RELATIVE REPRESENTATION EVALUATIONS AND COMPARISONS OF AUTOENCODERS AND VISUAL TRANSFORMERS

A further look into the paper **"Relative representations enable zero-shot latent space communication."** by Moschella, Luca, et al.

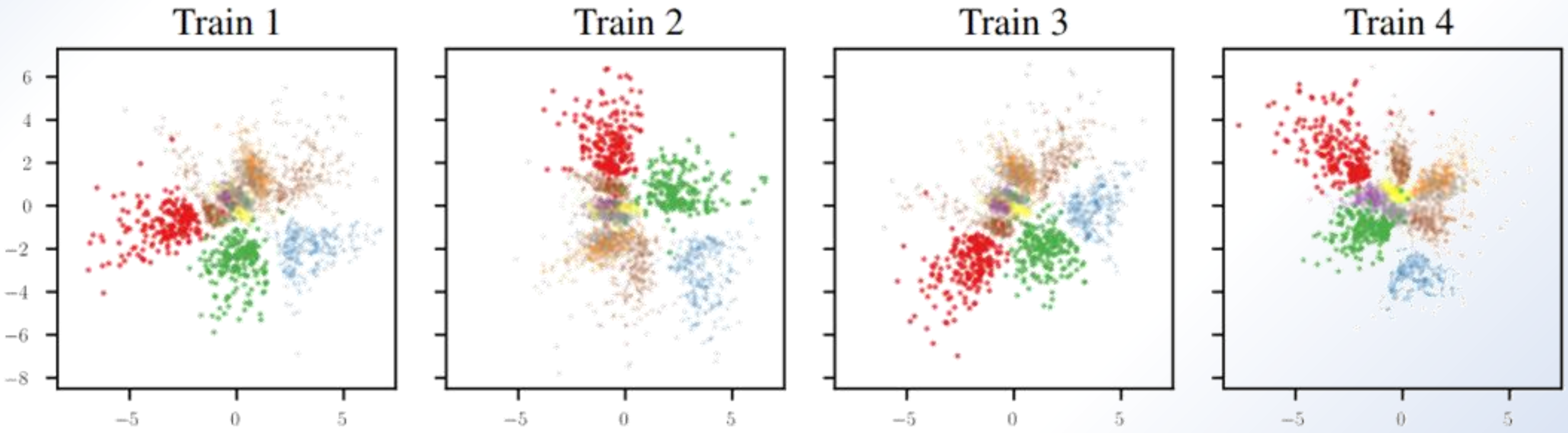
Michele SCOMINA

Models, like Autoencoders and Transformers, transform high dimensional data into a meaningful representation they can use to solve tasks. These learned representations depend on the initial state and hyperparameters of the given model.

- **Is there a meaningful way of comparing different learned representations?**
- **If there is, to what extent and to what architectures can this representation be used for?**

AUTOENCODERS

- Autoencoders' representations of a particular dataset reconstruction are **intrinsically similar**.
- They are extrinsically the same after an isometric correction.



COSINE SIMILARITY REPRESENTATION

$$r_{x^{(i)}} = (S_c\left(e_{x^{(i)}}, e_{a^{(1)}}\right), S_c\left(e_{x^{(i)}}, e_{a^{(2)}}\right), \dots, S_c\left(e_{x^{(i)}}, e_{a^{(|\mathbb{A}|)}}\right))$$

…where:

- $\mathbb{A}$ is a set of pre-defined anchor points from the dataset, used to build the representation. $a^{(n)} \in \mathbb{A} \;\; \forall n$
- $e_{x^{(i)}}, e_{a^{(n)}}$ are the input and n-th anchor representations in latent space respectively.
- $S_c(\boldsymbol{a}, \boldsymbol{b}) = \frac{\boldsymbol{ab}}{||\boldsymbol{a}|| * ||\boldsymbol{b}||} = cos\ \theta$, where θ is the angle between the two vectors.

Invariant representation to relative rotations!

