PCA-Guided Clustering for Optimal β-VAE

Syed Ebad Hyder
Department of Computer Engineering
Middle East Technical University
Ankara, Türkiye
ebad.hyder@metu.edu.tr

Affan Ahmed
Department of Computer Engineering
Middle East Technical University
Ankara, Türkiye
affan.ahmed@metu.edu.tr

Hazal Moğultay Özcan
Department of Computer Engineering
Middle East Technical University
Ankara, Türkiye
mogultay@metu.edu.tr

Abstract

This paper presents a comprehensive study on the disentanglement properties of Principal Component Analysis (PCA) and β-Variational Autoencoders (β-VAE) in the context of feature disentanglement within image datasets, introducing a systematic and computationally efficient pipeline designed to determine the optimal latent dimensions for disentangled representation learning. The pipeline begins with an analysis of the disentanglement potential using PCA [12], followed by the identification of optimal latent dimensions through k-means clustering metrics, including the silhouette score [13], Calinski-Harabasz index, and Davies-Bouldin index [14], with subsequent validation using β-VAE models [1]. Our experiments, conducted on the widely recognized dSprites [15] and MPI3D Toy [16] datasets, demonstrate a strong correlation between the optimal cluster count derived from PCA-transformed data and the most effective latent dimensionality for β-VAE, significantly reducing the need for extensive hyperparameter tuning while achieving competitive disentanglement scores [9, 10]. Providing a principled and efficient way to select latent dimensions before training complex disentanglement models, our research enhances the efficiency and effectiveness of feature disentanglement in unsupervised learning tasks.

Keywords: Feature Disentanglement, Principal Component Analysis (PCA), Beta-Variational Autoencoder (β-VAE), Representation Learning, K-means Clustering, Disentanglement Metrics

1 Introduction

Disentangled representation learning has emerged as a crucial paradigm in machine learning, where the goal is to encode independent generative factors of data variation into separate latent dimensions [11]. This capability is essential for applications like data compression and transfer learning [16], where disentangled representations can bridge the gap between synthetic and real-world data domains. While deep learning approaches like Beta-Variational Autoencoders (β-VAEs) [1] have shown promising results, they suffer from two key challenges: the need for extensive hyperparameter tuning, particularly for selecting the number of latent dimensions, and high computational costs during training and evaluation.

In this study, we address these challenges by proposing a novel two-stage approach that leverages Principal Component Analysis (PCA) and k-means clustering metrics to identify the optimal latent dimensions for β-VAE models. In the first stage, PCA Disentanglement Analysis, we apply PCA to the dataset and evaluate its disentanglement capability through Pearson correlation heatmaps between principal components and ground truth factors, visual inspection of 3D PCA scatter plots for separable clusters, and quantitative disentanglement metrics such as DCI [9], Explicitness, ZDiff, and CORDIS.

In the second stage, optimal dimension selection, we determine the optimal latent dimension count by applying k-means clustering to PCA-transformed data for a range of k values. We evaluate clustering quality using silhouette [13], Calinski-Harabasz, and Davies-Bouldin indices [14], and select the k with the best weighted combination of scores. Our key insight is that the optimal k in clustering space corresponds to the most effective latent dimensionality for nonlinear disentanglement methods like β-VAEs. We validate this through extensive experiments on the MPI3D Toy [16] and dSprites [15] datasets, demonstrating that our method reduces the need for trial-and-error in dimension selection, provides interpretable intermediate results through PCA analysis, and achieves comparable disentanglement to exhaustive β-VAE searches with significantly less computation.

2 Related Work

Our work bridges three research areas: disentangled representations, PCA applications, and clustering-based model selection.

2.1 Disentangled Representations

Modern disentanglement approaches extend beyond β-VAE [1] along several directions. Variational methods like FactorVAE [3] and TC-VAE [4] improve independence through adversarial training and direct total correlation minimization respectively, though with increased complexity. Supervised and semi-supervised approaches including AdaVAE and GroupVAE [5] leverage label information for better performance, while hierarchical models like LVAE [6] capture multi-scale features through careful architectural design. Recent paradigms incorporate contrastive learning [7] and diffusion processes [8], offering different trade-offs between computational requirements and disentanglement quality. Evaluation remains challenging despite metrics like DCI [9] and CORDIS, with Locatello et al. [11] highlighting fundamental limitations. Across all methods - from simple β-VAEs to complex diffusion models - the need to specify latent dimensionality beforehand persists as a critical limitation that our work directly addresses.

2.2 PCA for Disentanglement

While PCA is traditionally used for dimensionality reduction, recent work has explored its disentanglement capabilities. Abid et al. [12] showed PCA can identify interpretable directions in gene expression data. Our work extends this by providing a systematic evaluation of PCA's disentanglement potential across multiple benchmarks.

2.3 Clustering for Model Selection

Unsupervised clustering metrics like silhouette scores [13] and validity indices [14] have been adapted to evaluate disentangled representations. These methods assess latent structure by measuring cluster compactness and separation without requiring ground truth factors. Recent extensions specifically target disentanglement evaluation, though challenges remain in distinguishing truly factorized representations from merely well-clustered ones. Such approaches are particularly valuable when labeled data is unavailable.

3 Methodology

We developed a framework to evaluate unsupervised disentanglement by combining dimensionality reduction, clustering, and representation learning. Our pipeline consists of four key stages: (1) dataset preprocessing, (2) PCA-based feature extraction, (3) k-means clustering for latent structure analysis, and (4) β-VAE training with clustering-based evaluation metrics.

3.1 Datasets

In our study, we utilized two distinct datasets to evaluate the effectiveness of our proposed methodology. These datasets were specifically chosen for their complexity and relevance to the field of feature disentanglement in image analysis. Both datasets offer a rich variety of generative factors, providing a robust framework for evaluating the disentanglement capabilities of our models. By leveraging these datasets, we aim to establish a comprehensive understanding of how different methods perform under varied conditions.

3.1.1 dSprites Dataset

The dSprites dataset consists of 737,280 procedurally generated 2D shapes, each defined by six independent latent factors: color, shape, scale, rotation, and x and y positions. The dataset includes all possible combinations of these latent factors, with each image being unique. The latent factors and their possible values are as follows:

Latent Factor	**Values**
Color	white
Shape	square, ellipse, heart
Scale	six values linearly spaced between 0.5 and 1
Orientation	40 values ranging from 0 to 2π
Position X	32 values ranging from 0 to 1
Position Y	32 values ranging from 0 to 1

Table 1: dSprites Dataset Latent Factors and Values

3.1.2 MPI3D Toy Dataset

The MPI3D Toy dataset comprises 1,036,800 images, representing all possible combinations of various factors of variation. These factors include object color, shape, size, camera height, background color, horizontal axis, and vertical axis. The dataset is designed to benchmark representation learning algorithms across simulated and real-world environments. The factors and their possible values are as follows:

Latent Factor	**Values**
Object Color	white, green, red, blue, brown, olive
Object Shape	cone, cube, cylinder, hexagonal, pyramid, sphere
Object Size	small, large
Camera Height	top, center, bottom
Background Color	purple, sea green, salmon
Horizontal Axis	0 to 39
Vertical Axis	0 to 39

Table 2: MPI3D Toy Dataset Latent Factors and Values

3.2 Principal Component Analysis (PCA)

We applied Principal Component Analysis (PCA) to both the dSprites and MPI3D Toy datasets to identify disentangled attributes. PCA was performed using an incremental approach, known as IncrementalPCA, which allows efficient processing of large datasets in smaller batches [19, 20]. This method incrementally updates the principal components as new data batches are fed, making it suitable for datasets that do not fit entirely into memory.

