# PaCoNet: Deep Data Extraction for Parallel Coordinates

Poonam Poonam[1], Hannah Kniesel[1], Pere-Pau Vázquez[2], and Timo Ropinski[1]

Ulm University, Germany[1]
{poonam.poonam, hannah.kniesel, timo.ropinski}@uni-ulm.de
Universitat Politècnica de Catalunya, Barcelona, Spain[2]
pere.pau.vazquez@upc.edu

**Abstract.** Extracting data from visualizations has long challenged computer vision, with current research focused on bar, line, and pie charts, among other low-dimensional visualizations. However, parallel coordinates as a widely used high-dimensional data visualization approach, remain largely unexplored in this context. As parallel coordinate plots can quickly become cluttered and difficult to interpret when poorly designed or densely populated, automated data extraction from such visualizations is of particular interest. In this paper, we propose PaCoNet, the first approach for parallel coordinate data extraction. PaCoNet not only extracts line coordinates, but also enables the extraction of individual data samples for further analysis. Towards this end, we make the following contributions. We present the first deep learning approach tailored for parallel coordinate analysis, and demonstrate that it outperforms unadapted baselines by a significant margin. We further introduce a large-scale parallel coordinate dataset for training and testing. Together, these key contributions enable for the first time the automated analysis and redesign of parallel coordinate plots. PaCoNet thus lays the groundwork for complex visualization analysis, and further advances the intersection of computer vision and data visualization. All code, trained models, and data generation scripts will be made publicly available upon acceptance of the paper.

**Keywords:** Data Extraction · Charts · Deep Learning

## 1 Introduction

Data visualization plays a critical role in understanding complex, high-dimensional datasets. Parallel coordinates, originally proposed by Inselberg in 1985 [10], visualize high-dimensional data by plotting each dimension along a separate parallel axis, whereby individual data samples are represented as polylines intersecting each axis at their corresponding values (see Figure 1). Due to their benefits, parallel coordinates are widely employed for high-dimensional data analysis in domains such as finance [1], bioinformatics [2], and engineering [12].

Unfortunately, extracting data from a parallel coordinates plot is far more challenging than from simpler visualizations, such as bar charts or line charts [36,13,24]. Major challenges are the high density of overlapping lines and the inherent clutter

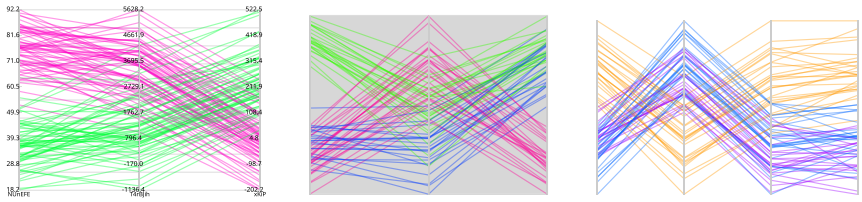

Fig. 1: Example parallel coordinate plots extracted from our introduced training dataset. Parallel coordinates simultaneously depict multiple data dimensions within a single visualization, by exploiting parallel coordinate axis, such that different parameter configurations and feature distributions can be intuitively compared and interpreted.

this creates. Furthermore, in bar or line charts, each data point or category can often be individually distinguished, but in a parallel coordinates plot, numerous data dimensions and records are condensed into a single view, causing significant overdraw that obscures precise values. This is also demonstrated by the difficulties, modern vision-language models (VLMs) [23,29] have to extract numerical values from parallel coordinates. As a result, while one may broadly infer data ranges or clustering patterns, extracting fine-grained, per-sample data currently requires direct access to the original dataset.

Building on these challenges, we present PaCoNet, the first deep learning approach specifically tailored to extracting data from parallel coordinates. Our method not only retrieves crucial information but also enables to apply more complex analysis to the extracted data. To achieve these goals, PaCoNet has been designed to support quantitative analysis by extracting the geometric structure and relative data trajectories encoded in the plots. Besides proposing PaCoNet, we further release the first large-scale parallel coordinate training dataset, which is an essential requirement for further research in this direction. We quantitatively and qualitatively analyze PaCoNet, and demonstrate that it not only is the first deep parallel coordinate analysis approach, but that it also outperforms all previous baselines. Thus, within this paper, we make the following contributions:

- We present PaCoNet as the first deep learning approach for extracting data from parallel coordinates.
- We introduce a large-scale parallel coordinate training and benchmark dataset.
- We demonstrate PaCoNet's capabilities on real-world datasets, by extracting data and showcasing chart redesign, including axis reordering and recoloring.

