Disentangled Representation of Data Distributions in Scatterplots

Jaemin Jo[*] Jinwook Seo[†]

Department of Computer Science and Engineering, Seoul National University

Abstract

We present a data-driven approach to obtain a disentangled and interpretable representation that can characterize bivariate data distributions of scatterplots. We first collect tabular datasets from the Web and build a training corpus consisting of over one million scatterplot images. Then, we train a state-of-the-art disentangling model, β-variational autoencoder, to derive a disentangled representation of the scatterplot images. The main output of this work is a list of 32 representative features that can capture the underlying structures of bivariate data distributions. Through latent traversals, we seek for high-level semantics of the features and compare them to previous human-derived concepts such as scagnostics measures. Finally, using the 32 features as an input, we build a simple neural network to predict the perceptual distances between scatterplots that were previously scored by human annotators. We found Pearson's correlation coefficient between the predicted and perceptual distances was above 0.75, which indicates the effectiveness of our representation in the quantitative characterization of scatterplots.

Index Terms: Human-centered computing—Visualization—Visualization theory, concepts and paradigms

1 Introduction

We aim to identify interpretable latent factors in data distributions of scatterplot images. Scatterplots are a core visualization technique in exploratory visual analysis as they give a succinct overview about the relationship between two quantitative variables. Among various visual queries one can perform on scatterplots, one of the most useful low-level tasks is to characterize its data distribution [3, 24]. Such characterization can be done either qualitatively (e.g., "As one variable increases, the other also increases") or quantitatively (e.g., "the Pearson correlation coefficient r is about 0.8"). Once done, it can be used to facilitate communication between people or interpretation of others, for example, by providing a caption. Especially, quantitative characterization enables useful queries to speed up the visual exploration of scatterplots, such as clustering similar scatterplots to reduce the number of scatterplots to inspect, or finding the most similar scatterplot to a target scatterplot.

To enable such characterization, a body of studies have attempted to devise features that capture interesting structures in the data distribution of a scatterplot. For example, consisting of nine hand-engineered measures with each ranging from zero to one, graph-theoretic scagnostics [28] is designed to capture the presence of specific structures in a scatterplot, such as *Outlying*, *Skewed*, *Clumpy*, *Convex*, *Skinny*, *Striated*, *Stringy*, *Straight*, and *Monotonic*. Using the measures, one can computationally characterize a scatterplot by describing it as a scagnostics vector of nine dimensions. Such vectors can be later used to efficiently approximate the similarity between two scatterplots; for example, ScagExplorer [7] curates representative scatterplots created by clustering similar scatterplots, considering the L_2 distance between two scagnostics vectors as the distance between the corresponding scatterplots.

However, one limitation of scagnostics measures is its coverage; the nine hand-engineered features may not be enough to faithfully capture interesting structures in a multitude of scatterplots. Moreover, the measures can be correlated, leaving a potential bias in similarity measurement. Indeed, in our corpus consisting of more than one million scatterplots, we could find a strong correlation between scagnostics measures (e.g., Pearson's $r_{Striated,Stringy} > 0.92$), which implies scagnostics tends to give a high priority to a specific structure. Consequently, the distance between two scagnostics vectors does not guarantee to reflect the actual distance perceived by a human, as a follow-up study [21] suggested that the L_2 distance between scagnostics vectors only explains small variance of the perceived distance (Pearson's $r_{predicted,perceived} < 0.26$).

Acknowledging the limitations of scagnostics, researchers have attempted to identify the structures that affect similarity judgements by observing how people actually cluster similar scatterplots. For example, Pandey et al. [21] asked 18 participants to cluster 247 representative scatterplots and identified six important terms that people used to describe the clusters: *Density*, *Orientation*, *Spread*, *Regularity*, *Grouping*, and *Edges*. Nonetheless, these terms are explanatory but not predictive as they are not sufficient to quantify the similarity between two unseen scatterplots.

We postulate the following three requirements for an effective representation of a scatterplot:

1. **Predictive:** The representation can be used to predict the perceptual similarity between two unseen scatterplots as scagnostics measures do (with a limited accuracy).
2. **Interpretable:** The representation should remain explainable, seeking for trustworthiness and reliability.
3. **Generalized:** The representation should be designed to faithfully capture the diverse structures of real data distributions.