3.2.1 Parameters for PCA

For both datasets, the PCA was configured with the following parameters:

Parameter	**Value**
Number of components	10
Batch size	50,000

Table 3: Parameters for PCA

We used the `IncrementalPCA` implementation from the `scikit-learn` Python library [19], which employs stochastic gradient descent for online eigenvector estimation. The methodology for this implementation is detailed by Lippi and Ceccarelli [20]. The training pipeline involved two main phases: First, the PCA model was incrementally fitted on image batches, with each image flattened into a vector. Second, the trained model was used to transform the entire dataset into a reduced 10-dimensional latent space. After training the PCA models, we created 2D (Figure 1, 3) and 3D (Figure 2, 4) scatter plots using the first two and three principal components, respectively, to visually inspect the separation of generative factors. Additionally, Pearson correlation heatmaps were generated between the PCA dimensions and known factors of variation, allowing for a quantitative evaluation of the effectiveness of disentanglement.

3.3 K-means Clustering

To determine the optimal number of latent dimensions for β-VAE, we applied K-means clustering to the PCA-transformed data. We evaluated clustering quality across different values of k using three internal validation metrics: Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index.

To aggregate these metrics into a single robust measure, we adopted the SPINEX methodology [18], which computes a composite clustering score from multiple internal indices. The steps are as follows:

Algorithm 1 Composite Clustering Score Calculation

1: **Input:** Silhouette Score $S(k)$, Calinski-Harabasz Index $CH(k)$, Davies-Bouldin Index $DB(k)$
2: **Output:** Composite Score Composite(k)
Step 1: Normalize Each Metric
3: Normalize $S(k)$ and $CH(k)$ using min-max normalization:

$$\tilde{S}(k) = \frac{S(k) - S_{\min}}{S_{\max} - S_{\min}}, \quad \tilde{CH}(k) = \frac{CH(k) - CH_{\min}}{CH_{\max} - CH_{\min}}$$

4: Normalize $DB(k)$ by inverting and applying min-max normalization:

$$\tilde{DB}(k) = 1 - \frac{DB(k) - DB_{\min}}{DB_{\max} - DB_{\min}}$$

Step 2: Compute the Composite Score
5: Compute the composite score as the average of the normalized metrics:

$$\text{Composite}(k) = \frac{1}{3}\left(\tilde{S}(k) + \tilde{CH}(k) + \tilde{DB}(k)\right)$$

6: **Return:** Composite Score Composite(k)

This score allows us to select the optimal k that balances compactness, separation, and overall cluster quality. We used this method to determine the most effective number of latent dimensions for each β-VAE model trained on the datasets.

3.4 Beta-Variational Autoencoder (β-VAE)

We trained Beta-Variational Autoencoder (β-VAE) models to evaluate their performance in disentangling attributes compared to PCA. The architecture of the β-VAE models varied slightly between the dSprites and MPI3D Toy datasets to account for differences in data structure and complexity.

The β-VAE extends the standard Variational Autoencoder (VAE) by introducing a hyperparameter $\beta > 1$ to control the strength of the Kullback–Leibler divergence penalty. This encourages the latent representations to become more disentangled at the cost of some reconstruction fidelity [1]. The modified loss function is defined as:

$$\mathcal{L}_{\beta\text{-VAE}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] - \beta \cdot D_{\text{KL}}(q_\phi(z|x) \| p(z)), \tag{1}$$

where $q_\phi(z|x)$ is the approximate posterior, $p(z)$ is the prior (typically a unit Gaussian), and $p_\theta(x|z)$ is

the decoder likelihood. This formulation allows β-VAE to promote statistical independence across latent dimensions, which is a key goal in disentanglement learning [2].

3.4.1 β-VAE Architecture for dSprites Dataset

For the dSprites dataset, the β-VAE model consisted of fully connected layers for both the encoder and decoder networks. The architecture is as follows:

Component	Layer Configuration	Activation
Encoder		
	Input: 64×64 flattened to 4096	-
	Hidden Layer 1: $4096 \rightarrow 1200$	ReLU
	Hidden Layer 2: $1200 \rightarrow 1200$	ReLU
	Output μ: $1200 \rightarrow$ latent_dim	Linear
	Output $\log \sigma^2$: $1200 \rightarrow$ latent_dim	Linear
Decoder		
	Input: latent_dim-dimensional latent vector	-
	Hidden Layer 1: latent_dim $\rightarrow 1200$	ReLU
	Hidden Layer 2: $1200 \rightarrow 1200$	ReLU
	Hidden Layer 3: $1200 \rightarrow 1200$	ReLU
	Output: $1200 \rightarrow 4096$, reshaped to 64×64	Sigmoid

Table 4: VAE Network Architecture Specification for dSprites

3.4.2 β-VAE Architecture for MPI3D Toy Dataset

For the MPI3D Toy dataset, the β-VAE model utilized convolutional layers for the encoder and transposed convolutional layers for the decoder. This choice was made to more effectively capture spatial hierarchies in the image data compared to using fully connected layers, which are less suited for preserving spatial relationships. The architecture is as follows:

Component	Layer Configuration	Activation
Encoder		
	Input: $64 \times 64 \times 3$	-
	Conv2d: $3 \rightarrow 32$, 4×4, stride 2, padding 1	ReLU
	Conv2d: $32 \rightarrow 64$, 4×4, stride 2, padding 1	ReLU
	Conv2d: $64 \rightarrow 128$, 4×4, stride 2, padding 1	ReLU
	Conv2d: $128 \rightarrow 256$, 4×4, stride 2, padding 1	ReLU
	Flatten: $256 \times 4 \times 4$	-
	Output μ: $256 \times 4 \times 4 \rightarrow$ latent_dim	Linear
	Output $\log\sigma^2$: $256 \times 4 \times 4 \rightarrow$ latent_dim	Linear
Decoder		
	Input: latent_dim-dimensional latent vector	-
	FC: latent_dim $\rightarrow 256 \times 4 \times 4$	-
	ConvTranspose2d: $256 \rightarrow 128$, 4×4, stride 2, padding 1	ReLU
	ConvTranspose2d: $128 \rightarrow 64$, 4×4, stride 2, padding 1	ReLU
	ConvTranspose2d: $64 \rightarrow 32$, 4×4, stride 2, padding 1	ReLU
	ConvTranspose2d: $32 \rightarrow 3$, 4×4, stride 2, padding 1	Sigmoid

Table 5: ConvVAE Network Architecture Specification for MPI3D Toy

3.4.3 Parameters for β-VAE Training

For both datasets, the β-VAE models were trained with the following parameters:

Parameter	Value
Latent dimensions	3, 5, and 7
Beta (β)	2
Batch size	128
Number of iterations (dSprites)	300,000
Number of iterations (MPI3D Toy)	405,000

Table 6: Parameters for β-VAE Training

3.5 Evaluation Metrics

The evaluation of model performance in disentangling various dataset attributes was conducted using several disentanglement metrics provided by Ubisoft's Laforge repository [17]. Specifically, the assessment of the β-VAE models' disentanglement capabilities involved the use of multiple metrics: Disentanglement, Completeness, and Informativeness (collectively referred to as the DCI framework), as well as Explicitness, ZDiff, and CORDIS.

In the DCI framework [9, 21], each metric captures a distinct aspect of the quality of learned representations. Disentanglement measures whether each latent dimension is associated with only one ground truth factor, indicating that the model has successfully separated the underlying factors of variation. Completeness evaluates whether each ground truth factor is primarily captured by a single latent variable, rather than being spread across several dimensions. Informativeness quantifies how well the latent representation can predict the ground truth factors, often evaluated using the performance of a simple predictive model such as a linear classifier or regressor.