## 2 Related Work

*Parallel Coordinates.* Parallel coordinates, introduced by Inselberg [10], are a foundational visualization technique for high-dimensional data, mapping multidimensional samples to polylines intersecting parallel axes. Extensive prior work

has focused on improving their visual interpretability through rendering strategies and interactive techniques [8]. However, these efforts primarily target human interaction and visualization quality [37,30] and highlight the active evolution of parallel coordinates visualization [15,31], rather than automated data extraction from rendered plots.

*Deep Chart Analysis.* Automated chart analysis has attracted significant attention, with early systems such as ReVision [28] focusing on chart classification and structural understanding. More recent deep learning approaches target precise numerical extraction, including methods for bar charts [36] and line charts [13,24], as well as comprehensive pipelines such as ChartOCR [18] that integrate detection and OCR. Despite significant progress across bar, line, and pie charts [3,11,22,16], parallel coordinates plots have not been previously addressed by deep learning approaches. Broader chart-mining pipelines, such as those developed in the ChartInfo and ChartQA benchmarks [4,19], typically focus on recovering semantic information by parsing textual elements, axis scales, and legends. In contrast, PaCoNet targets the geometric extraction of dense polyline structures in parallel coordinates plots, which pose unique challenges due to heavy overdraw and frequent crossings and are not explicitly addressed in these benchmarks.

*Deep Line Detection.* Line detection is a fundamental computer vision task that has advanced significantly with deep learning. Beyond classical approaches such as the Hough Transform [9], modern methods incorporate global reasoning through learnable Hough-like formulations [7,14], transformer-based architectures [33], and holistic wireframe parsing [34]. General-purpose line and segment detectors such as HAWP, LETR, and classical Hough-based approaches [33,9,7] are primarily designed to detect sparse, isolated line structures in natural images. In contrast, parallel coordinates plots contain extremely dense, overlapping, and highly intersecting polylines, where these methods tend to produce fragmented or noisy detections that are difficult to associate into coherent data trajectories.

Among existing approaches, DHLP [14] is well suited for extracting lines from parallel coordinate plots due to their structural regularity. Such plots consist of densely overlapping, predominantly straight line segments that often span the full vertical extent of the image. DHLP incorporates global geometric priors and aggregates evidence across entire line extents in a Hough-transform–inspired manner, making it robust to dense crossings, anti-aliasing artifacts, and low-contrast regions. As a result, DHLP provides structured line representations that can be naturally adapted to recover line coordinates in this domain.

## 3 Dataset Construction

Training models for automatic data extraction from parallel coordinates plots requires large amounts of annotated data, which are not publicly available. To address this limitation, we construct a large-scale synthetic dataset whose design is informed by statistical properties observed in real-world parallel coordinates

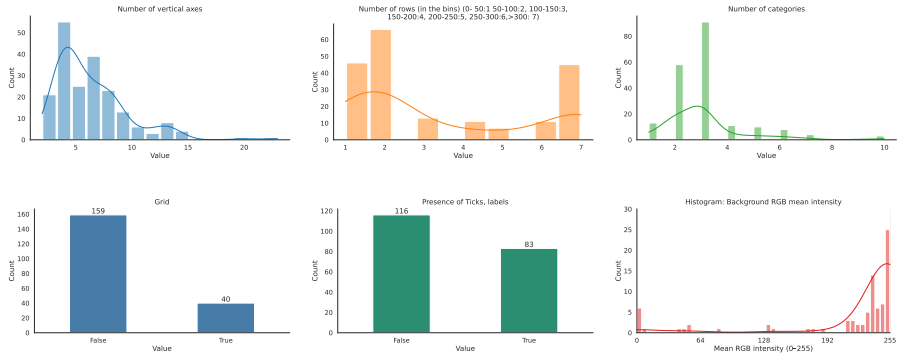


Fig. 2: Distribution of key visual and structural properties in the real-world chart dataset. Top row (left to right) shows the distributions of the number of vertical axes, the number of rows (binned), and the number of categories. Bottom row shows the distribution of grid presence, tick presence, label presence, and the background RGB mean intensity.

plots. The later we have carefully collected and curated to form a real-world parallel coordinate data set. We first summarize this real-world dataset used for statistical analysis, and then describe the synthetic data generation pipeline guided by these statistics.