To find a generalized and factorized representation, we present scatterplot images of a variety of real data distributions to a machine agent. The machine agent learned a disentangled representation of the data distributions without supervision. As a training dataset, we collect tabular datasets from the Web and build a large corpus of normalized scatterplot images which has about 1.1 million scatterplots. By traversing the latent space that the agent learned, we seek the interpretation of each latent factor that could possibly be overlooked in previous studies. Finally, we present a simple neural network with one hidden layer that can predict the human-annotated distances obtained from a previous study [21] with a high correlation coefficient (Pearson's $r_{predicted,perceived} > 0.75$).

2 Related Work

Our work stands between the visualization and machine learning fields. On the visualization side, we cover previous work for modeling the perceptual similarity between scatterplots. Then, from a machine learning perspective, we elaborate on the recent advances in finding a disentangled representation of images.

2.1 Perceptual Similarity between Scatterplots

Inspired by scagnostics by John and Paul Tukey [26], Wilkinson et al. [28] realized nine graph-theoretic scagnostics measures that are

[*] e-mail: jmjo@hcil.snu.ac.kr

[†] e-mail: jseo@snu.ac.kr

computationally efficient enough to be used in practice. Scagnostics has proven its effectiveness in follow-up studies and extended to various data types. For example, ScagExplorer [7] facilitated the exploration of large scatterplot matrices (SPLOM) by clustering similar scatterplots based on the distance between their scagnostics vectors. Scagnostics has been extended to three-dimensional spaces [9] and high-dimensional time series [6]. Researchers also attempted to understand the characteristics of scagnostics; for example, Wilkinson et al. [29] demonstrated the distribution of each scagnostics measure from 1,000 synthetic scatterplots. More recently, Dang et al. [8] presented a method to revealing hidden structures in data by inspecting changes in scagnostics measures after scale transformations.

Despite the widespread use of scagnostics, it is also found that the distance between scagnostics vectors has a weak correlation with the distance people actually perceive [21]. To understand factors that affect similarity perception, Pandey et al. [21] identified six important concepts, such as *Density* and *Orientation*, which people commonly used to describe similar scatterplots. Based on human annotators' perception on similarity, Ma et al. [19] presented a subjective similarity model that can recommend scatterplots perceptually similar to a target using deep neural networks. Finally, Matute et al. [20] presented skeleton-based scagnostics consisting of two novel measures D_H and D_F, based on the Hausdorff and Fréchet distances respectively, and showed their measures better predict the perceptual distance between 29 test scatterplot images.

Nonetheless, we found that studies meeting our three requirements in the introduction are rare. For example, scagnostics [28] is interpretable, but it is unknown to what extent it covers general structures of data distributions. The six important concepts in similarity judgements [21] are interpretable but not predictive as they cannot quantify the distance between scatterplots. ScatterNet [19] is predictive and generalized, but it is still challenging to explain what structural features were captured, since the features were not disentangled well. Finally, skeleton-based scagnostics [20] leaves a question on its generalizability due to the limited number of scatterplots tested in a user study. Seeking for an interpretable and generalized representation, we use a β-variational autoencoder (β-VAE) [10], a deep neural network that demonstrated the state-of-the-art disentangling performance, on a training corpus consisting of over one million scatterplot images. We also test the prediction performance of our model by modeling scatterplot similarities from a previous study [21].

2.2 Disentangled Representation

From a machine learning perspective, our problem can be seen as finding independent generative factors of scatterplot images. This problem is often called *disentangling* the factors of variation in observations (in our case, scatterplot images) [4]. Let a vector $\mathbf{z}$ denote a disentangled representation of a scatterplot image $\mathbf{x}$. If $\mathbf{z}$ is a successful disentangled representation, changing the value of one dimension of $\mathbf{z}$ (e.g., the first dimension of $\mathbf{z}$, $\mathbf{z}_1$) will result in changes in $\mathbf{x}$ by a single generative factor, keeping other factors relatively invariant [10]. For example, assume that we have obtained a disentangled representation of MNIST handwritten digits [18]. Changing one dimension of $\mathbf{z}$ (e.g., $\mathbf{z}_1$) may generate digits with different rotations but with a consistent stroke width. Similarly, changing another dimension (e.g., $\mathbf{z}_2$) may only change the stroke width of digits, keeping the rotations invariant.