Explicitness measures how well each latent variable corresponds to a single generative factor. High explicitness indicates that each latent dimension is predominantly influenced by one specific factor, which is desirable for interpretability and control in generative models [21].

ZDiff assesses disentanglement by comparing latent codes across data subsets where factors are fixed. Lower code differences indicate better alignment between latent variables and generative factors [21]. This intervention-based approach complements statistical independence analyses [22].

CORDIS evaluates the consistency and robustness of disentangled representations. It measures how well the learned latent representations can be used to predict the ground truth factors across different datasets or conditions. High CORDIS scores indicate that the model's latent representations are robust and generalize well across different scenarios [23].

The Pearson correlation coefficient, which ranges from -1 to 1, captures the strength of a linear relationship between two variables. In the context of disentanglement, high absolute values indicate that a latent variable is strongly associated with a particular ground truth attribute, suggesting effective disentanglement.

Subsequently, the CORDIS score was determined, and min-max normalization was applied to derive the composite result. This process is detailed in Algorithm 2. This approach follows a framework similar to that introduced in [9], where correlation-based metrics are used to quantitatively evaluate disentangled representations.

Algorithm 2 Composite Disentanglement Score Calculation

1: **Input:** DCI-D $D(k)$, DCI-C $C(k)$, DCI-I $I(k)$, Explicitness $E(k)$, ZDiff $Z(k)$, CORDIS $CO(k)$
2: **Output:** Composite Score Composite(k)
Step 1: Normalize Each Metric
3: Normalize $D(k)$, $C(k)$, $I(k)$, $E(k)$, $Z(k)$, and $CO(k)$ using min-max normalization:

$$\tilde{D}(k) = \frac{D(k) - D_{\min}}{D_{\max} - D_{\min}}, \quad \tilde{C}(k) = \frac{C(k) - C_{\min}}{C_{\max} - C_{\min}}, \quad \tilde{I}(k) = \frac{I(k) - I_{\min}}{I_{\max} - I_{\min}}$$

$$\tilde{E}(k) = \frac{E(k) - E_{\min}}{E_{\max} - E_{\min}}, \quad \tilde{Z}(k) = \frac{Z(k) - Z_{\min}}{Z_{\max} - Z_{\min}}, \quad \tilde{CO}(k) = \frac{CO(k) - CO_{\min}}{CO_{\max} - CO_{\min}}$$

Step 2: Compute the Composite Score
4: Compute the composite score as the average of the normalized metrics:

$$\text{Composite}(k) = \frac{1}{6}\left(\tilde{D}(k) + \tilde{C}(k) + \tilde{I}(k) + \tilde{E}(k) + \tilde{Z}(k) + \tilde{CO}(k)\right)$$

5: **Return:** Composite Score Composite(k)

4 Results and Analysis

This section presents the comprehensive evaluation of our proposed pipeline across both the dSprites and MPI3D Toy datasets. We organize our findings into four key areas: PCA disentanglement analysis,

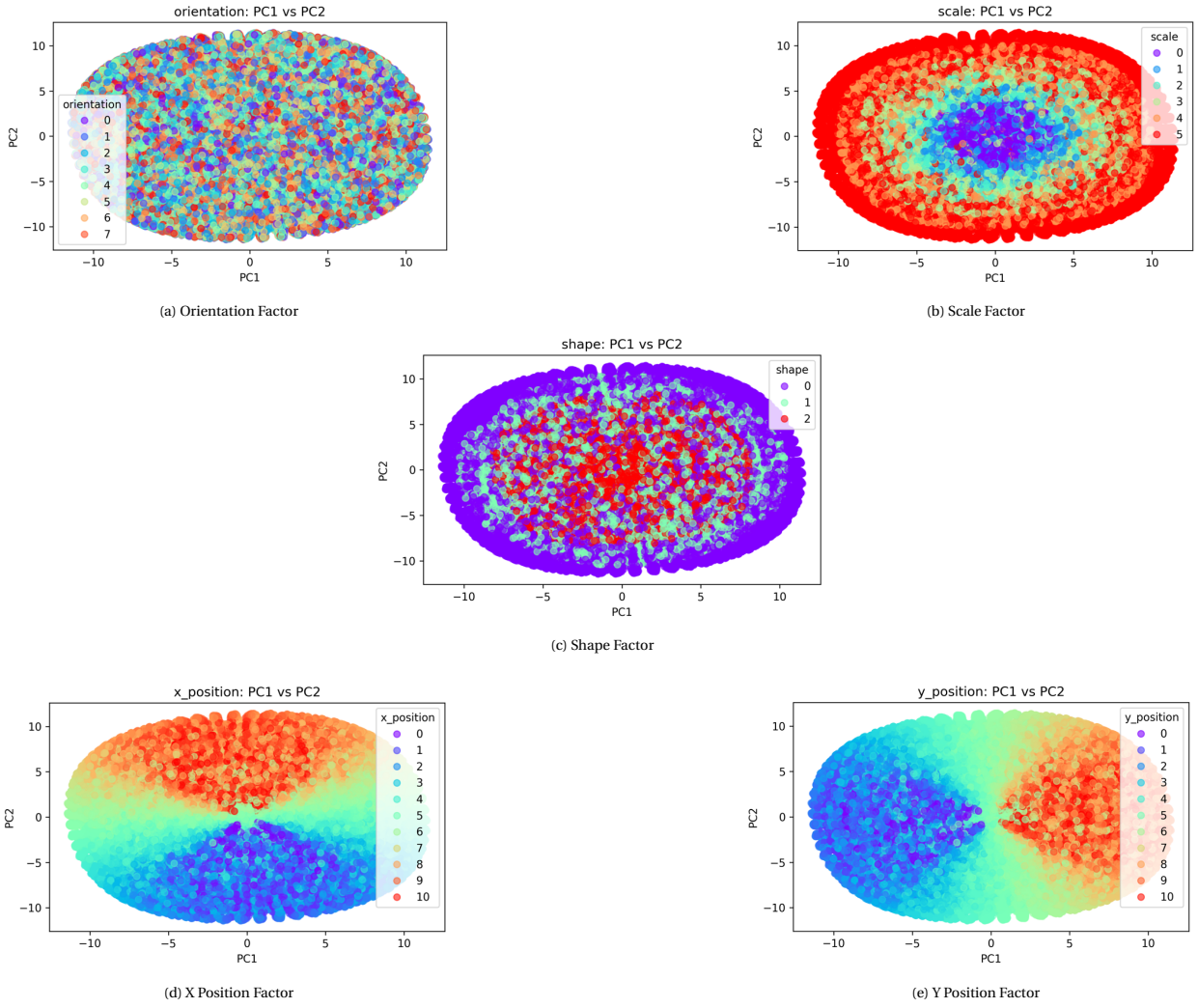
optimal dimension selection through clustering, β-VAE validation results, and comparative performance analysis.

