EuroVis Workshop on Visual Analytics (2026)
D. Archambault and A. Vilanova (Editors)

Same Stats, Different Graphs Made Simple and Fast by Linear Transformations

S. van Wageningen[1] and A. C. Telea[1] and T. Mchedlidze[1]

[1]Utrecht University, The Netherlands

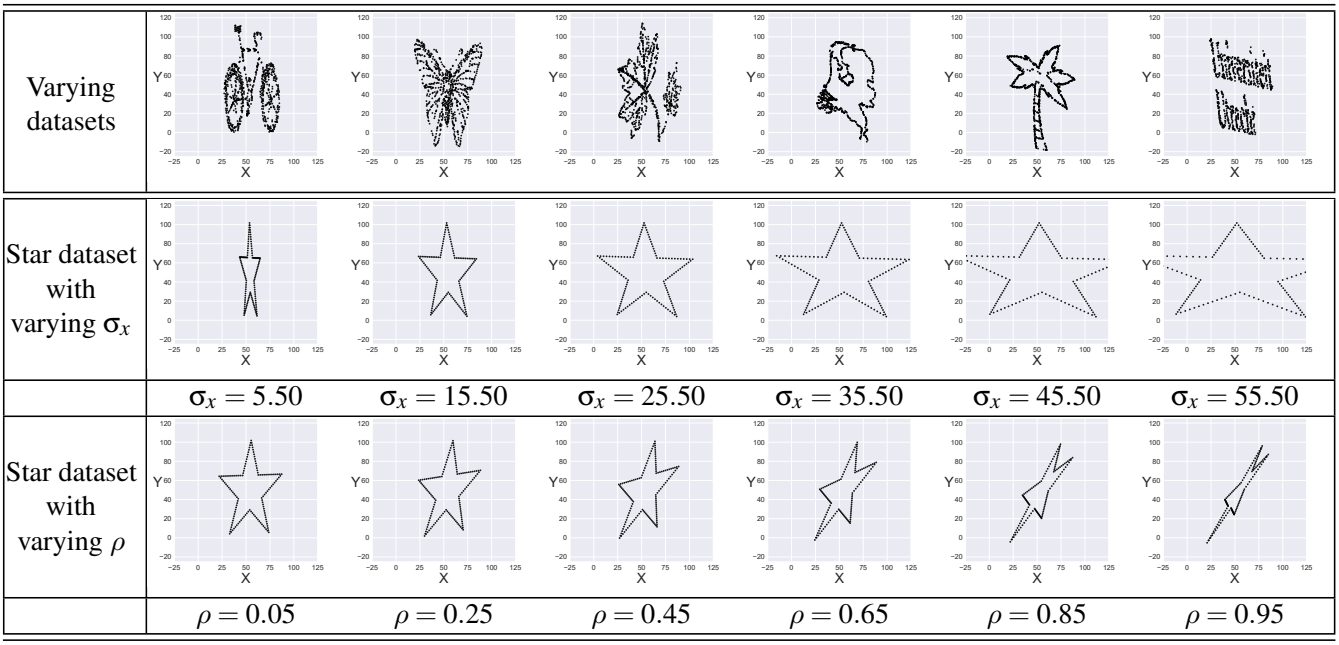

Figure 1: *First row: Scatterplots of different datasets created by the proposed transformation method (`LT`). All have the same summary statistics ($\mu_x = 52.26, \mu_y = 47.83, \sigma_x = 16.76, \sigma_y = 26.93, \rho = -0.06$). Bottom two rows: Scatterplots of the Star dataset created by `LT`. All have the same summary statistics as the first row but with varying σ_x and varying ρ.*

Abstract

While summary statistics are used consistently to report data results, their limitations (and the importance of visualizing data) are well known. Yet, current methods that aim to approximate shapes to match some desired summary statistics are quite complex and can fail the approximation for complex shapes. We present a simple linear transformation `LT` that performs the same approximation more effectively and efficiently. We show that this method works well for complicated shapes and large datasets, visually outperforms almost all baseline algorithms for most shapes, while being significantly faster and simpler. We also present an iterative approach that improves the method `LT` in case of extreme statistical values.

CCS Concepts

• ***Human-centered computing*** → ***Visualization techniques;***

1. Introduction

Summary statistics, such as mean and standard deviation, are widely used to capture the main trends of large datasets. However, several works have shown the limitations of summary statistics and underlined the importance of visualizing data. In particular, Anscombe [Ans73] showed simple examples where quite different visual scatter plots yield similar statistics. Later on, Chatterjee and Firat [CF07] showed that they could *generate* datasets with similar statistics but completely dissimilar graphics using a Genetic

Algorithm. The Datasaurus Dozen [MF17] followed up on this with a Simulated Annealing Algorithm that morphs a given scatterplot into a target shape, such as the Datasaurus [Cai16], with the same summary statistics up to 2 digits.

In dimensionality reduction (*DR*), large high-dimensional datasets are reduced to 2D or 3D scatterplots. Quality metrics are used to gauge how well the DR process preserves the data structure [EMK*19] – thus playing a similar summarization role to statistics. Yet, such summary quality metrics can also be misleading. Chari and Pachter [CP23] showed how to morph an existing DR plot into specified shapes, such as an elephant, while keeping quality metrics relatively constant. Machado et al. [MBT25] used Feed-Forward Neural Network to massively distort DR plots into pseudorandom shapes while again keeping quality metrics similar.

Related findings were shown in the field of graph drawing (*GD*). Similar to DR, GD has its own set of quality metrics which are used during optimization and evaluation of the drawings. Inspired by the Datasaurus Dozen [MF17], van Wageningen et al. [vWMT25] used a Simulated Annealing algorithm to morph graph drawings into several unique target shapes, including a video, while keeping a set of quality metrics nearly the same.

While all above methods showed that summary statistics are not truly descriptive of the input data, they do this via quite *complex* ways – either iterative approaches, such as Simulated Annealing or Genetic Algorithms; or required careful training of a non-trivial Neural Network. Also, these methods are quite computationally *expensive* and do not guarantee desired results for more complicated target shapes. As such, these methods raise the question whether summary statistics are indeed inherently *hard* to fool – in which case, their perceived 'strength' would be larger.

In this work, we show that given a scatterplot, we can approximate *any* target shape, regardless of the original dataset, while keeping summary statistics exactly constant, using simple and fast algebra transformations. Our contributions include:

- A *near-instant* linear transformation method `LT` able to approximate shapes to a desired set of summary statistics;
- A fast iterative gradient descent method which outperforms all other existing iterative algorithms when combined with `LT`;
- A replication of the results from the Datasaurus Dozen, comparing the proposed method `LT` with three baseline algorithms;
- Two new dataset collections used for testing the fooling of summary statistics: (1) The DataDance DoubleDozen: A sequence of scatterplots with the same summary statistics which can be displayed in a playful video. (2) A collection of eight large complicated scatterplots.

While the suggested transformation method is not novel, its use in demonstrating the untrustworthiness of summary statistics has thus far not been explored. With this work, we show that generating dissimilar scatterplots with similar statistics is a mathematical problem with a trivial algebraic solution.