A Generative adversarial network (GAN) is a common and useful method to learning a disentangled representation of images. The original GAN consists of two deep neural networks, a generator G and a discriminator D. G learns to map a latent feature vector $\mathbf{z}$ to a realistic input image so as to deceive its adversarial network D. Many variants of GAN have been suggested to capture more disentangled and interpretable factors of input images; for example, InfoGAN [5] modified the original architecture by rewarding the mutual information between the observations and a subset of latent factors. Deep convolutional inverse graphics network (DC-IGN) is another semi-supervised approach to learn a disentangled representation of data [17]. DC-IGN encourages neurons to learn specific graphical transformation by presenting mini-batches of data corresponding to changes in only one scene variable (e.g., azimuth of faces). In this paper, we chose β-VAE for its high disentanglement score [10] and capability of unsupervised training.

Consisting of an encoder g and a decoder f, autoencoders [11] are designed to discover an efficient coding $\mathbf{z}$ to compress and reconstruct the original input $\mathbf{x}$ in an unsupervised way. Instead of directly encoding $\mathbf{x}$ as $\mathbf{z}$, a variational autoencoder (VAE) [16] learns to compute the mean and variance of $\mathbf{z}$, $\mu(\mathbf{z})$ and $\sigma^2(\mathbf{z})$, for a stable and scalable training process. More recently, β-VAE [10] introduced a hyperparameter β that controls the trade-off between the capacity of reconstruction and learning statistically independent latent factors, showing the state-of-the-art disentangling performance comparable to other recent models, such as InfoGAN [5] and DC-IGN [17]. With this encouraging result, we chose to use β-VAE to elicit a disentangled representation from scatterplot images.

To understand the learned representation, it is necessary to interpret the meaning of the latent space found. A variety of quantitative measures for assessing the quality of disentanglement have been presented [10]. However, we are more interested in deriving high-level semantics of each dimension in the latent space as in scagnostics. One popular method is to inspect latent traversals [14]; we will visualize a series of scatterplot images reconstructed from a latent code $\mathbf{z}$ by gradually changing its value at only one dimension (e.g., $\mathbf{z}_1$) while fixing the values at the others (e.g., $\mathbf{z}_{[2,dim(\mathbf{z})]}$). Then, we inspect the reconstructed scatterplot images (from the manipulated codes) to see the meaning of the dimension. We believe our semantics garnered by latent traversals is more interpretable than a previous deep neural network [19], but a more systemic approach to pinpointing the meaning would be possible in the future.

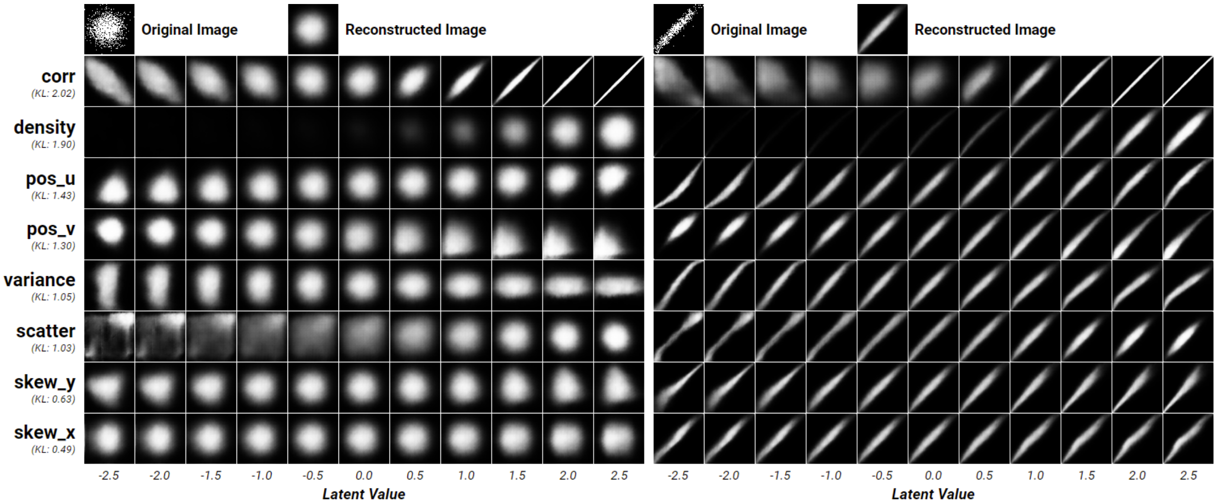
3 Disentangled Representation of Scatterplots

We elaborate on the data collection and training process of our work as well as the rationale behind the choice of a machine agent.