4.1 PCA Disentanglement Analysis

4.1.1 Visual Disentanglement Assessment

To provide an intuitive understanding of PCA's ability to create separable clusters for different factor values, we begin with visual assessments through 2D (Figure 1, 3) and 3D scatter plots (Figure 2, 4) of the principal components, colored by individual generative factors. In these plots, each axis represents one principal component, and each color within a plot represents a different value of that attribute.

dSprite Visual Analysis



(a) Orientation Factor

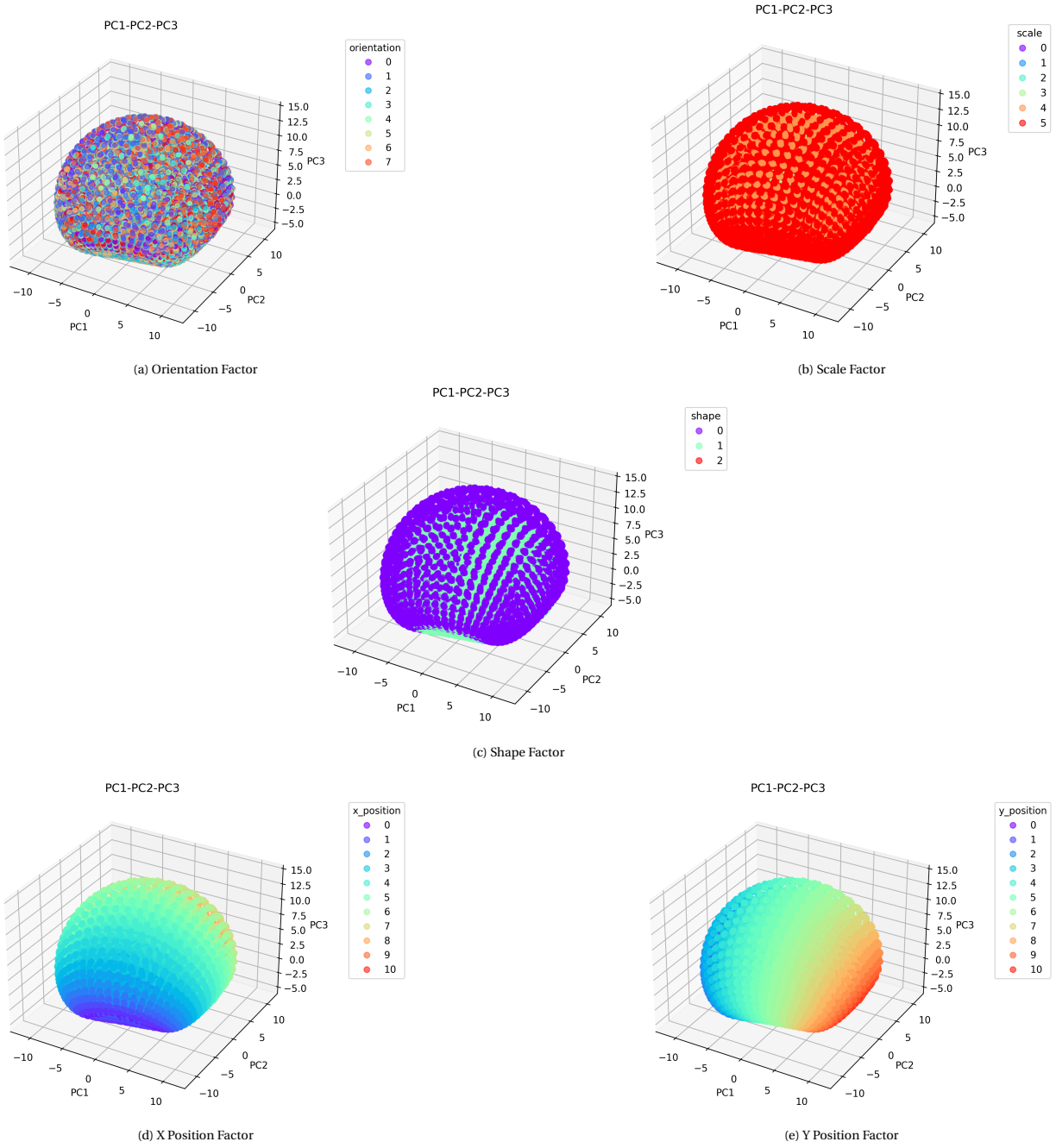
(b) Scale Factor

(c) Shape Factor

(d) X Position Factor

(e) Y Position Factor

Figure 1: 2D PCA scatter plots for the dSprites dataset, illustrating the distribution of data points colored by different generative factors: (a) Orientation, (b) Scale, (c) Shape, (d) X Position, and (e) Y Position. Each plot visualizes the first two principal components, highlighting the degree of linear separability achieved by PCA for each factor.



(a) Orientation Factor

(b) Scale Factor

(c) Shape Factor

(d) X Position Factor

(e) Y Position Factor

Figure 2: 3D PCA scatter plots for the dSprites dataset, illustrating the distribution of data points colored by different generative factors: (a) Orientation, (b) Scale, (c) Shape, (d) X Position, and (e) Y Position. Each plot visualizes the first three principal components, highlighting the degree of linear separability achieved by PCA for each factor.

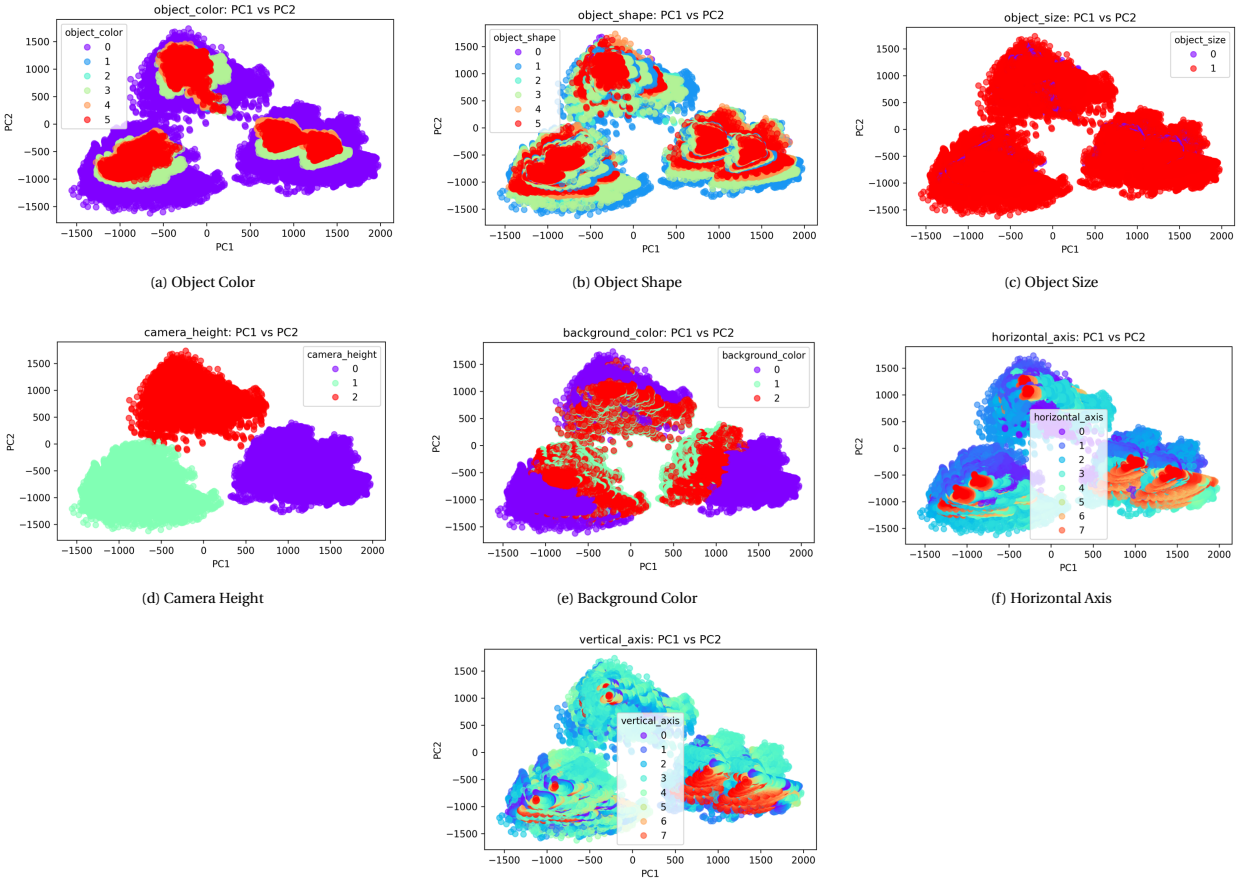
In Figures 1d and 2d, the attribute of Position X demonstrates distinct linear separability within the principal component space. This clear separation indicates a strong level of disentanglement, meaning that this factor is well distinguished in the reduced dimensional space provided by PCA. Similarly, Position Y, as depicted in Figures 1e and 2e, also shows strong linear separability and thus strong disentanglement.

