordered pairs of LDs. Then*

$$\text{LSDI}(A) = \frac{\sum_{(i,j)\in\mathcal{P}} \| |A_i| - |A_j| \|_1}{\sum_{(i,j)\in\mathcal{P}} (\|A_i\|_1 + \|A_j\|_1)} \in [0,1]; \tag{7}$$

$$= 0 \text{ if } \|A\|_1 = 0; \tag{8}$$

$$= 0 \text{ if there exists a row } i \text{ such that } A_{i,k} \geq A_{j,k} \text{ for all } k \text{ and } j \neq i. \tag{9}$$

LSDI is 1 for a perfectly disentangled latent space, and 0 for a completely entangled latent space. A higher LSDI indicates stronger latent structural separation, while lower values suggest overlapping or entangled feature encoding. Eq. 7 contains contribution from each pair out of a total of $C(^2_K)$ pairs. For a pair consisting of two informative LDs, if they encode perfectly separated distinct sets of input features, then $\|A_i| - |A_j\|_1$ equals $\|A_i\|_1 + \|A_j\|_1$, so the contribution from that pair to the LSDI is 1. For most pairs that involving two informative LDs, it is expected its contribution to the total LSDI is $\in (0,1)$. For a pair consisting of two non-informative LDs, both rows are zero and such pairs therefore have no effect on the overall LSDI. If a pair consists of one informative LD and one non-informative LD, the non-informative row contains only zeros. Consequently, $\|A_i - A_j\|_1 = \|A_i\|_1$ and $\|A_i\|_1 + \|A_j\|_1 = \|A_i\|_1$, yielding a unit contribution toward LSDI. Eqs. 8 and 9 represents two degenerative cases, as illustrated in Fig. 4.

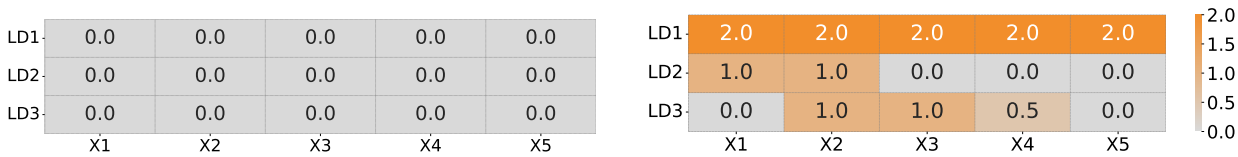


Figure 4: Two degenerate cases resulting in LSDI of 0. Left: All entries of matrix $\mathbf{A}$ are 0, indicating the complete absence of informative LDs. Right: A single LD dominates all other LDs across all input features in $\mathbf{A}$, resulting in no disentanglement.

LSDI provides a direct and convenient quantitative measure of structural disentanglement derived from the latent–feature association matrices produced by FVH-LT and DBSR-LS, making comparison among different methods that acomparison across different objective functions in terms of how effectively the two procedures separate and organize latent representations.

4 Experiments

The experiments empirically evaluate the proposed bfVAE framework in its disentanglement efficiency and the robustness of FVH-LT, DBSR-LS, and LSDI in quantifying the disentanglement effect. We examine different types of diverse datasets, with particular emphasis on tabular data. We consider settings with and without known generative factors, as well as scenarios where target variables are available, to demonstrate that the proposed procedures remain applicable and informative in various data scenarios. The seven datasets examined in this sections are summarized in Tab. 1. The three simulated datasets vary in both feature dimensionality and sample size; the white wine dataset (Cortez et al., 2009) includes features describing white wine characteristics, with a label on the quality score; the FIFA 2018 dataset (Mathan, 2018) contains various soccer match statistics; and MNIST(Deng, 2012) and CelebA(Liu et al., 2015) are widely used benchmark image data.

Table 1: Datasets used in the experiments

Dataset	**Total n; Training n**	**No. of Features p**	**Known Ground Truth?**
FA15 (tabular)†	1,000; 800	15	Yes (4 factors)
FA24 (tabular)†	10,000; 8,000	24	Yes (6 factors)
FA100 (tabular)†	50,000; 40,000	100	Yes (6 factors)
White wine (tabular)#	4,898; 3918	12	No
FIFA 2018 (tabular)#	128 (64 matches); 102 (51 matches)	16	No
MNIST (image)#	60,000; 50,000	$28 \times 28 \times 1 = 784$	No
CelebA (image)#	40,000; 35,000	$64 \times 64 \times 3 = 12288$	No

† We synthesized the FA15, FA24, and FA100 datasets from factor analysis models.
# Publicly available at Kaggle, PyTorch, and UC Irvine Machine Learning Repository.

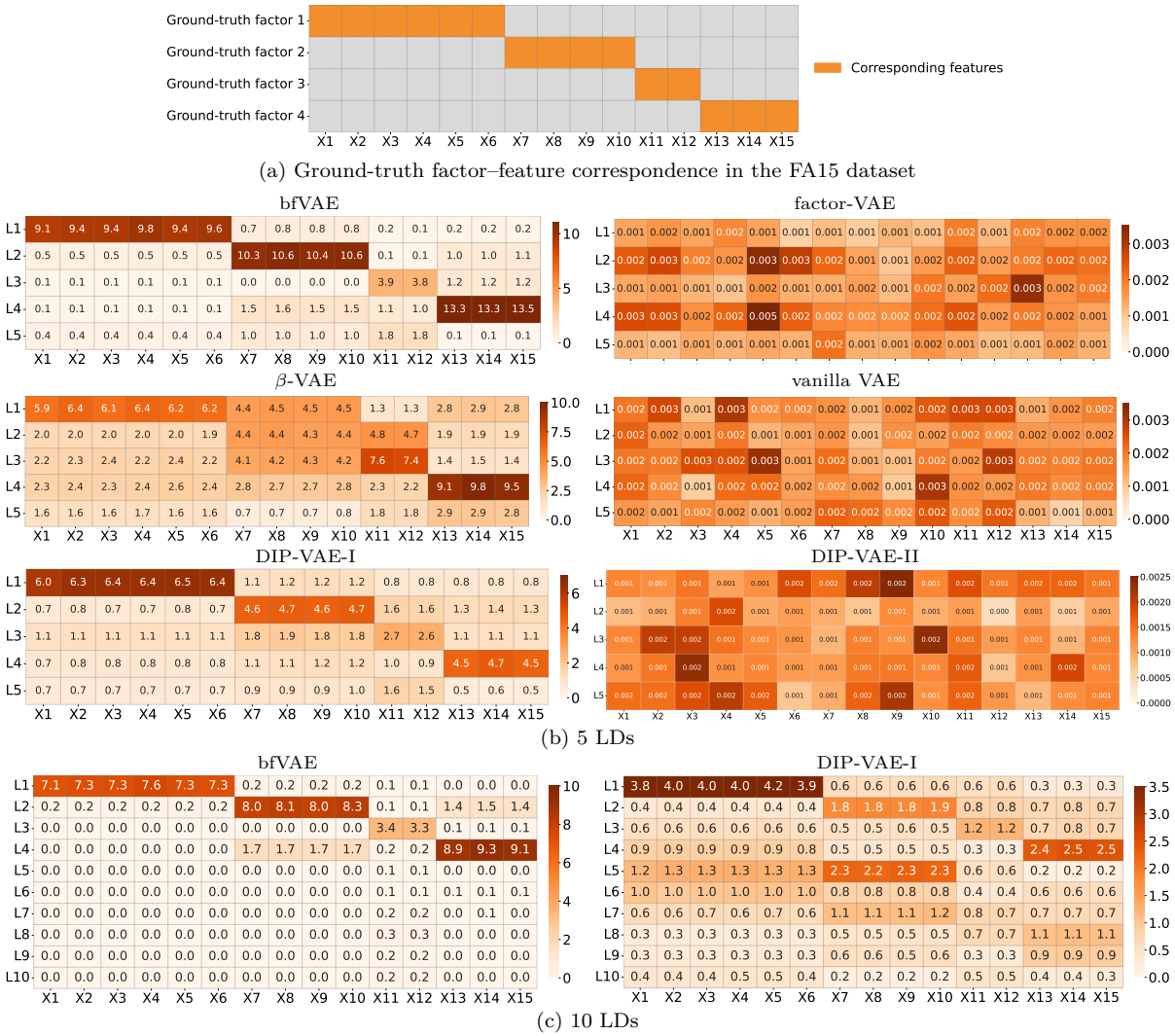
In each experiment, the dataset was split into a training set and a test set. The training set was used to train VAE and perform the FVH-LT and DBSR-LS procedures and the testing set was used to evaluate modeling fitting and convergence during VAE training. bfVAE was trained to optimize the objective function in Eq. 1 ($C = 0$ in the five tabular datasets and $C > 0$ in the MNIST and CelebA datasets). We used $\beta < 1$ for bfVAE in the tabular data experiment R was set at 10 in all experiments except for CelebA, where $R = 5$ due to computational constraints. More details on model configurations and algorithmic hyperparameters are provided in App. A; computational time is listed in App. B. We provide a subset of the experimental results in the main text and list the comprehensive results in App. I.

In summary, the results on disentanglement effectiveness evaluation are generally consistent between FVH-LT and DBSR-LS, but FVH-LT is slightly better in that the informative LD signals are stronger in some cases with more clear-cut disentanglement. DBSR-LS is computationally more costly. FVH-LT only requires sample variance calculation and subsequent averaging in each run, leading to linear time complexity $O(Ln)$ in training size n and LT steps L. DBSR-LS, on the other hand, requires iterative optimization to solve for $\mathbf{D}$ from the regression model in Eq. 6, the time complexity per run is $O(Tp(n + K))$, where the iteration number T depends on the solver. When T or p is large, DBSR-LS can be costly to run. In the CelebA dataset ($p = 12,288$), the high computational costs actually exceed our resources. Due to space limitations and for the reasons listed above, we present the FVH-LT results in the main text; the DBSR-LS results can be found in Apps. C to P.

4.1 Benchmark comparison and ablation study for bfVAE

We first demonstrate that bfVAE outperforms the state-of-the-art VAE disentanglement frameworks: vanilla VAE, factor-VAE, β-VAE, DIP-VAE-I, and DIP-VAE-II (see Apps. A.8 and A.9 for the corresponding loss functions) using the FA15 dataset. There are 4 latent factors in the FA15 dataset, each associated with a different subset of $\mathbf{X}$, as depicted Fig. 5(a). The block structure provides a clear ground-truth

reference for evaluating whether each learned LD aligns with a distinct subset of input features. Fig. 5(b)-(c) presents the FVH-LT results. Since vanilla VAE, β-VAE, and factor-VAE are special cases of bfVAE, their comparisons with bfVAE also comprise an ablation study by removing the β-VAE and factor-VAE components, respectively.



(a) Ground-truth factor–feature correspondence in the FA15 dataset

(b) 5 LDs

(c) 10 LDs

Figure 5: Comparison between bfVAE vs other disentangled VAE frameworks in FA15. (b) also represents an ablation study as factor-, β-, vanilla- VAEs are special cases of bfVAE by removing a certain component. Darker cells indicate stronger associations between LDs and input features (cell values not shown due to readability reasons).

In summary, bfVAE successfully discovers the full ground-truth factor structure in FA15. bfVAE is similar to DIP-VAE-I and outperforms all the other frameworks when $K = 5$ (Fig. 5(b)) and is better than DIP-VAE-I when $K = 10$ (Fig. 5(c)). In contrast, factor-VAE and vanilla VAE fail to separate the underlying factors, with LDs exhibiting mixed associations across features. The posterior distributions learned by factor-VAE and vanilla VAE remain close to the prior, yielding near-zero KL divergence and indicating posterior collapse. β-VAE exhibits partial separation across LDs and is able to discover some factor-related structure at the same β value and $C = 0$ that bfVAE adopts. This observation confirms the importance of term T3 in Eq. 1. When K is over-specified, as presented in Fig. 5(c), DIP-VAE-I over-picks informative dimensions with high false discovery rates on informative LDs while bfVAE has a zero false discovery rate. Consistent results are observed for the DBSR-LS procedure (App. C).

We further compute the LSDI values based on the outputs of FVH-LT and DBSR-LS to compare bfVAE and existing disentangled VAE formulations. In addition, given that FA15 is a synthetic dataset with known ground-truth generative factors, we further computed the disentanglement score in Higgins et al. (2017) (see Sec. 1.3 for details) As shown in Tab. 2, bfVAE consistently achieves the highest LSDI scores, indicating the best structural separation among LDs. Similarly in the case of the disentanglement score, bfVAE attains the highest score, followed by DIP-VAE-I. Overall, these findings demonstrate that bfVAE, together with FVH-LT and DBSR-LS, reliably uncovers disentangled and interpretable latent structure across dimensions.

Table 2: LSDI scores for different disentanglement VAE formulations ($K = 5$) in the FA15 data

		bfVAE	FactorVAE	β-VAE	Vanilla VAE	DIP-VAE-I	DIP-VAE-II
LSDI‡	FVH-LT	***0.842***	0.188	0.362	0.146	0.500	0.166
	DBSR-LS	***0.796***	0.359	0.725	0.321	0.630	0.260
disentanglement	score‡	***1.000***	0.253	0.867	0.253	0.987	0.253

bold italic indicates the best scores, indicating our bfVAE has the best disentanglement performance. LSDI was also computed based on the FVH-LT output for bfVAE and DIP-VAE-I with $K = 10$, bfVAE outperforms DIP-VAE-I with a LSDI of 0.933 compared to 0.432 for the latter.

‡ The computation of LSDI does not use ground-truth generative factors information, whereas disentanglement score (Higgins et al., 2017) does (the worst score is 0.25, as there are 4 ground-truth factors).

4.2 Tabular data with knowledge on true generative factors

We evaluate the effectiveness of FVH-LT and DBSR-LS on more complex synthetic tabular datasets FA24 and FA100 with known generative factors. Fig. 6 visualizes the factor-feature association of FA24 and FA100. There are 6 generative factors in both, but FA100 has a substantially larger number input features.

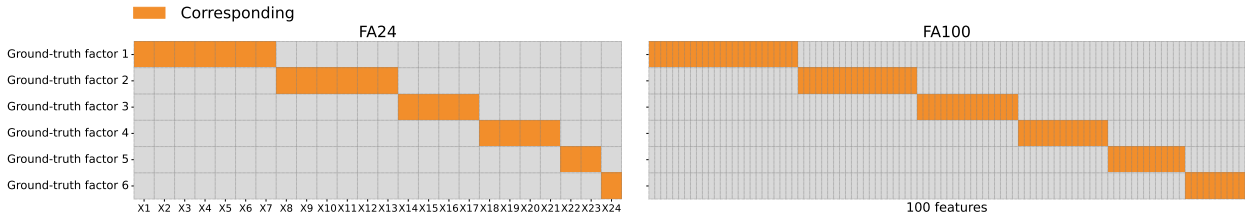


Figure 6: Ground-truth factor–feature correspondence in the FA24 and FA100 datasets.

The FVH-LT results in Fig. 7 suggest that bfVAE successfully disentangle the latent structure, with the learned associations between informative LDs and input features closely aligning with the ground truth. It is also worth noting the robustness of FVH-LT to over-specified K, similar to the observations in Fig. 5(b); that is, it does not mis-identify non-informative LDs to be informative, resulting in a low false positive rate. The DBSR-LS results are consistent with FVH-LT and are provided in Apps. E and F.

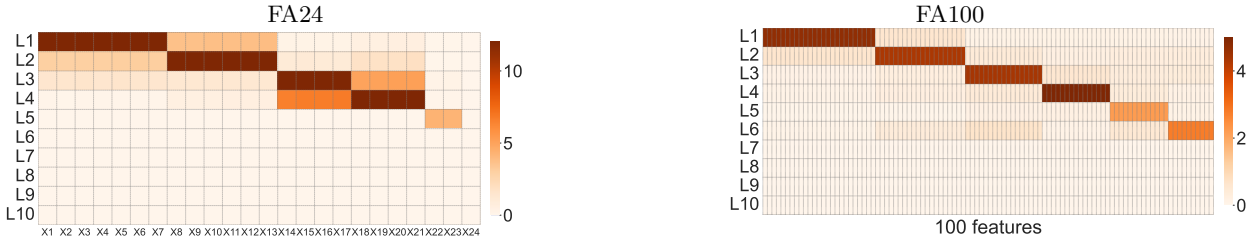


Figure 7: Disentanglement effects of bfVAE assessed by FVH-LT in FA24 and FA100. Darker cells indicate stronger associations between latent dimensions (rows) and input features (columns) (cell values not shown due to readability reasons).

4.3 Tabular data without knowledge on generative factors

We next consider real-world tabular datasets where the underlying generative factors are not explicitly known, and evaluate whether our procedures can still provide meaningful and domain-consistent latent interpretations. The white wine data consists of 11 physicochemical features of white wine and a label on

sensory quality ratings from expert tasters. The input features reflect key aspects of white wine composition and production, and can be grouped into categories: acidity-related attributes (fixed acidity, citric acid, and pH); flavor and fermentation-related properties (residual sugar, total sulfur dioxide, density, and alcohol); and individual chemical compounds (volatile acidity, chlorides, free sulfur dioxide, and sulphates). The 2018 FIFA statistics data contains 16 match-level features on team performance from the 2018 FIFA World Cup. Fig. 8 presents the FVH-LT results on the two datasets.

In summary, although the true generative factors underlying the two real datasets are unknown, the latent space disentanglement via bfVAE assessed by FVH-LT is interpretable and consistent with domain knowledge. Specifically, for the white wine data, LD1 is associated with fixed acidity, citric acid, and pH, reflecting the wine acidity profile. LD2 is associated with chlorides, and LD3 with volatile acidity. LD4 encodes free sulfur dioxide and sulphates that are common preservatives to stabilize and protect wine from microbial activity. LD5 is strongly associated with density, alcohol, residual sugar, and total sulfur dioxide, which are characteristics related to wine flavor. LD6 is non-informative. In the 2018 FIFA data, LD1 is strongly associated with Distance Covered, Fouls Committed, and Yellow cards – reflecting physical exertion and disciplinary intensity. LD2 encodes Goal Scored and On-Target, representing attacking play. LD3 encodes Ball Possession, Attempts, Blocked, Corners, Pass Accuracy, and Passes, encapsulating possession-based play and attacking build up. LD4 to LD6 are noninformative.

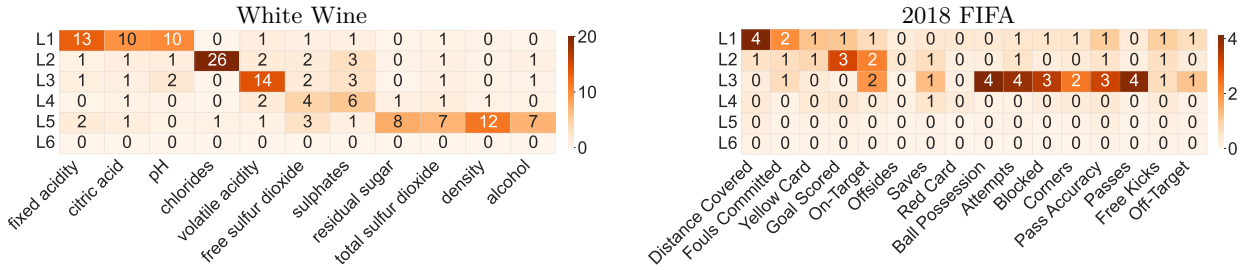


Figure 8: Disentanglement effects of bfVAE assessed by FVH-LT in datasets without knowledge of ground-truth generative factors. Darker cells indicate stronger associations between latent dimensions (rows) and input features (columns).

We also explored the application of the FVH-LT in bf-CVAE on the white wine data, given that it contains label information Y (wine quality). We aim to leverage $Y$to identify features that are most relevant to Y, so as to improve the efficiency of targeted content generation given a certain condition in Y. The detailed implementation of bf-CVAE and the full FVH-LT results are provided in App. J. Tab. 3 reports the last row of the FVH-LT heatmap, which summarizes how each feature varies when perturbing the conditioning variable Y. We observe that volatile acidity (VC), chlorides (C), density (D), and alcohol (A) exhibit the highest variance from traversing Y, indicating strong relations with wine quality and consistency with domain knowledge (see App. Q). Leveraging these findings, a wine manufacture can fix Y at a high-quality score, randomly sample along LDs that exhibits strong associations between the four input features (VC, C, D, A) and Y, and feed these samples to the decoder to generate feature combinations that could potentially yield a new wine with the targeted high-quality score.

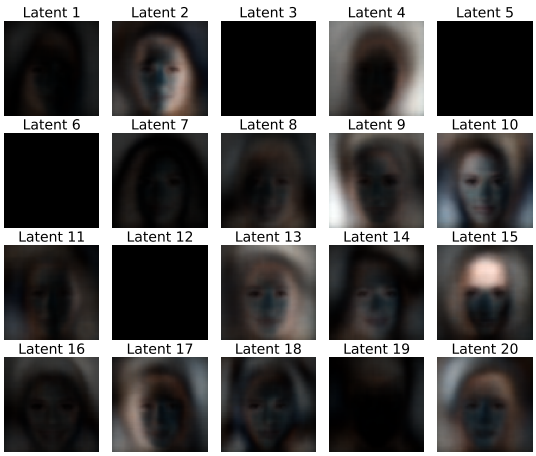
Table 3: Variance of reconstructed features with LT on wine quality score via FVH-LT.

fixed acidity	**volatile acidity**	citric acid	residual sugar	**chlorides**	free SO_2
0.08	**0.17**	0.08	0.06	**0.14**	0.05
total SO_2	**density**	pH	sulphates	**alcohol**	**quality**
0.09	**0.25**	0.07	0.05	**0.54**	**2.9**

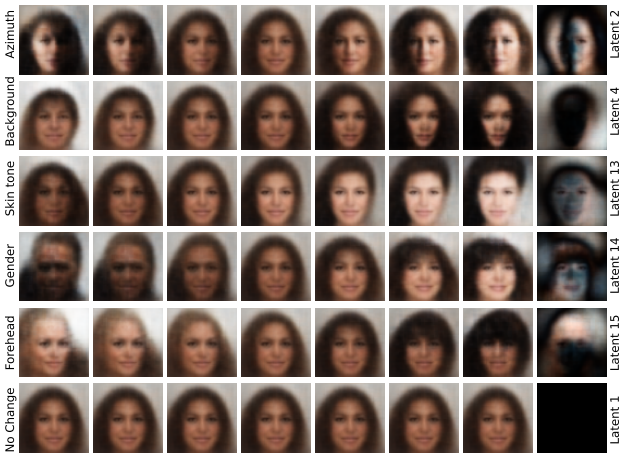
4.4 Image data CelebA and MNIST

To demonstrate that bfVAE combined with FVH-LT and DBSR-LS can uncover meaningful disentangled latent structure beyond tabular settings, we examine image data in this section, Importantly, our focus is not on optimizing reconstruction fidelity, but on identifying interpretable and semantically distinct factors encoded in the latent space.

CelebA. In the FVH-LT analysis of CelebA that contains three color channels (red, green, and blue), heatmaps were first computed separately by channel, then globally normalized and merged to produce a composite visualization in Fig. 9(a). The LT of a representative image (Fig. 9(b)) illustrates how informative LDs correspond to meaningful facial features in CelebA images. Specifically, FVH-LT accurately displays which meaningful facial features correspond to which informative LD, aiding the interpretability. For example, LT of LD1 produces no visible change in the reconstruction and results in a completely dark heatmap in Fig. 9(a) and 9(b), indicating the non-informativeness of LD1. In contrast, LT of LD2 reflects changes in azimuth (horizontal head orientation) given that the FVH-LT heatmap highlights the left and right regions of the face; LD4 primarily encodes background; LD5 encodes skin complexity, LD14 seems to encode gender information, and LD15 encodes forehead.