3.1 Data Collection

Scatterplot images on the Web are often noisy due to the use of extra visual elements, such as annotations, and multiple visual channels such as colors of points. To control such variation, we chose to collect raw data from the Web and convert them into "normalized" scatterplots. We used 2,101,990 datasets from the UCI Machine Learning repository [2] and a previous study [12] that collected JSON files from a public visualization gallery, Plotly Feed [1]. To identify quantitative fields in data to be shown in a scatterplot, we use the following criteria: a field is considered quantitative if all values in the field can be converted to floating point numbers with more than 20 unique values, ensuring not to include a categorical field with numeric class names. We discarded datasets that did not have a pair of quantitative fields (about 64% of the datasets) and had more than 30 quantitative fields (about 1.9%), since they could be generated by machines and result in a large number of synthetic scatterplots. After removing datasets that were exact duplicates of each other, we obtained 477,177 unique datasets.

We combined all possible pairs of the quantitative fields in each dataset to generate 64 × 64 single-channel black-and-white scatterplot images. For each pair of fields, rows with missing values on either field were discarded. We computed the minimum and maximum values for each of the two fields and used them to compute the position of a data point in a scatterplot. To avoid distracting the machine agent by visual embellishments, we plotted each data point as an 1 × 1 pixel without axes, legends, etc. A pixel was marked as white if at least one data point belonged to the pixel. Otherwise, it



(a) Gaussian

(b) Linear

Figure 1: **Manipulating latent representations of two scatterplot images.** Among the 32 latent factors, we selected the top eight most informative factors with the largest Kullback-Leibler divergence to a unit normal distribution. We visualize the effect of the factors on the output images as their values vary from -2.5 to 2.5.

remained as black. Note that such a policy may outweigh outlying points than multiple points mapped to the same pixel, since it does not consider their density. Through the data collection procedure, we could obtain a corpus of 1,189,038 unique scatterplot images.

3.2 Model Selection and Implementation

As a machine agent, we used β-VAE [10] for its disentangling performance and stability in training. We initially tested several recent models, such as DCGAN [23], InfoGAN [5], and DC-IGN [17]. However, with our corpus, we found the GAN-based models suffered from the mode collapse problem [25] where the generator learns to generate only a few modes of scatterplot images neglecting many other modes. DC-IGN requires a semi-supervised training procedure where input images differ in only one generative factor. However, in our case, we do not know a priori generative factors underlying scatterplot images, and even if we knew, it would be infeasible to gather images with only one factor varying due to the sheer size of our corpus.

β-VAE consists of two deep neural networks, an encoder g and a decoder f. The encoder transforms a scatterplot image, a $64 \times 64 \times 1$ tensor where the last dimension denotes the channel of the image (i.e., black-and-white, 0.0 for black and 1.0 for white), to a latent representation $\mathbf{z}$ of lower dimensionality by estimating its mean and variance in each dimension, i.e., $\mu(\mathbf{z})$ and $\sigma^2(\mathbf{z})$. The decoder learns to reconstruct $\mathbf{z}$ back to a scatterplot image as similarly as possible to the original image (i.e., minimizing the difference between $\mathbf{x}$ and $f(g(\mathbf{x}))$). Since β-VAE internally projects input images on a lower-dimensional space, the latent representation $\mathbf{z}$ is expected to capture important features in characterizing the images.