2. Method

A dataset $D \in \mathbb{R}^{n\times 2}$ consists of a set of n (x,y) values in $\mathbb{R}^2$. Let $S(D) = \{\mu_x^D, \mu_y^D, \sigma_x^D, \sigma_y^D, \rho^D\}$ denote summary statistics computed on D – means μ, standard deviations σ with respect to both x and y, and the Pearson correlation coefficient ρ between x and y. We denote by M^{AB} the $n \times n$ pair-wise Euclidean distance matrix of two such datasets A and B. Finally, let D_x, D_y denote the two n-dimensional columns of D.

Methods aiming to prove limitations of summary statistics transform a dataset A to acquire the shape of a dataset B, which differs strongly from A, so that $S(B) \simeq S(A)$. The method, described next, works slightly differently – it transforms B so as to yield a set of statistical values (either computed from a given A or provided by the user). Our full implementation is available online.

2.1. Linear transformation method

Let B be a dataset and $S(A) = \{\mu_x^A, \mu_y^A, \sigma_x^A, \sigma_y^A, \rho^A\}$, a set of summary statistics acquired from either dataset A or manually specified. Let $S(B)$ be the analogous statistics computed from dataset B. We next transform B to achieve the statistics $S(A)$ in two steps: First, we translate B to have zero mean ($\mu^B = 0$) and apply a so-called *whitening transform* [AH01] to set its variance to one. Second, we *recolor* the resulting dataset to have statistics $S(A)$.

Given a statistics-set $S(D)$, let cov^D denote the covariance matrix of a dataset D

$$\mathrm{Cov^D} = \begin{pmatrix} \sigma_x^D \sigma_x^D & \rho^D \sigma_x^D \sigma_y^D \\ \rho^D \sigma_x^D \sigma_y^D & \sigma_y^D \sigma_y^D \end{pmatrix}.$$

We denote by E_D and $\lambda^D = (\lambda_1^D, \lambda_2^D)$ the matrix of eigenvectors and the eigenvalues of $\mathrm{Cov^D}$, respectively.

With the above, we first center B by performing $B_x \leftarrow B_x - \mu_x^B$, $B_y \leftarrow B_y - \mu_y^B$. We then apply a whitening transform $B \leftarrow (B \cdot W^T)$ where

$$W = E_B \cdot \begin{pmatrix} \frac{1}{\sqrt{\lambda_1^B}} & 0 \\ 0 & , \frac{1}{\sqrt{\lambda_2^B}} \end{pmatrix} \cdot E_B.$$

Next, we recolor the dataset B with the statistics of A by computing $B \leftarrow (B \cdot C^T)$ where

$$C = E_A \cdot \begin{pmatrix} \sqrt{\lambda_1^A} & 0 \\ 0 & , \sqrt{\lambda_2^A} \end{pmatrix} \cdot E_A.$$

Finally, we translate B to the desired means $B_x \leftarrow B_x + \mu_x^A$, $B_y \leftarrow B_y + \mu_y^A$. At this point, the resulting dataset B has statistics $S(A)$. All the above steps, taken together, result in a *linear* transformation (called `LT` next) which combines translation, scaling, and shearing, as next illustrated in our experiments.

2.2. Baseline Algorithms

To determine the effectiveness of the proposed transformation method `LT`, we will compare its results with other algorithms used to transform scatterplots into target shapes while maintaining similar summary statistics. For all these algorithms we remove boundary constraints – that is, the created datasets can have any coordinates in $\mathbb{R}^2$.

$\texttt{SA}_{D^2}$: The Simulated Annealing algorithm from the Datasaurus Dozen [MF17] reads a dataset A with desired summary statistics $S(A)$ and a target dataset B with a specified shape. It then adds noise to a single data point and accepts that point if its fitness is better. Fitness is defined as the Euclidean distance, to *any* target point, i.e. if the noisy data point is closer to any target point it is accepted. Points that are not closer to any target point are also accepted occasionally so as to escape local minima, as per the simulated annealing logic. The entire process is repeated for a set number of iterations.

$\texttt{SA}_{D^4}$: The DataDance DoubleDozen [vW25] takes the same inputs as the Simulated Annealing algorithm $\texttt{SA}_{D^2}$ but adapts some parameters and the fitness functions. The algorithm adds noise to *multiple* data points and only accepts points if the fitness of *all* data points improves. Here, fitness is computed via either the Chamfer distance [AS03] or the Hungarian Algorithm [Kuh55]. This increases computation times but allows the algorithm to recreate more complex shapes. The algorithm also applies a simulated annealing cooling step to the amount of noise added to data points, and the number of data points sampled. The entire process is repeated for a set number of iterations.

SGD : In addition to the above algorithms, we propose a simpler and faster Stochastic Gradient Descent (SGD) algorithm which can reconstruct most target shapes while keeping the summary statistics *roughly* the same. Algorithm 1 shows the pseudocode of SGD. We start by computing the summary statistics of the input dataset $S_{init}(A)$. We next consider a sampled subset $A_s \in A$ of s samples to accelerate computations. We compute the fitness of our current dataset using the Chamfer distance by $\text{Chamfer}(A_s,B) = \frac{1}{s}\sum_{i=1}^{s} \min_j M_{ij}^{AB} + \frac{1}{n}\sum_{j=1}^{n} \min_i M_{ij}^{AB}$, where we sum the means of the minimum values of the rows and columns of the pair-wise distance matrix M^{AB}. We then sum up the squared differences of the current summary statistics of A and the target summary statistics ($\text{loss\_stats}(A, S_{init}(A))$). We take the sum of the fitness returned by the Chamfer distance and the squared sum of errors as our loss and backpropagate this loss to modify our current dataset. We repeat these steps for a set number of *iterations*. Note that our approach is a *soft-constrained* optimization algorithm – that is, the algorithm can return datasets with summary statistics that deviate more than our hard constraint of never varying more than 2 decimals from S_{init}. We fix this by applying our transformation method LT described in Sec. 2.1 to A to acquire the final transformed dataset A_f.

Algorithm 1 SGD

$S_{init}(A) \leftarrow \{\mu_x^A, \mu_y^A, \sigma_x^A, \sigma_y^A, \rho^A\}$
$\eta = [0.5, \ldots, 1e{-}5]^{iterations}$
while $i < iterations$ **do**
  $A_s \leftarrow \text{sample}(A, s)$
  $\text{fit} \leftarrow \text{Chamfer}(A_s, B)$
  $L_s \leftarrow \text{loss\_stats}(A, S_{init}(A))$
  $\ell \leftarrow \text{fit} + L_s$
  $A \leftarrow A - \eta_i \nabla_A \ell$
  $i \leftarrow i + 1$
end while
$A_f \leftarrow \texttt{TF}(A, S_{init})$
return A_f