Conversely, Scale and Shape, as depicted in Figures 1b, 2b and Figures 1c, 2c respectively, show partial clustering. This suggests that while there is some level of disentanglement, it is not as clearly separated

as Position X and Position Y, indicating that Scale and Shape are only partially disentangled.

On the other hand, the factor of Orientation, as shown in Figures 1a and 2a, exhibits a uniformly scattered distribution. This overlapping pattern suggests challenges in disentanglement, indicating that this factor is not well separated in the principal component space and may be more intertwined with other factors. Additionally, the dataset includes the attribute Color, but since there was only one color present, there is no need to create a graph for it. According to the plots above, the predicted number of latent dimensions is 5.

MPI3D Toy Visual Analysis

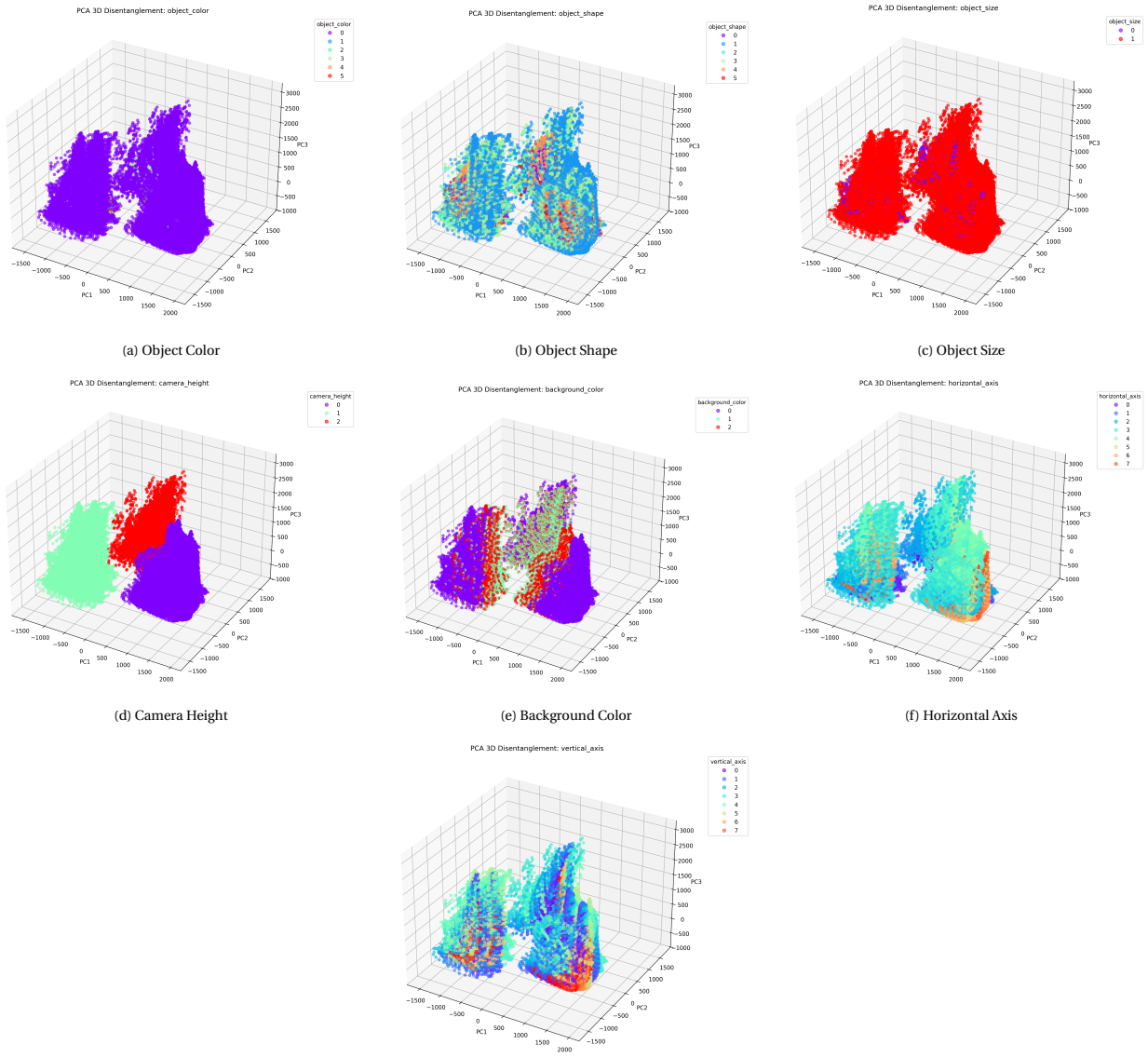


(a) Object Color (b) Object Shape (c) Object Size

(d) Camera Height (e) Background Color (f) Horizontal Axis

(g) Vertical Axis

Figure 3: 2D PCA scatter plots for the MPI3D Toy dataset, illustrating the distribution of data points colored by different generative factors: (a) Object Color, (b) Object Shape, (c) Object Size, (d) Camera Height, (e) Background Color, (f) Horizontal Axis, and (g) Vertical Axis. Each plot visualizes the first two principal components, highlighting the degree of linear separability achieved by PCA for each factor.



(a) Object Color

(b) Object Shape

(c) Object Size

(d) Camera Height

(e) Background Color

(f) Horizontal Axis

(g) Vertical Axis

Figure 4: 3D PCA scatter plots for the MPI3D Toy dataset, illustrating the distribution of data points colored by different generative factors: (a) Object Color, (b) Object Shape, (c) Object Size, (d) Camera Height, (e) Background Color, (f) Horizontal Axis, and (g) Vertical Axis. Each plot visualizes the first three principal components, highlighting the degree of linear separability achieved by PCA for each factor.

In the visualizations presented in Figures 3d and 4d, the attribute of Camera Height demonstrates distinct separability within the principal component space. This clear separation indicates a strong level of disentanglement, meaning that this factor is well distinguished in the reduced dimensional space provided by PCA.