β-VAE introduces a hyperparameter β that controls the trade-off between reconstruction capacity and the extent of disentanglement. Inspired by a previous study [10], we set the value of β to 4 and used 32 latent dimensions, i.e., $\mathbf{z} \in \mathscr{R}^{32}$. We also employed a network architecture similar to the previous study [10] as shown in Table 1. We implemented the model on the PyTorch framework [22] and trained it using an Adam optimizer [15] with a learning rate of 2e-3. The source codes for this work are available at `https://github.com/jaeminjo/Disentangling-Scatterplots`.

4 Results and Discussion

After the training process, we can use the encoder of β-VAE to map a scatterplot image to a lower-dimensional latent representation $\mathbf{z}$ of 32 dimensions. The representation can be seen as a feature vector that describes important characteristics of the original image. We qualitatively analyze what structures were captured by β-VAE through latent traversals. We also demonstrate such feature vectors can be used to predict the perceptual distances between scatterplots.

4.1 Latent Traversals

Fig. 1 shows the results of latent traversals for two scatterplot images: one has a Gaussian distribution at the center (Fig. 1a) and the other has a strong linear relationship with several scattered outliers (Fig. 1b). In each example, the original and reconstructed images (through β-VAE) are shown on the top. Since the dimensionality of a latent representation $\mathbf{z}$ is much lower than that of the original image, information loss is inevitable; one can see that the details of the input image (i.e., scattered points of Fig. 1b) are lost in the reconstructed one. Among the 32 latent dimensions, we chose the top eight most informative dimensions with the largest Kullback-Leibler divergence to a unit normal distribution (labels on the vertical axis in Fig. 1). We named each dimension by interpreting its effect in reconstructing images; for a representation, we changed the value at each of the eight dimensions over $[-2.5, 2.5]$, decoded the manipulated representation using the decoder of β-VAE, and compared the manipulated image with the original reconstruction.

The most informative feature in characterizing scatterplot images was related to the correlation between the two variables in a scatterplot. On the first row of Fig. 1, one can see that negative values on the first dimension, which we named D_{corr}, generated scatterplots with negative correlation, while positive values produced images with positive correlation. Note that the effect of D_{corr} was not symmetric; large positive values on D_{corr} produced images with almost perfect positive correlation, but we could not produce images with perfect negative correlation. This may be due to the charac-

Table 1: Architecture of β-VAE

Input	64×64×1
Encoder	Conv 32×4×4 (stride 2), 32×4×4 (stride 2), 64×4×4 (stride 2), 64×4×4 (stride 2), FC 256, ReLU activation
Latents	32
Decoder	Deconv reverse of encoder, ReLU, Sigmoid

teristic of datasets on the Web as scatterplots with strong positive correlation were more frequent. The second most informative dimension, $D_{density}$, described the overall density of a scatterplot from the sparsest image to the densest image (see the second row of Fig. 1). Note that $D_{density}$ did not affect the correlation of scatterplots; it is successfully disentangled from D_{corr}.

The next dimensions, D_{pos_u} and D_{pos_v}, affected the position of the "center" of a cluster. D_{pos_u} determined the position on one diagonal axis u (from top-left to bottom-right) while D_{pos_v} affected the position on the other v (from top-right to bottom-left). Although their directions differ from ones people commonly use (i.e., x and y), β-VAE successfully found two orthogonal axes u and v.

Interestingly, the fifth dimension, $D_{variance}$, encoded the relative variance between the x and y axes. For example, a positive value on $D_{variance}$ produced horizontally long distributions where $\sigma^2(X) > \sigma^2(Y)$, while a negative value generated vertically long distributions where $\sigma^2(Y) > \sigma^2(X)$. The next dimension $D_{scatter}$ was related to the area on which data points are distributed. For instance, negative values on $D_{scatter}$ generated data points scattered overall especially with two clusters on the top-right and bottom-left corners, while positive values left the points to gather at the center. The last two dimensions, D_{skew_x} and D_{skew_y}, generated "tails" on distributions, which may be related to statistical skewness [13]. For example, negative values on D_{skew_y} generated images of an inverted triangle shape whose points are skewed to the bottom. D_{skew_x} also left data points skewed to the left (when negative) and to the right (when positive), but the amount of disentanglement was less significant as its lower KL divergence ($KL = 0.49$) suggested. See the supplementary material for latent traversals with more input images.