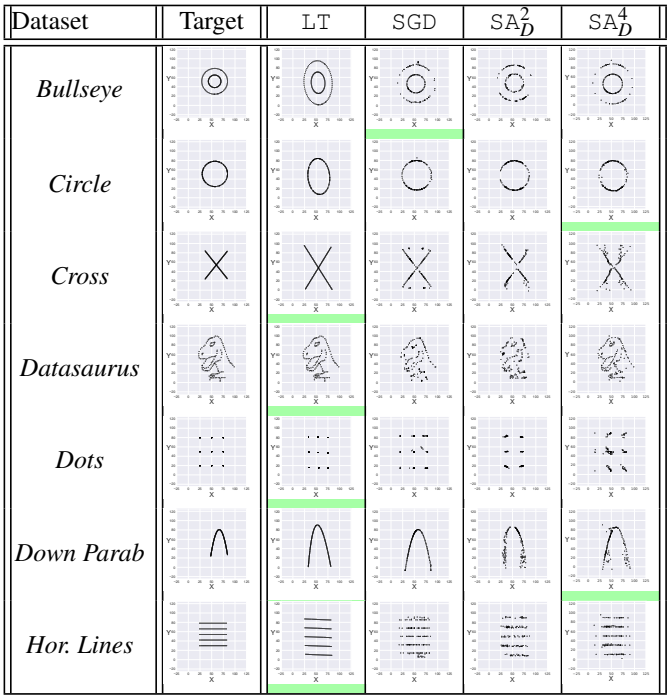

Figure 2: *Scatterplots of transformed datasets from the Datasaurus Dozen produced by the LT method and three baseline methods (SGD, $\texttt{SA}_D^2$, and $\texttt{SA}_D^4$). The summary statistics of all datasets are identical. Green-marked cells show the best results (lowest δ value).*

2.3. Evaluation

We show the effectiveness of the simple transformation method LT by conducting three experiments.

Exp. 1: We replicate the shapes from the Datasaurus Dozen [MF17] and those from the DataDance DoubleDozen [vW25]).

Exp. 2: We vary the target standard deviation σ_x and the target correlation ρ for one target shape to see how well our transformation works for different summary statistics. We do not vary σ_y, as the results of σ_x will hold true for σ_y as well. Also, we do not experiment with varying μ, as simple translations can be made to a dataset to get *any* target μ.

Exp. 3: We apply the proposed transformation LT to larger datasets with more complicated shapes.

For all experiments, we start from the same dataset A, which is a dataset with random coordinates and no inherent shape. We fix the target summary statistics based on the exact values from the Datasaurus Dozen ($\mu_x^A = 52.26, \mu_y^A = 47.83, \sigma_x^A = 16.76, \sigma_y^A = 26.93, \rho^A = -0.06$). We compare the results of the method LT with the baseline algorithms $\texttt{SA}_D^2$, $\texttt{SA}_D^4$, and SGD (see Sec. 2.2). The latter three algorithms are iterative approaches and have the number of iterations set to $4 \cdot 10^5$, $2 \cdot 10^5$, and $1.5 \cdot 10^4$, respectively. We ask these iterative algorithms to keep summary statistics consistent to two decimal points. In Exp. 3, we only compare LT and SGD as the other methods had drastically longer computation times and worse results. We observe the quality of the resulting scatterplots through visual inspection. To quantitatively assess how well a shape

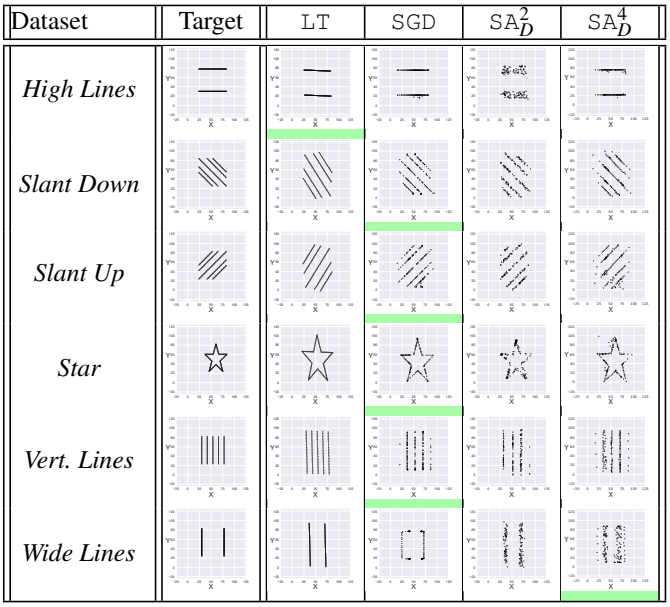

Figure 3: *Scatterplots of the transformed datasets from the Datasaurus Dozen produced by the* LT *method and three baseline methods (*SGD*,* SA^2_D*, and* SA^4_D*). The summary statistics of all datasets (minus the target shape dataset) are identical. Green-marked cells show the best results (lowest* δ *value).*

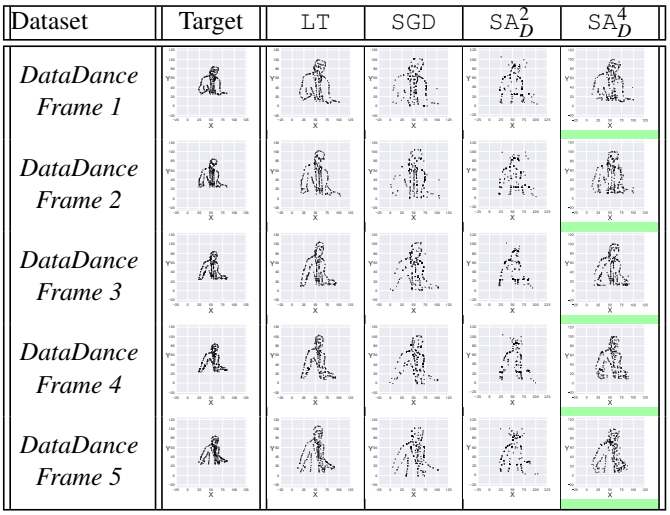

Figure 4: *The first five scatterplots of the transformed datasets from the DataDance DoubleDozen produced by the* LT *method and three baseline methods (*SGD*,* SA^2_D*, and* SA^4_D*). The summary statistics of all datasets (minus the target shape dataset) are identical. Green-marked cells show the best results (lowest* δ *value).*

was recreated we compute the difference δ between shapes by $\delta = \sum_{i=1}^{n} M^{AB}_{i,\pi^\star(i)}$, here the optimal (lowest-cost) permutation $\pi^\star$ of the pair-wise distance matrix M^{AB} is computed using the Hungarian algorithm: $\pi^\star = Hungarian(M^{AB})$.

3. Results and Discussion

We next discuss our obtained results. For larger versions of all figures, we refer the reader to the supplementary material.

Exp. 1: We reproduce all scatterplots from the Datasaurus Dozen and the DataDance DoubleDozen and compare the method LT with three baselines SGD, SA^2_D, and SA^4_D. Figures 2-4 show these results. We observe that the results of LT are consistently high in visual quality – that is, visually very similar to the target shapes. In particular, the *Datasaurus*, *Dots*, *Vert. lines*, and *Star* results of LT are near perfect recreations of their target shapes. Yet we notice that LT does not always provide 'faithful' recreations according to the computed metric δ. This can be seen for the *Bullseye* and *Circle* datasets where, by design, LT transforms the shape to fit the desired summary statistics and therefore warps the proportions of the shapes via shearing. In contrast, for these datasets, SGD and SA^4_D reach shapes which are closer to the target (according to the metric δ) – though, these shapes create local disturbances, *i.e.*, not not all points are accurately placed on the radii of the circles. We notice similar effects for the datasets *Datadance Frame 3* and *Frame 4* (Fig. 4). Here, LT 'slims' the target shape while SA^4_D acquires better δ results. When all frames are visualized in a video, SA^4_D produces more time-coherent frames. Finally, we see that all iterative algorithms do introduce various amounts of noise in their created shapes (most for SA^2_D, less for SA^4_D, and least for SGD). In contrast, LT, by design, never adds such noise.