EVALUATION METRICS

The following metrics have been used to compare the representations:

- **Cosine Similarity Index:**

$$\textbf{Cosine}(\mathrm{s}) = \frac{f_{\mathbb{X}}(s) \cdot f_{\mathbb{Y}}(s)}{\|f_{\mathbb{X}}(s)\| \|f_{\mathbb{Y}}(s)\|}$$

- **Jaccard Index:**

$$\textbf{Jaccard}(\mathrm{s}) = \frac{|\,\mathrm{KNN}_k^{\mathbb{X}}(f_{\mathbb{X}}(s)) \cap \mathrm{KNN}_k^{\mathbb{Y}}(f_{\mathbb{X}}(s))|}{|\,\mathrm{KNN}_k^{\mathbb{X}}(f_{\mathbb{X}}(s)) \cup \mathrm{KNN}_k^{\mathbb{Y}}(f_{\mathbb{X}}(s))|}$$

Where $\mathbb{X}$, $\mathbb{Y}$ are source and target space respectively, and $f_{\mathbb{X}} : \mathbb{S} \to \mathbb{X}$, $f_{\mathbb{Y}} : \mathbb{S} \to \mathbb{Y}$ are the encoding functions.

ARCHITECTURES AND DATASETS

The type of models used in this project are the following:

- **Convolutional Autoencoder** (with variable bottleneck size)
- **Visual Transformer**

The architectures are then trained on one of 4 datasets as reconstruction models:

- **MNIST**
- **kMNIST**
- **FashionMNIST**
- **CIFAR-10**

HYPERPARAMETERS

Optimizer: Adam

Loss function: Mean Square Error (MSE)