### 3.1 Real-World Data

To obtain representative real-world parallel coordinates plots, we collect and curate a set of approximately 200 images from publicly available sources, ensuring that the dataset reflects realistic and interpretable examples encountered in practice. We analyze the real-world dataset to estimate key visual and structural properties of parallel coordinates plots, including the number of axes, data density, categorical color usage, grid and label presence, and background appearance. The resulting distributions, summarized in Figure 2, are used to guide the parameter ranges and design decisions in our synthetic data generation process. Detailed dataset collection procedures and statistics extraction protocols are provided in the supplementary material.

### 3.2 Synthetic Data Generation

Due to its limited size and lack of ground-truth annotations, the real-world dataset cannot be used directly for supervised learning. We therefore generate a large synthetic dataset of parallel coordinates plots with exact ground-truth annotations. The design of the synthetic data generator is informed by the statistical analysis of the real-world dataset ( Figure 2). Rather than generating plots with arbitrary parameters, we sample key properties from distributions observed in real-world visualizations, ensuring realistic variation while retaining

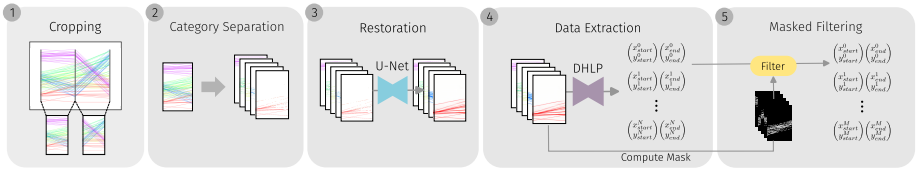


Fig. 3: Overview of the proposed PaCoNet framework for data extraction from parallel coordinates plots. The pipeline consists of five stages: (1) **Cropping**, where the input plot is partitioned into regions between adjacent vertical axes; (2) **Category Separation**, which separates individual line categories to mitigate overdraw; (3) **Restoration**, using a U-Net-based network to enhance line continuity and suppress artifacts; (4) **Data Extraction**, where a deep neural network (DHLP) [14] predicts polyline coordinates; and (5) **Masked Filtering**, which applies mask-based filtering to refine the final set of extracted coordinates.

full control and precise annotations. For each synthetic plot, we sample the number of axes, data rows, and categories from empirical distributions estimated from the real-world dataset, ensuring coverage of both sparse and densely populated plots. In addition, we model key visual attributes affecting extraction performance, including grid lines, ticks, labels, background brightness, and line colors, by sampling from observed real-world distributions.

*Data Generation and Annotation.* Using the sampled parameters, we generate parallel coordinates plots programmatically using the Vega-Lite visualization grammar [27] and render them as raster images a fixed resolution of 600 $\times$ 300 pixels. Since the underlying data values and plot layout are known by construction, we obtain exact ground-truth annotations for each plot, including axis locations, data values, and polyline correspondences. This process yields a large, diverse, and fully annotated synthetic dataset. We use 5,000 images for training and validation (80%/20% split) and an additional 1,000 images for testing. Additional implementation details of the synthetic data generator are provided in the supplementary material.

## 4 Method

Our pipeline for data extraction from parallel coordinates plots is designed to address challenges such as visual clutter, line overdraw, and multi-category representations. The method comprises a sequence of modular steps that progressively reduce ambiguity and improve line separability. First, the parallel coordinates plot is cropped at the vertical axes, next we separate individual categories by their color. Since this separation may introduce artifacts, we subsequently apply a line restoration step. The coordinates are then extracted using a combination of learning-based and algorithmic components. Finally, a postprocessing stage

suppresses falsely identified lines. An overview of the workflow is shown in Figure 3. Throughout this section, we refer to a category as a group of polylines sharing the same visual color encoding.

### 4.1 Cropping

To reduce task complexity, we detect vertical axes and crop the plot into regions between adjacent axes, each corresponding to the space where polylines transition between dimensions ( Figure 3(1)). For synthetic data, axis positions are obtained directly from the generation parameters, while for real-world images vertical axes are detected using a lightweight geometric procedure based on vertical line detection and projection-profile analysis. This allows subsequent processing stages to operate on localized line segments with no interference from unrelated axes. Although extraction is performed on crops between adjacent axes, polyline identities are preserved globally by associating corresponding segments across successive axis pairs based on axis order and spatial continuity.