Exp. 2: We now keep the same summary statistics S_{init} but vary σ_x and ρ. Figures 5-6 show the results. We see now better how LT warps the proportions of its target shape to acquire the desired

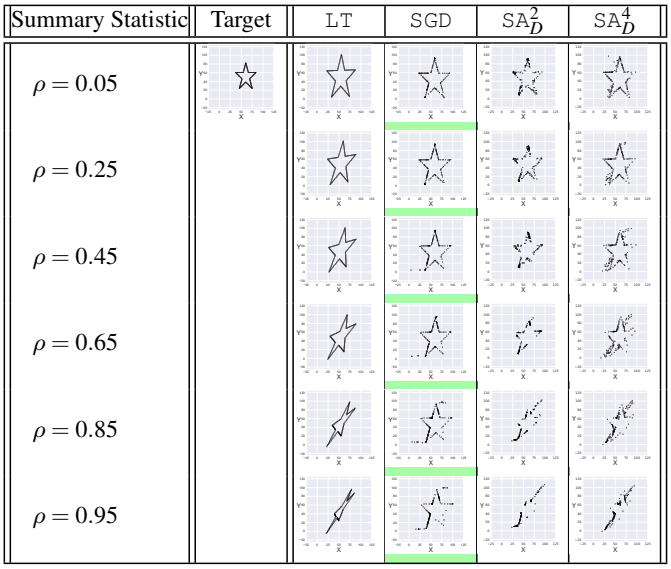

Figure 5: *Scatterplots of the transformed Star datasets produced by the* LT *method and three baseline methods (*SGD*,* SA^2_D*, and* SA^4_D*). The summary statistics of all datasets (minus the target shape dataset) are identical except for the varying* ρ*. Green-marked cells show the best results (lowest* δ *value).*

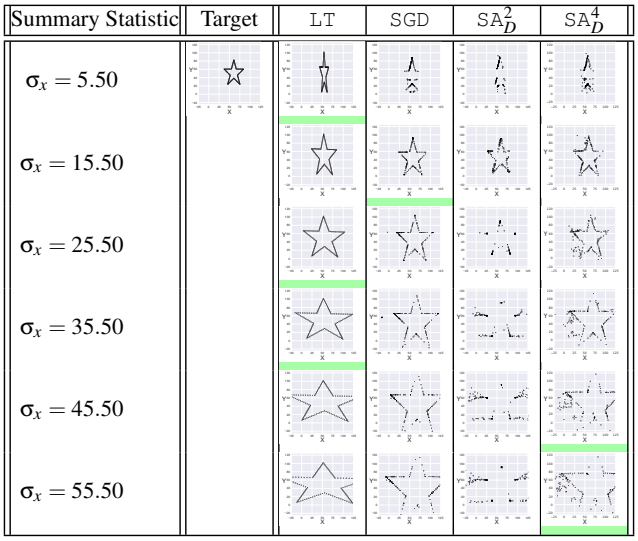

Figure 6: *Scatterplots of the transformed Star datasets produced by the* LT *method and three baseline methods (*SGD*,* SA^2_D*, and* SA^4_D*). The summary statistics of all datasets (minus the target shape dataset) are identical except for the varying* σ_x*. Green-marked cells show the best results (lowest* δ *value).*

summary statistics. For instance, the *Star* dataset with $\rho = 0.95$ in Fig. 5 is strongly warped by LT to the point where a star can no longer be recognized. SGD, on the other hand, still produces a *star-like* shape. In contrast, both SA^2_D and SA^4_D have greater difficulties in generating a shape which resembles the input star: when varying σ, the star structure exhibits many gaps and noisy points; when varying ρ, we see the same warping as for LT but, in addition, significant noise is added.

Exp. 3: Figure 7 shows the results of LT and SGD on eight larger and more complex shapes (855 points each). LT was able to recover the structure of all datasets without introducing spurious noise – but with the already discussed warping being present. In contrast, SGD produces less warping but creates significant amounts of noise – for some datasets like *UU* the input shape is barely recognizable.

Altogether, our experiments show that LT can adapt all tested datasets to precisely yield the desired target statistics while keeping the target shapes intact (modulo shearing and scaling which are inherent to the method's design). The SGD method uses the LT method for a final correction, leading to shapes with less warping but more noise. Speed-wise, LT handles all tested shapes in under 10 milliseconds on a commodity PC; SGD requires about 60 seconds for the largest shapes; and the earlier methods presented in the literature, SA^2_D and SA^4_D both require 10 to 20 minutes for small shapes (142 points) and 2 to 3 hours for the largest shapes (855 points).

4. Conclusion

We presented a simple linear transformation approach LT that *nearly instantly* approximates *any* given target shape for a desired set of summary statistics. LT works well for complicated, large, shapes and visually outperforms almost all existing algorithms for most shapes (modulo a warping factor). We also presented a gradient-descent-based method SGD that, when combined with LT, reduces the warping inherent to LT in return for noise.

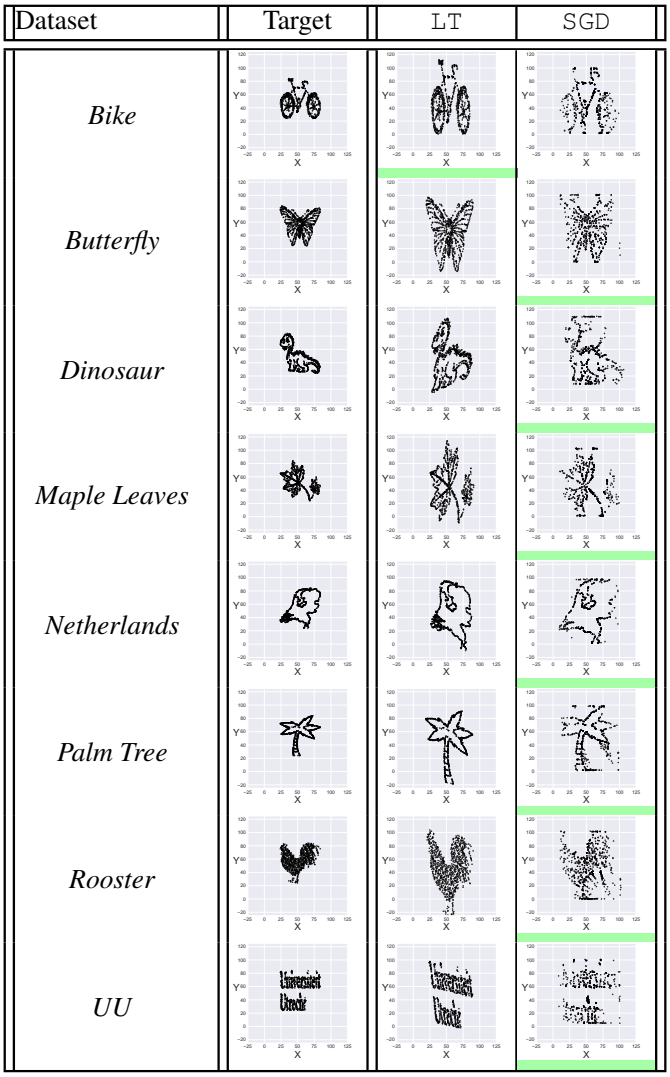

Figure 7: *Scatterplots of the transformed datasets produced by the* LT *method and one baseline method* SGD*. The summary statistics of all datasets (minus the target shape dataset) are identical. Green-marked cells show the best results (lowest* δ *value).*

Our work shows that one does not need complex and expensive algorithms to faithfully create scatterplots that have given statistical values – hence further underlines the need for better measures to characterize scatterplots. Future work can explore whether the simple linear transformations can be adapted to analogous ends for more complicated statistics such as graph drawing or dimensionality reduction quality metrics. Separately, it is worth considering refining the LT approach to reduce warping, and to explore its use for 3D scatterplots. Finally, other existing whitening transforms (besides the current PCA-based one) could be explored in the design of LT.