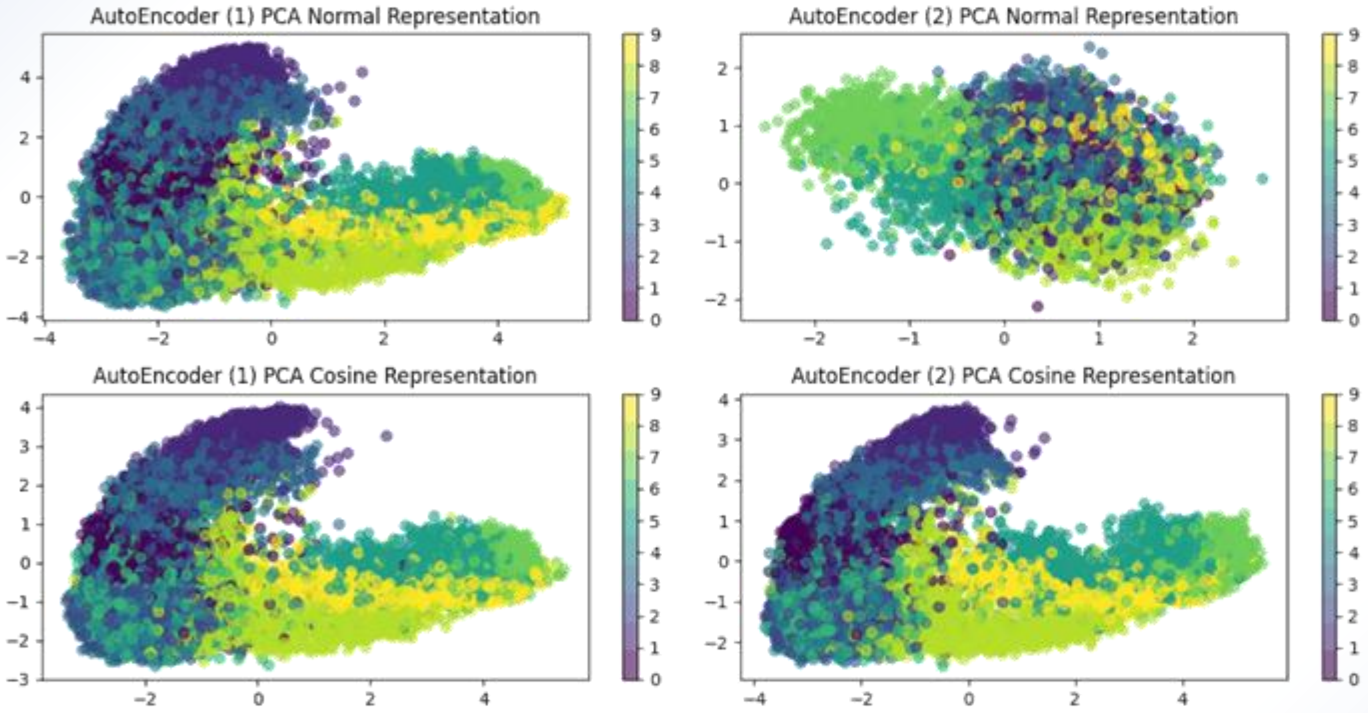
Number of anchors: 30 per class

Number of k-neighbours for Jaccard index: 10

Regularization: Dropout (0.5), Batch Normalization, Early Stopping

RESULTS

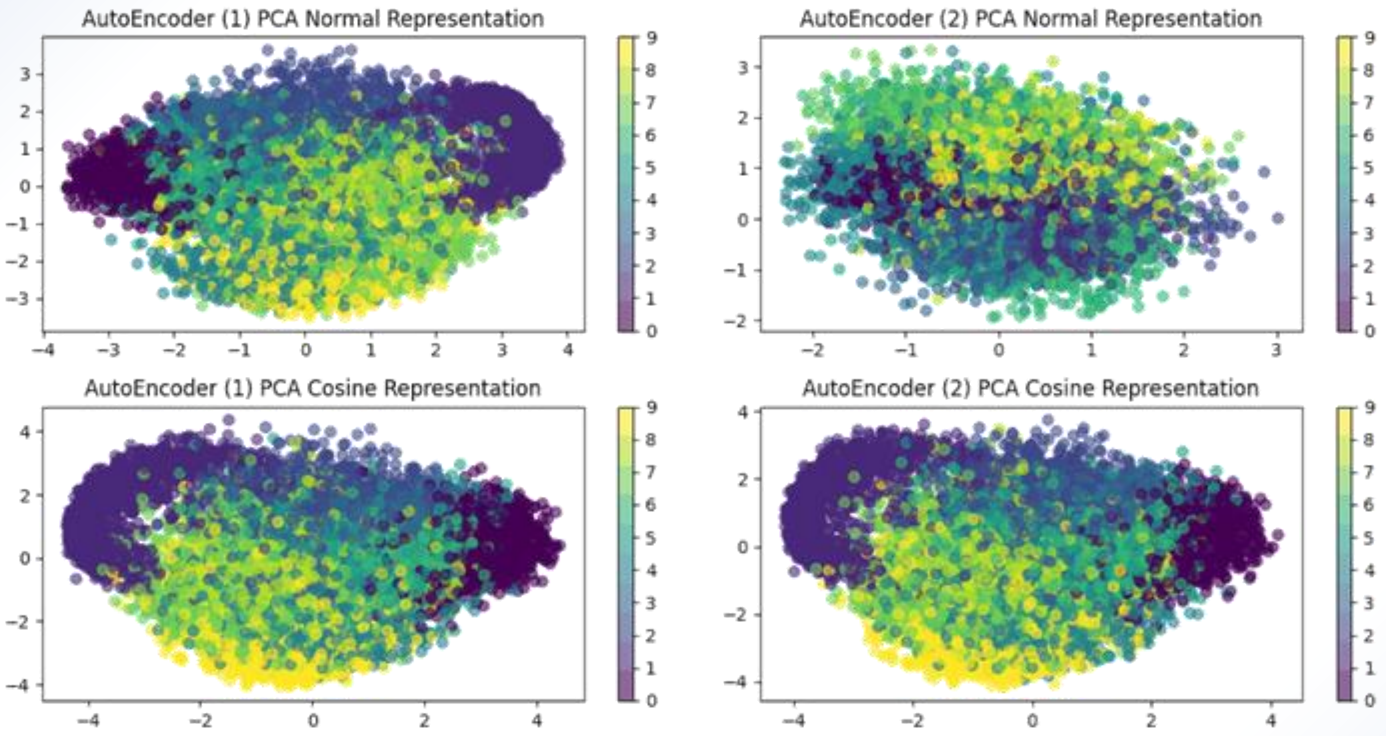
2 Autoencoders – Bottleneck of size 48 - FashionMNIST