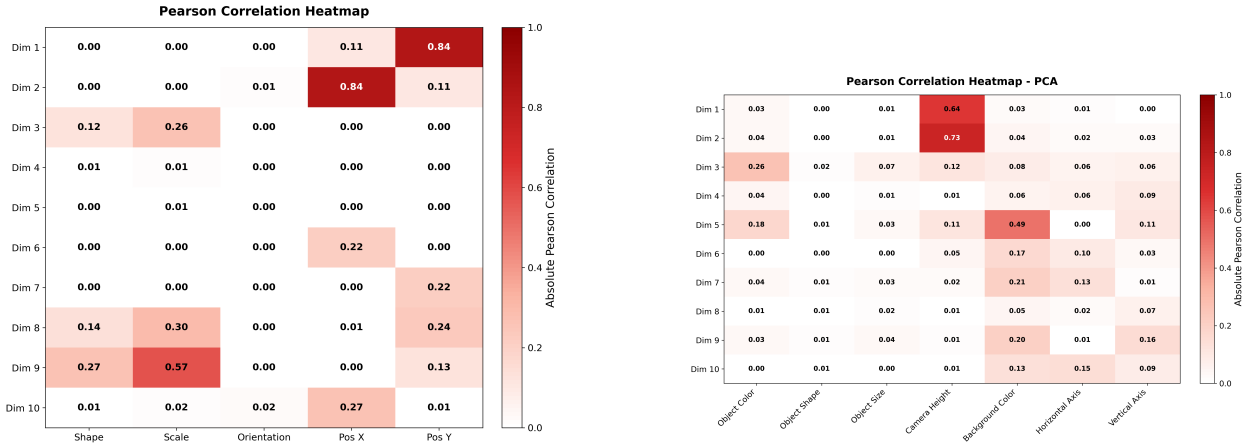
Conversely, Background Color and Object Color, as depicted in Figures 3e, 4e and Figures 3a, 4a respectively, show partial clustering. This suggests that while there is some level of disentanglement, it is not as clearly separated as Camera Height, indicating that Background Color and Object Color are only partially disentangled.

On the other hand, factors such as Horizontal Axis (Figures 3f and 4f), Vertical Axis (Figures 3g and 4g),

Object Shape (Figures 3b and 4b), and Object Size (Figures 3c and 4c) exhibit more complex and overlapping distributions. These overlapping patterns suggest challenges in disentanglement, indicating that these factors are not well separated in the principal component space and may be more intertwined with other factors. According to the plots above, the predicted number of latent dimensions is 3.

4.1.2 Correlation Analysis

Following the visual assessments, we evaluate the disentanglement capabilities of PCA through correlation analysis between principal components and ground truth factors. Figure 5 presents the Pearson correlation heatmaps for both datasets, revealing the extent to which linear dimensionality reduction can capture the underlying generative factors.



(a) dSprites Dataset

(b) MPI3D Toy Dataset

Figure 5: PCA correlation heatmaps showing Pearson correlation coefficients between principal components and ground truth factors. Darker colors indicate stronger correlations, revealing which factors PCA can effectively disentangle.

The heatmap analysis of the dSprites dataset, as shown in Figure 5a, reveals successful disentanglement of four out of five features. Specifically, Position X and Position Y are strongly correlated with Dimension 2 and Dimension 1, each showing a high correlation score of 0.84. Scale is also well disentangled in Dimension 9 with a correlation score of 0.57, while Shape shows a partial disentanglement in Dimension 9 with a score of 0.27. The color feature was not included in the heatmap as it uniformly resulted in a perfect correlation score of 1 due to the lack of variance, making it trivial for disentanglement purposes. The predicted number of latent dimensions is 5, as is also evident from the Figure 1 and Figure 2.

The heatmap analysis of the MPI3D Toy dataset, illustrated in Figure 5b, demonstrates the model's ability to disentangle several key features within the latent space. Specifically, the analysis successfully disentangled three distinct features: Camera Height is well disentangled in Dimension 1 and Dimension 2, with correlation scores of 0.64 and 0.73, respectively. Background Color is disentangled in Dimension 5, showing a correlation score of 0.49. Object Color is partially disentangled in Dimension 3, with a correlation score of 0.26. This analysis highlights the model's capacity to distinguish and isolate these generative factors in the MPI3D Toy dataset. The predicted number of latent dimensions is 2, as is also evident from the Figure 3 and Figure 4.

4.2 Optimal Dimension Selection via Clustering

Following our PCA analysis, we applied k-means clustering to the PCA-transformed data to identify optimal latent dimensions for β-VAE training. Tables 7 and 8 present the clustering quality metrics across different values of k for the dSprites and MPI3D Toy datasets, respectively.

K	Silhouette Score	Calinski-Harabasz Index	Davies-Bouldin Index	Composite Score
2	0.1480	77816.5469	2.2804	0.000000
3	0.1872	89174.6641	1.9017	0.437081
5	0.2438	96575.0547	1.4862	0.871364
7	0.2516	96773.6562	1.3150	**0.950241**
10	0.2497	94858.6875	1.3091	0.912189
15	0.2236	87977.8359	1.1544	0.752663
20	0.2244	80714.1250	1.1456	0.630100

Table 7: Clustering Metrics for Different Cluster Sizes - dSprites Dataset

K	Silhouette	Calinski-Harabasz Index	Davies-Bouldin Index	Composite Score
2	0.4352	641224.3125	1.0002	0.218699
3	0.6277	1385187.5000	0.5747	**1.000000**
5	0.5991	824339.6875	1.1169	0.506506
7	0.5188	668113.8750	1.2693	0.243132
10	0.5090	529342.0625	1.2202	0.204254
15	0.4529	429738.4062	1.2190	0.075050
20	0.4511	367900.7812	1.1656	0.077297

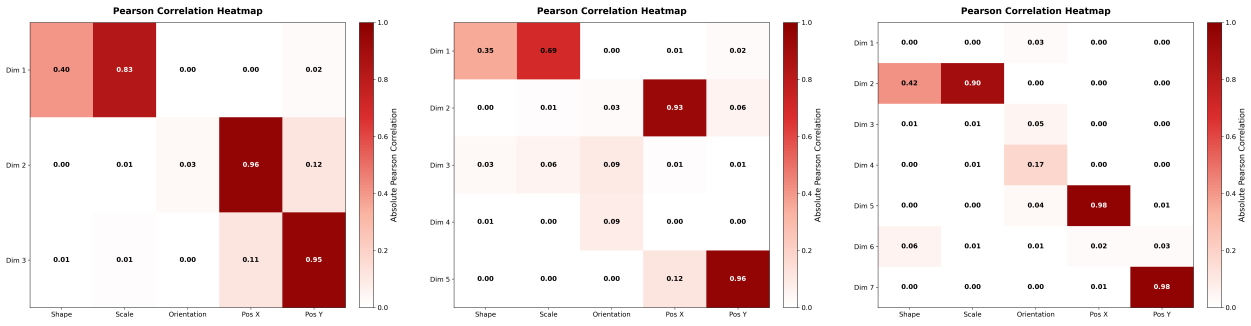
Table 8: Clustering Metrics for Different Cluster Sizes - MPI3D Toy Dataset

After determining the Silhouette [13], Calinski-Harabasz Index [14], and Davies-Bouldin Index scores [14], the composite score [18] was calculated using Algorithm 1 . The clustering analysis reveals distinct optimal points for both datasets. For the dSprites dataset, the metrics converge on an optimal cluster count of 7 (Table 7), with a composite score of **0.950241**. In contrast, the MPI3D Toy dataset suggests 3 clusters (Table 8), with a composite score of **1**. These values serve as the predicted optimal latent dimensions for β-VAE training, representing a principled approach to hyperparameter selection based on the underlying data structure revealed through PCA.

4.3 β-VAE Validation Results

To validate our clustering-based predictions, we trained β-VAE models with different latent dimensionalities (3, 5, and 7) and evaluated their disentanglement performance using correlation analysis between latent representations and ground truth factors.