Acknowledgments We would like to thank Alister Machado for his suggestion on using the Hungarian Algorithm, and his helpful explanations on the linear transformation method.

References

[AH01] AAPO HYVÄRINEN JUHA KARHUNEN E. O.: *Principal Component Analysis and Whitening*. John Wiley and Sons, Ltd, 2001, ch. 6, pp. 125–144. doi:https://doi.org/10.1002/0471221317.ch6. 2

[Ans73] ANSCOMBE F. J.: Graphs in Statistical Analysis. *The American Statistician 27*, 1 (1973), 17–21. doi:10.1080/00031305.1973.10478966. 1

[AS03] ATHITSOS V., SCLAROFF S.: Estimating 3D hand pose from a cluttered image. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* (June 2003), vol. 2, pp. II–432. doi:10.1109/CVPR.2003.1211500. 3

[Cai16] CAIRO A.: Download the Datasaurus: Never trust summary statistics alone; always visualize your data, 2016. URL: https://thefunctionalart.blogspot.com/2016/08/download-datasaurus-never-trust-summary.html. 2

[CF07] CHATTERJEE S., FIRAT A.: Generating Data with Identical Statistics but Dissimilar Graphics. *The American Statistician 61*, 3 (2007), 248–254. doi:10.1198/000313007X220057. 1

[CP23] CHARI T., PACHTER L.: The specious art of single-cell genomics. *Computational Biology 19*, 8 (2023), 1–20. doi:10.1371/journal.pcbi.1011288. 2

[EMK*19] ESPADOTO M., MARTINS R., KERREN A., HIRATA N., TELEA A.: Toward a quantitative survey of dimension reduction techniques. *IEEE TVCG 27*, 3 (2019), 2153–2173. doi:10.1109/TVCG.2019.2944182. 2

[Kuh55] KUHN H. W.: The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly 2*, 1-2 (1955), 83–97. doi:https://doi.org/10.1002/nav.3800020109. 3

[MBT25] MACHADO A., BEHRISCH M., TELEA A. C.: Necessary but not Sufficient: Limitations of Projection Quality Metrics. *Computer Graphics Forum* (2025), e70101. doi:10.1111/cgf.70101. 2

[MF17] MATEJKA J., FITZMAURICE G.: Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing. In *CHI* (2017), Association for Computing Machinery, p. 1290–1294. doi:10.1145/3025453.3025912. 2, 3

[vW25] VAN WAGENINGEN S.: Datadancedoubledozen. https://github.com/simonvw95/DataDanceDoubleDozen, 2025. 3

[vWMT25] VAN WAGENINGEN S., MCHEDLIDZE T., TELEA A. C.: Same Quality Metrics, Different Graph Drawings. In *Graph Drawing* (Dagstuhl, Germany, 2025), Dujmović V., Montecchiani F., (Eds.), vol. 357 of *LIPIcs*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 7:1–7:13. doi:10.4230/LIPIcs.GD.2025.7. 2

5. Supplementary Material

We provide the large scale figures of all results presented in the paper:

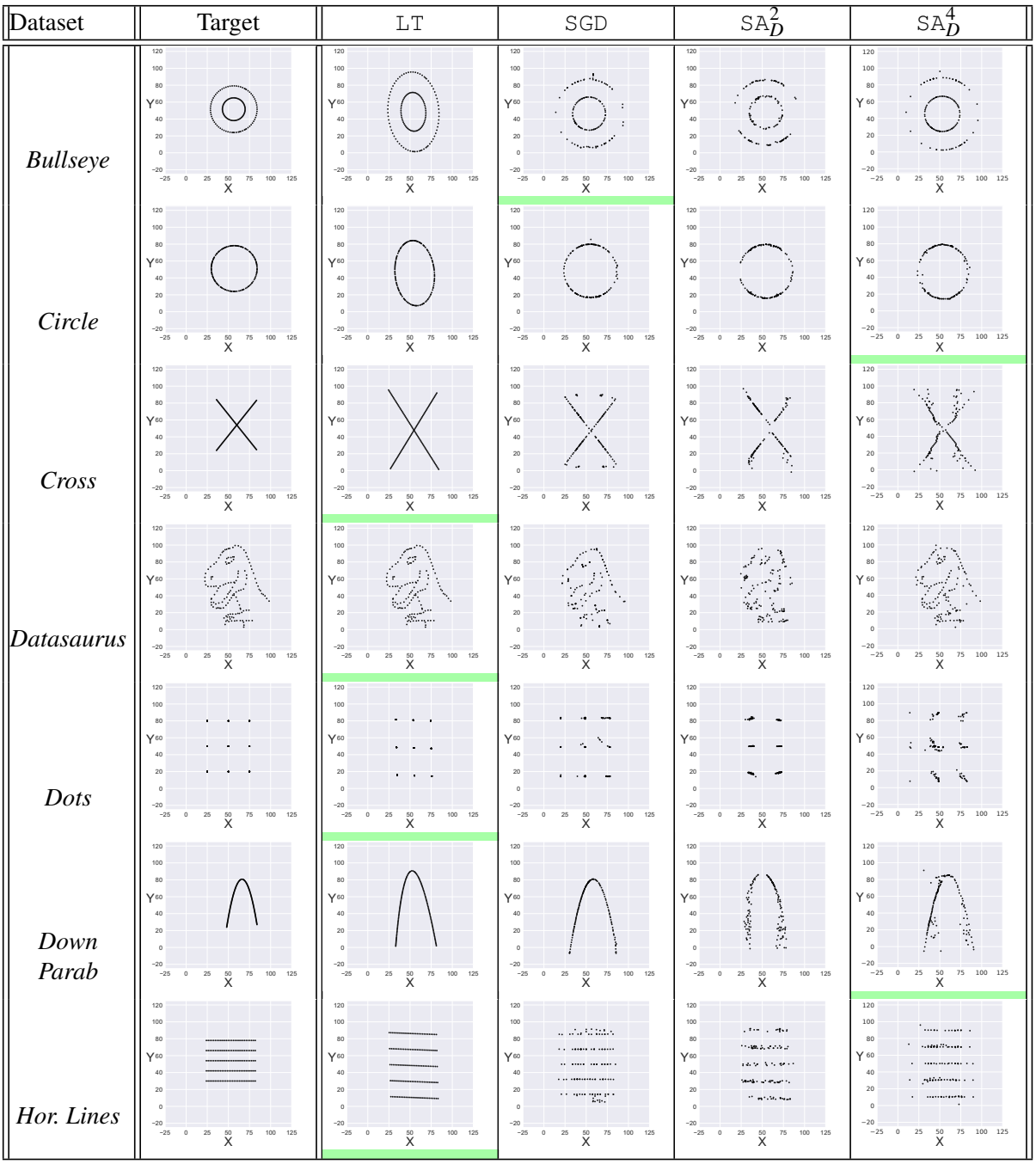

Figure 8: *Scatter plots of the transformed datasets from the Datasaurus Dozen produced by the* LT *method and three baseline methods (*SGD*,* SA^2_D*, and* SA^4_D*). The summary statistics of all datasets (minus the target shape dataset) are identical . Green-marked cells show the best results (lowest* δ *value).*