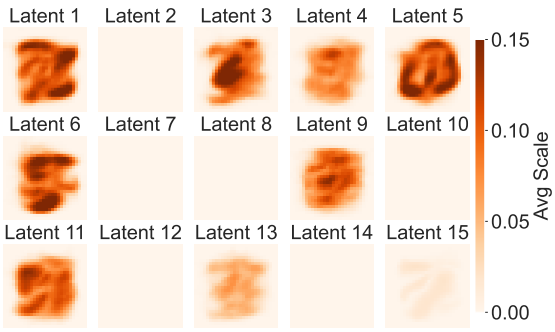


(a) FVH-LT results averaged over 1000 images and 5 runs. Brighter regions indicate higher variance, highlighting image areas that are sensitive to perturbations in the latent space.

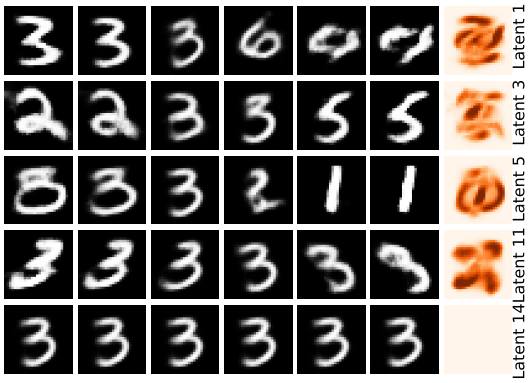


(b) Example images in Columns 1 to 7 reconstructed from LT on 6 LDs in a single run; the corresponding FVH-LT heatmaps are in Column 8.

Figure 9: Disentanglement effects of bfVAE assessed by FVH-LT on CelebA



(a) FVH-LT results (averaged over 1000 images and 10 runs)



(b) Example images in Columns 1 to 6 reconstructed from LT on 5 LDs in a single run; the corresponding FVH-LT heatmaps are in Column 7

Figure 10: Disentanglement effects of bfVAE assessed by FVH-LT in MNIST

MNIST. Figs. 10(a) shows the FVH-LT results and 10(b) displays the LT results for an image with digit "3". Fig. 10(a) suggests there are 7 informative LDs, where meaningful pixels are concentrated at the image center, reflecting the fact that the digits are centered in the original images. Fig. 10(a) and the last column of Fig. 10(b) also suggest that the informative LDs tend to capture combinations of digit shapes (this phenomenon is more apparent in the signed $\hat{\mathbf{D}}$ from the DBSR-LS procedure presented in App. P). Specifically, LD1 appears as a combination of a 3 and an ellipse; LD3 resembles a blend of digits 2 and 5; LD5 shows features combining 8 and 1; and LD11 presents two slanted versions of the digit 3. Key interpretable structural elements that differentiate the digits are clearly evident, such as the circular loops and the center

vertical line in LD5 and curved regions near the top-left and bottom-right in LD1, indicating the importance of the stroke directionality in digit formation. The 7 informative LDs identified in Fig. 10(a) produce obvious changes in reconstructed images under LT (first four rows in Fig. 10(b)). In contrast, reconstructed images by varying a non-informative LD in LT exhibit negligible variation (last row in Fig. 10(b)), and correspond to uniformly inactive a FVH-LT heatmap. In addition, LT in the informative LDs yields interpretable semantic patterns. For example, LT of LD3 leads to generation of digits "2","3","5" and LT of LD5 leads to generation of "3", "2", "1". The FVH-LT heatmap reflects this morphing process in that, for example, the patten exhibited in the last column of Fig. 10(b) in the LD3 row suggest a blend of digits "2","3","5", and a similar effect is observed for the LD5 row. LD11 is interesting as it captures variations in handwriting style within the same digit, a critical latent factor in handwritten digit representation, and this is clearly reflected in the corresponding FVH-LT heatmap.

4.5 Visualizing posterior distributions of informative and non-informative LDs

Algs. 1 and 2 calculate the posterior-prior KL divergence for each LD as intermediate product, on which the informative vs non-informative LDs in Definitions 1 and 2 are based. To understand what drives the posterior distributions of informative LDs to deviate from the priors, we plot the posterior distributions of the LDs from a single run of bfVAE with $\gamma = 0$ on a random subset of 1,000 MNIST training images in Fig. 11. We set the dimensionality of LDs at 15. The KL-divergence suggests there are seven informative LDs, the posterior distributions of which exhibit low variance (< 0.1) and many also have posterior means shifted from 0, the main reason behind their deviation from prior $\mathcal{N}(0, 1)$ and large prior-posterior KL divergence (~ 8 on average). The rest eight LDs are non-informative with an KL divergence of $\sim 3 \times 10^{-3}$ on average. Similar patterns are consistently observed in other runs.

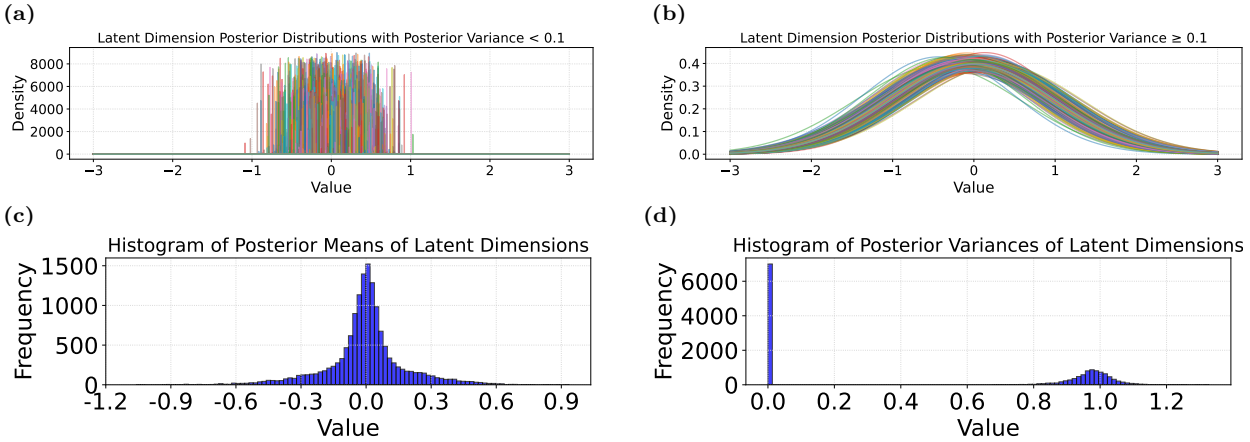


Figure 11: Posterior statistics across LDs on MNIST (15 LDs $\times$ 1000 MNIST images): (a) posterior distributions of informative LDs, (b) posterior distributions of non-informative LDs, (c) histogram of posterior means, and (d) histogram of posterior variances (among the 15,000 LDs, exactly 7,000 display near-zero variances in their distribution and correspondingly exhibit large KL divergence).

5 Conclusion and Discussion

We proposed a unified framework – bfVAE – for VAE latent space disentanglement with an information-bottleneck perspective. We introduced and evaluated two robust and general-purpose techniques, FVH-LT and DBSR-LS, for interpreting and quantifying the effect of VAE latent space disentanglement techniques, along with GAS to address label switching in latent space from multiple runs to improve result consistency and robustness. We also introduced the LSDI, an overall quantitative metric derived from the latent-feature association matrix produced by FVH-LT and DBSR-LS to evaluate structural separation in the latent space, without knowledge of ground-truth generative factors. Our extensive experiments on data of various types and sizes show that bfVAE outperforms existing VAE disentanglement frameworks in disentangling and understanding the semantics of the LDs in VAEs.

Both FVH-LT and DBSR-LS effectively uncover semantically meaningful, domain-aligned latent structures, even in the absence of precise knowledge of true generative factors. That said, FVH-LT is preferable from both methodological and computational perspectives: it is assumption-lean and thus more general and robust methodologically and computationally it is cheaper than DBSR-LS.

Future directions include applying our procedures to other data types (e.g., graphs), adapting the procedures to other deep learning models that may benefit from interpretable latent spaces disentanglement, and extending DBSR-LS to non-linear regression settings.

References

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Seungheun Baek, Soyon Park, Yan Ting Chok, Junhyun Lee, Jueon Park, Mogan Gim, and Jaewoo Kang. Cradle-vae: Enhancing single-cell gene perturbation modeling with counterfactual reasoning-based artifact disentanglement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39 (15), pp. 15445–15452, 2025.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-VAE. *arXiv preprint arXiv:1804.03599*, 2018.

Clément Chadebec, Louis Vincent, and Stephanie Allassonniere. Pythae: Unifying generative autoencoders in python-a benchmarking use case. *Advances in Neural Information Processing Systems*, 35:21575–21589, 2022.

Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018.

Paulo Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Wine Quality. UCI Machine Learning Repository, 2009. DOI: https://doi.org/10.24432/C56S3T.

Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. doi: 10.1109/MSP.2012.2211477.

Cian Eastwood and Christopher KI Williams. A framework for the quantitative evaluation of disentangled representations. In *6th International Conference on Learning Representations*, 2018.

William Harvey, Saeid Naderiparizi, and Frank Wood. Conditional image generation by conditioning variational auto-encoders. *arXiv preprint arXiv:2102.12037*, 2021.

Lucas Heublein, Simon Kocher, Tobias Feigl, Alexander Rügamer, Christopher Mutschler, and Felix Ott. VAE-based feature disentanglement for data augmentation and compression in generalized gnss interference classification. *arXiv preprint arXiv:2504.10556*, 2025.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International conference on learning representations*, 2017.

Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

Ali Jalali, Sujay Sanghavi, Chao Ruan, and Pradeep Ravikumar. A dirty model for multi-task learning. *Advances in neural information processing systems*, 23, 2010.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International conference on machine learning*, pp. 2649–2658. PMLR, 2018.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Chengrui Li, Yunmiao Wang, Yule Wang, Weihan Li, Dieter Jaeger, and Anqi Wu. A revisit of total correlation in disentangled variational auto-encoder with partial disentanglement. *arXiv preprint arXiv:2502.02279*, 2025.

Xiao Liu, Pedro Sanchez, Spyridon Thermos, Alison Q O'Neil, and Sotirios A Tsaftaris. Learning disentangled representations in the imaging domain. *Medical Image Analysis*, 80:102516, 2022.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

F Locatello, S Bauer, M Lucic, G Rätsch, S Gelly, B Schölkopf, and O Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. arxiv preprint arxiv: 1811.12359, 2018.

Mathan. Predict FIFA 2018 man of the match. `https://www.kaggle.com/datasets/mathan/fifa-2018-match-statistics`, 2018. Accessed: 2025-05-02.

Mozhgan Pourkeshavarz, Junrui Zhang, and Amir Rasouli. CaDeT: a causal disentanglement approach for robust trajectory prediction in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14874–14884, 2024.

Evan Poworoznek, Niccolo Anceschi, Federico Ferrari, and David Dunson. Efficiently resolving rotational ambiguity in Bayesian matrix sampling with matching. *arXiv preprint arXiv:2107.13783*, 2021.

Yijian Qin, Xin Wang, Ziwei Zhang, and Wenwu Zhu. Disentangled representation learning with large language models for text-attributed graphs. *arXiv preprint arXiv:2310.18152*, 2023.

Xinwei Shen, Furui Liu, Hanze Dong, Qing Lian, Zhitang Chen, and Tong Zhang. Weakly supervised disentangled generative causal representation learning. *Journal of Machine Learning Research*, 23(241): 1–55, 2022.

Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Anshuk Uppal, Yuhta Takida, Chieh-Hsin Lai, and Yuki Mitsufuji. Denoising multi-beta vae: Representation learning for disentanglement and generation. *arXiv preprint arXiv:2507.06613*, 2025.

Xin Wang, Hong Chen, Si'ao Tang, Zihao Wu, and Wenwu Zhu. Disentangled representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Ziye Zhang, Li Sun, Zhilin Zheng, and Qingli Li. Disentangling the spatial structure and style in conditional VAE. In *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 1626–1630. IEEE, 2020.

A Detailed model parameters in the experiments

A.1 Simulated tabular data with ground-truth knowledge

For the synthetic tabular experiments, we used a fully connected VAE architecture with a symmetric encoder-decoder design, trained using the loss in Eq. (1). The encoder consists of three hidden layers with dimensions 256, 128, and 64, each followed by a ReLU activation and dropout with a rate of 0.2. The encoder outputs are passed through two parallel linear layers to produce the mean and log-variance for the latent space (5-dimension for FA15, 10-dimension for FA24). The decoder mirrors the encoder, with hidden layers of

size 64, 128, and 256, followed by ReLU and dropout activations, and concludes with a linear output layer mapping to the input dimension. Samples from the latent space were obtained using the reparameterization trick (Kingma & Welling, 2013). A discriminator network is used to estimate the Total Correlation (TC) of the aggregated posterior (Kim & Mnih, 2018). The discriminator consists of five fully connected layers with 1000 units each, interleaved with LeakyReLU activations (negative slope = 0.2), and ends with a linear layer outputting two logits that distinguish between real and permuted latent.

For data FA15, we used $\beta = 2 \times 10^{-3}$ and $\gamma = 0.3$ for FVH-LT, and $\beta = 1 \times 10^{-3}$ and $\gamma = 0.2$ for DBSR-LS. For DBSR-LS, the regularization coefficients were set to $\lambda_B = 0.1$ and $\lambda_S = 0.1$.

For data FA24, we used $\beta = 2 \times 10^{-3}$ and $\gamma = 0.25$ for both FVH-LT and DBSR-LS. The regularization parameters for DBSR-LS were set to $\lambda_B = 0.2$ and $\lambda_S = 0.2$.

For data FA100, we used $\beta = 1 \times 10^{-4}$ and $\gamma = 0.3$ for FVH-LT, and $\beta = 7 \times 10^{-4}$ and $\gamma = 0.3$ for DBSR-LS. For DBSR-LS, the regularization coefficients were set to $\lambda_B = 0.3$ and $\lambda_S = 0.3$.

For all datasets, models were trained with a batch size of 16 using the Adam optimizer with a learning rate of 1×10^{-4}. The correlation threshold for alignment was set to $\rho_{\min} = 0.5$. LT was performed over the fixed range $[-15, 15]$, and each experiment was repeated $R = 10$ times with identical configurations to ensure stability and enable consistent latent alignment.

A.2 Real-world tabular data without ground-truth knowledge

For the real-world data experiments, we used a fully connected VAE architecture with a symmetric encoder-decoder design, trained using the objective in Eq.(1). The encoder consists of three hidden layers with dimensions 256, 128, and 64, each followed by a ReLU activation and dropout with a rate of 0.2. The encoder outputs are passed through two parallel linear layers to produce the mean and log-variance of a 6-dimensional latent space. The decoder mirrors the encoder with hidden layers of size 64, 128, and 256, followed by ReLU and dropout activations, and concludes with a linear output layer mapping to the input dimension. Samples from the latent space were obtained using the reparameterization trick. A discriminator network is used to estimate the Total Correlation (TC) of the aggregated posterior. The discriminator consists of five fully connected layers with 1000 units each, interleaved with LeakyReLU activations (negative slope = 0.2), and ends with a linear layer outputting two logits that distinguish between real and permuted latent.

For white wine data, we used $\beta = 2 \times 10^{-3}$ and $\gamma = 0.3$ for FVH-LT, and $\beta = 3 \times 10^{-3}$ and $\gamma = 0.3$ for DBSR-LS. The DBSR-LS regularization coefficients were set to $\lambda_B = 1 \times 10^{-2}$ and $\lambda_S = 1 \times 10^{-2}$. Models was trained with a batch size of 16 using the Adam optimizer with a learning rate of 1×10^{-4}. The correlation threshold for alignment was set to $\rho_{\min} = 0.5$.

For 2018 FIFA statistics data, we used $\beta = 1 \times 10^{-3}$ and $\gamma = 0.3$ for FVH-LT, and $\beta = 1 \times 10^{-3}$ and $\gamma = 0.3$ for DBSR-LS. The regularization coefficients for DBSR-LS were also set to $\lambda_B = 1 \times 10^{-2}$ and $\lambda_S = 1 \times 10^{-2}$. Model was trained with a batch size of 8 using the Adam optimizer with a learning rate of 5×10^{-5}. The correlation threshold for alignment was set to $\rho_{\min} = 0.5$.

LT was performed over the fixed range $[-15, 15]$, and each experiment was repeated $R = 10$ times with identical configurations to ensure stability and enable consistent latent alignment.

For the CVAE applications, we used a fully connected VAE architecture with a mirrored encoder-decoder design and objective function in Eq.(1). The encoder consists of three hidden layers with dimensions 256, 128, and 64, each followed by a ReLU activation and dropout with a rate of 0.2. The encoder outputs are passed through two parallel linear layers to produce the mean and log-variance of a 6-dimensional latent space. The decoder contains three hidden layers of size 64, 128, and 256, followed by ReLU and dropout activations, and concludes with a linear output layer mapping to the input dimension. Samples from the latent space were obtained using the reparameterization trick.

We used $\beta = 1 \times 10^{-2}$ and $\gamma = 0.3$. Model was trained with a batch size of 16 using the Adam optimizer with a learning rate of 1×10^{-4}.LT was conducted over the quality latent using a data-driven range of $[-3.2, 3.5]$, while the remaining latent dimensions were traversed over a fixed interval of $[-15, 15]$.To ensure stability

and enable consistent alignment of latent dimensions, each experiment was repeated $R = 10$ times under identical configurations.

A.3 MNIST

We employed VAE architecture in the Pythae (Chadebec et al., 2022) with a disentangled β-VAE loss function. Specifically, we used a ResNet-based encoder-decoder design tailored for image inputs. The encoder consists of four sequential blocks: (1) a convolutional layer with 64 output channels, 4×4 kernel, stride 2, and padding 1; (2) a convolutional layer with 128 output channels, 4×4 kernel, stride 2, padding 1; (3) a convolutional layer with 128 output channels, 3×3 kernel, stride 2, padding 1; (4) two residual blocks, each composed of ReLU → Conv2d(128, 32, 3×3) → ReLU → Conv2d(32, 128, 1×1). The output is flattened and passed through two fully connected layers (2048 → 10) to generate the latent mean and log-variance vectors for a 10-dimensional latent space.

The decoder mirrors the encoder structure in reverse. It begins with a fully connected layer (10 → 2048), followed by a transposed convolution: ConvTranspose2d(128, 128, 3×3, stride 2, padding 1). This is followed by two residual blocks identical in structure to those in the encoder, and then two upsampling layers: (1) ConvTranspose2d(128, 64, 3×3, stride 2, padding 1, output padding 1) with ReLU activation; (2) ConvTranspose2d(64, 1, 3×3, stride 2, padding 1, output padding 1) with Sigmoid activation to reconstruct the 28×28 grayscale image.

The model was trained for 64 epochs with a MSE reconstruction loss, $\beta = 50$, and target capacity $C = 60$ (Eq.(2)) for both FVH-LT and DBSR-LS methods. Latent traversals were performed within the fix range $[-2, 2]$. Each experiment was repeated with $R = 10$ independent runs to ensure stability and enable consistent latent alignment. The regularization coefficients were set to $\lambda_B = 1 \times 10^{-3}$ and $\lambda_S = 1 \times 10^{-3}$. The correlation threshold for alignment was set to $\rho_{\min} = 0.3$.

A.4 CelebA

We employed the disentangled β-VAE architecture implemented in the Pythae library. Specifically, we used a ResNet-based encoder-decoder design tailored for CelebA images. The encoder consists of four sequential convolutional blocks: (1) a convolutional layer with 64 output channels, 4×4 kernel, stride 2, and padding 1; (2) a convolutional layer with 128 output channels, 4×4 kernel, stride 2, and padding 1; (3) a convolutional layer with 128 output channels, 3×3 kernel, stride 2, and padding 1; (4) a convolutional layer with 128 output channels, 3×3 kernel, stride 2, and padding 1.

This is followed by two residual blocks, each composed of ReLU → Conv2d(128, 32, 3×3) → ReLU → Conv2d(32, 128, 1×1). The output is flattened and passed through two fully connected layers (2048 → 16) to generate the latent mean and log-variance vectors for a 16-dimensional latent space.

The decoder mirrors the encoder structure in reverse. It begins with a fully connected layer (16 → 2048), followed by a transposed convolution: ConvTranspose2d(128, 128, 3×3, stride 2, padding 1). This is followed by two residual blocks identical in structure to those in the encoder, and then three upsampling layers: (1) ConvTranspose2d(128, 128, 5×5, stride 2, padding 1) with Sigmoid activation; (2) ConvTranspose2d(128, 64, 5×5, stride 2, padding 1, output padding 1); (3) ConvTranspose2d(64, 3, 4×4, stride 2, padding 1) with Sigmoid activation to reconstruct the 64×64 RGB image.

The model was trained for 64 epochs with a MSE reconstruction loss, $\beta = 250$, and target capacity $C = 150$ (Eq.(1)) for the FVH-LT method. Latent traversals were performed using a data-dependent range of $[\mu - 1, \mu + 1]$, where μ denotes the posterior mean of each latent dimension. Each experiment was repeated with $R = 5$ independent runs to ensure stability and enable consistent latent alignment. For visualization, FVH-LT was applied separately to each color channel, and the resulting variance heatmaps were globally normalized and combined into a single composite heatmap. DBSR-LS was not applied to CelebA due to the high dimensionality of image features.

A.5 Factor-VAE on FA15

For the factor-VAE on FA15 experiments, we used a fully connected VAE architecture with a symmetric encoder-decoder design, trained using the loss in Eq.(1) with $\beta = 1$, indicating a factor-VAE objective function. The encoder consists of three hidden layers with dimensions 256, 128, and 64, each followed by a ReLU activation and dropout with a rate of 0.2. The encoder outputs are passed through two parallel linear layers to produce the mean and log-variance for a 5-dimensional latent space. The decoder mirrors the encoder, with hidden layers of size 64, 128, and 256, followed by ReLU and dropout activations, and concludes with a linear output layer mapping to the input dimension. Samples from the latent space were obtained using the reparameterization trick. A discriminator network is used to estimate the Total Correlation (TC) of the aggregated posterior. The discriminator consists of five fully connected layers with 1000 units each, interleaved with Leaky-ReLU activations (negative slope = 0.2), and ends with a linear layer outputting two logits that distinguish between real and permuted latent.

We used $\gamma = 0.3$ for FVH-LT and $\gamma = 0.2$ for DBSR-LS. For DBSR-LS, the regularization coefficients were set to $\lambda_B = 1 \times 10^{-4}$ and $\lambda_S = 1 \times 10^{-4}$.

Models were trained with a batch size of 16 using the Adam optimizer with a learning rate of 1×10^{-4}. The correlation threshold for alignment was set to $\rho_{\min} = 0.5$. LT was performed over the fixed range $[-15, 15]$, and each experiment was repeated $R = 10$ times with identical configurations to ensure stability and enable consistent latent alignment.

A.6 β-VAE on FA15

For the β-VAE on FA15 experiments, we used a fully connected VAE architecture with a symmetric encoder-decoder design, trained using the loss in Eq.(1) with $\gamma = 0$, indicating a β-VAE objective function. The encoder consists of three hidden layers with dimensions 256, 128, and 64, each followed by a ReLU activation and dropout with a rate of 0.2. The encoder outputs are passed through two parallel linear layers to produce the mean and log-variance for a 5-dimensional latent space. The decoder mirrors the encoder, with hidden layers of size 64, 128, and 256, followed by ReLU and dropout activations, and concludes with a linear output layer mapping to the input dimension. Samples from the latent space were obtained using the reparameterization trick.

We used $\beta = 2 \times 10^{-3}$ for FVH-LT and $\beta = 1 \times 10^{-3}$ for DBSR-LS. For DBSR-LS, the regularization coefficients were set to $\lambda_B = 1 \times 10^{-2}$ and $\lambda_S = 1 \times 10^{-2}$.