	Average Cosine(s) Index	Average Jaccard(s) Index
Normal	0.0747	0.0008
Cosine	0.8285	0.2202
Cosine/Normal	**11.09**	**271.85**

RESULTS

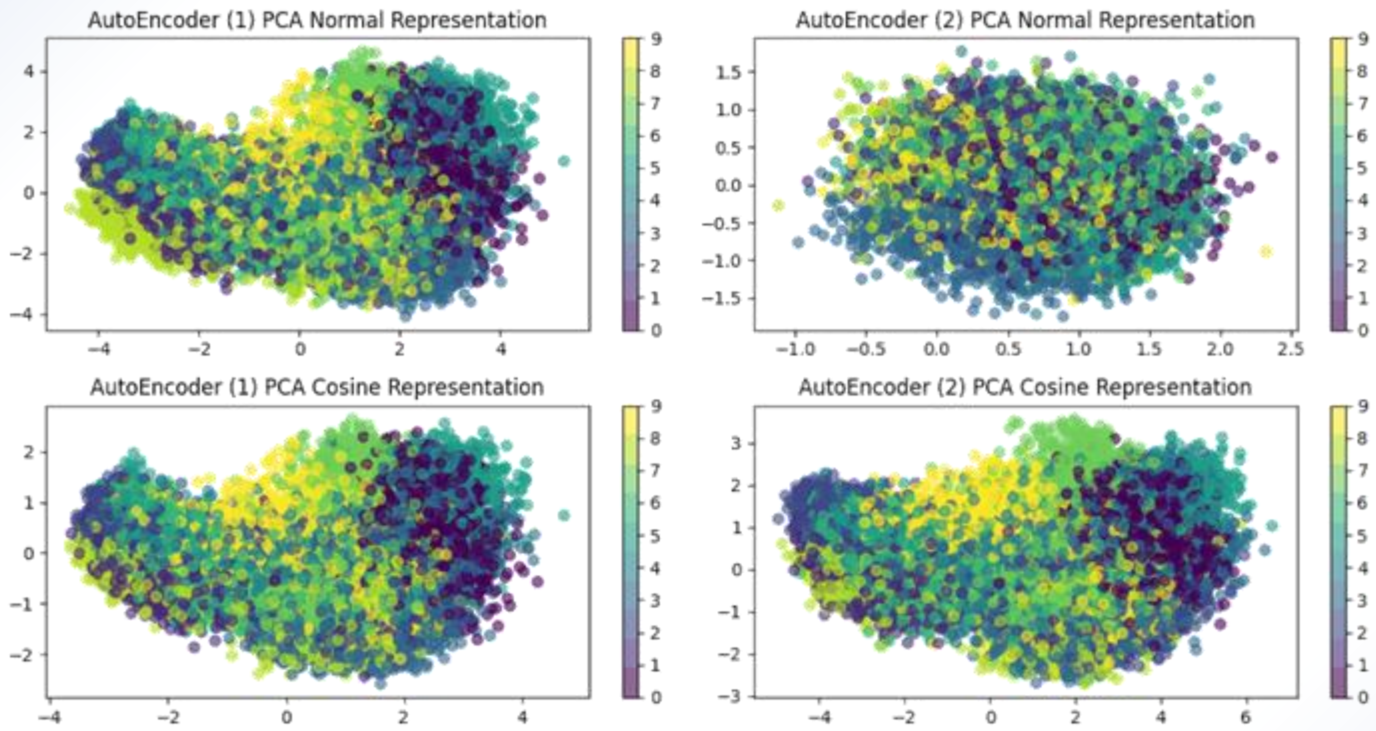
2 Autoencoders – Bottleneck of size 24 - MNIST