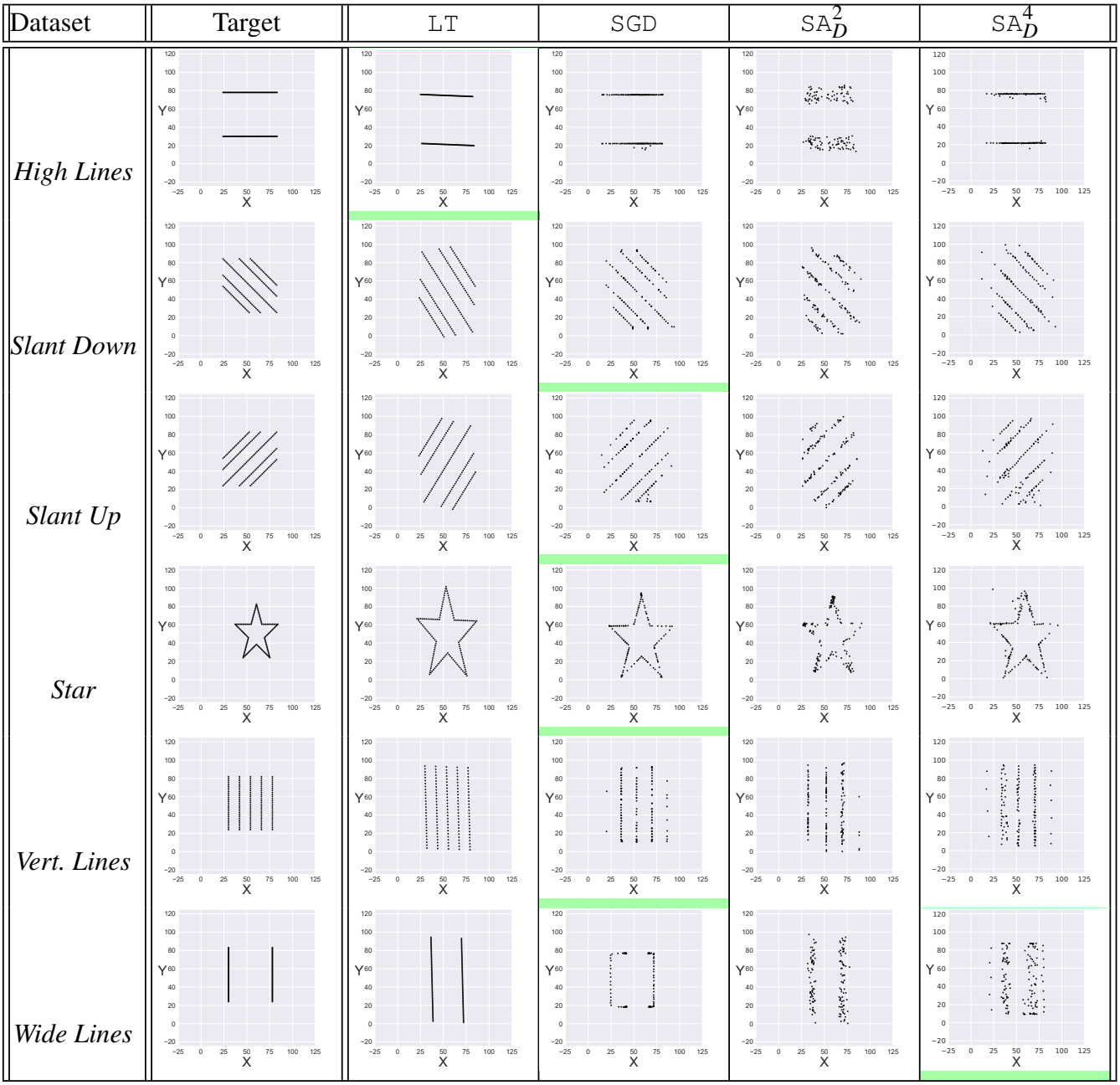

Figure 9: *Scatter plots of the transformed datasets from the Datasaurus Dozen produced by the* LT *method and three baseline methods (*SGD*,* SA_D^2*, and* SA_D^4*). The summary statistics of all datasets (minus the target shape dataset) are identical . Green-marked cells show the best results (lowest* δ *value).*