Compared with previous studies [21, 28], our approach automatically captures important structures in the decreasing order of importance. We could identify common concepts in ours and previous ones; for example, *Density* from Pandey et al. [21] indicates the concentration of data points in a certain region which is relevant to $D_{density}$ and $D_{scatter}$. Similarly, *Orientation* refers to the characterization of trends in scatterplots, which is similar to D_{corr}. However, concepts indicating strong edges in scatterplots (*Striated* in scagnostics and *Edges* in Pandey et al. [21]) were missing in our model. The reason may be that such high-frequency and non-aligned patterns were averaged when convolutional layers extracted features. In the next section, we show how our representation can be complemented by borrowing such missing concepts from scagnostics.

4.2 Perceptual Distance Prediction

As reported in a previous experiment [21], the L_2 distance between two scagnostics vectors does not explain the perceptual distance between scatterplots well (Pearson's $r < 0.26$). To better predict the distance, we made the following two improvements:

1. **β-VAE Representation:** Instead of nine scagnostics measures, we use the encoder of our model to extract 32 latent features from a scatterplot image.
2. **Neural Network-based Approximator:** We found the empirical distribution of scagnostics measures is skewed, even having strong correlation between measures. Instead of giving equal weights to measures (e.g., L_2 distance), we adopt a simple neural network to better determine the weights.

We conducted an experiment as follows: we first collected 247 scatterplots from a previous study [21] with their perceptual distances known through a user study in a range between 0 (most similar) and 1 (most dissimilar). We filtered out the scatterplots according to our filtering criteria (Sect. 3.1), which resulted in 180 scatterplots and 16,110 pairs of them. We used five-fold cross validation; we separated the pairs into a training set and a test set. For a scatterplot pair in the training set, we extracted the *code* of each scatterplot. We used three code conditions: 1) scagnostics measures (9 dimensions), 2) a representation from the encoder of our model (32 dimensions), and 3) concatenation of both ($9 + 32 = 41$ dimensions). Then, we concatenated the codes of the two scatterplots in a pair into a *pairwise code*; the three code conditions generated a pairwise code of 18, 64, and 82 dimensions, respectively.

Table 2: Correlation between Predicted and Perceived Distances

Code	*Approximator*	*Pearson's r*
Scagnostics	L_2 Distance	< 0.26 [21]
Scagnostics	1-hidden-layer NN	0.637
β-VAE	1-hidden-layer NN	0.706
β-VAE + Scagnostics	**1-hidden-layer NN**	**0.751**

We trained a simple neural network (hereafter, *Approximator*) so that it predicts the perceived distance of a pair from its pairwise code. To make Approximator less obscure, we only used one hidden layer with 32 neurons. Using the test set, we computed the Pearson's correlation coefficient between the predicted distances and perceived distances as in a previous study [21]. The correlation was averaged over five folds. Approximator had layers of BatchNorm-Linear(32)-ReLU-Linear(1) and was trained using the MSE loss function and an Adam optimizer [15] with a learning rate of 5e-3.

Table 2 shows the prediction performance depending on code conditions in terms of Pearson's r. Overall, using a neural network instead of L_2 distance resulted in a higher correlation coefficient even though the network was simple (i.e., only one hidden layer). Our code ($r_{\beta\text{-}VAE} = 0.706$) outperformed scagnostics measures ($r_{scagnostics} = 0.637$). Surprisingly, we could achieve the highest correlation coefficient when the two codes were combined and used together ($r_{\beta\text{-}VAE+scagnostics} = 0.751$).

As discussed in Sect. 4.1, the results suggest that the β-VAE and scagnostics codes can complement each other. It seems that β-VAE is able to capture the overall structures of scatterplots, although there also exist generative factors not captured by β-VAE but that play an important role in human judgement. For instance, it is shown that people are sensitive to strong edges in scatterplot images when judging the similarity [21], but such a feature was not present in β-VAE, while scagnostics has dedicated measures for this such as *Skinny* and *Striated*. Another reason would be the capability of capturing outliers. We found outliers were often missing after reconstruction, which may result from the denoising nature of autoencoders [27]. However, such information could be preserved better in scagnostics, for example, through the *Outlying* measure. In summary, we found that our β-VAE model can produce an interpretable representation that is also effective in predicting the perceived distance between a pair of scatterplots. We also discovered that the prediction performance can be further improved by complementing our representation by previous scagnostics measures.