4.3.1 dSprites β-VAE Results



(a) 3 Latent Dimensions (b) 5 Latent Dimensions (c) 7 Latent Dimensions

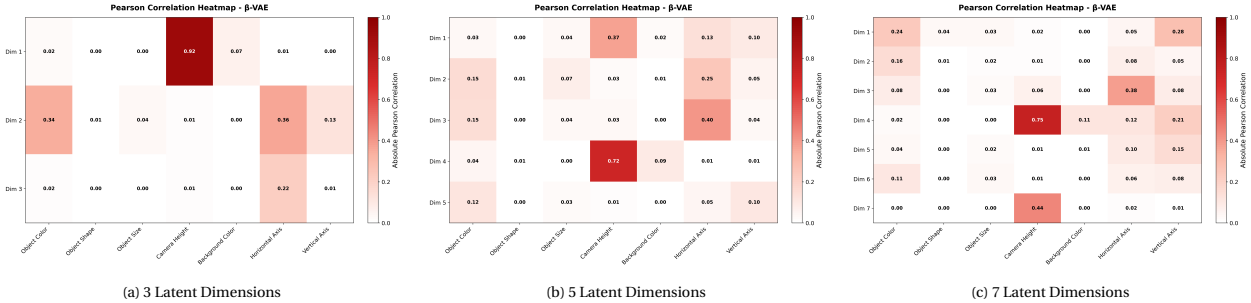
Figure 6: β-VAE correlation heatmaps for dSprites dataset across different latent dimensionalities. The heatmaps show Pearson correlations between learned latent dimensions and ground truth factors.

The β-VAE results for dSprites demonstrate that 7 latent dimensions perform best and strongly validate our clustering prediction. The correlation patterns in Figure 6 reveal that:

- 3 latent dimensions (Figure 6a) show limited disentanglement capability, with only partial separation of spatial factors (Pos X: 0.96, Pos Y: 0.95) and shape-scale conflation (Dim 1 captures both shape: 0.40 and scale: 0.83).
- 5 latent dimensions (Figure 6b) achieve improved disentanglement with cleaner factor separation, particularly for spatial positioning (Pos X: 0.93, Pos Y: 0.96) and better isolation of shape and scale factors.
- 7 latent dimensions (Figure 6c) demonstrate the clearest disentanglement with each factor predominantly captured by a single latent dimension: Pos X (0.98), Pos Y (0.98), and improved separation of shape, scale, and orientation factors.

The progression from 3 to 7 dimensions shows increasingly sparse correlation matrices, indicating better factor isolation and supporting our clustering-based prediction of 7 as the optimal dimensionality.

4.3.2 MPI3D Toy β-VAE Results



(a) 3 Latent Dimensions (b) 5 Latent Dimensions (c) 7 Latent Dimensions

Figure 7: β-VAE correlation heatmaps for MPI3D dataset across different latent dimensionalities. The increased complexity of the MPI3D environment is reflected in the correlation patterns.

For the MPI3D Toy dataset, the β-VAE results show 3 latent dimensions provide the most focused disentanglement, confirming our clustering predictions. The correlation patterns in Figure 7 indicate:

- 3 latent dimensions (Figure 7a) achieve the strongest single-factor correlations, with camera height showing exceptional disentanglement (0.92) and cleaner factor separation overall.
- 5 latent dimensions (Figure 7b) show more distributed correlations across factors, with camera height correlation dropping to 0.72, suggesting over-parameterization begins to dilute factor representations.
- 7 latent dimensions (Figure 7c) exhibit further correlation dilution, with camera height at 0.75 and increased cross-factor interference, indicating that additional dimensions beyond the optimal count introduce noise rather than improved disentanglement.

The MPI3D Toy results demonstrate that the optimal cluster count of 3 identified through our k-means analysis corresponds precisely to the latent dimensionality that achieves the most concentrated and interpretable factor representations.

4.4 Comparative Performance Analysis

Table 9 presents a comprehensive comparison of disentanglement performance across different methods and dimensionalities using established metrics including DCI Disentanglement, DCI Completeness, DCI Informativeness, Explicitness, ZDiff, and CORDIS scores.

Method	**DCI-D**	**DCI-C**	**DCI-I**	**Explicit.**	**ZDiff**	**CORDIS**
			dSprites Dataset			
PCA	0.7317	**0.8706**	0.2785	0.3545	0.6215	0.4528
β-VAE (z=3)	0.645	0.8138	0.4821	0.5107	0.8952	**0.6445**
β-VAE (z=5)	0.6481	0.6784	0.4159	0.4955	0.9047	0.4855
β-VAE (z=7)	**0.7945**	0.7746	**0.5249**	**0.6044**	**0.914**	0.5162
			MPI3D Toy Dataset			
PCA	0.4734	0.1809	**0.1373**	0.4215	0.4818	0.1402
β-VAE (z=3)	**0.6601**	**0.7164**	0.1208	0.2886	0.5249	**0.2931**
β-VAE (z=5)	0.4977	0.287	0.0719	0.3569	0.6159	0.1673
β-VAE (z=7)	0.4848	0.2382	0.1114	**0.4396**	**0.7573**	0.1657

Table 9: Quantitative disentanglement metrics comparison across PCA and β-VAE methods

Analysis for dSprites Dataset

The results for this dataset are as follows:

- **PCA**: This method shows a relatively high DCI Completeness score of 0.8706, indicating that it captures the ground truth factors well in a complete manner. However, its performance in other metrics such as DCI Disentanglement (0.7317) and Informativeness (0.2785) is moderate. The Explicitness and ZDiff scores are also not the highest, suggesting that while PCA can capture complete information, it may not disentangle the factors as effectively as other methods.
- **β-VAE (z=3)**: This configuration shows balanced performance across most metrics but does not excel in any particular one. The CORDIS score of 0.6445 is the highest among all methods for this dataset, indicating robust and consistent disentangled representations.

- **β-VAE (z=5)**: This configuration has moderate scores across all metrics. It does not particularly stand out in any metric, suggesting that while it can capture some aspects of disentanglement, it is not as effective as the 7-dimensional configuration.
- **β-VAE (z=7)**: This configuration shows the highest DCI Disentanglement score (0.7945) and DCI Informativeness score (0.5249), indicating that it effectively separates the underlying factors of variation and predicts the ground truth factors well. It also has the highest Explicitness (0.6044) and ZDiff (0.914) scores, suggesting that each latent dimension is predominantly influenced by one specific factor and that the latent variables are more independent of each other.

The superior performance of the β-VAE model with 7 latent dimensions suggests that it effectively captures the underlying data structure of the dSprites dataset. This finding aligns with research indicating that higher dimensionality can capture more nuanced features [9], as validated in comprehensive metric evaluations [21].

Analysis for MPI3D Toy Dataset

The results for this dataset are as follows:

- **PCA**: This method shows moderate performance across most metrics. The DCI Informativeness score of 0.1373 is the highest among all methods for this dataset, indicating that PCA can predict the ground truth factors relatively well. However, its performance in other metrics such as DCI Disentanglement (0.4734) and Completeness (0.1809) is not as high, suggesting that it may not capture the underlying factors of variation as effectively as other methods.
- **β-VAE (z=3)**: This configuration shows the highest DCI Disentanglement score (0.6601) and DCI Completeness score (0.7164), indicating that it effectively separates the underlying factors of variation and captures the ground truth factors well. It also has the highest CORDIS score (0.2931), suggesting robust and consistent disentangled representations.
- **β-VAE (z=5)**: This configuration shows moderate performance across most metrics. It does not particularly stand out in any metric, suggesting that while it can capture some aspects of disentanglement, it is not as effective as the 3-dimensional configuration.
- **β-VAE (z=7)**: This configuration shows the highest Explicitness score (0.4396) and ZDiff score (0.7573), indicating that each latent dimension is predominantly influenced by one specific factor and that the latent variables are more independent of each other. However, its performance in other metrics such as DCI Disentanglement (0.4848) and Completeness (0.2382) is not as high, suggesting that it may not capture the underlying factors of variation as effectively as the 3-dimensional configuration.