Models were trained with a batch size of 16 using the Adam optimizer with a learning rate of 1×10^{-4}. The correlation threshold for alignment was set to $\rho_{\min} = 0.5$. LT was performed over the fixed range $[-15, 15]$, and each experiment was repeated $R = 10$ times with identical configurations to ensure stability and enable consistent latent alignment.

A.7 Vanilla VAE on FA15

For the vanilla VAE on FA15 experiments, we used a fully connected VAE architecture with a symmetric encoder-decoder design, trained using the loss in Eq.(1) with $\gamma = 0$ and $\beta = 1$, indicating a vanilla VAE objective function. The encoder consists of three hidden layers with dimensions 256, 128, and 64, each followed by a ReLU activation and dropout with a rate of 0.2. The encoder outputs are passed through two parallel linear layers to produce the mean and log-variance for a 5-dimensional latent space. The decoder mirrors the encoder, with hidden layers of size 64, 128, and 256, followed by ReLU and dropout activations, and concludes with a linear output layer mapping to the input dimension. Samples from the latent space were obtained using the reparameterization trick.

We used the regularization coefficients were set to $\lambda_B = 1 \times 10^{-4}$ and $\lambda_S = 1 \times 10^{-4}$ for DBSR-LS method.

Models were trained with a batch size of 16 using the Adam optimizer with a learning rate of 1×10^{-4}. The correlation threshold for alignment was set to $\rho_{\min} = 0.5$. LT was performed over the fixed range $[-15, 15]$,

and each experiment was repeated $R = 10$ times with identical configurations to ensure stability and enable consistent latent alignment.

A.8 DIP-VAE-I on FA15

For the DIP-VAE-I on FA15 experiments, we used a fully connected VAE architecture with a symmetric encoder-decoder design, trained using the loss function

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\omega}) = \frac{1}{n} \sum_{i=1}^{n} \left[-\mathbb{E}_{q(\mathbf{z}_i|\mathbf{x}_i,\boldsymbol{\phi})} \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\omega}) + \mathrm{KL}(q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}) \,\|\, p(\mathbf{z}_i)) \right] + \lambda_{\mathrm{od}} \sum_{j \neq j' = 1}^{K} \left[\mathrm{Cov}_{p(\mathbf{x})}[\boldsymbol{\mu}_\phi(\mathbf{x})] \right]_{jj'}^2 + \lambda_d \sum_{j=1}^{K} \left(\left[\mathrm{Cov}_{p(\mathbf{x})}[\boldsymbol{\mu}_\phi(\mathbf{x})] \right]_{jj} - 1 \right)^2.$$

The encoder consists of three hidden layers with dimensions 256, 128, and 64, each followed by a ReLU activation and dropout with a rate of 0.2. The encoder outputs are passed through two parallel linear layers to produce the mean and log-variance for a 5-dimensional latent space. The decoder mirrors the encoder, with hidden layers of size 64, 128, and 256, followed by ReLU and dropout activations, and concludes with a linear output layer mapping to the input dimension. Samples from the latent space were obtained using the reparameterization trick.

We used the $\lambda_{od} = 1$ and $\lambda_D = 1$ for both DBSR-LS and FVH-LT methods. The regularization coefficients were set to $\lambda_B = 1 \times 10^{-3}$ and $\lambda_S = 1 \times 10^{-3}$ for DBSR-LS method.

Models were trained with a batch size of 16 using the SGD optimizer with a learning rate of 1×10^{-4}. The correlation threshold for alignment was set to $\rho_{\min} = 0.5$. LT was performed over the fixed range $[-15, 15]$, and each experiment was repeated $R = 10$ times with identical configurations to ensure stability and enable consistent latent alignment.

A.9 DIP-VAE-II on FA15

For the DIP-VAE-II on FA15 experiments, we used a fully connected VAE architecture with a symmetric encoder-decoder design, trained using the loss function

$$\mathcal{L}(\boldsymbol{\phi}, \boldsymbol{\omega}) = \frac{1}{n} \sum_{i=1}^{n} \left[-\mathbb{E}_{q(\mathbf{z}_i|\mathbf{x}_i,\boldsymbol{\phi})} \log p(\mathbf{x}_i|\mathbf{z}_i, \boldsymbol{\omega}) + \mathrm{KL}(q(\mathbf{z}_i|\mathbf{x}_i, \boldsymbol{\phi}) \,\|\, p(\mathbf{z}_i)) \right] + \lambda_{\mathrm{od}} \sum_{j \neq j' = 1}^{K} \left[\mathrm{Cov}_{q_\phi(\mathbf{z})}[\mathbf{z}] \right]_{jj'}^2 + \lambda_d \sum_{j=1}^{K} \left(\left[\mathrm{Cov}_{q_\phi(\mathbf{z})}[\mathbf{z}] \right]_{jj} - 1 \right)^2.$$

The encoder consists of three hidden layers with dimensions 256, 128, and 64, each followed by a ReLU activation and dropout with a rate of 0.2. The encoder outputs are passed through two parallel linear layers to produce the mean and log-variance for a 5-dimensional latent space. The decoder mirrors the encoder, with hidden layers of size 64, 128, and 256, followed by ReLU and dropout activations, and concludes with a linear output layer mapping to the input dimension. Samples from the latent space were obtained using the reparameterization trick.

We used the $\lambda_{od} = 1$ and $\lambda_D = 1$ for both DBSR and FVHLT methods. The regularization coefficients were set to $\lambda_B = 5 \times 10^{-5}$ and $\lambda_S = 5 \times 10^{-5}$ for DBSR-LS method.

Models were trained with a batch size of 16 using the SGD optimizer with a learning rate of 1×10^{-4}. The correlation threshold for alignment was set to $\rho_{\min} = 0.5$. LT was performed over the fixed range $[-15, 15]$, and each experiment was repeated $R = 10$ times with identical configurations to ensure stability and enable consistent latent alignment.

B Computation

All experiments were conducted on a local machine equipped with an Apple M2 Max CPU and 96 GB of RAM. No GPU acceleration or cloud computing resources were utilized. The following three tables summarize the approximate training times for each dataset under the VAE model, FVH-LT, and DBSR-LS methods, respectively. Reported times correspond to a single run, averaged over R repetitions ($R = 10$ or $R = 5$, depending on the dataset). All experiments were implemented in Python 3.12.

Approximate Training Time for VAE (per run)

Dataset	n	**Epochs**	**Time (minutes)**
FA15	800	100	3
FA24	8,000	100	40
FA100	40,000	150	150
White wine	3918	75	30
2018 FIFA statistics	102	800	3
CVAE on white wine	3,918	75	30
MNIST	50,000	20	60
CelebA	35,000	300	1,500
factor-VAE on FA15	800	100	3
β-VAE on FA15	800	100	2
vanilla VAE on FA15	800	100	2
DIP-VAE-I on FA15	800	300	< 1
DIP-VAE-II on FA15	800	300	< 1

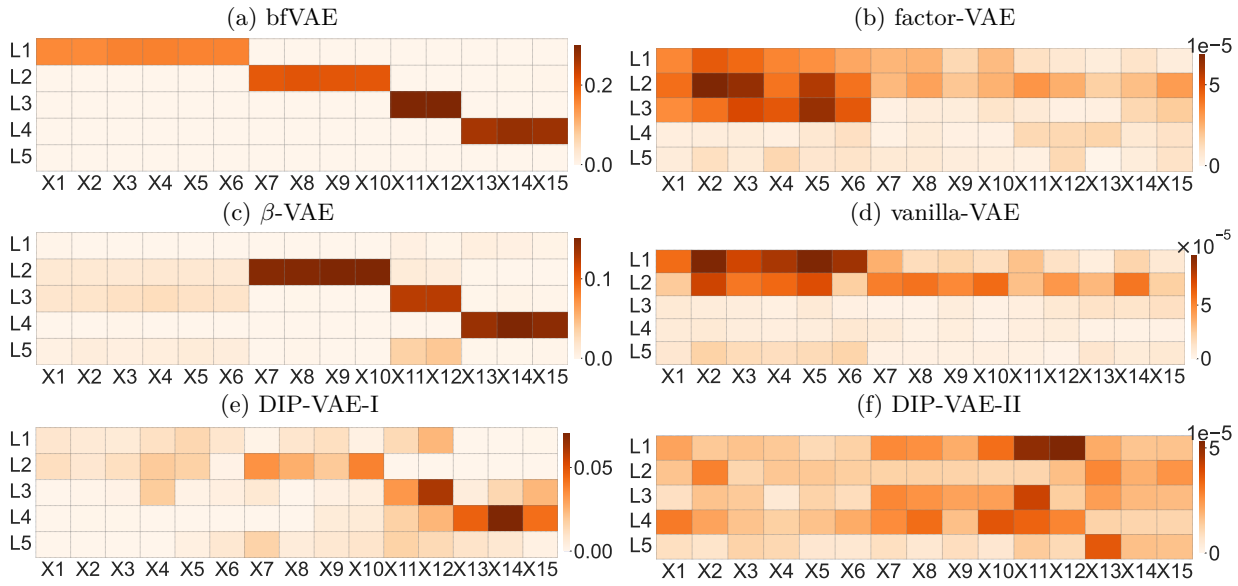
Approximate Computation Time for FVH-LT method (per run)

Dataset	n	**Time (minutes)**
FA15	800	2
FA24	8,000	30
FA100	40,000	400
White wine	3,918	15
2018 FIFA statistics	102	< 1
CVAE on white wine	3,918	16
MNIST	1,000	80
CelebA	1,000	1,200
factor-VAE on FA15	800	2
β-VAE on FA15	800	2
vanilla VAE on FA15	800	2
DIP-VAE-I on FA15	800	2
DIP-VAE-II on FA15	800	2

Approximate Computation Time for DBSR-LS method (per run)

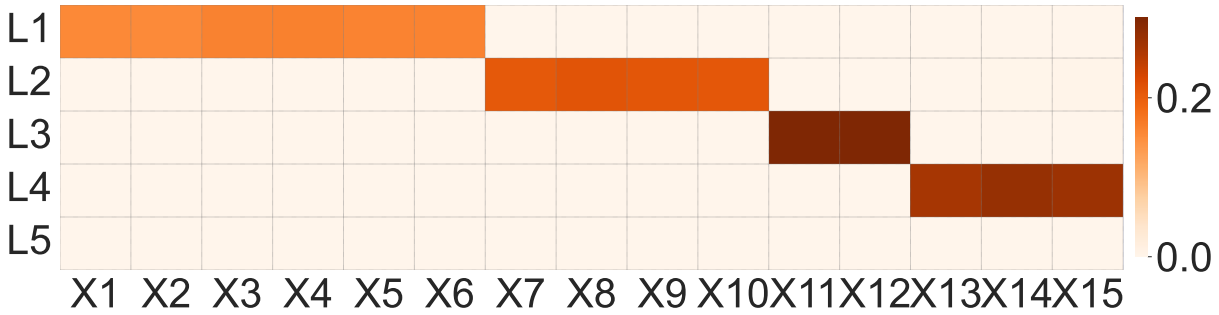
Dataset	n	**Time (minutes)**
FA15	800	< 1
FA24	8,000	2
FA100	40,000	20
White wine	3,918	1
2018 FIFA statistics	102	< 1
MNIST	1,000	10
factor-VAE on FA15	800	< 1
β-VAE on FA15	800	< 1
vanilla VAE on FA15	800	< 1
DIP-VAE-I on FA15	800	< 1
DIP-VAE-II on FA15	800	< 1

C DBSR-LS results on FA15 using different disentanglement frameworks

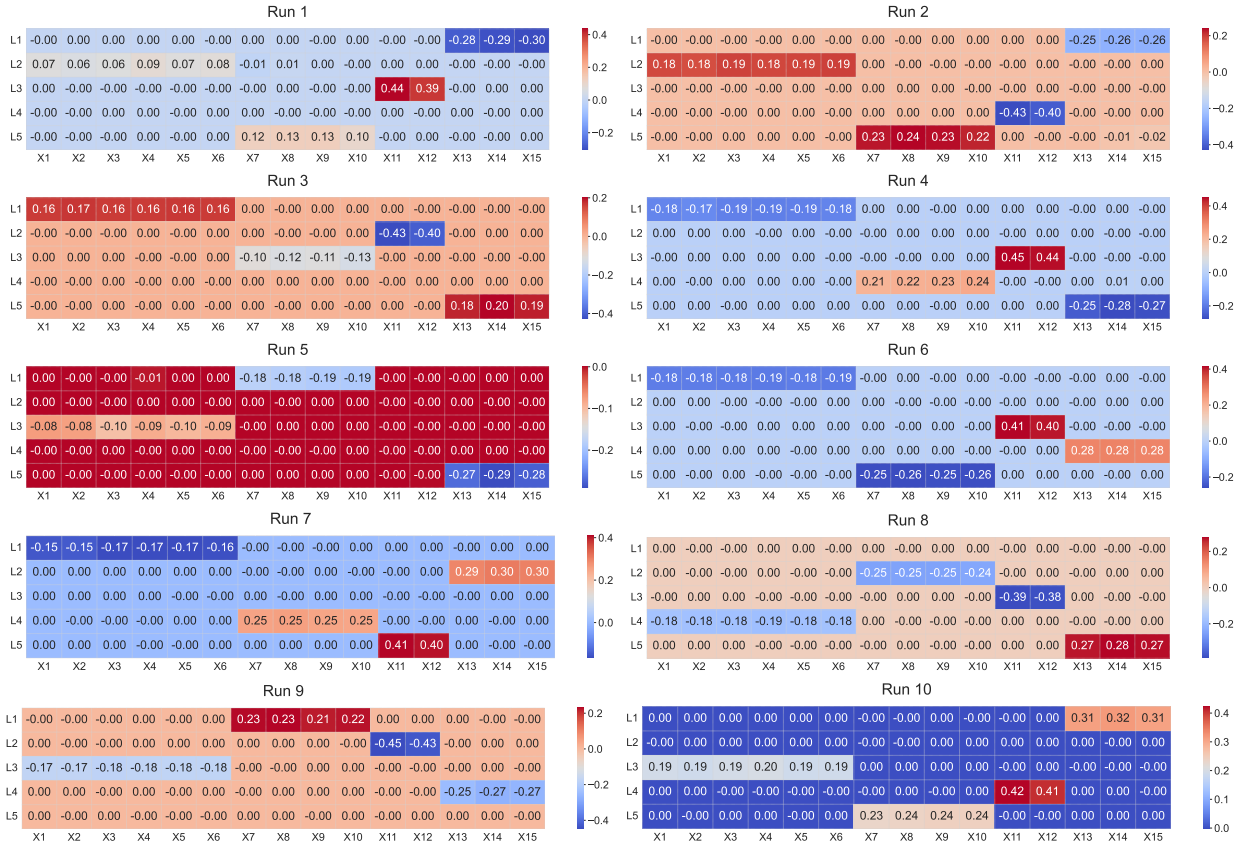


D FA15: DBSR-LS results

D.1 Results over 10 runs

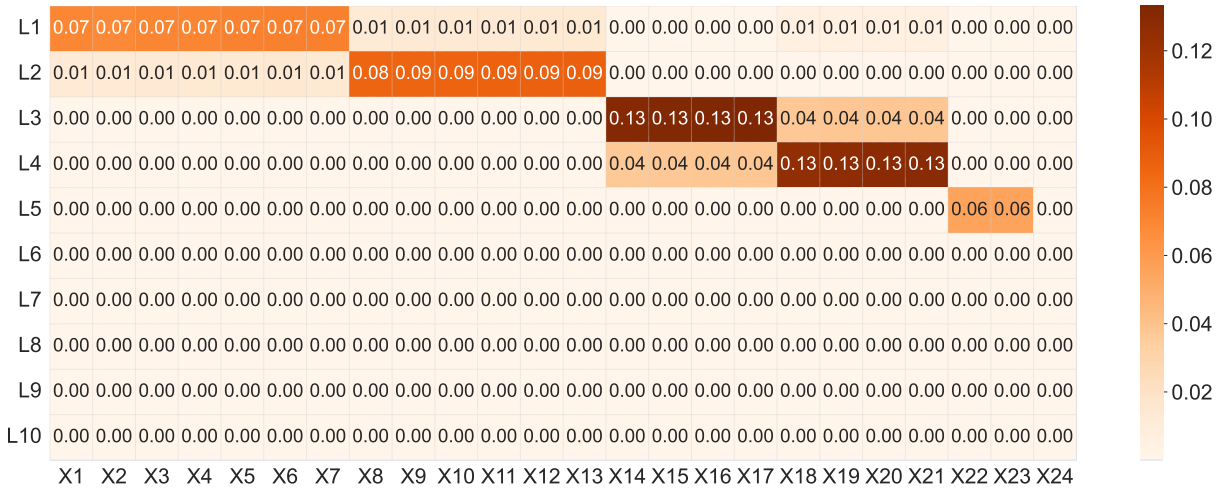


D.2 Single-run results

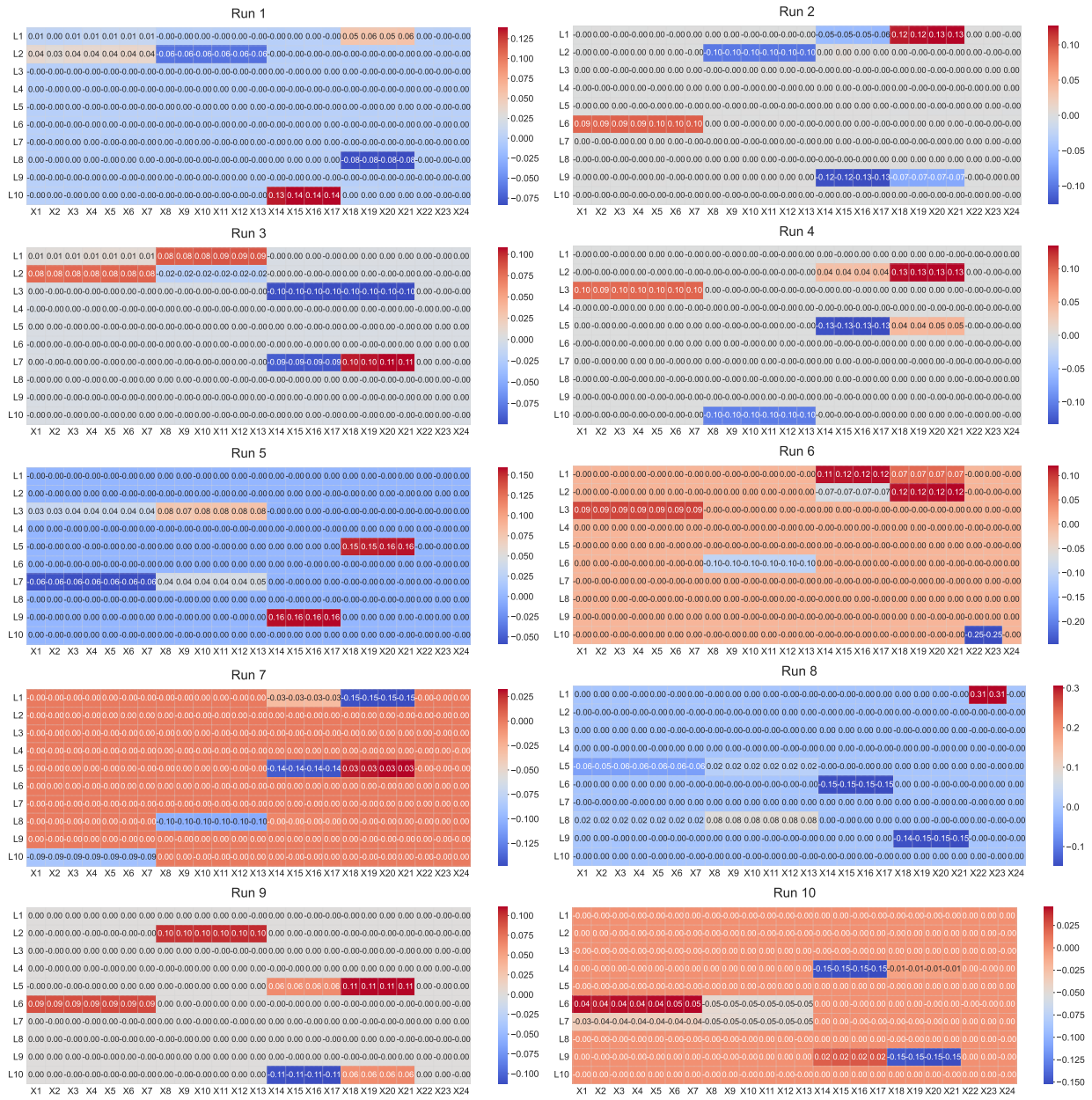


E FA24: DBSR-LS results

E.1 Result over 10 runs

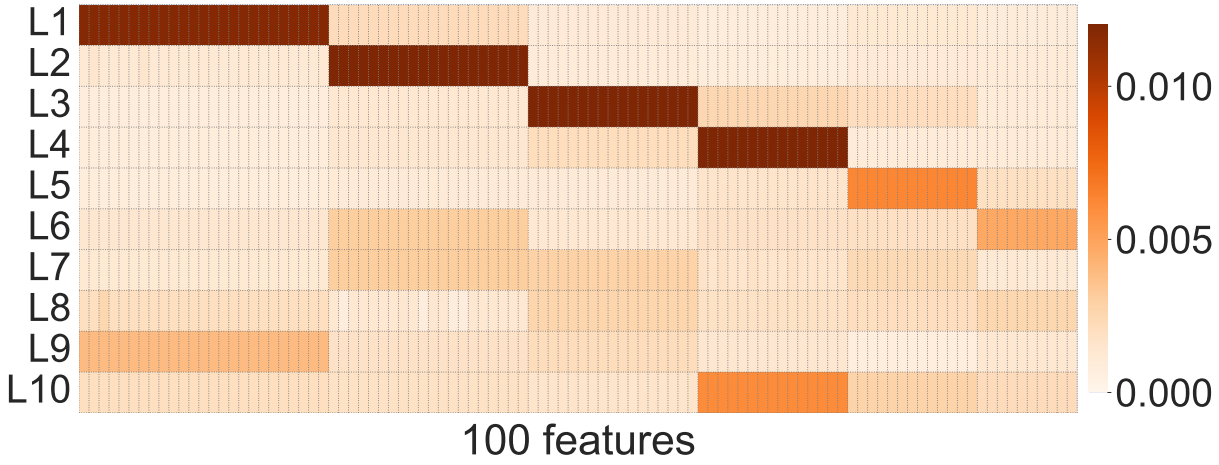


E.2 Single-run results

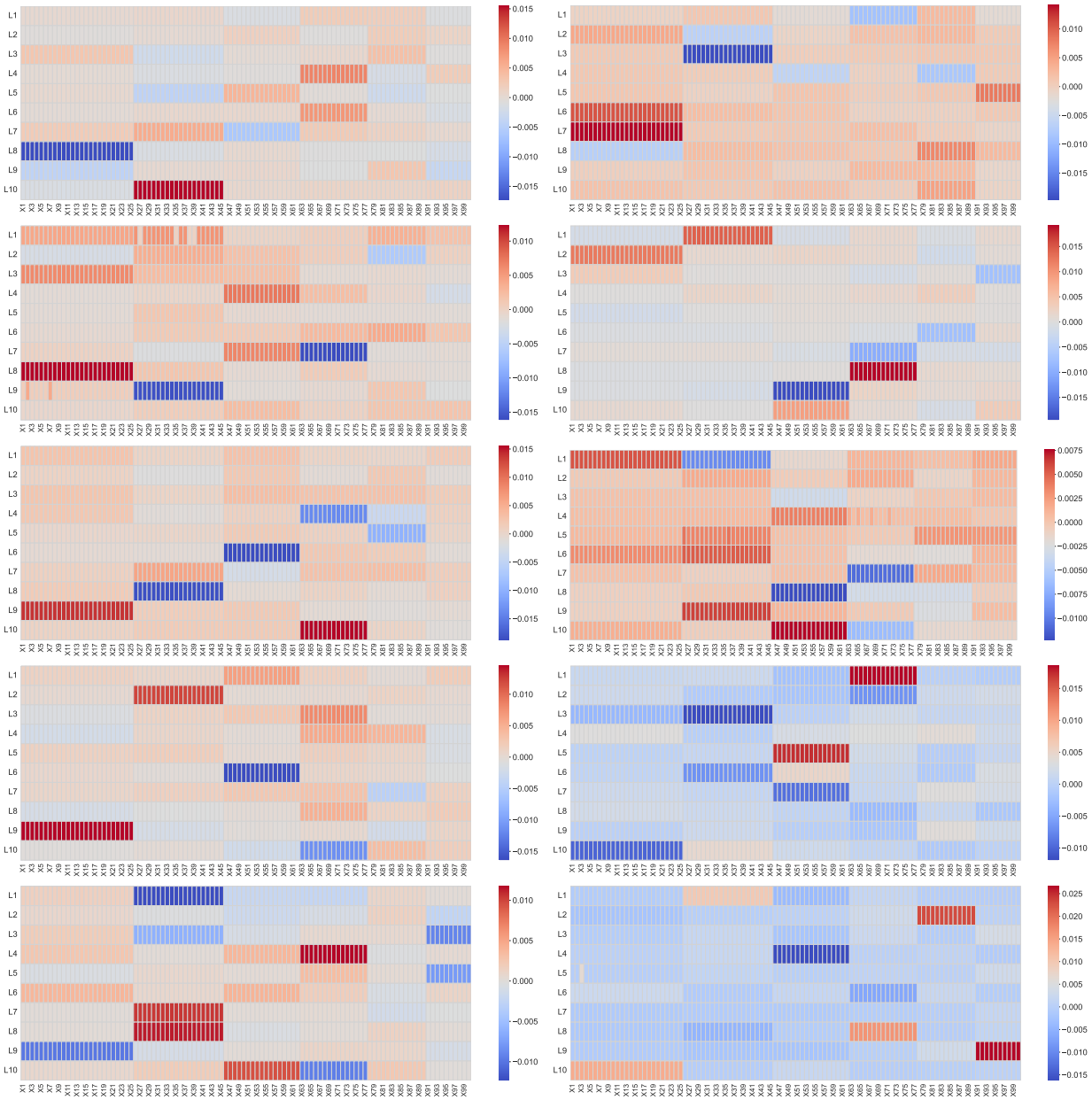

Run 1
Run 2
Run 3
Run 4
Run 5
Run 6
Run 7
Run 8
Run 9
Run 10
L1 L2 L3 L4 L5 L6 L7 L8 L9 L10
X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19 X20 X21 X22 X23 X24