5 Conclusion and Future Work

We present a data-driven approach to identify latent factors that can describe the characteristics of empirical data distributions in scatterplot images. Our representation is derived using β-variational autoencoder on a training corpus consisting of 1M+ scatterplot images. Our representation is interpretable and predictive; we could not only interpret its most informative features, but also predict the perceptual distance between scatterplots with high correlation ($r > 0.75$) complemented by scagnostics measures.

For robustness in training, we simplified scatterplot images to be binary and use only positional channels (i.e., x and y). It would be interesting to identify generative factors of scatterplot images with multiple visual channels, such as opacity and size, used simultaneously. We are also excited to compare our model to another deep learning-based model such as ScatterNet [19].

References

[1] Plotly feed. `https://plot.ly/feed`. Accessed: 2019-06-04.

[2] Uci machine learning repository. `https://archive.ics.uci.edu/ml/`. Accessed: 2019-06-04.

[3] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 111–117, Oct 2005. doi: 10.1109/INFOVIS.2005.1532136

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

[5] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems*, pp. 2172–2180, 2016.

[6] T. N. Dang, A. Anand, and L. Wilkinson. Timeseer: Scagnostics for high-dimensional time series. *IEEE Transactions on Visualization and Computer Graphics*, 19(3):470–483, 2012.

[7] T. N. Dang and L. Wilkinson. Scagexplorer: Exploring scatterplots by their scagnostics. In *2014 IEEE Pacific Visualization Symposium*, pp. 73–80. IEEE, 2014.

[8] T. N. Dang and L. Wilkinson. Transforming scagnostics to reveal hidden features. *IEEE transactions on visualization and computer graphics*, 20(12):1624–1632, 2014.

[9] L. Fu. Implementation of three-dimensional scagnostics. *Univ. of Waterloo, Dept. of*, 2009.

[10] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, vol. 3, 2017.

[11] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[12] K. Hu, M. A. Bakker, S. Li, T. Kraska, and C. Hidalgo. Vizml: A machine learning approach to visualization recommendation. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, p. 128. ACM, 2019.

[13] D. Joanes and C. Gill. Comparing measures of sample skewness and kurtosis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(1):183–189, 1998.

[14] H. Kim and A. Mnih. Disentangling by factorising. *arXiv preprint arXiv:1802.05983*, 2018.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[16] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[17] T. D. Kulkarni, W. F. Whitney, P. Kohli, and J. Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pp. 2539–2547, 2015.

[18] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010.

[19] Y. Ma, A. K. Tung, W. Wang, X. Gao, Z. Pan, and W. Chen. Scatternet: A deep subjective similarity model for visual analysis of scatterplots. *IEEE transactions on visualization and computer graphics*, 2018.

[20] J. Matute, A. C. Telea, and L. Linsen. Skeleton-based scagnostics. *IEEE transactions on visualization and computer graphics*, 24(1):542–552, 2017.

[21] A. V. Pandey, J. Krause, C. Felix, J. Boy, and E. Bertini. Towards understanding human similarity perception in the analysis of large sets of scatter plots. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pp. 3659–3669. ACM, 2016.

[22] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[23] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[24] A. Sarikaya and M. Gleicher. Scatterplots: Tasks, data, and designs. *IEEE transactions on visualization and computer graphics*, 24(1):402–412, 2017.

[25] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton. Veegan: Reducing mode collapse in gans using implicit variational learning. In *Advances in Neural Information Processing Systems*, pp. 3308–3318, 2017.

[26] J. W. Tukey and P. A. Tukey. Computer graphics and exploratory data analysis: An introduction. *The Collected Works of John W. Tukey: Graphics: 1965-1985*, 5:419, 1988.

[27] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103. ACM, 2008.

[28] L. Wilkinson, A. Anand, and R. Grossman. Graph-theoretic scagnostics. In *IEEE Symposium on Information Visualization, 2005. INFOVIS 2005.*, pp. 157–164. IEEE, 2005.

[29] L. Wilkinson and G. Wills. Scagnostics distributions. *Journal of Computational and Graphical Statistics*, 17(2):473–491, 2008.