### 4.2 Category Separation

Parallel coordinates plots often encode multiple categories using color, which exacerbates visual clutter and overdraw as the number of categories increases. To reduce this complexity, we introduce an explicit category separation step ( Figure 3(2)). As preprocessing, we remove non-essential visual elements such as axis labels, ticks, and grid annotations, retaining only the colored polylines. Since category information is not explicitly encoded, we infer category structure from color distributions using two strategies: **peak-based separation** and **clustering-based separation**. Both operate without prior knowledge of the number of categories and are robust to low color contrast. This formulation assumes categories are encoded using discrete, stable color hues, reflecting common practice in categorical parallel coordinates plots; grayscale, monochrome, or continuously varying colormaps are outside the scope of this work.

*Peak-Based Separation.* To estimate the number of categories $N$, we compute a one-dimensional histogram over all hue values in the image. Let $h_i \in [0, 1]$ denote the hue value of pixel $i$, and let

$$H(k) = \sum_i 1\big(h_i \in [b_k, b_{k+1})\big)$$

be the histogram count of bin $k$ with bin boundaries $\{b_k\}$. Then, we apply peak detection to identify a set of local maxima $\{p_1, \ldots, p_N\}$, where each peak corresponds to a dominant color mode in the plot and thus defines one category. Given the detected peaks, each pixel is assigned to the nearest peak in hue space,

$$c_i = \arg \min_{j \in \{1,\ldots,N\}} |h_i - p_j|,$$

thereby grouping pixels with similar colors into a common category. Using this assignment, the original parallel coordinates plot is decomposed into $N$ category-specific images by retaining only pixels belonging to a given category and suppressing all others. As a result, we obtain $N$ cropped images, one for each inferred category.

*Clustering-Based Separation.* In addition to peak detection, we investigate clustering as an alternative mechanism for category separation that directly assigns pixels or line fragments to category groups. As before, clustering is performed on the hue component in HSV color space. Since the number of categories present in a parallel coordinates plot is unknown a priori, density-based clustering methods are particularly well suited to this task. Specifically, we employ DBSCAN [5] and HDBSCAN [21], which identify clusters as dense regions in feature space without requiring the number of clusters to be specified in advance. Given hue values $\{h_i\}$, DBSCAN groups samples based on a neighborhood radius $\varepsilon$ and a minimum number of samples $m$. A pixel $i$ is considered a core point if

$$|\{\, j \mid |h_i - h_j| \leq \varepsilon \,\}| \geq m,$$

and clusters are formed by connecting density-reachable core points. Points that are not assigned to any cluster are treated as noise. The resulting clusters correspond to the inferred categories and are used to decompose the original plot into category-specific images. These strategies define two variants of our method, $\text{PaCoNet}_{Peak}$ and $\text{PaCoNet}_{DBScan}$.

### 4.3 Restoration

Due to visual clutter and strong overdraw, particularly in parallel coordinate plots with many categories, the previous category separation step can introduce artifacts, resulting in noisy lines (see Figure 4, first column) and complicating subsequent analysis. These imperfections primarily arise from cluster and peak-based separation methods, where overlapping lines, anti-aliasing effects, and compression artifacts can produce broken polylines, spurious gaps, color bleeding between categories, and isolated noise pixels.

To address these issues, we integrate a restoration step using a UNet-based neural network [26], as illustrated in Figure 3(3). Let $I_s$ denote the separated, artifact-prone image for a given category, and let $I_c$ denote the corresponding clean image derived from our synthetic ground truth. The UNet is trained to learn a mapping

$$f_\theta : I_s \mapsto I_c,$$

where $\theta$ represents the network parameters. The training objective is to minimize a pixel-wise mean squared error (MSE) loss:

$$\mathcal{L}(\theta) = \frac{1}{|P|} \sum_{p \in P} \| f_\theta(I_s)_p - I_{c,p} \|_2^2,$$

where $P$ denotes the set of pixels and $\|\cdot\|_2$ denotes the $\ell_2$ norm. After training, the U-Net restores noisy lines and corrects visual artifacts, producing clean, category-specific plots suitable for downstream analysis.

### 4.4 Data Extraction

With the now category-separated and restored image segments, we proceed to detect the individual data lines and extract data using DHLP [14]. DHLP is particularly well suited for parallel coordinates plots, as discussed in Section 2, because it embeds global geometric priors that allow robust detection of densely overlapping, mostly straight lines, even in the presence of low-contrast regions. Global polylines are reconstructed by linking extracted line segments across adjacent axis crops in sequence, yielding a single continuous polyline per category that spans all axes. The network is built upon a fully convolutional architecture, and to ensure consistency and reproducibility, we follow the same network architecture and training setup as described in [14]. This step ( Figure 3(4)) is pivotal for accurately identifying and extracting the lines, which is crucial for reconstructing the underlying data samples from parallel coordinate charts. The resulting data representation captures relative positions and trends across axes in image coordinates, without requiring explicit parsing of axis ticks, labels, or numeric scales.