F FA100: DBSR-LS results

F.1 Results over 10 runs

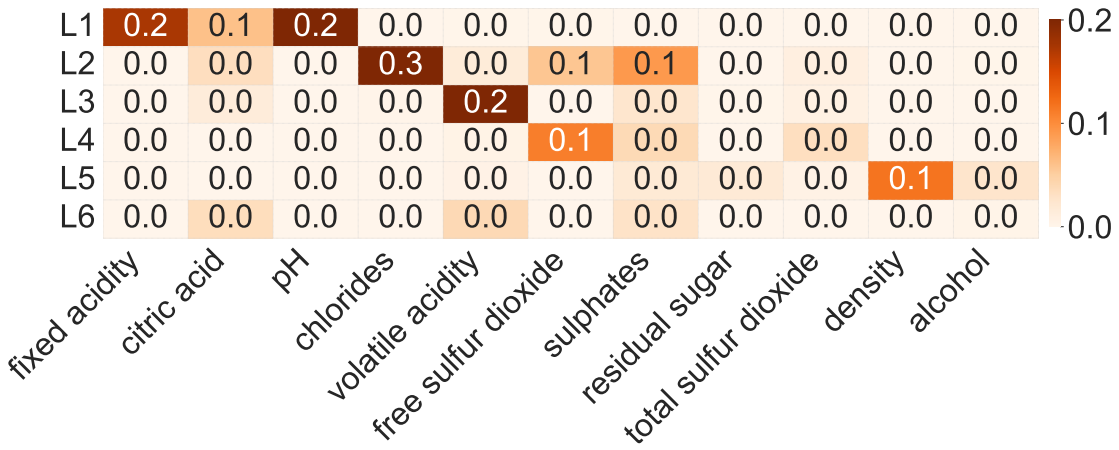


F.2 Single-run results

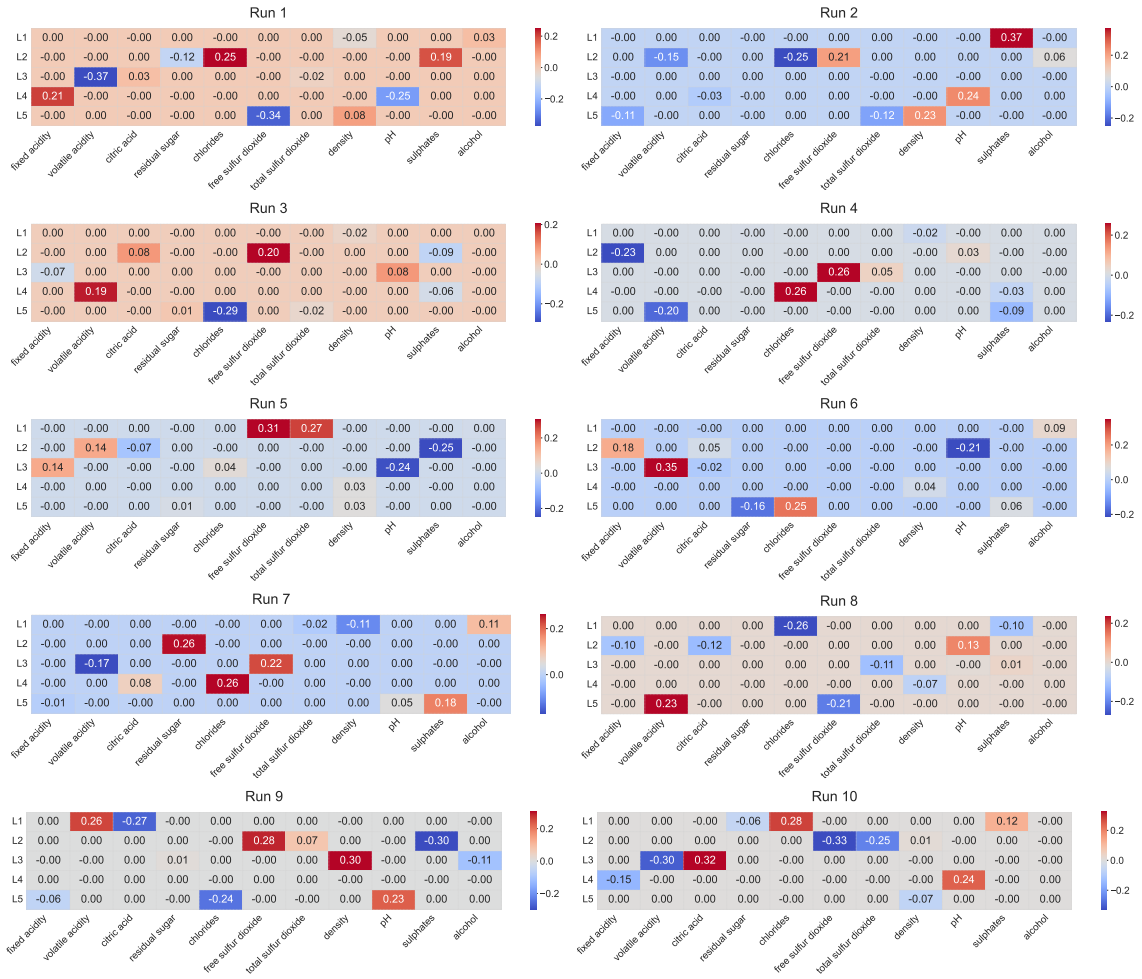


G White wine data: DBSR-LS results

G.1 Results over 10 runs

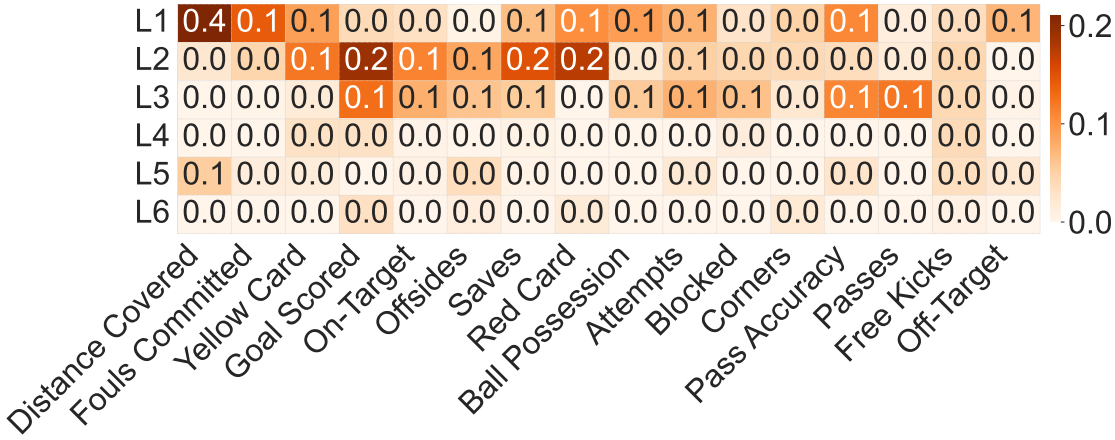


G.2 Single-run results

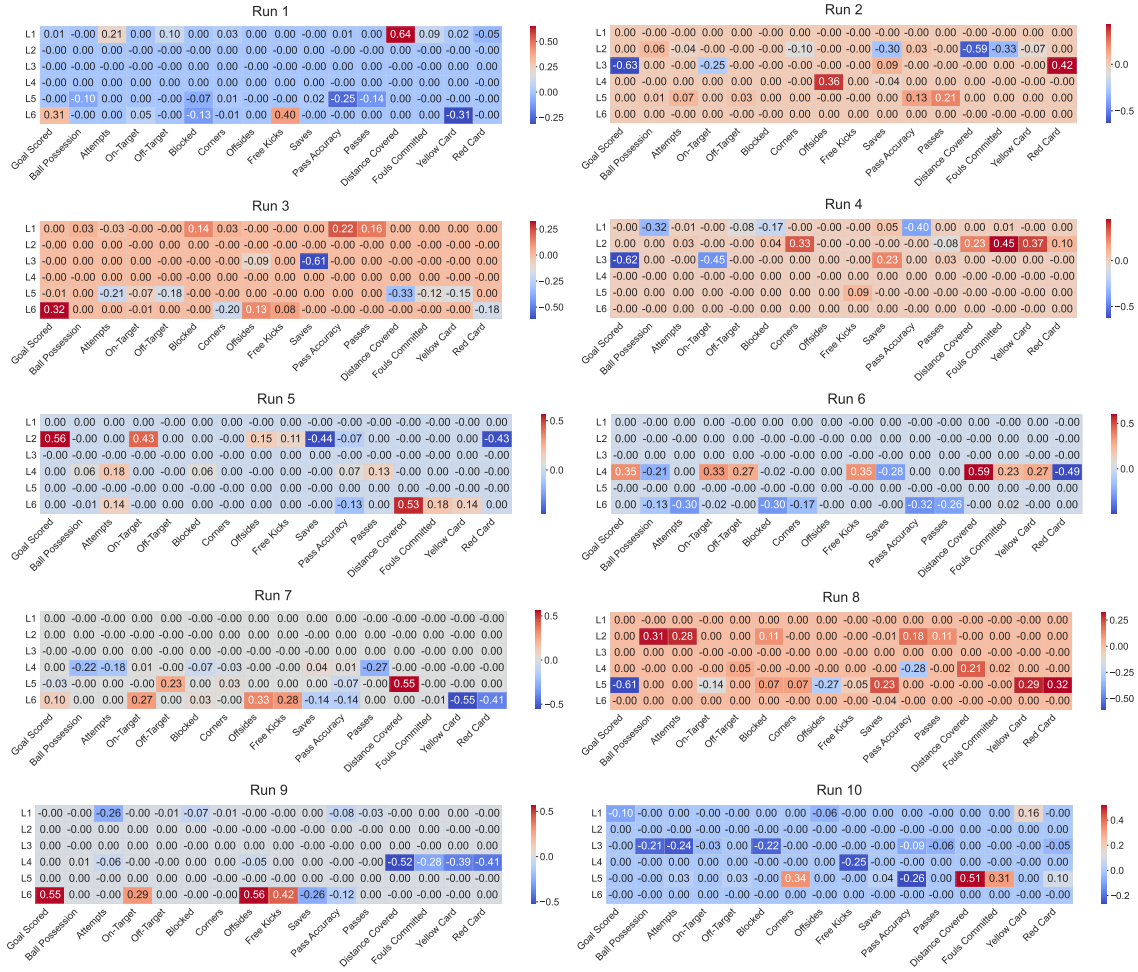


H 2018 FIFA statistics data: DBSR-LS results

H.1 Results over 10 runs

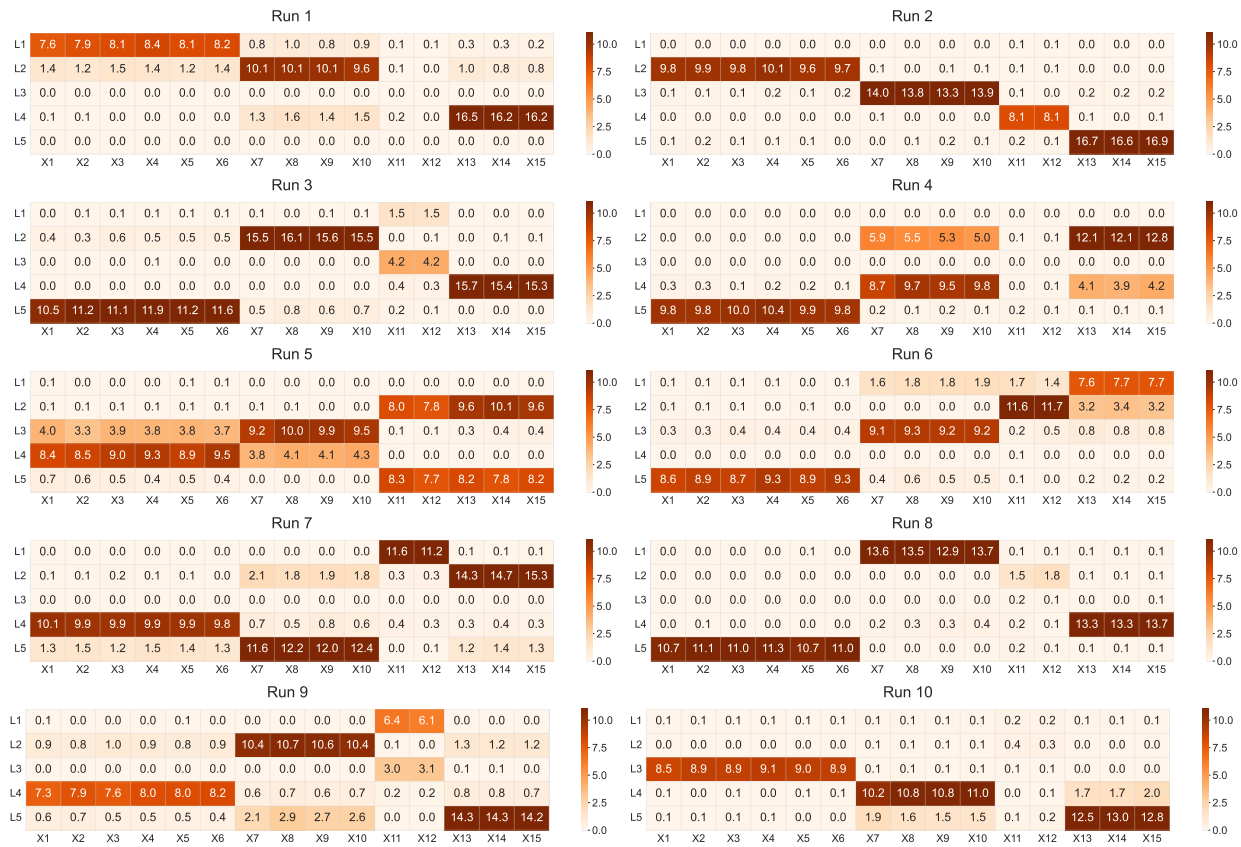


H.2 Single-run results

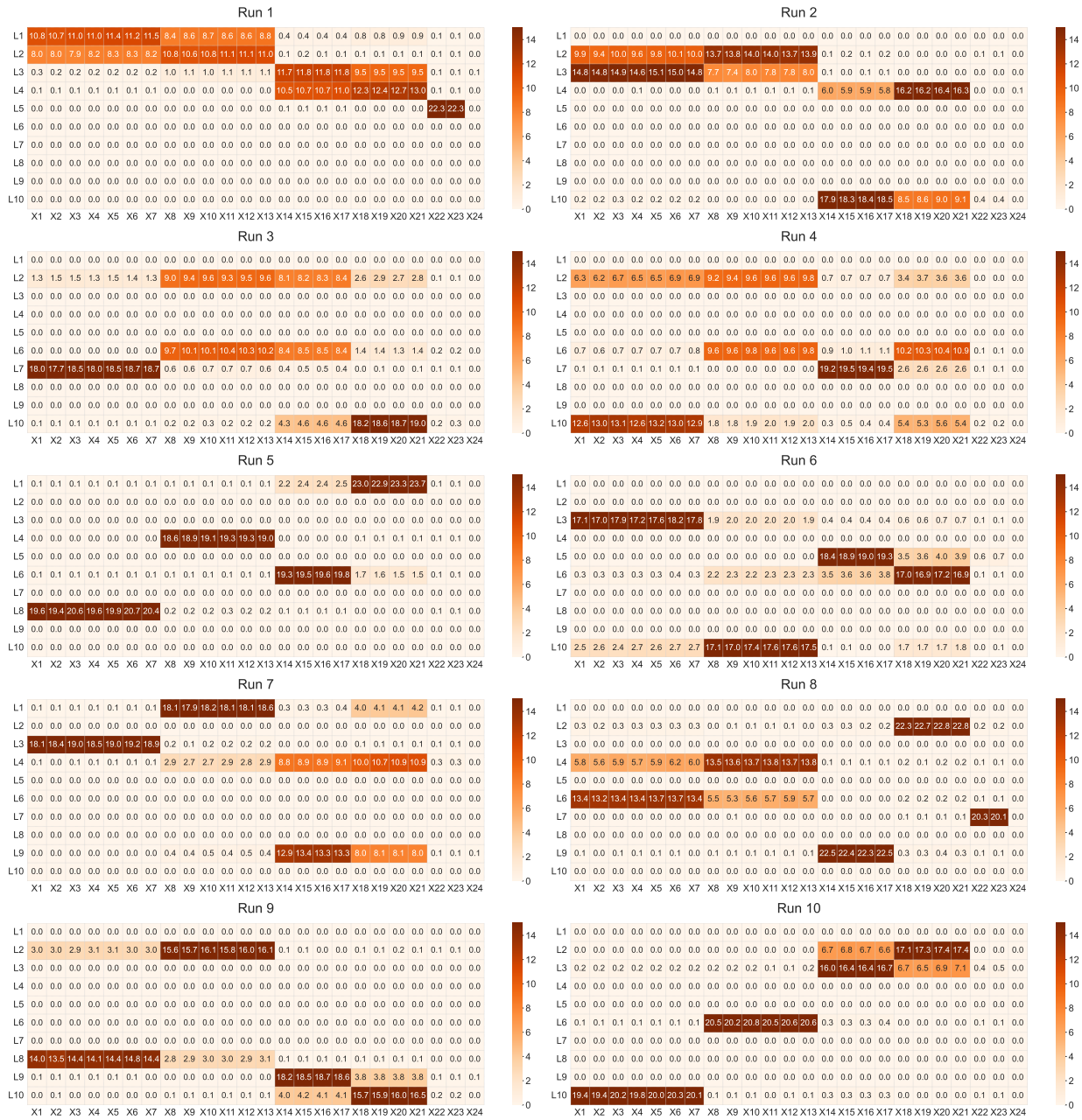


I Single-run results corresponding to aggregate findings presented in the main text