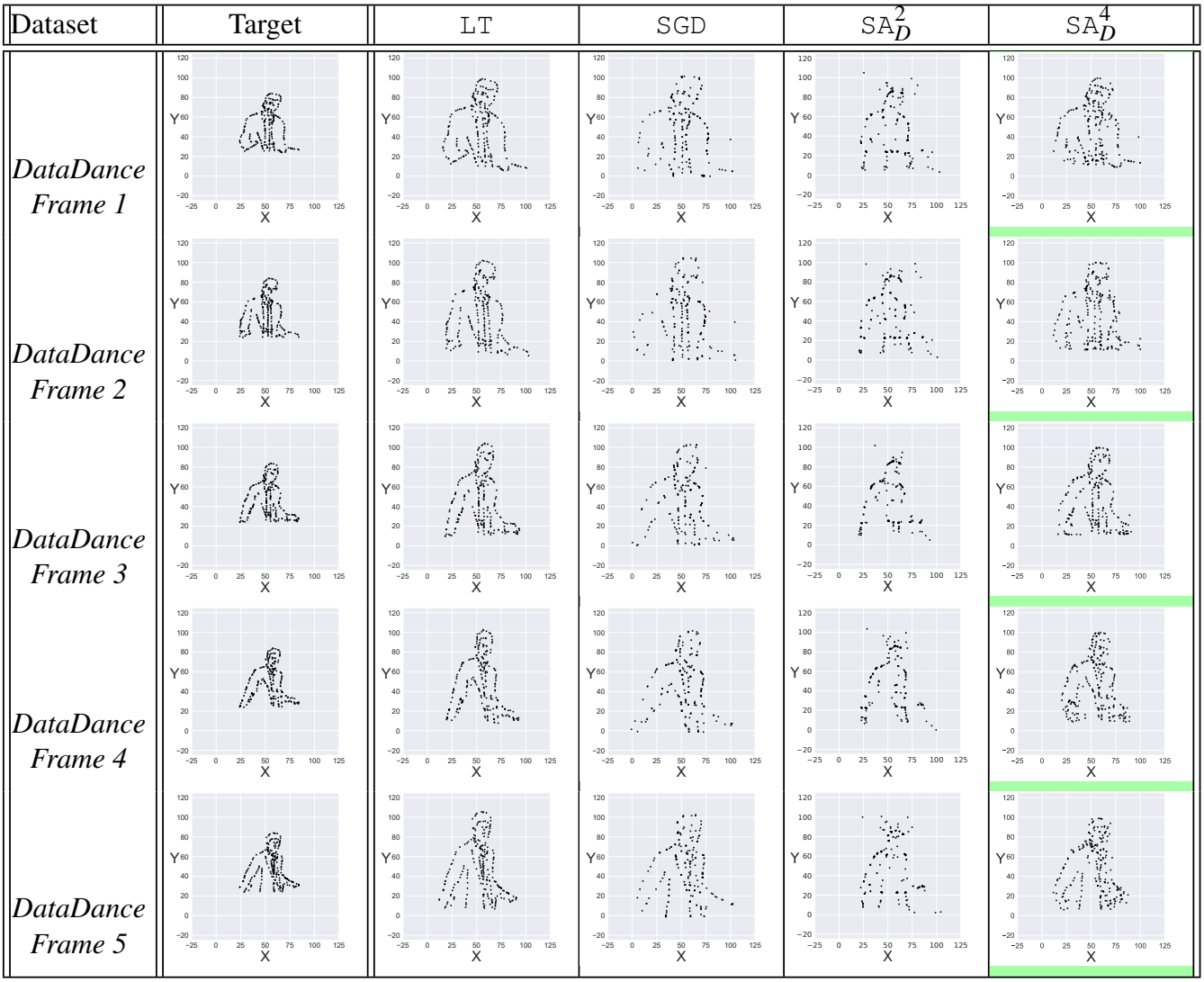

Figure 10: *Scatter plots of the transformed datasets from the DataDance DoubleDozen produced by the* `LT` *method and three baseline methods (*`SGD`*,* SA_D^2*, and* SA_D^4*). The summary statistics of all datasets (minus the target shape dataset) are identical . Green-marked cells show the best results (lowest* δ *value).*

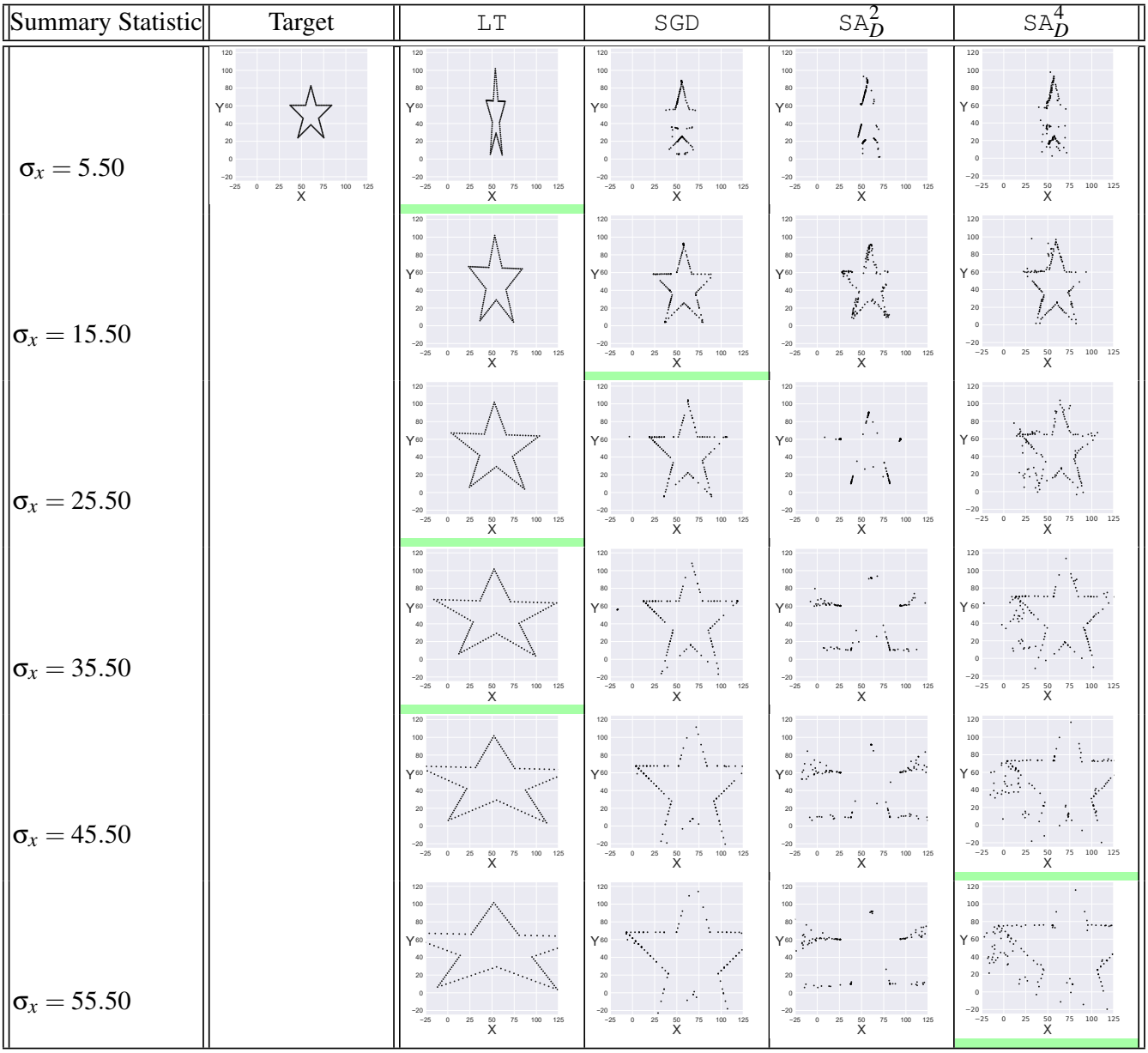

Figure 11: *Scatter plots of the transformed Star datasets produced by the* `LT` *method and three baseline methods (*`SGD`*,* SA^2_D*, and* SA^4_D*). The summary statistics of all datasets (minus the target shape dataset) are identical except for the varying* σ_x*. Green-marked cells show the best results (lowest* δ *value).*