	Average Cosine(s) Index	Average Jaccard(s) Index
Normal	-0.0364	0.0002
Cosine	0.5575	0.5187
Cosine/Normal	**15.33**	**2247.46**

RESULTS

2 Autoencoders – Bottleneck of sizes 48,24 – kMNIST

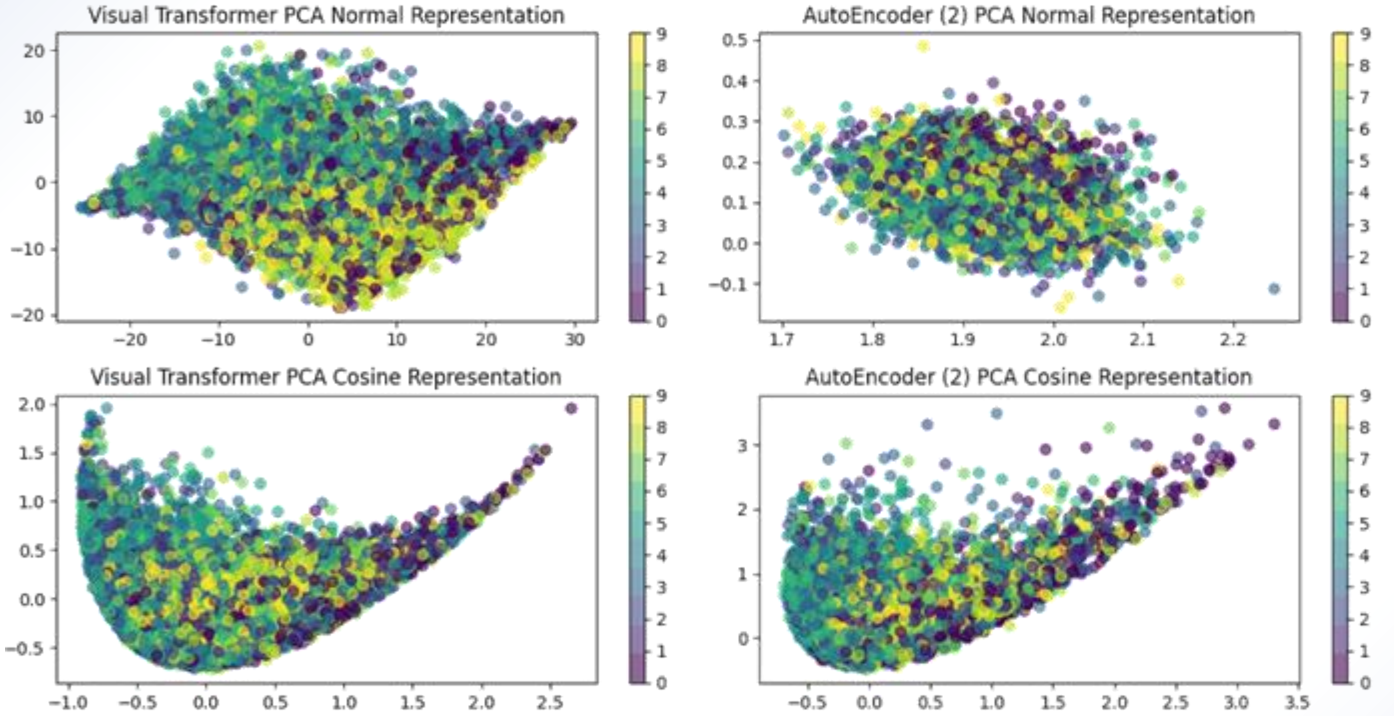


	Average Cosine(s) Index	Average Jaccard(s) Index
Normal	N/A	0.0005
Cosine	0.2642	0.0714
Cosine/Normal	**N/A**	**155.99**

Different scales in Cosine representation when the bottlenecks have different dimensionalities!