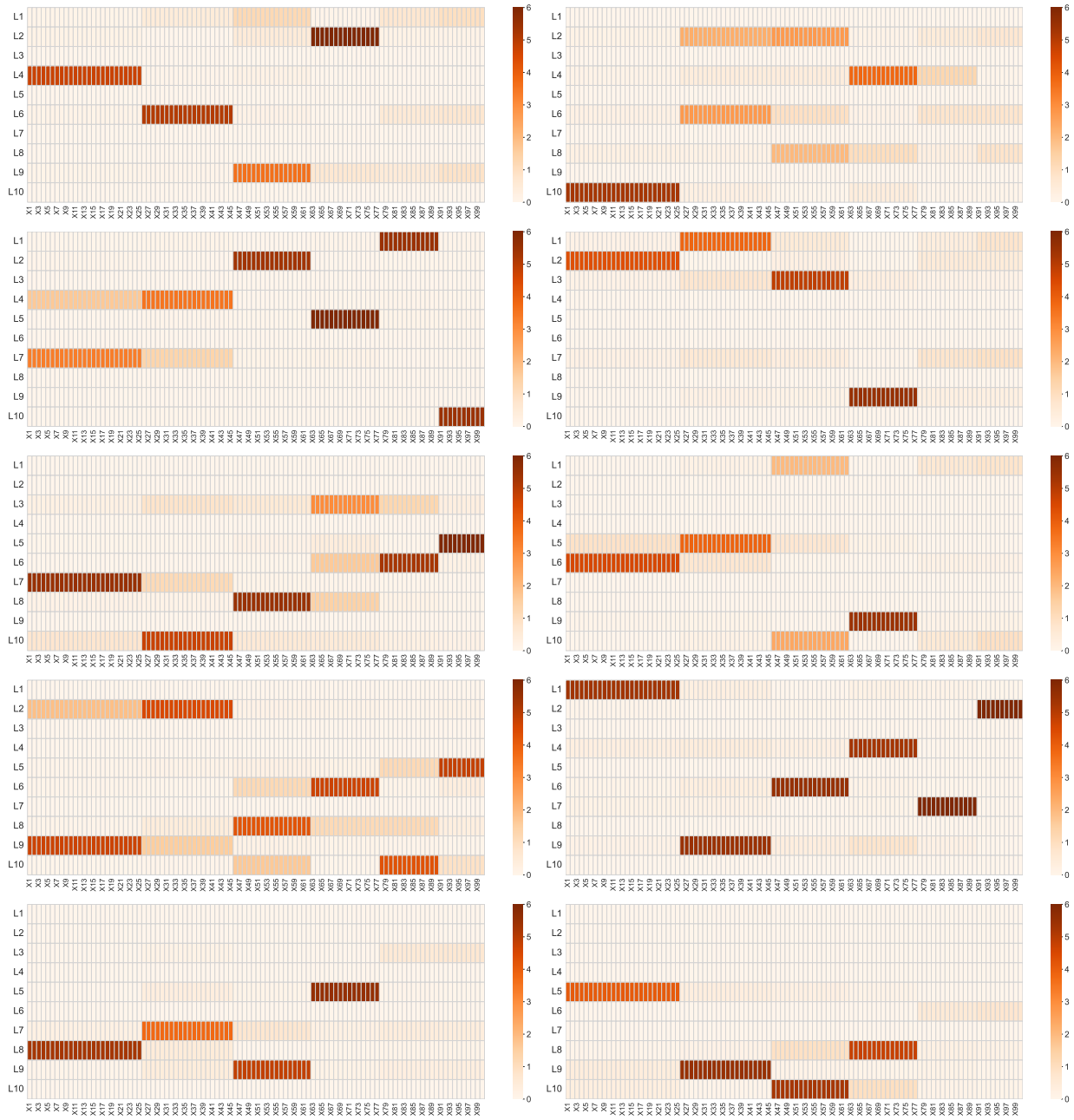
I.1 FA15: FVH-LT result with bfVAE



I.2 FA24: FVH-LT single-run results

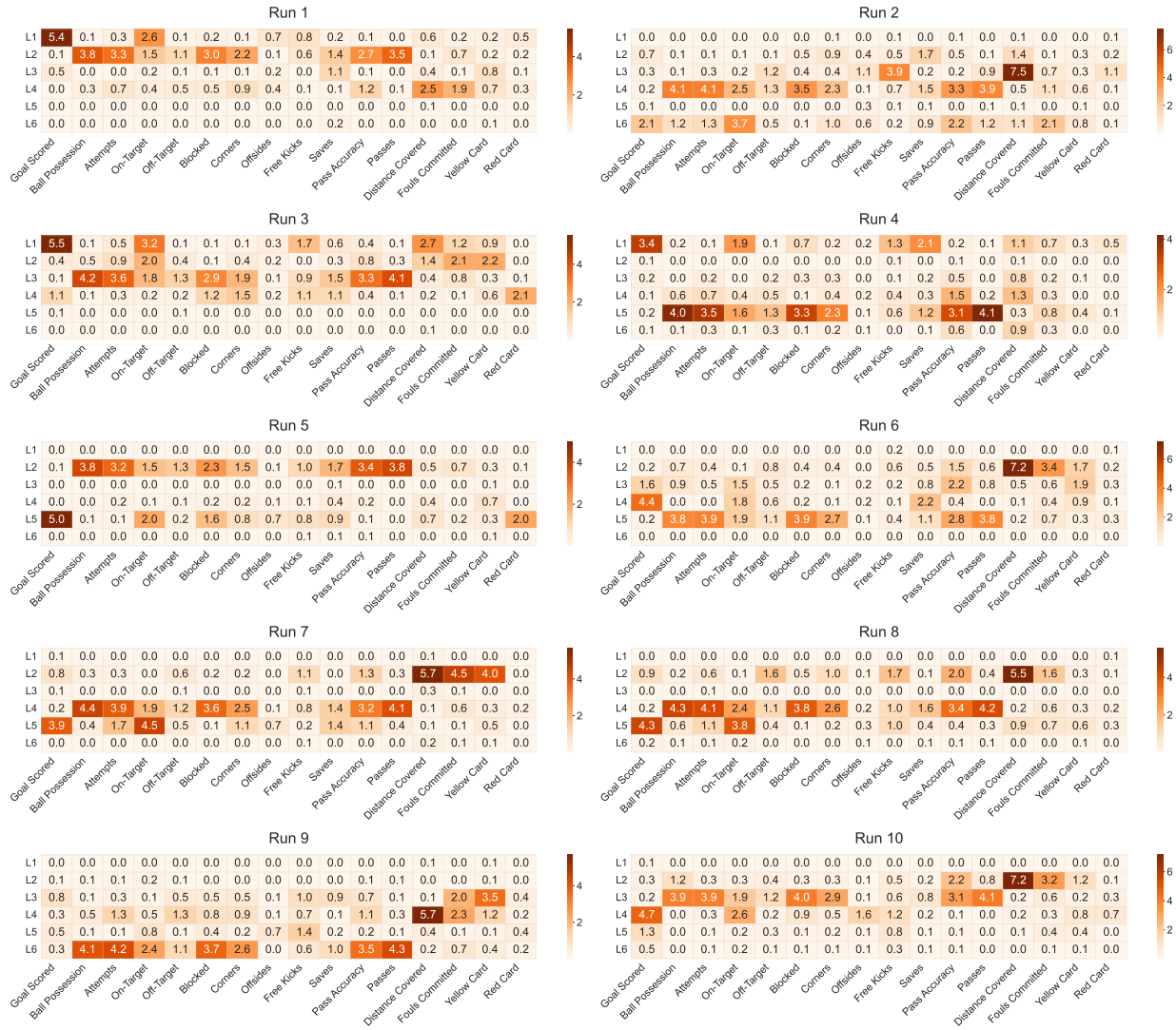


I.3 FA100: FVH-LT result

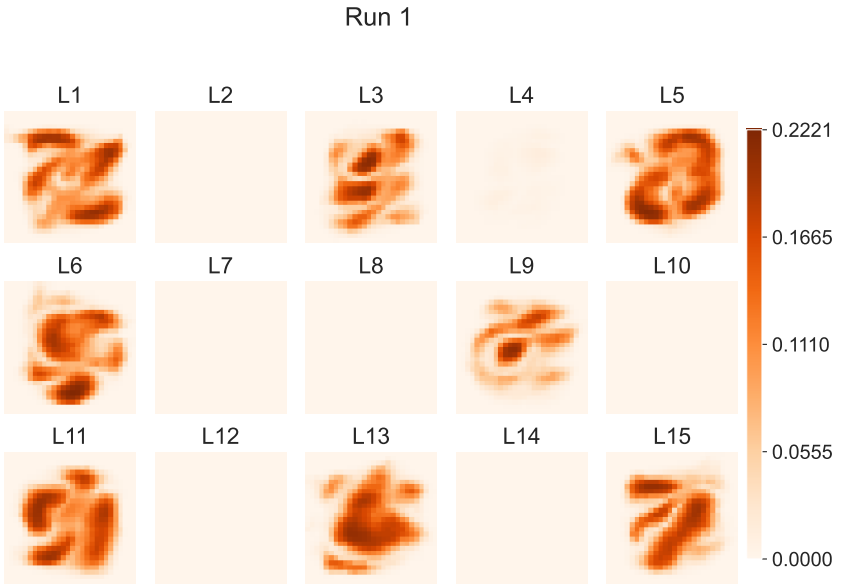


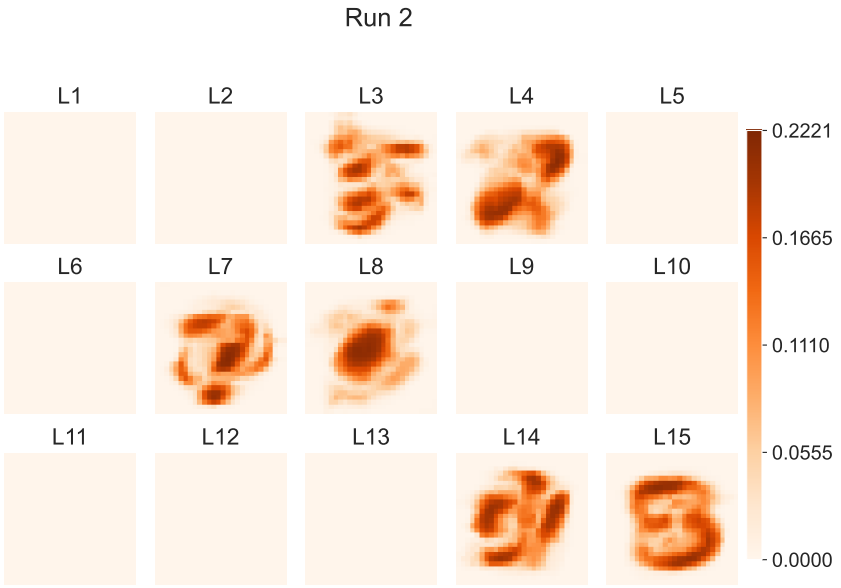
I.4 White wine data: FVH-LT result

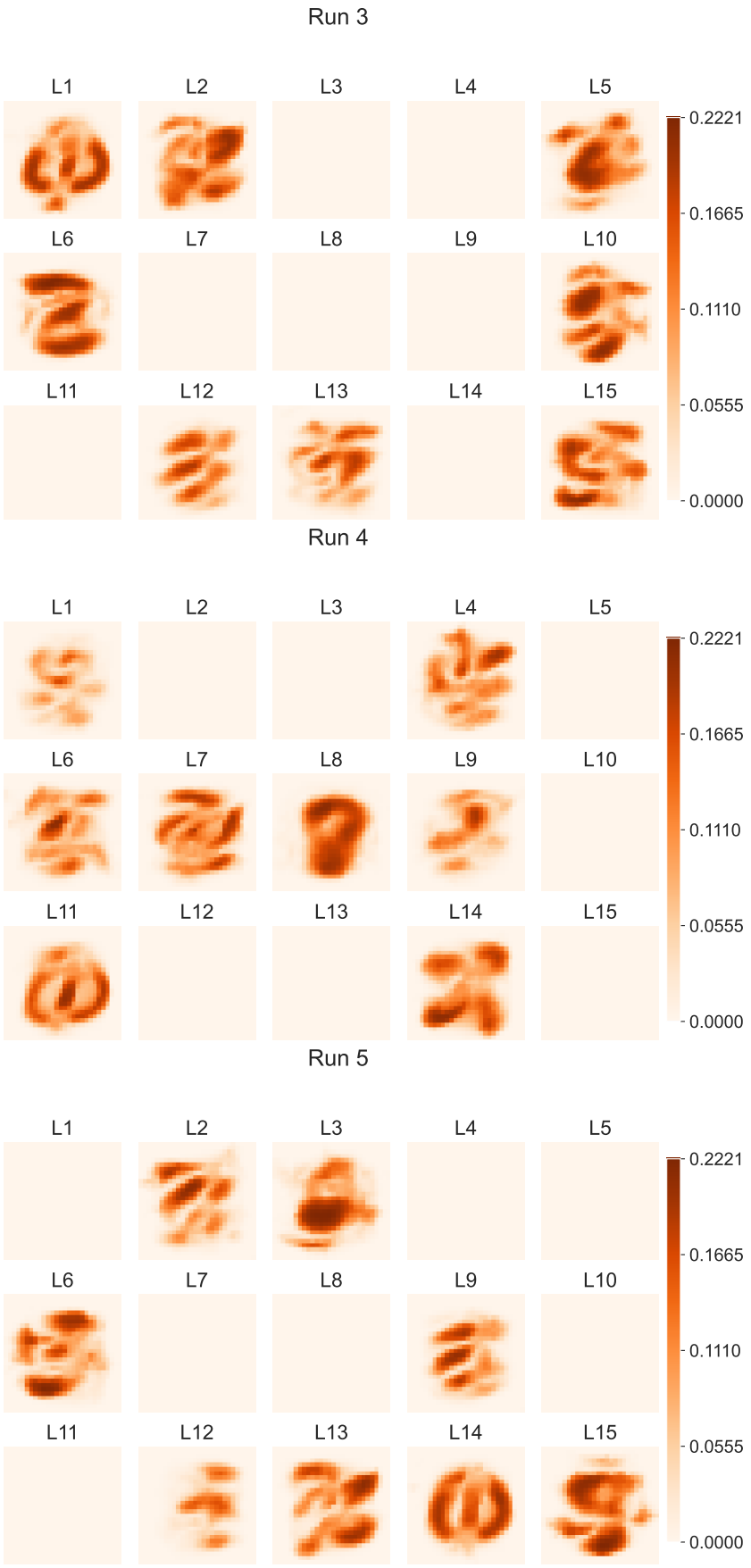
I.5 2018 FIFA statistics data: FVH-LT result

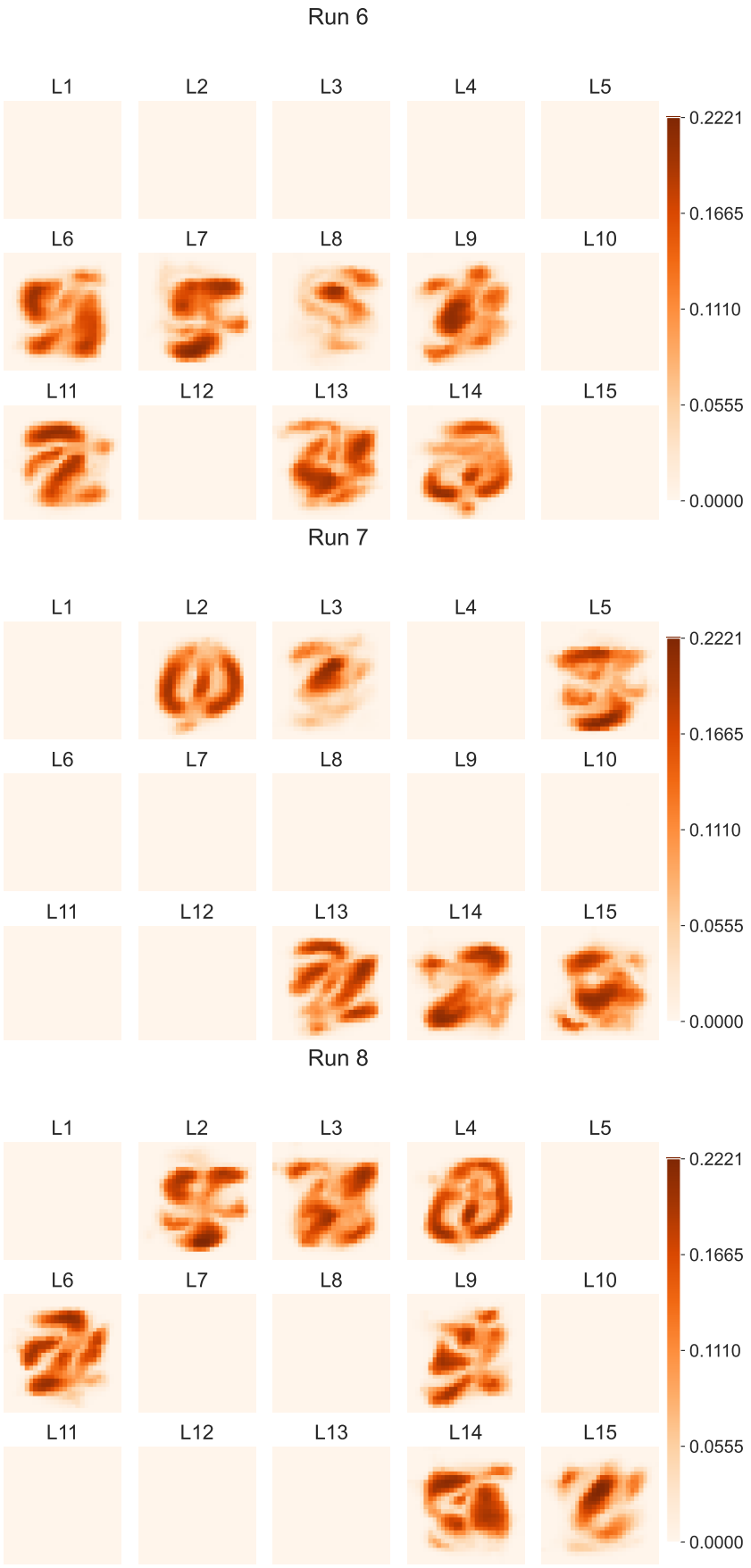


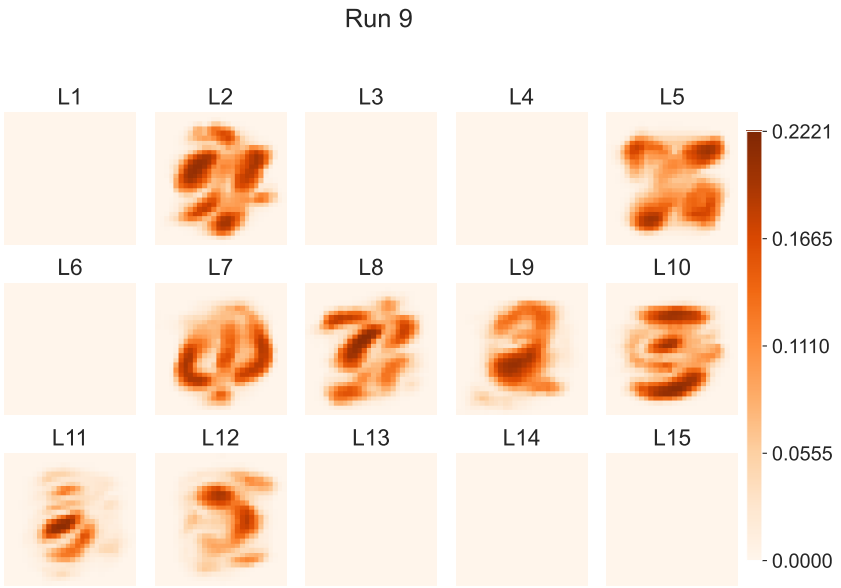
I.6 MNIST: FVH-LT result

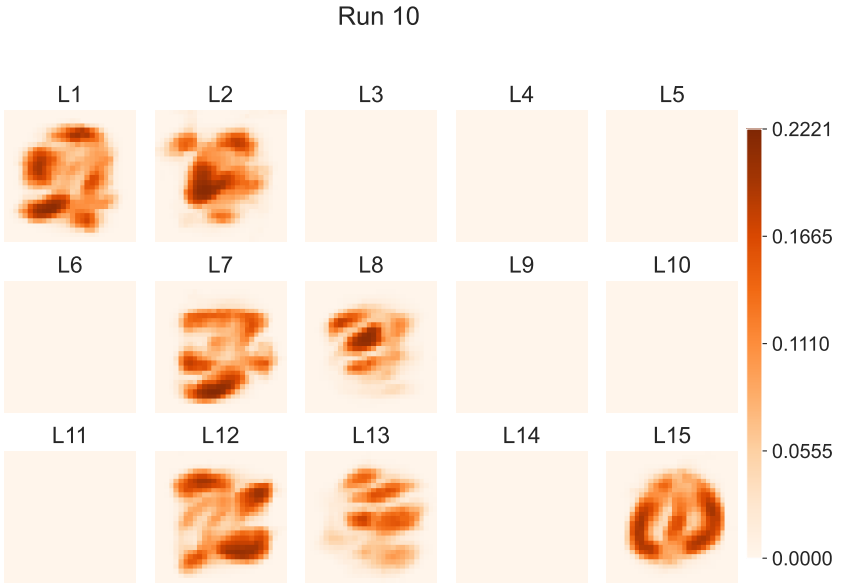





Run 3
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.2221
0.1665
0.1110
0.0555
0.0000
Run 4
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.2221
0.1665
0.1110
0.0555
0.0000
Run 5
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.2221
0.1665
0.1110
0.0555
0.0000


Run 6
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.2221
0.1665
0.1110
0.0555
0.0000
Run 7
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.2221
0.1665
0.1110
0.0555
0.0000
Run 8
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.2221
0.1665
0.1110
0.0555
0.0000


Run 9
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.2221
0.1665
0.1110
0.0555
0.0000


Run 10
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.2221
0.1665
0.1110
0.0555
0.0000

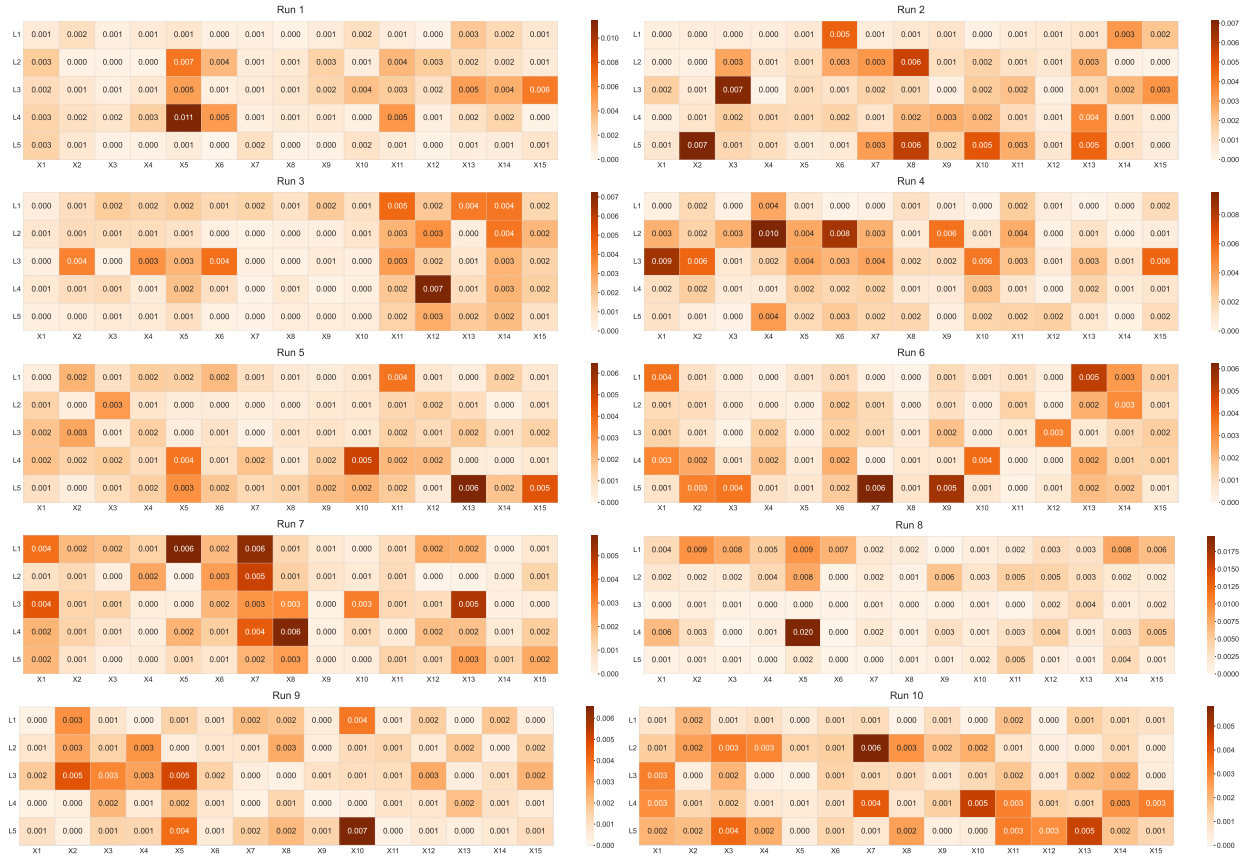
I.7 CelebA: FVH-LT result

Latent 1
Latent 2
Latent 3
Latent 4
Latent 5
Latent 6
Latent 7
Latent 8
Latent 9
Latent 10
Latent 11
Latent 12
Latent 13
Latent 14
Latent 15
Latent 16
Latent 17
Latent 18
Latent 19
Latent 20