The results for the MPI3D Toy dataset align with the clustering prediction, with the 3-dimensional β-VAE configuration achieving the highest scores in key metrics. This confirmation strengthens the confidence in the methodology, particularly when dealing with more complex data structures where the optimal dimensionality is less immediately apparent [22, 23].

Latent Dimension	Composite Result
dSprites Dataset	
β-VAE (z=3)	0.457820
β-VAE (z=5)	0.087676
β-VAE (z=7)	**0.817262**
MPI3D Toy Dataset	
β-VAE (z=3)	**0.666667**
β-VAE (z=5)	0.172013
β-VAE (z=7)	0.467962

Table 10: Composite Disentanglement Results for β-VAE Models Across Datasets

Using Algorithm 2, the Composite Result scores were derived and are presented in Table 10. These scores align with the analysis of the detailed metrics in Table 9, reinforcing the effectiveness of the clustering-based predictions for optimal latent dimensionality. For the dSprites dataset, the β-VAE model with 7 latent dimensions achieved the highest Composite Result score, outperforming the initial prediction of 5 latent dimensions. This discrepancy suggests that the method may be identifying not just the most obvious factors of variation but also more subtle or complex interactions within the data. This indicates that the approach is sensitive enough to detect nuanced patterns that might be missed by simpler analyses. In contrast, the MPI3D Toy dataset results align perfectly with the predictions, with the 3-dimensional configuration achieving the highest score, further validating the methodology.

5 Conclusion

This research has demonstrated the effectiveness of a novel two-stage pipeline that leverages Principal Component Analysis (PCA) and k-means clustering metrics to determine optimal latent dimensions for Beta-Variational Autoencoders (β-VAEs) in disentangled representation learning. Through experiments on the dSprites and MPI3D Toy datasets, we have shown that our approach provides a computationally efficient alternative to exhaustive hyperparameter searches while achieving competitive disentanglement performance.

Our key contributions include demonstrating PCA's effectiveness as a preliminary tool for assessing disentanglement potential and establishing a strong connection between clustering quality metrics and optimal latent dimensionality for β-VAE models. Our composite scoring methodology successfully identified optimal configurations, aligning with superior disentanglement performance. This approach not only reduces computational resources but also accelerates the development of disentangled representation models. Quantitative evaluations using established metrics confirm that models trained with our predicted dimensions achieve competitive or superior performance.

Future research directions include extending evaluations to a broader range of datasets and architectures, exploring nonlinear dimensionality reduction techniques, and developing adaptive clustering metrics. In conclusion, our pipeline represents a significant step toward making disentangled representation learning more accessible and efficient, contributing to the broader goal of developing robust methods for unsupervised feature disentanglement.

References

[1] Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., Mohamed, S., and Lerchner, A., "β-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework," in *Proc. Int. Conf. on Learning Representations (ICLR)*, 2017. https://www.cs.toronto.edu/~bonner/courses/2022s/csc2547/papers/generative/disentangled-representations/beta-vae,-higgins,-iclr2017.pdf

[2] Burgess, C. P., Higgins, I., Pal, A., Matthey, L., Watters, N., Desjardins, G., and Lerchner, A., "Understanding Disentangling in β-VAE," *arXiv preprint arXiv:1804.03599*, 2018. arXiv:1804.03599.

[3] Kim, H. and Mnih, A., "Disentangling by Factorizing," in *Proc. Int. Conf. on Machine Learning (ICML)*, 2018. arXiv:1802.05983.

[4] Chen, T. Q., Li, X., Grosse, R., and Duvenaud, D., "Isolating Sources of Disentanglement in Variational Autoencoders," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018. arXiv:1802.04942.

[5] Hosoya, H., "Group-Based Learning of Disentangled Representations with Generalizability for Novel Contents," *IJCAI*, 2019. arXiv:1809.02383

[6] Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., & Winther, O. "Ladder Variational Autoencoders," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 29, 2016. arXiv:1602.02282

[7] Li, H., Wang, X., Zhang, Z., Yuan, Z., Li, H., & Zhu, W. "Disentangled Contrastive Learning on Graphs," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021. https://proceedings.neurips.cc/paper/2021/hash/b6cda17abb967ed28ec9610137aa45f7-Abstract.html

[8] Preechakul, K., Chatthee, N., Wizadwongsa, S., and Suwajanakorn, S., "Diffusion Autoencoders: Toward a Meaningful and Decodable Representation," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pp. 10619–10628, 2022. arXiv:2111.15640

[9] Eastwood, C. and Williams, C., "A Framework for the Quantitative Evaluation of Disentangled Representations," *ICLR*, 2018. https://openreview.net/forum?id=By-7dz-AZ

[10] Ridgeway, K. and Mozer, M., "Learning Deep Disentangled Embeddings with the F-Statistic Loss," *NeurIPS*, 2018. arXiv:1802.05312

[11] Locatello, F., et al., "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations," *ICML*, 2019. arXiv:1811.12359

[12] Abid, A., Zhang, M. J., Bagaria, V. K., & Zou, J., "Exploring Patterns Enriched in a Dataset with Contrastive PCA," *Nature Communications*, vol. 9, p. 2134, 2018, doi: 10.1038/s41467-018-04608-8.

[13] Rousseeuw, P., "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Computational and Applied Mathematics*, 1987, doi: 10.1016/0377-0427(87)90125-7

[14] Halkidi, M., Batistakis, Y., & Vazirgiannis, M., "On Clustering Validation Techniques," *Journal of Intelligent Information Systems*, vol. 17, no. 2–3, pp. 107–145, 2001, doi: 10.1023/A:1012801612483.

[15] Matthey, L., Higgins, I., Hassabis, D., & Lerchner, A., "dSprites: Disentanglement testing Sprites dataset," DeepMind, 2017, https://github.com/deepmind/dsprites-dataset

[16] M. W. Gondal, M. Wüthrich, D. Miladinović, F. Locatello, M. Breidt, V. Volchkov, J. Akpo, O. Bachem, B. Schölkopf, and S. Bauer. On the transfer of inductive bias from simulation to the real world: A new disentanglement dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[17] Ubisoft. (2021). Ubisoft-Laforge Disentanglement Metrics. GitHub repository. Retrieved from https://github.com/ubisoft/ubisoft-laforge-disentanglement-metrics

[18] M. Z. Naser and A. Z. Naser. SPINEX-clustering: Similarity-based predictions with explainable neighbors exploration for clustering problems. *Cluster Computing*, 28:335, 2025. https://doi.org/10.1007/s10586-024-04981-8.

[19] Scikit-learn Developers, "IncrementalPCA - Online principal component analysis," *Scikit-learn Documentation*, 2023. https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.IncrementalPCA.html

[20] V. Lippi and G. Ceccarelli. Incremental Principal Component Analysis: Exact implementation and continuity corrections. *arXiv preprint arXiv:1901.07922*, 2019. https://arxiv.org/abs/1901.07922. Matlab implementation: https://www.mathworks.com/matlabcentral/fileexchange/69844-incremental-principal-component-analysis.

[21] M.-A. Carbonneau, J. Zaidi, J. Boilard, and G. Gagnon, "Measuring Disentanglement: A Review of Metrics," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 35, no. 7, pp. 3747–3760, Jul. 2024. doi: 10.1109/TNNLS.2022.3218982.

[22] Tokui, S., and Sato, I., "Disentanglement Analysis with Partial Information Decomposition," *arXiv preprint arXiv:2108.13753*, 2021. arXiv:2108.13753

[23] Julka, S., Wang, Y., and Granitzer, M., "Towards an Improved Metric for Evaluating Disentangled Representations," *arXiv preprint arXiv:2410.03056*, 2024. arXiv:2410.03056