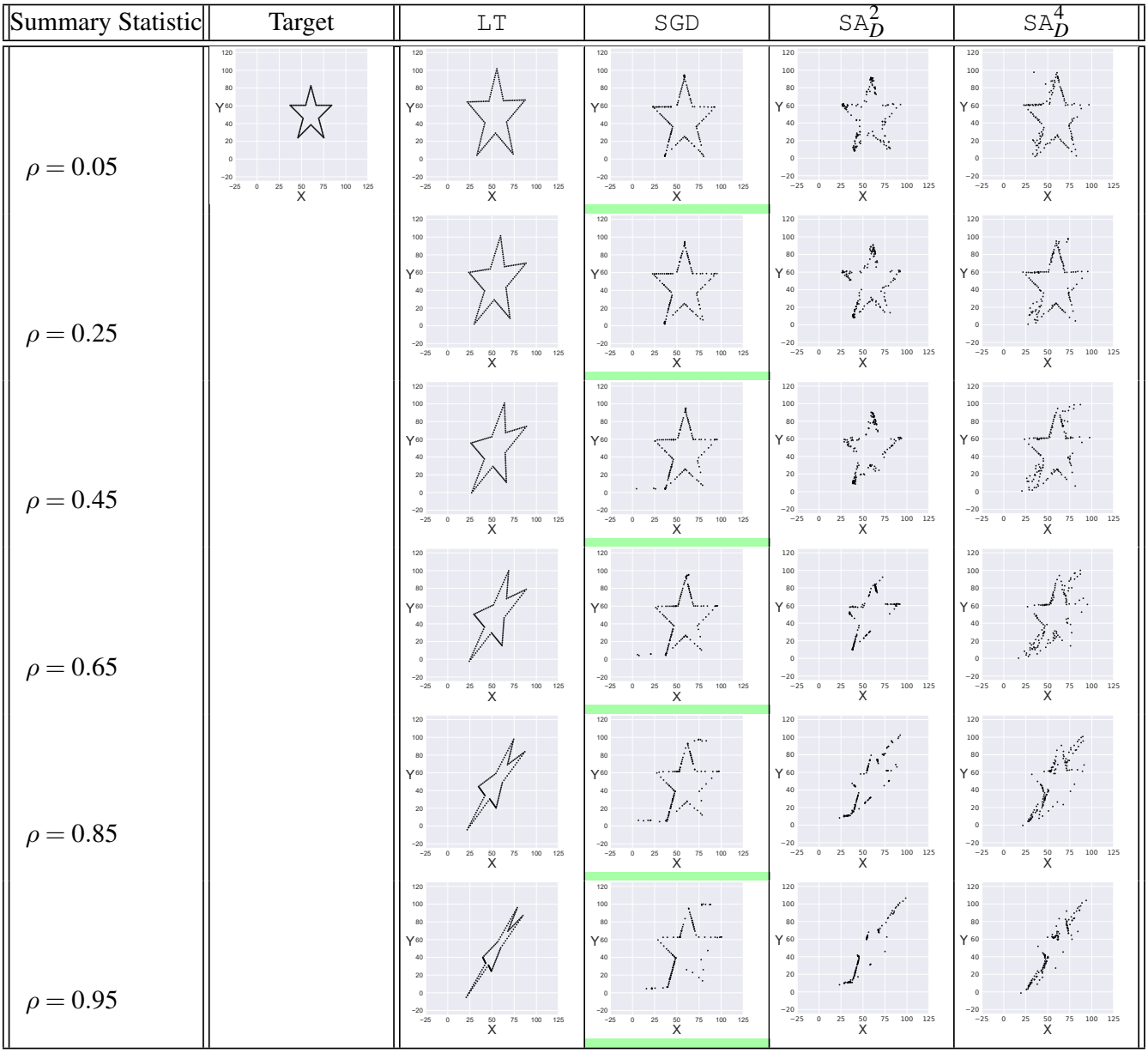

Figure 12: *Scatter plots of the transformed Star datasets produced by the* `LT` *method and three baseline methods (*`SGD`*,* SA_D^2*, and* SA_D^4*). The summary statistics of all datasets (minus the target shape dataset) are identical except for the varying* ρ*. Green-marked cells show the best results (lowest* δ *value).*

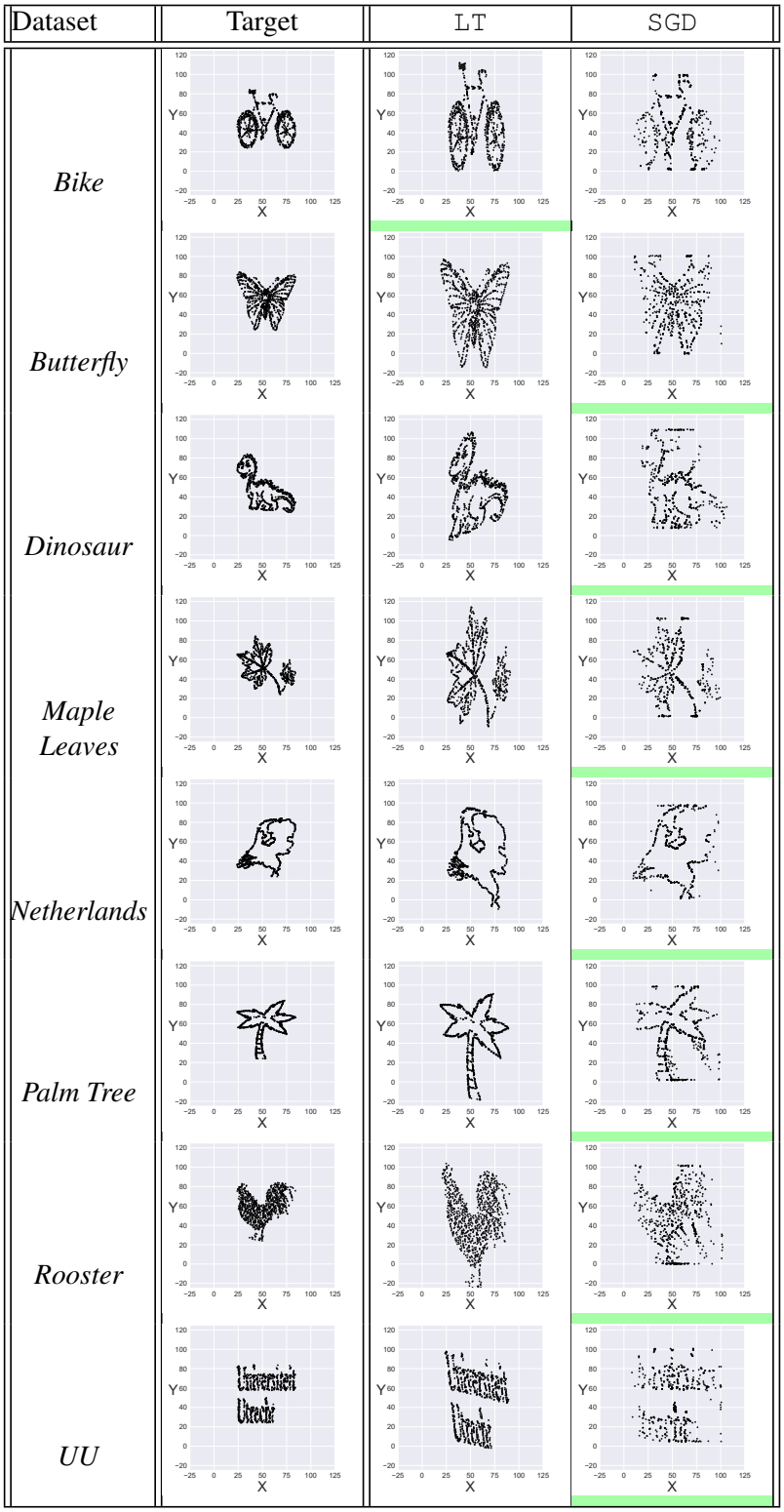

Figure 13: *Scatter plots of the transformed datasets produced by the proposed* `LT` *method and one baseline method* `SGD`*. The summary statistics of all datasets (minus the target shape dataset) are identical . Green-marked cells show the best results (lowest* δ *value).*