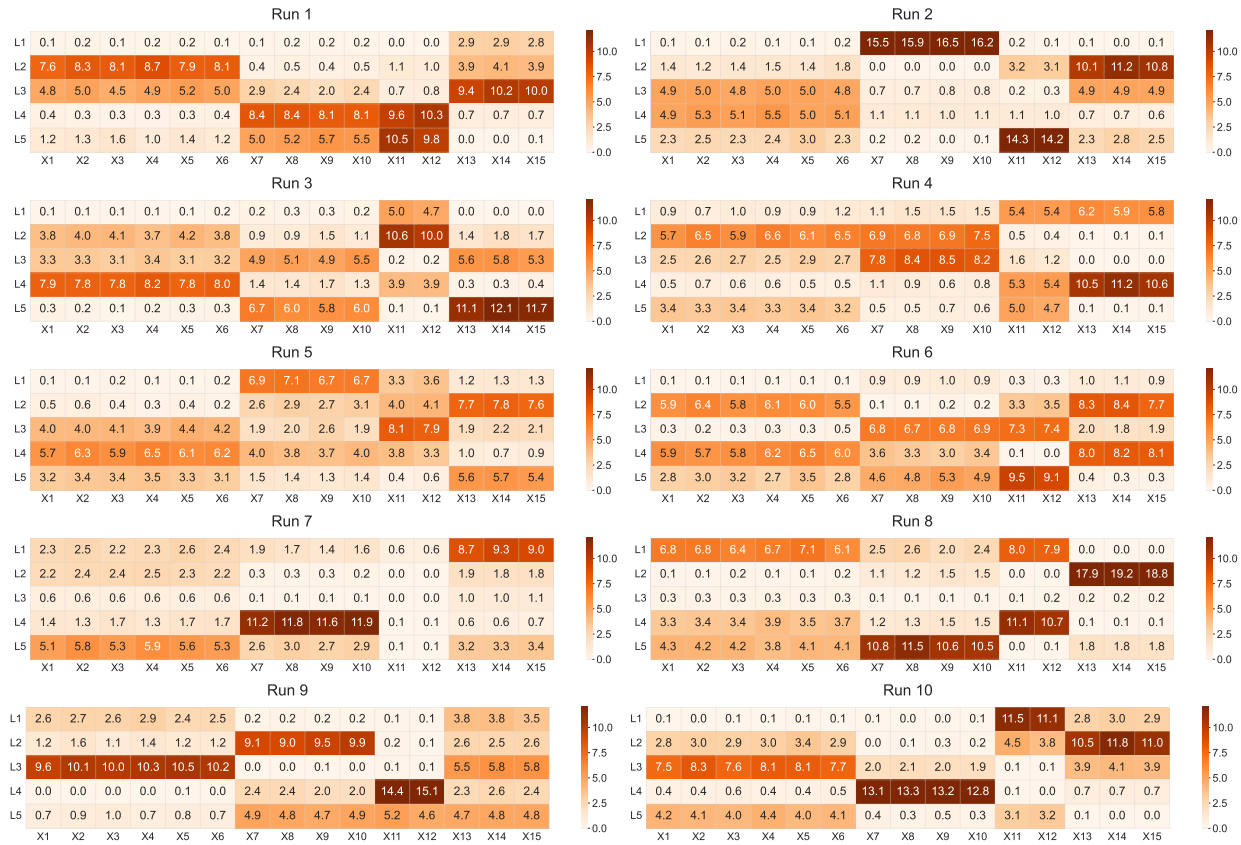
Latent 1
Latent 2
Latent 3
Latent 4
Latent 5
Latent 6
Latent 7
Latent 8
Latent 9
Latent 10
Latent 11
Latent 12
Latent 13
Latent 14
Latent 15
Latent 16
Latent 17
Latent 18
Latent 19
Latent 20

Latent 1
Latent 2
Latent 3
Latent 4
Latent 5
Latent 6
Latent 7
Latent 8
Latent 9
Latent 10
Latent 11
Latent 12
Latent 13
Latent 14
Latent 15
Latent 16
Latent 17
Latent 18
Latent 19
Latent 20

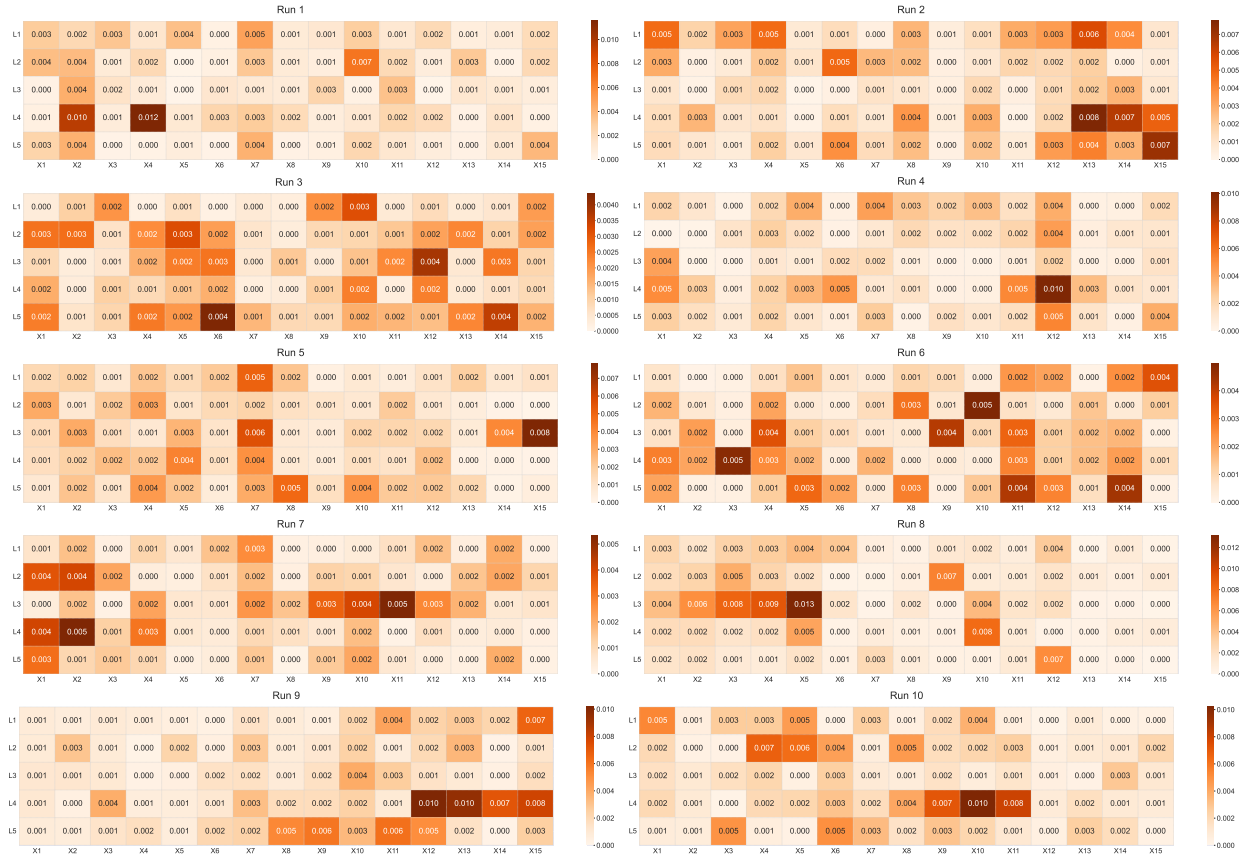
I.8 FA15: FVH-LT result with factor VAE



I.9 FA15: FVH-LT result with β-VAE



I.10 FA15: FVH-LT result with vanilla VAE



I.11 FA15: FVH-LT result with DIP-VAE-I

Run 1

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	0.2	0.2	0.1	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.9	1.1	0.8
L2	0.1	0.2	0.2	0.2	0.1	0.1	4.5	4.7	4.7	4.8	2.4	2.0	0.9	0.8	0.9
L3	0.0	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.1	0.0	1.6	1.4	6.0	6.4	6.2
L4	0.8	0.9	0.8	0.9	1.1	0.7	3.7	3.7	3.8	3.9	3.5	3.5	0.8	0.6	0.5
L5	8.9	9.3	9.5	9.5	9.6	9.3	0.1	0.1	0.2	0.1	0.3	0.3	0.3	0.3	0.4

Run 2

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	7.7	8.0	7.8	8.5	8.2	8.0	1.8	1.7	1.7	1.6	0.0	0.0	0.0	0.0	0.0
L2	0.5	0.5	0.5	0.6	0.5	0.5	1.0	1.0	0.9	1.0	2.7	2.7	0.1	0.1	0.1
L3	0.4	0.3	0.5	0.4	0.6	0.5	4.1	4.6	4.7	4.5	0.3	0.2	2.5	2.6	2.6
L4	0.2	0.2	0.2	0.2	0.1	0.1	1.8	1.9	1.8	1.8	0.3	0.3	4.8	5.2	5.0
L5	0.9	1.2	1.1	0.9	1.0	0.9	0.3	0.4	0.4	0.4	4.0	4.1	0.5	0.5	0.4

Run 3

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	1.4	1.5	1.6	1.7	1.9	1.9	1.9	2.3	2.4	2.3	0.8	0.9	2.6	3.0	2.7
L2	1.8	1.9	2.0	2.0	2.0	1.8	0.9	0.9	0.8	0.9	2.8	2.9	2.1	2.4	2.3
L3	0.3	0.4	0.2	0.4	0.4	0.5	6.1	6.2	6.0	6.4	1.7	1.5	0.1	0.2	0.2
L4	5.0	5.5	5.3	5.2	5.1	5.2	0.3	0.3	0.3	0.3	0.0	0.0	3.4	3.1	3.2
L5	0.3	0.2	0.3	0.3	0.2	0.3	0.3	0.4	0.4	0.3	1.4	1.4	0.1	0.1	0.1

Run 4

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	5.2	5.8	5.7	5.7	5.6	6.0	4.2	4.3	5.1	4.5	0.1	0.0	0.3	0.2	0.2
L2	2.2	2.2	1.8	2.1	2.0	2.2	3.3	3.7	3.4	3.8	2.8	2.4	0.3	0.3	0.3
L3	0.6	0.7	0.7	0.6	0.6	0.6	0.9	0.9	1.0	1.0	3.2	3.3	0.2	0.2	0.2
L4	0.3	0.5	0.3	0.4	0.4	0.4	1.8	1.7	1.9	1.8	2.6	2.4	0.4	0.4	0.4
L5	0.6	0.6	0.5	0.6	0.8	0.6	0.2	0.2	0.1	0.1	0.0	0.0	6.4	6.6	6.5

Run 5

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	0.4	0.4	0.3	0.3	0.4	0.3	6.4	6.6	6.6	6.5	0.8	1.0	0.0	0.1	0.0
L2	2.1	2.3	2.4	2.3	2.1	2.0	0.0	0.0	0.0	0.0	0.0	0.0	5.8	6.2	6.2
L3	6.0	6.4	6.2	6.3	6.7	6.4	0.2	0.2	0.2	0.2	0.0	0.0	2.3	2.6	2.5
L4	0.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1.0	1.0	0.7	0.8	0.1	0.1	0.0
L5	0.1	0.0	0.0	0.1	0.0	0.1	2.0	1.9	2.0	1.9	5.2	5.1	0.1	0.1	0.1

Run 6

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	0.4	0.4	0.4	0.4	0.4	0.4	2.9	3.0	3.2	3.6	4.1	4.0	0.9	0.9	0.9
L2	1.9	2.1	2.2	2.2	2.0	2.1	5.0	5.0	5.0	4.9	1.3	1.4	0.0	0.1	0.0
L3	5.8	6.3	6.1	6.3	6.4	6.1	0.6	0.9	0.6	0.8	2.7	2.6	0.0	0.0	0.0
L4	0.1	0.1	0.1	0.1	0.2	0.2	0.0	0.0	0.0	0.0	1.0	1.0	5.6	5.3	5.3
L5	0.4	0.5	0.5	0.5	0.5	0.5	0.7	0.5	0.5	0.5	0.2	0.2	2.3	2.7	2.5

Run 7

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	0.1	0.1	0.0	0.1	0.0	0.0	4.8	4.8	5.4	5.0	0.5	0.4	2.3	2.7	2.4
L2	4.3	4.3	4.2	4.0	4.6	4.5	0.6	0.8	0.4	0.7	2.3	2.3	0.8	1.0	0.9
L3	3.6	3.4	3.6	3.6	3.0	3.6	2.9	3.0	2.8	2.8	0.8	0.8	2.1	2.1	2.1
L4	0.4	0.5	0.4	0.4	0.5	0.4	0.0	0.0	0.0	0.0	1.8	1.7	0.8	0.8	0.8
L5	1.6	1.8	1.5	1.7	1.7	1.6	0.6	0.6	0.5	0.6	1.5	1.3	2.3	2.5	2.3

Run 8

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	1.7	1.4	1.6	1.5	1.7	1.6	0.1	0.1	0.1	0.1	2.1	1.7	0.1	0.1	0.2
L2	0.1	0.3	0.2	0.2	0.2	0.2	1.1	1.0	1.1	1.0	3.3	3.1	1.2	1.4	1.3
L3	8.1	8.3	8.6	8.4	8.2	8.5	0.1	0.2	0.2	0.1	0.4	0.4	0.2	0.1	0.1
L4	0.3	0.3	0.2	0.3	0.3	0.3	6.0	6.4	5.4	6.2	0.0	0.0	2.3	2.7	2.5
L5	0.0	0.0	0.0	0.0	0.0	0.0	2.1	2.0	2.2	2.1	1.3	1.3	4.6	4.6	4.2

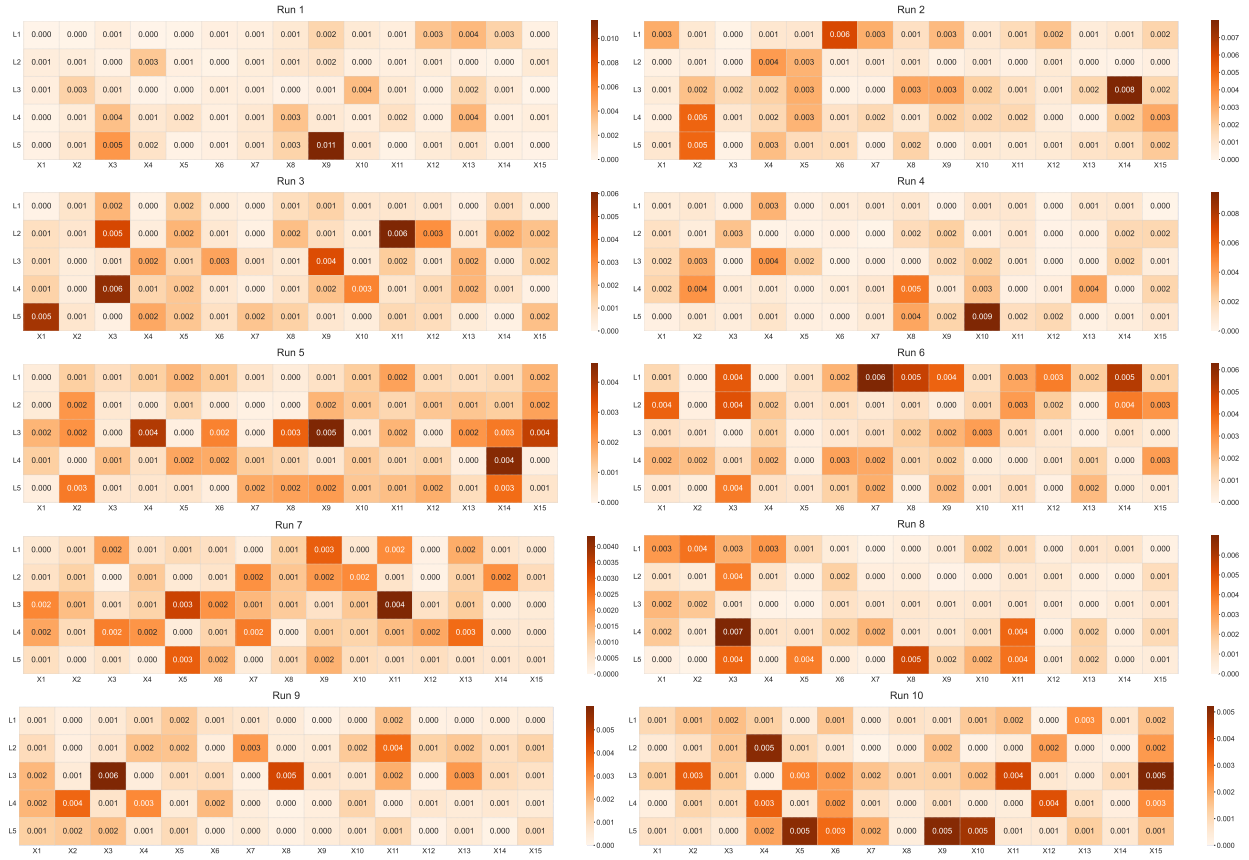
Run 9

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	0.1	0.2	0.2	0.1	0.2	0.2	1.0	1.1	1.1	1.0	2.8	2.3	1.5	1.3	1.3
L2	6.6	6.8	7.2	7.0	7.5	7.2	1.8	1.9	2.3	2.1	0.2	0.1	0.6	0.8	0.6
L3	0.1	0.1	0.1	0.1	0.1	0.1	1.5	1.5	1.3	1.4	0.2	0.2	2.0	1.9	1.8
L4	0.4	0.7	0.6	0.6	0.7	0.5	5.2	4.9	4.7	4.8	3.1	3.2	0.0	0.0	0.0
L5	1.2	1.3	1.3	1.2	1.0	1.1	1.4	1.4	1.4	1.5	1.8	1.7	3.5	3.7	3.6

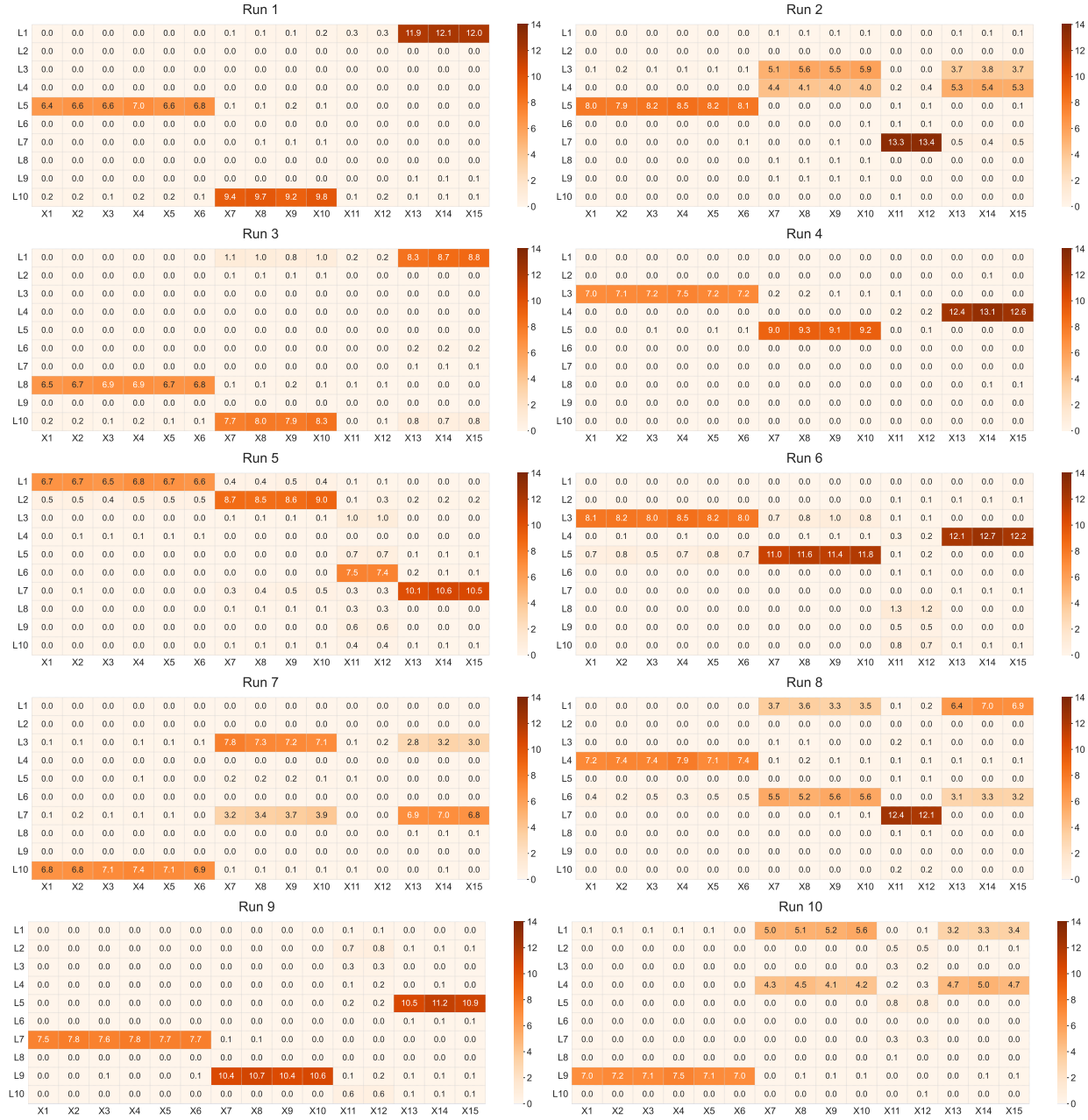
Run 10

	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	X15
L1	0.4	0.4	0.5	0.4	0.5	0.4	3.5	3.5	3.6	3.7	2.2	2.0	1.8	1.9	1.8
L2	2.7	2.7	3.0	2.9	2.8	3.2	1.1	1.3	1.3	1.2	2.4	2.5	0.3	0.3	0.3
L3	3.1	3.0	3.0	3.2	3.3	2.8	4.0	4.2	3.8	4.0	2.1	2.3	0.0	0.0	0.0
L4	0.0	0.0	0.0	0.0	0.0	0.0	0.4	0.4	0.6	0.6	1.3	1.2	5.7	6.3	5.9
L5	2.8	2.9	3.0	2.9	3.0	2.7	0.2	0.3	0.4	0.3	0.1	0.0	0.1	0.1	0.1

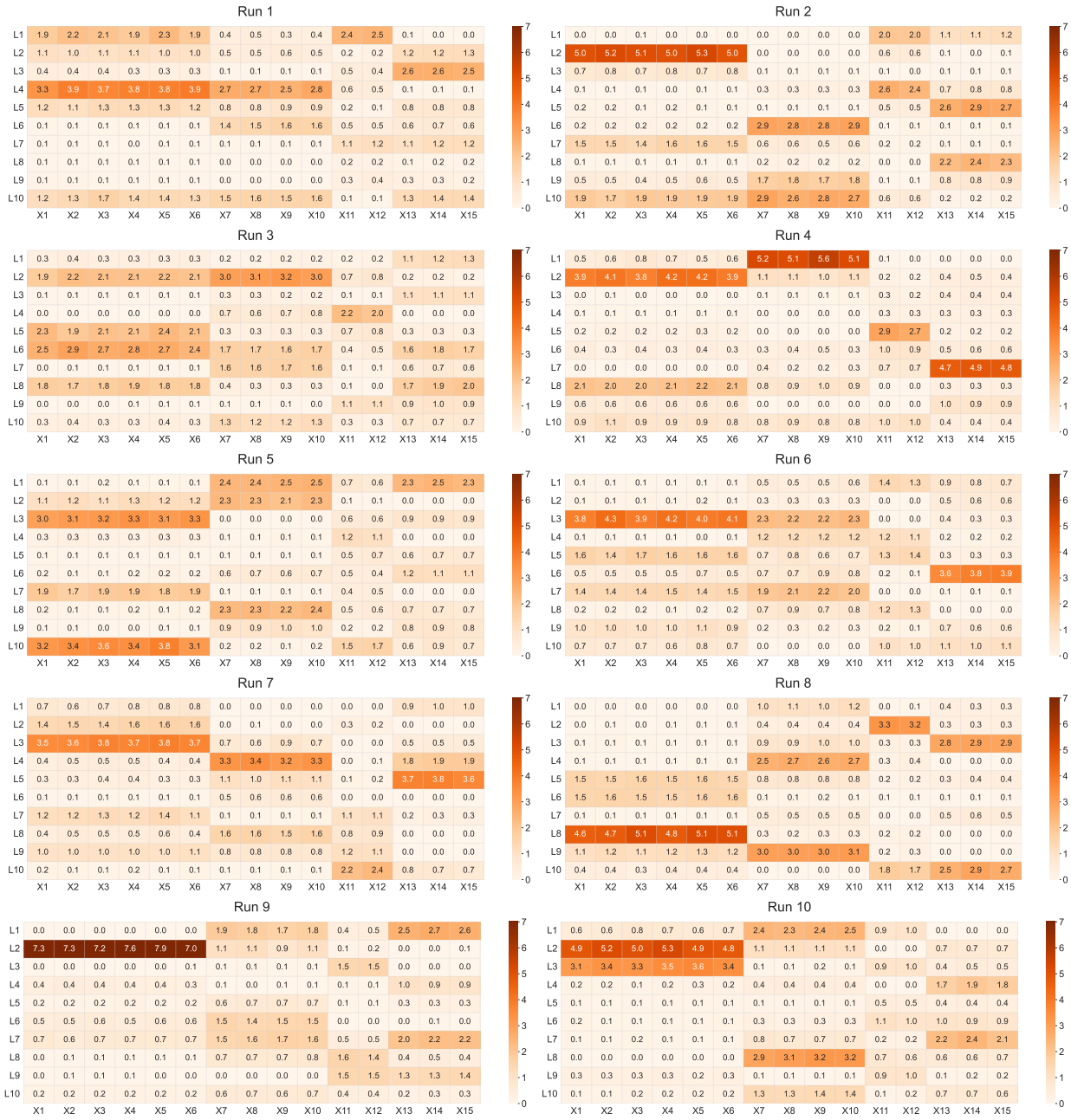
I.12 FA15: FVH-LT result with DIP-VAE-II