RESULTS

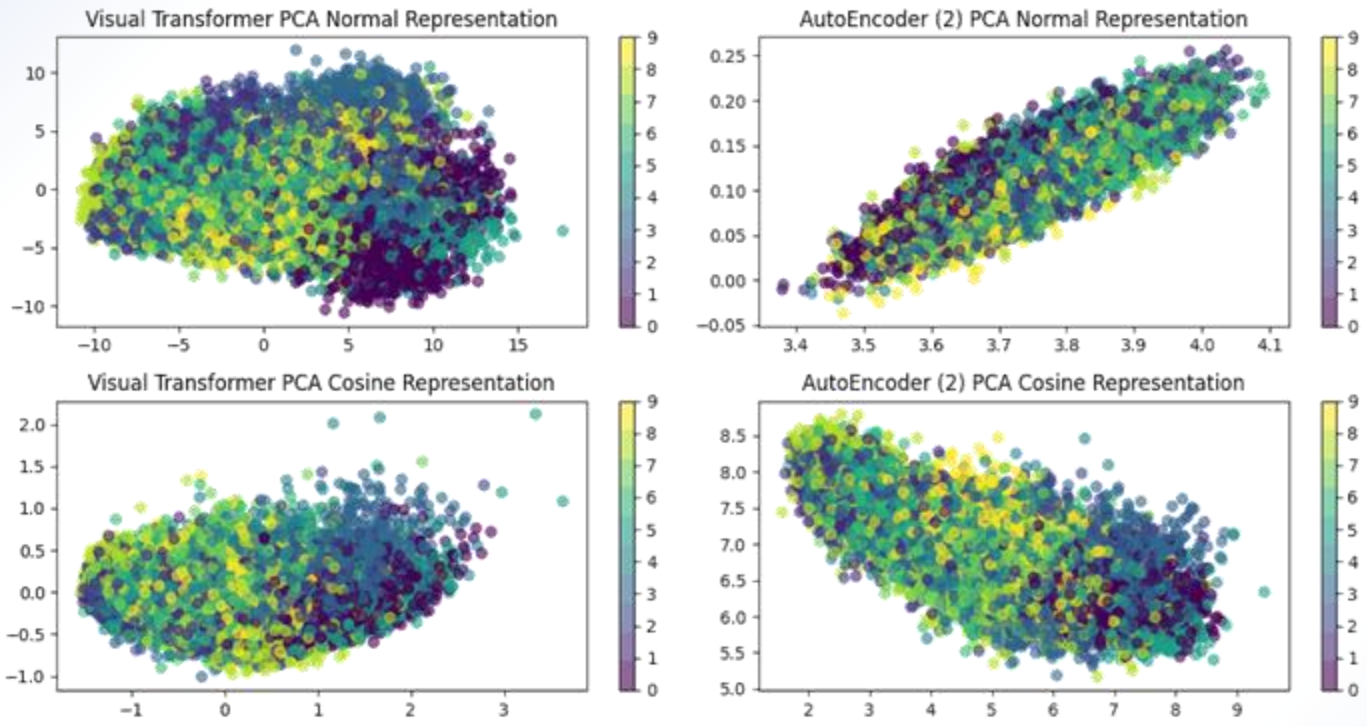
Visual Transformer and Autoencoder – Bottleneck of size 72 – CIFAR10



	Average Cosine(s) Index	Average Jaccard(s) Index
Normal	N/A	0.0002
Cosine	0.9983	0.0263
Cosine/Normal	**N/A**	**133.99**

RESULTS

Visual Transformer and Autoencoder – Bottleneck of size 48 – kMNIST



	Average Cosine(s) Index	Average Jaccard(s) Index
Normal	N/A	0.0005
Cosine	0.7779	0.0008
Cosine/Normal	**N/A**	**1.61**

CONCLUSIONS

- Autoencoders’ latent spaces are comparable through a rotationally invariant representation.
- Autoencoders of different latent space dimensionalities scale with respect to the number of dimensions.
- Transformers and Autoencoders do share inconsistent similarities, further research is required in order to have a definitive answer regarding the possibility of alternative representations that might fit the models better.