I.13 FA15: FVH-LT results with bfVAE when $K = 10$



I.14 FA15: FVH-LT results with DIP-VAE-I when $K = 10$



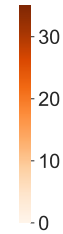
J bf-CVAE Implementation and FVH-LT Results

J.1 bf-CVAE Implementation

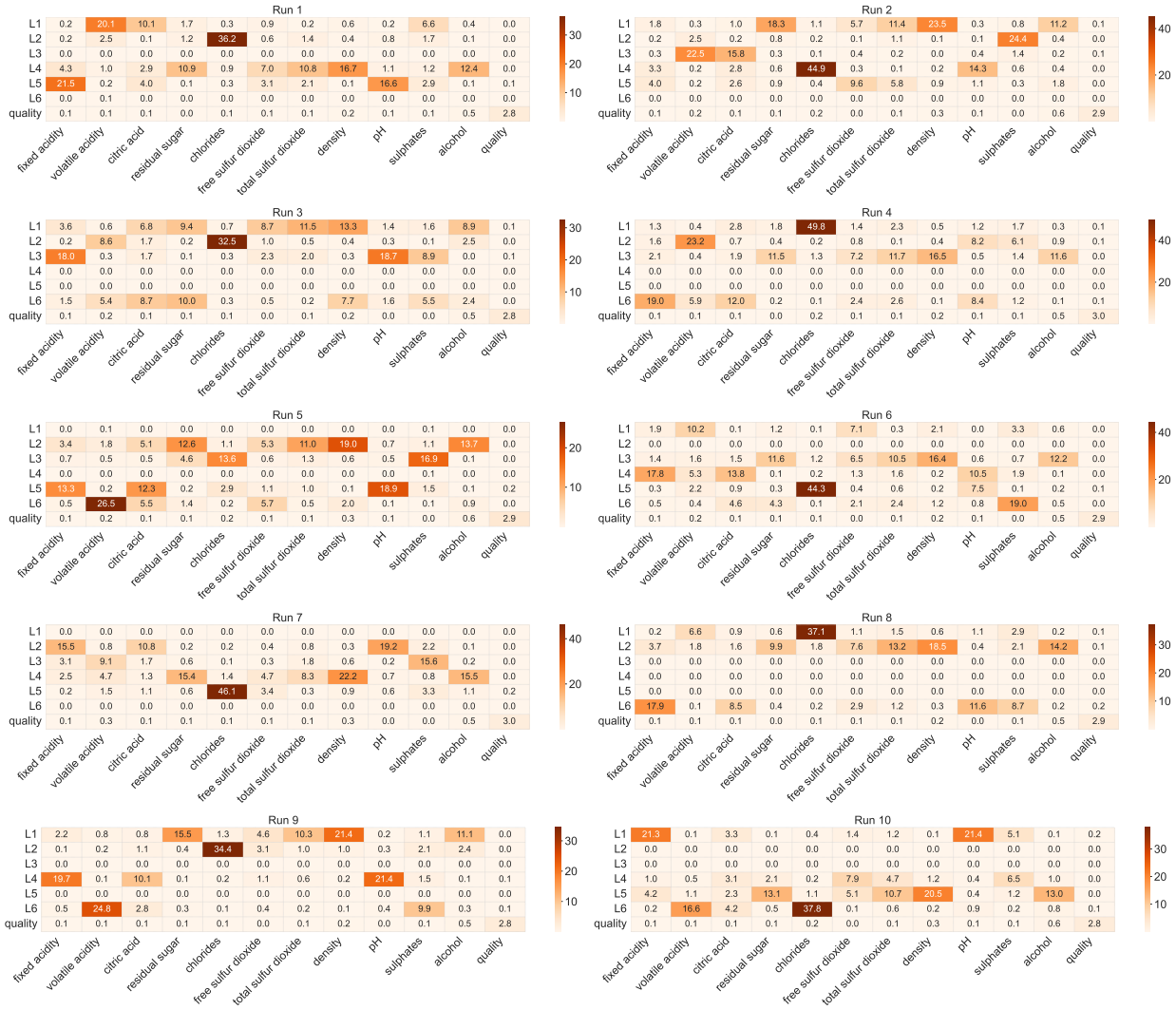
We slightly modified the bfVAE structure and the FVH-LT procedure by incorporating the wine quality score y as part of the input, i.e., $\mathbf{x}' \leftarrow (\mathbf{x}, y)$, to the encoder. We then appended y to the learned latent space $\mathbf{z}$ i.e., $\mathbf{z}' \leftarrow (\mathbf{z}, y)$, before being passed to the decoder. We performed LT on y with its range in the observed data while fixing the latent $\mathbf{z}$ at a randomly drawn sample from its posterior distribution. Variances of the generated $\hat{\mathbf{x}}$ were computed.

J.2 FVH-LT Results across 10 runs

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
L1	2.9	1.4	2.5	12.8	1.2	6.2	10.9	18.8	0.6	1.2	12.4	0.0
L2	13.0	1.4	7.0	0.4	0.5	2.9	2.1	0.3	11.5	2.7	0.3	0.1
L3	0.7	3.9	1.6	1.1	37.7	1.2	0.9	0.5	2.8	3.0	0.8	0.1
L4	4.9	14.2	5.7	1.6	0.2	2.1	0.6	1.4	4.4	6.2	0.6	0.1
L5	0.1	0.3	0.5	0.5	0.0	0.2	0.4	0.1	0.1	4.4	0.1	0.0
L6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
quality	0.1	0.2	0.1	0.1	0.1	0.0	0.1	0.3	0.1	0.0	0.5	2.9

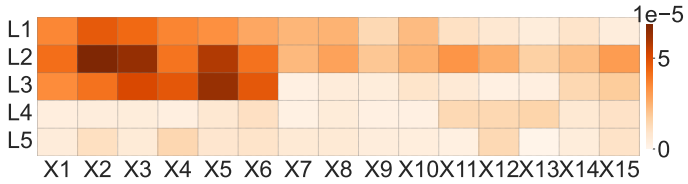


J.3 Single run

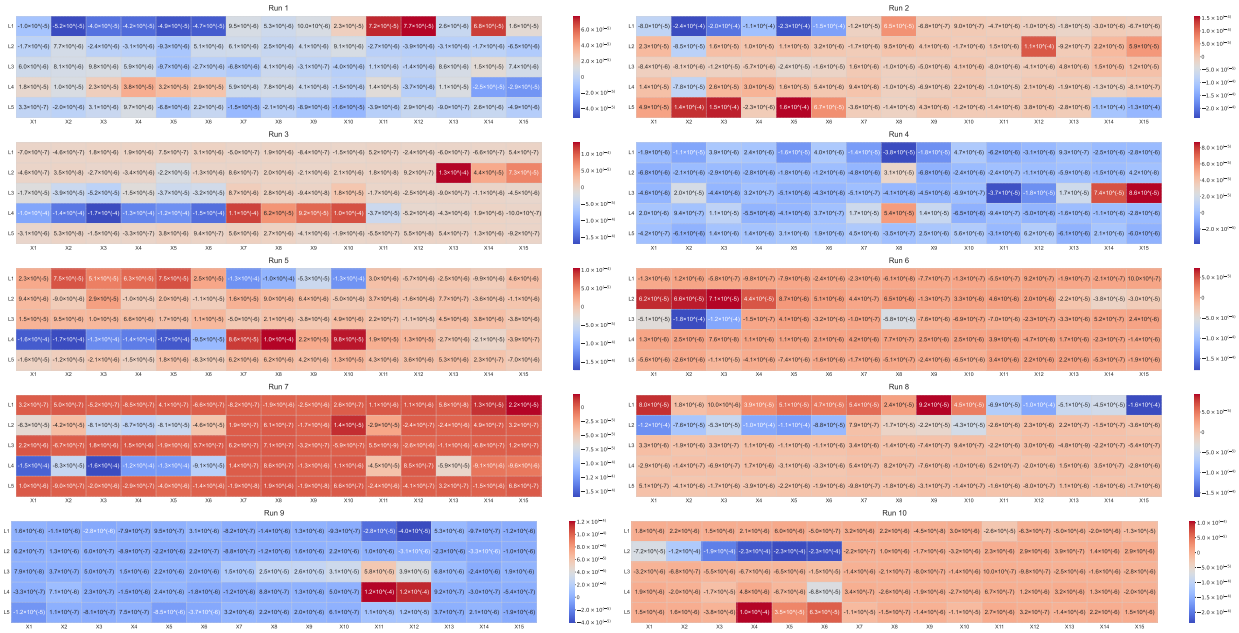

Run 1
L1 0.2 20.1 10.1 1.7 0.3 0.9 0.2 0.6 0.2 6.6 0.4 0.0
L2 0.2 2.5 0.1 1.2 36.2 0.6 1.4 0.4 0.8 1.7 0.1 0.0
L3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L4 4.3 1.0 2.9 10.9 0.9 7.0 10.8 16.7 1.1 1.2 12.4 0.0
L5 21.5 0.2 4.0 0.1 0.3 3.1 2.1 0.1 16.6 2.9 0.1 0.1
L6 0.0 0.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
quality 0.1 0.1 0.1 0.0 0.1 0.1 0.1 0.2 0.1 0.1 0.5 2.8
Run 2
L1 1.8 0.3 1.0 18.3 1.1 5.7 11.4 23.5 0.3 0.8 11.2 0.1
L2 0.2 2.5 0.2 0.8 0.2 0.1 1.1 0.1 0.1 24.4 0.4 0.0
L3 0.3 22.5 15.8 0.3 0.1 0.4 0.2 0.0 0.4 1.4 0.2 0.1
L4 3.3 0.2 2.8 0.6 44.9 0.3 0.1 0.2 14.3 0.6 0.4 0.0
L5 4.0 0.2 2.6 0.9 0.4 9.6 5.8 0.9 1.1 0.3 1.8 0.0
L6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
quality 0.1 0.2 0.1 0.1 0.2 0.0 0.1 0.3 0.1 0.0 0.6 2.9
Run 3
L1 3.6 0.6 6.8 9.4 0.7 8.7 11.5 13.3 1.4 1.6 8.9 0.1
L2 0.2 8.6 1.7 0.2 32.5 1.0 0.5 0.4 0.3 0.1 2.5 0.0
L3 18.0 0.3 1.7 0.1 0.3 2.3 2.0 0.3 18.7 8.9 0.0 0.1
L4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L6 1.5 5.4 8.7 10.0 0.3 0.5 0.2 7.7 1.6 5.5 2.4 0.0
quality 0.1 0.2 0.1 0.1 0.1 0.0 0.1 0.2 0.0 0.0 0.5 2.8
Run 4
L1 1.3 0.4 2.8 1.8 49.8 1.4 2.3 0.5 1.2 1.7 0.3 0.1
L2 1.6 23.2 0.7 0.4 0.2 0.8 0.1 0.4 8.2 6.1 0.9 0.1
L3 2.1 0.4 1.9 11.5 1.3 7.2 11.7 16.5 0.5 1.4 11.6 0.0
L4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L6 19.0 5.9 12.0 0.2 0.1 2.4 2.6 0.1 8.4 1.2 0.1 0.1
quality 0.1 0.1 0.1 0.0 0.2 0.1 0.1 0.2 0.1 0.1 0.5 3.0
Run 5
L1 0.0 0.1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.1 0.0 0.0
L2 3.4 1.8 5.1 12.6 1.1 5.3 11.0 19.0 0.7 1.1 13.7 0.0
L3 0.7 0.5 0.5 4.6 13.6 0.6 1.3 0.6 0.5 16.9 0.1 0.0
L4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.1 0.0 0.0
L5 13.3 0.2 12.3 0.2 2.9 1.1 1.0 0.1 18.9 1.5 0.1 0.2
L6 0.5 26.5 5.5 1.4 0.2 5.7 0.5 2.0 0.1 0.1 0.9 0.0
quality 0.1 0.2 0.1 0.1 0.2 0.1 0.1 0.3 0.1 0.0 0.6 2.9
Run 6
L1 1.9 10.2 0.1 1.2 0.1 7.1 0.3 2.1 0.0 3.3 0.6 0.0
L2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L3 1.4 1.6 1.5 11.6 1.2 6.5 10.5 16.4 0.6 0.7 12.2 0.0
L4 17.8 5.3 13.8 0.1 0.2 1.3 1.6 0.2 10.5 1.9 0.1 0.0
L5 0.3 2.2 0.9 0.3 44.3 0.4 0.6 0.2 7.5 0.1 0.2 0.1
L6 0.5 0.4 4.6 4.3 0.1 2.1 2.4 1.2 0.8 19.0 0.5 0.0
quality 0.1 0.2 0.1 0.1 0.1 0.0 0.1 0.2 0.1 0.0 0.5 2.9
Run 7
L1 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L2 15.5 0.8 10.8 0.2 0.2 0.4 0.8 0.3 19.2 2.2 0.1 0.0
L3 3.1 9.1 1.7 0.6 0.1 0.3 1.8 0.6 0.2 15.6 0.2 0.0
L4 2.5 4.7 1.3 15.4 1.4 4.7 8.3 22.2 0.7 0.8 15.5 0.0
L5 0.2 1.5 1.1 0.6 46.1 3.4 0.3 0.9 0.6 3.3 1.1 0.2
L6 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
quality 0.1 0.3 0.1 0.1 0.1 0.1 0.1 0.3 0.0 0.0 0.5 3.0
Run 8
L1 0.2 6.6 0.9 0.6 37.1 1.1 1.5 0.6 1.1 2.9 0.2 0.1
L2 3.7 1.8 1.6 9.9 1.8 7.6 13.2 18.5 0.4 2.1 14.2 0.1
L3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L4 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L6 17.9 0.1 8.5 0.4 0.2 2.9 1.2 0.3 11.6 8.7 0.2 0.2
quality 0.1 0.1 0.1 0.0 0.1 0.1 0.1 0.2 0.0 0.1 0.5 2.9
Run 9
L1 2.2 0.8 0.8 15.5 1.3 4.6 10.3 21.4 0.2 1.1 11.1 0.0
L2 0.1 0.2 1.1 0.4 34.4 3.1 1.0 1.0 0.3 2.1 2.4 0.0
L3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L4 19.7 0.1 10.1 0.1 0.2 1.1 0.6 0.2 21.4 1.5 0.1 0.1
L5 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L6 0.5 24.8 2.8 0.3 0.1 0.4 0.2 0.1 0.4 9.9 0.3 0.1
quality 0.1 0.1 0.1 0.1 0.1 0.0 0.1 0.2 0.0 0.1 0.5 2.8
Run 10
L1 21.3 0.1 3.3 0.1 0.4 1.4 1.2 0.1 21.4 5.1 0.1 0.2
L2 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L3 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0
L4 1.0 0.5 3.1 2.1 0.2 7.9 4.7 1.2 0.4 6.5 1.0 0.0
L5 4.2 1.1 2.3 13.1 1.1 5.1 10.7 20.5 0.4 1.2 13.0 0.0
L6 0.2 16.6 4.2 0.5 37.8 0.1 0.6 0.2 0.9 0.2 0.8 0.1
quality 0.1 0.1 0.1 0.1 0.2 0.0 0.1 0.3 0.1 0.1 0.6 2.8
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality

K factor-VAE: DBSR-LS result

K.1 Results across 10 runs

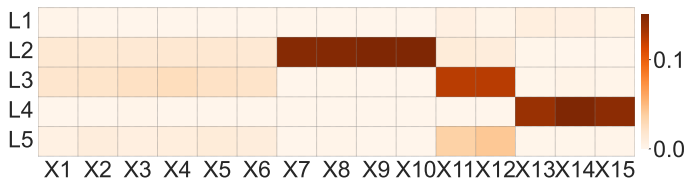


K.2 Single run

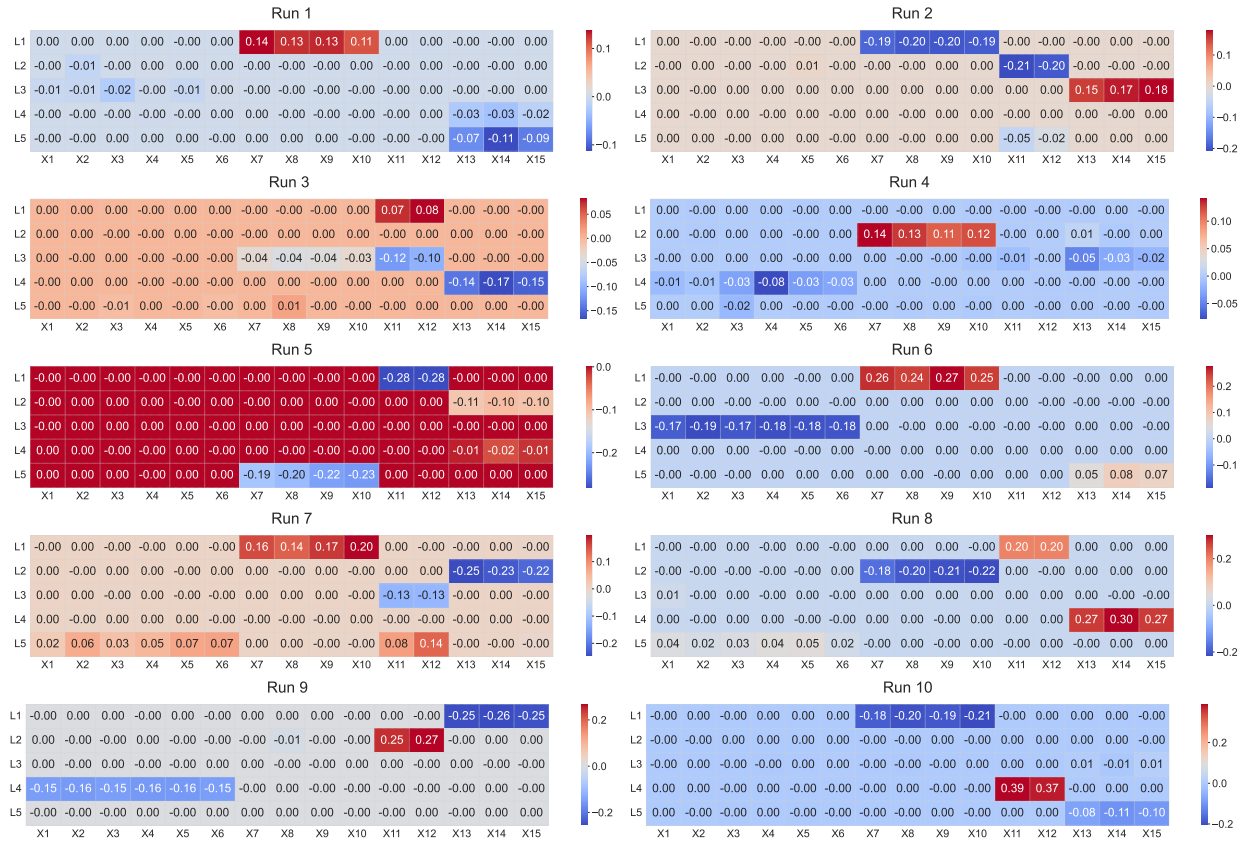


L β-VAE: DBSR-LS results

L.1 Results across 10 runs

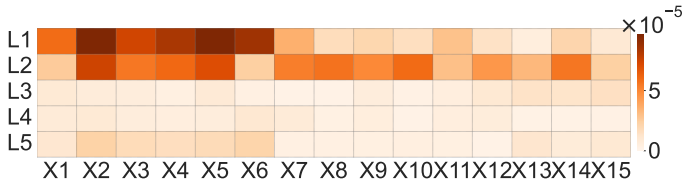


L.2 Single run

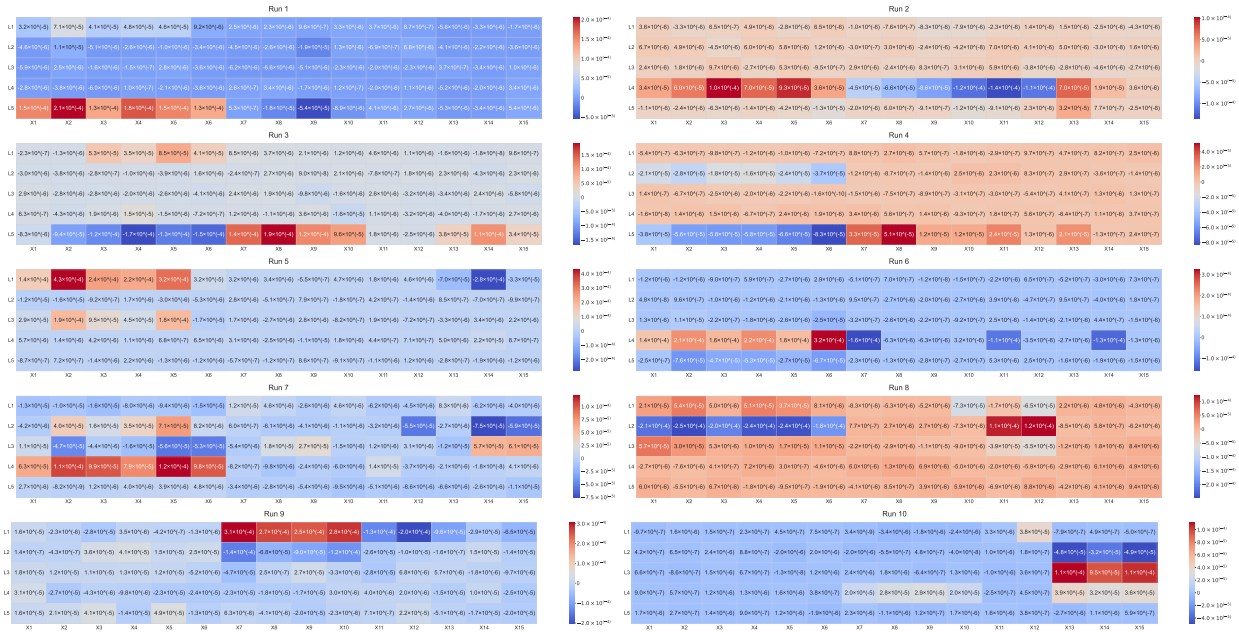


M Vanilla VAE: DBSR-LS result

M.1 Results across 10 runs

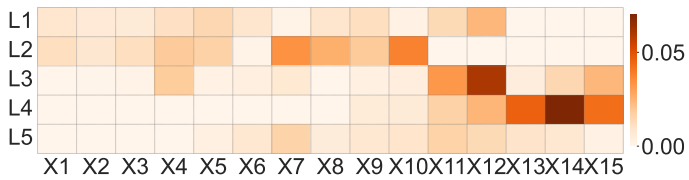


M.2 Single run

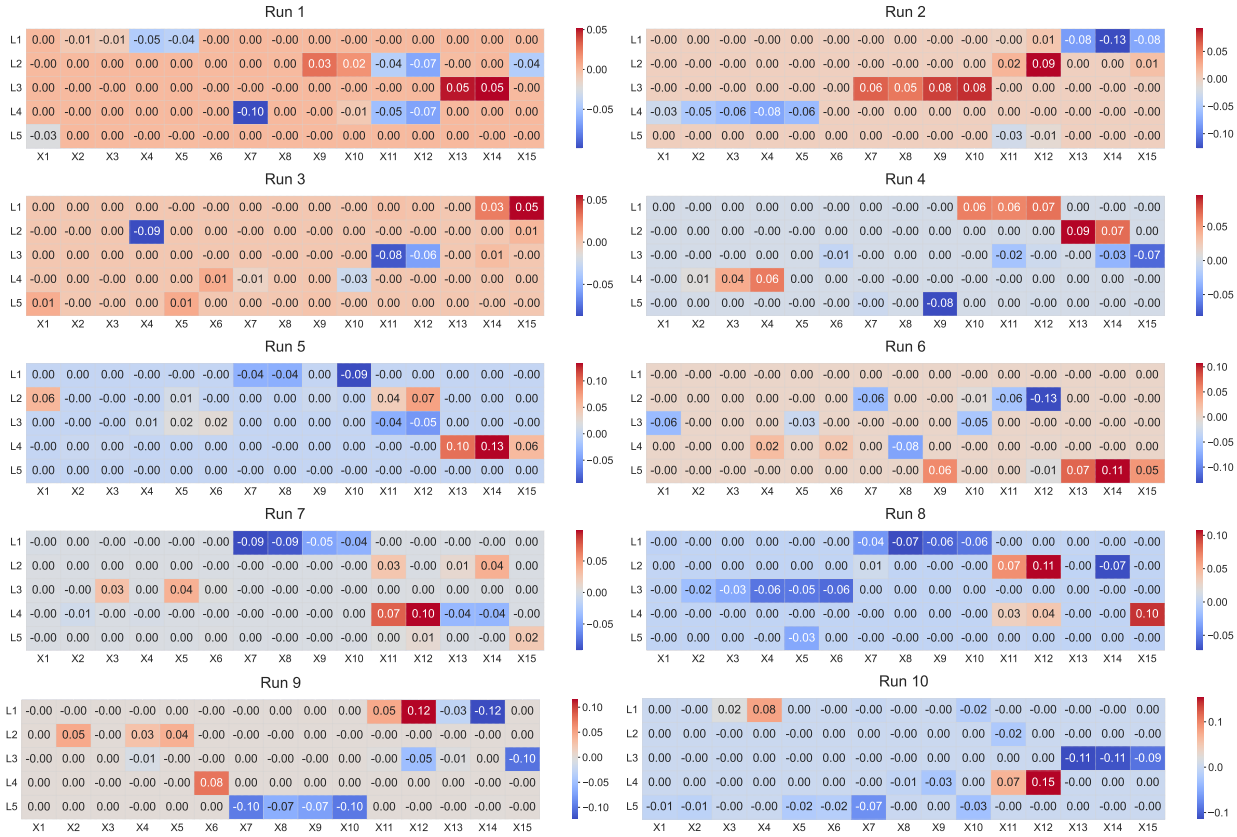


N DIP-VAE-I: DBSR-LS result

N.1 Results across 10 runs

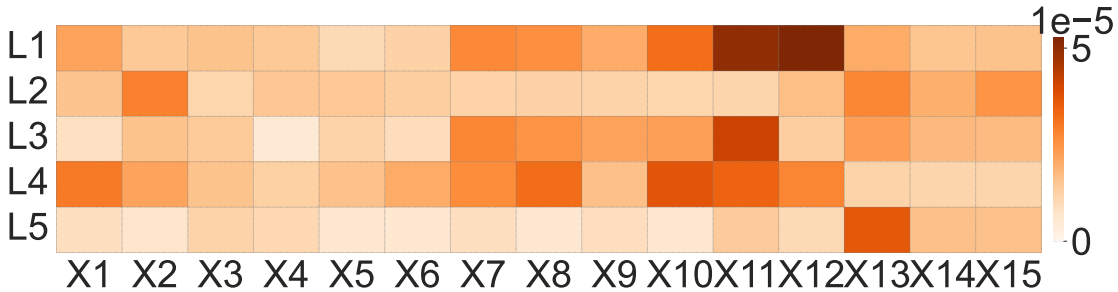


N.2 Single run

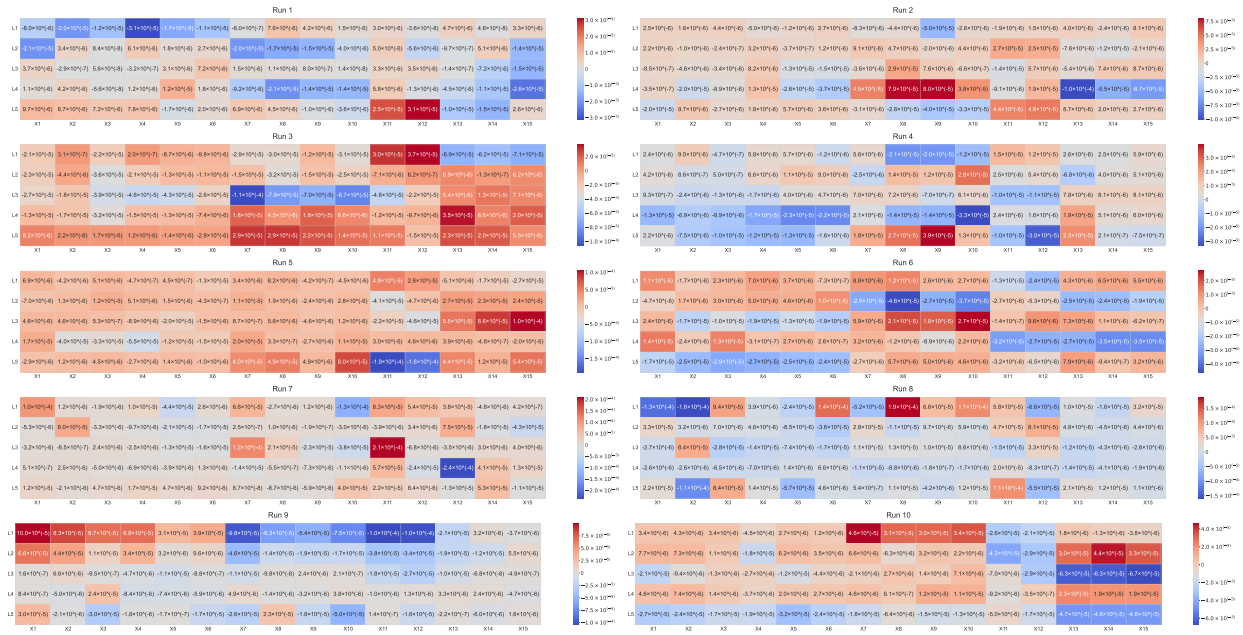


O DIP-VAE-II: DBSR-LS result

O.1 Results across 10 runs

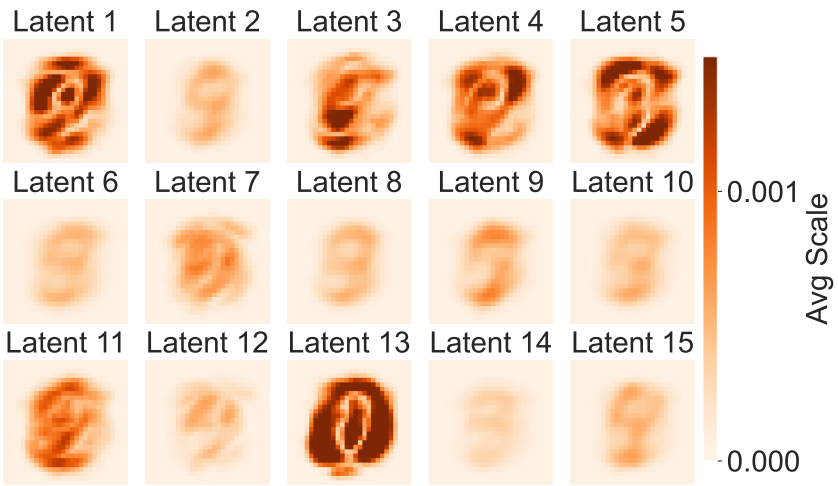


O.2 Single run

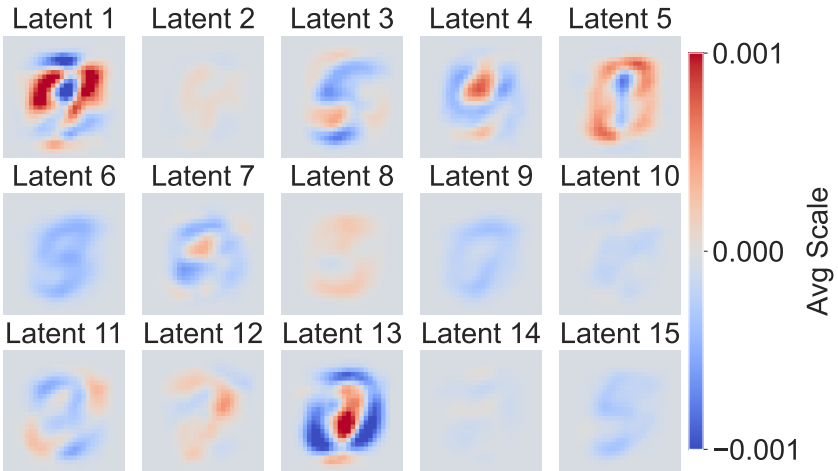


P MNIST: DBSR-LS results

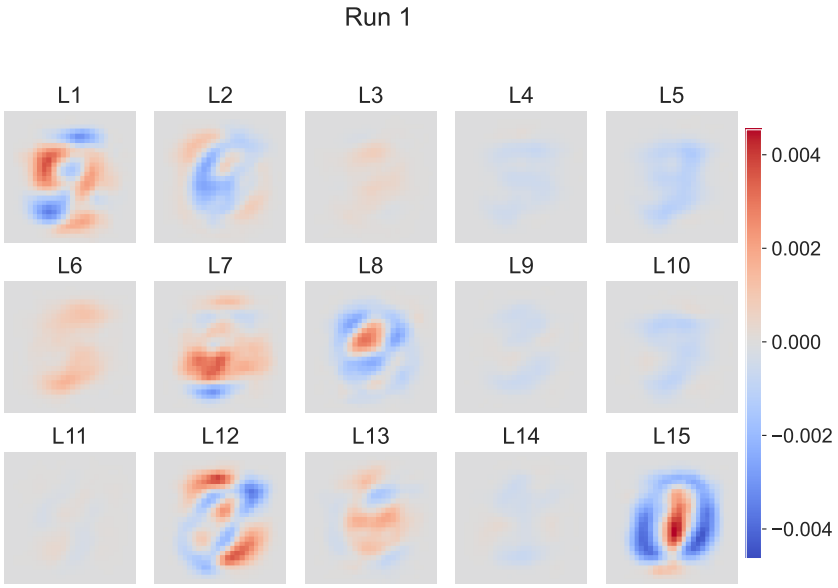
P.1 Results over 10 runs (with matrix $|\hat{D}|$)

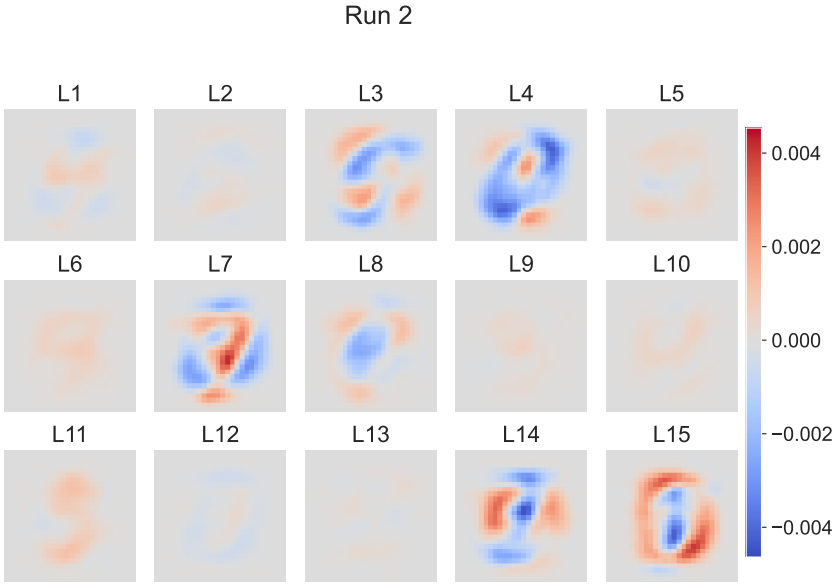


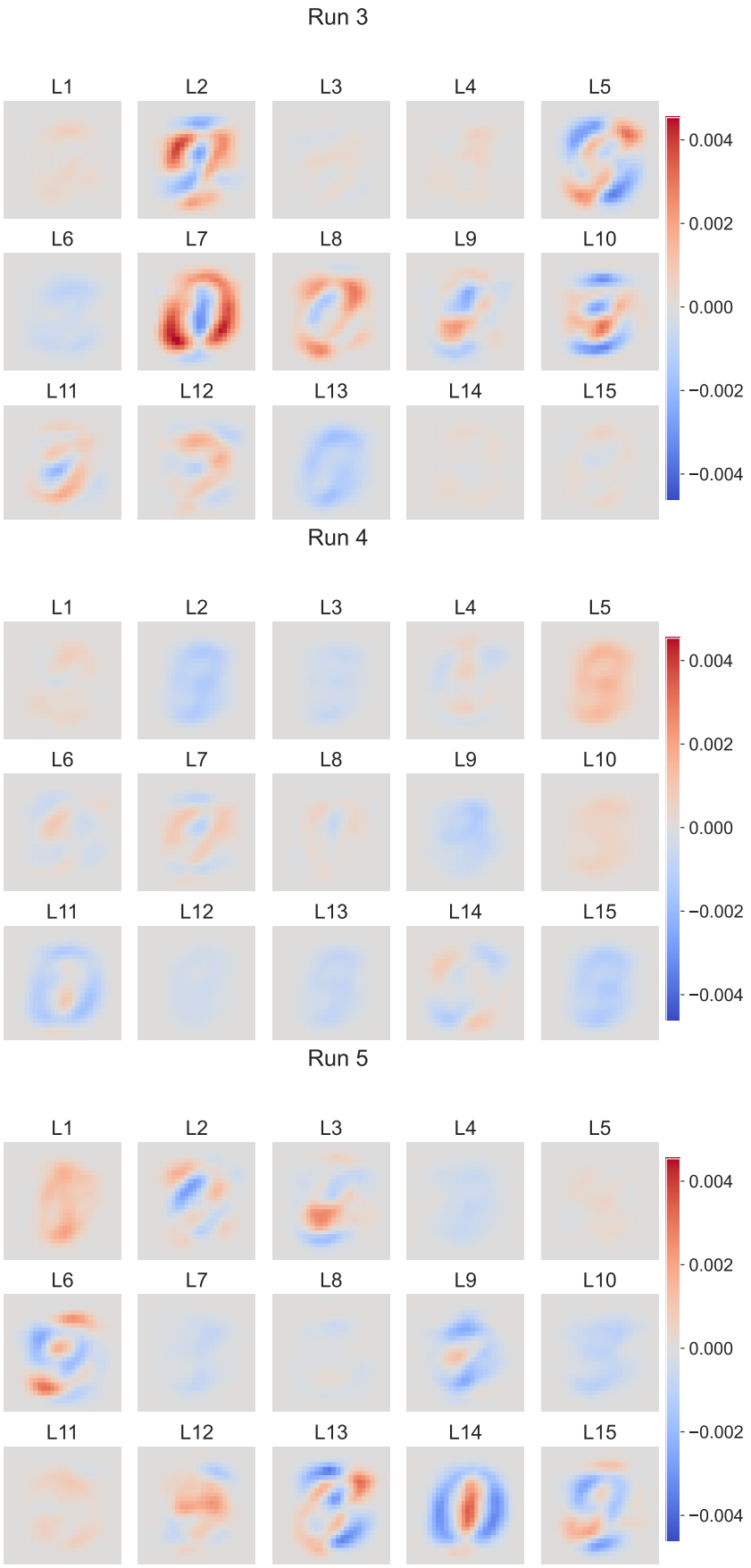
P.2 Results over 10 runs (with matrix $\hat{D}$)

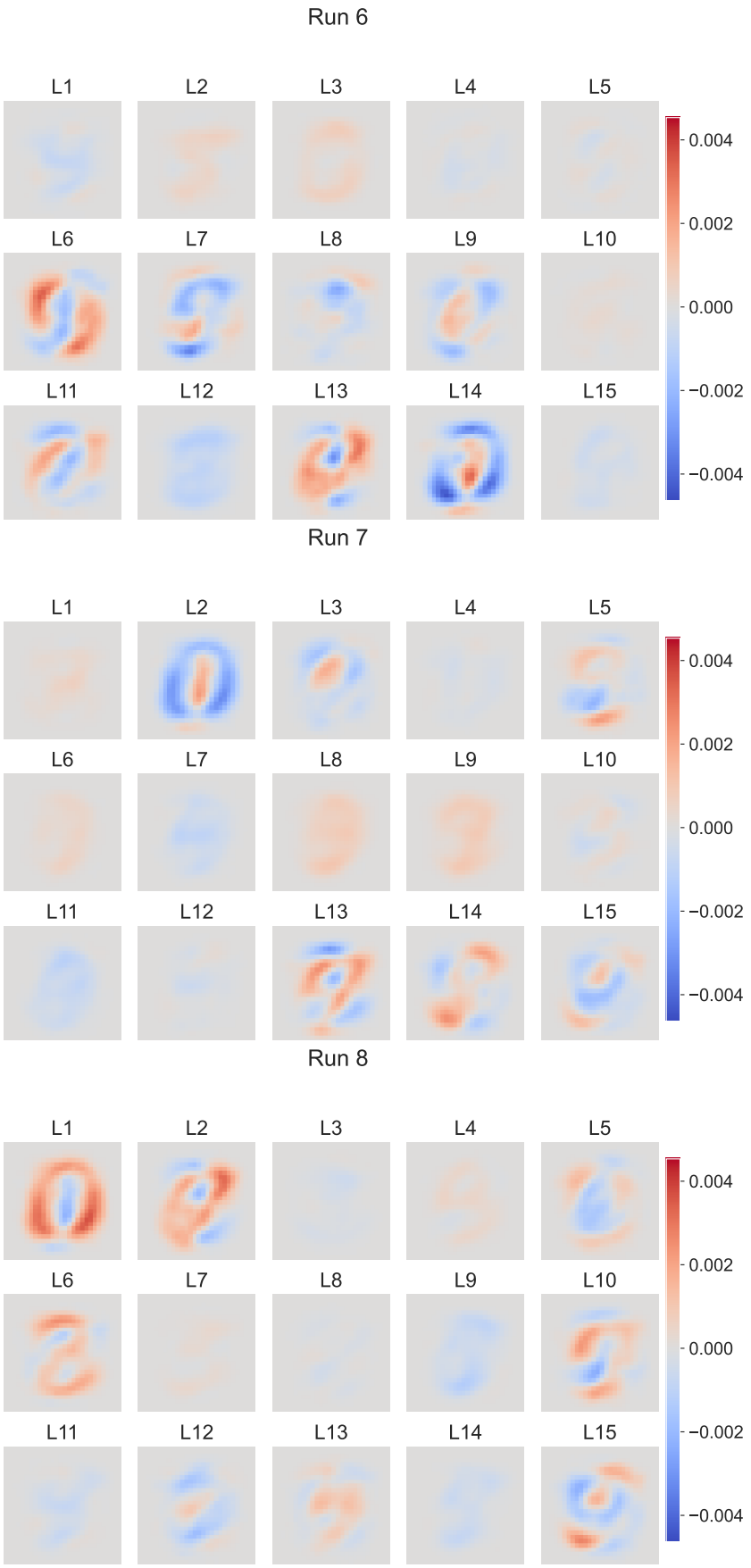


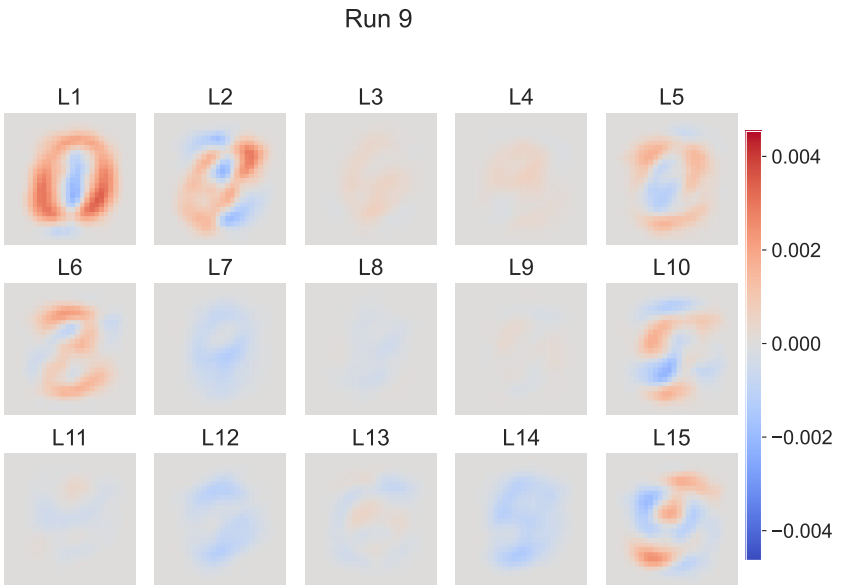
P.3 Single-run results

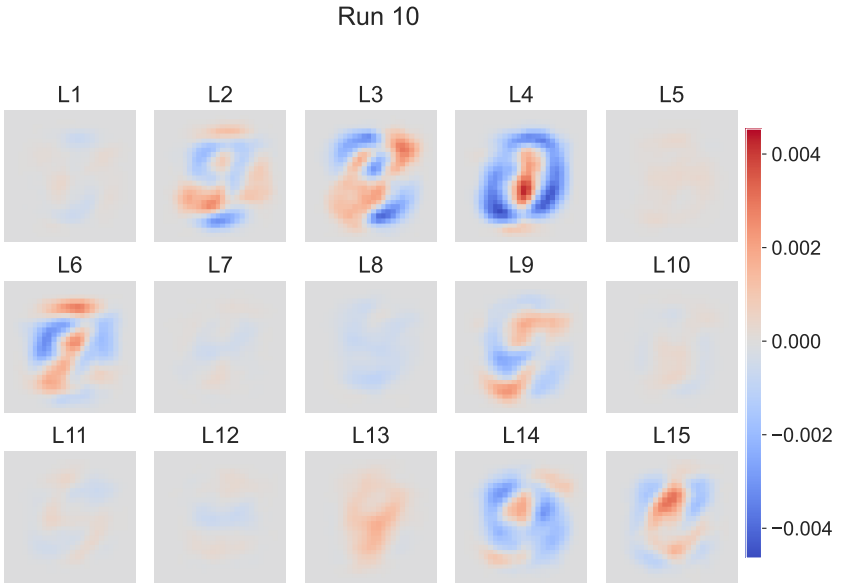





Run 3
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.004
0.002
0.000
−0.002
−0.004
Run 4
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.004
0.002
0.000
−0.002
−0.004
Run 5
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.004
0.002
0.000
−0.002
−0.004


Run 6
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.004
0.002
0.000
−0.002
−0.004
Run 7
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.004
0.002
0.000
−0.002
−0.004
Run 8
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.004
0.002
0.000
−0.002
−0.004


Run 9
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.004
0.002
0.000
−0.002
−0.004


Run 10
L1
L2
L3
L4
L5
L6
L7
L8
L9
L10
L11
L12
L13
L14
L15
0.004
0.002
0.000
−0.002
−0.004

Q domain knowledge on wine quality data

VC is a key indicator of fermentation quality and excessive levels often lead to sensory defects. C contributes to the perceived wine saltiness and may reflect suboptimal production conditions. D captures the balance between residual sugar and A, influencing wine texture. A strongly affects mouthfeel and overall quality perception (higher levels are generally associated with higher quality).