### 4.5 Masked Filtering

Finally, we refine the predicted lines using our proposed masked filtering ( Figure 3(5)). After restoration with the UNet, we obtain a set of cleaned, category-specific images, denoted as $\{I_c^{(k)}\}_{k=1}^{N}$, where $N$ is the number of categories. Each restored image is then binarized to generate a mask highlighting the detected lines:

$$M^{(k)}(x,y) = \begin{cases} 1, & \text{if } I_c^{(k)}(x,y) \geq \tau, \\ 0, & \text{otherwise,} \end{cases}$$

where $I_c^{(k)}(x,y)$ is the pixel intensity at position $(x,y)$ in category $k$ and $\tau$ is a binarization threshold.

Let $\mathcal{L} = \{L_1, L_2, \ldots, L_m\}$ denote the set of lines detected by DHLP in the restored images. Each line $L_i$ is represented as a sequence of pixel coordinates $\{(x_{i,j}, y_{i,j})\}_{j=1}^{n_i}$. To enforce spatial consistency with the restored mask, we retain only those lines for which a sufficient fraction of pixels lies within the mask:

$$\hat{\mathcal{L}} = \left\{ L_i \in \mathcal{L} \;\middle|\; \frac{1}{n_i} \sum_{j=1}^{n_i} M^{(k)}(x_{i,j}, y_{i,j}) \geq \alpha \right\},$$

where $\alpha \in [0,1]$ is a threshold specifying the minimum fraction of pixels that must lie inside the mask for the line to be retained. Lines not satisfying this criterion are removed.

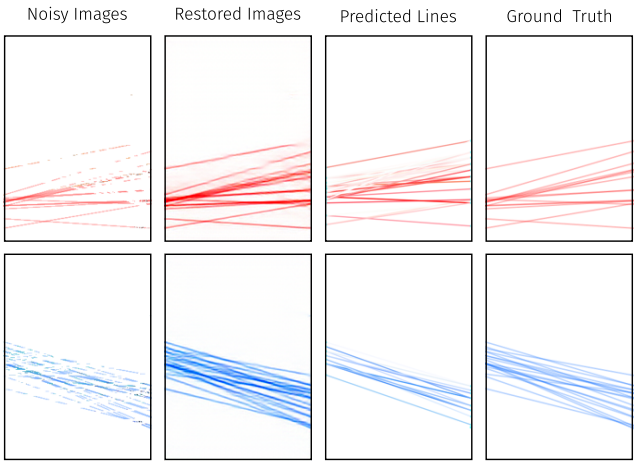


Fig. 4: Qualitative results on the synthetic test set before chart reconstruction. From left to right: noisy input images, restored images, predicted line structures, and ground truth. The results highlight the effectiveness of restoration and line prediction.

This masked filtering step ensures that only lines corresponding to valid, restored regions are kept, effectively suppressing spurious detections outside the category-specific regions. The output $\hat{\mathcal{L}}$ forms the final set of lines used for reconstruction and analysis.

## 5 Experiments

We evaluate PaCoNet on both synthetic and real-world datasets. Since no prior method addresses end-to-end data extraction from parallel coordinates plots, our experiments validate feasibility, analyze key design choices, and assess robustness under varying visual conditions. Quantitative evaluation is restricted to synthetic data, where precise ground truth is available, while real-world results are assessed qualitatively.

### 5.1 Experimental Setup

*Evaluation Protocol.* All models are trained exclusively on synthetically generated parallel coordinate plots with complete ground-truth annotations. Training is performed on NVIDIA A6000 and A100 GPUs, with each model requiring approximately 10–12 hours. We adopt a standard U-Net [26] with established training hyperparameters for the restoration network and retain the hyperparameter configuration as in the original DHLP paper [14]. We evaluate PaCoNet quantitatively and qualitatively on the synthetic test set, which reflects real-world plot distributions (see Section 3), and additionally report qualitative results on real-world examples where ground-truth annotations are unavailable.

Table 1: Quantitative evaluation of PaCoNet on the synthetic test set. **(a)** LC-MAE comparison against VLMs and the DHLP baseline. **(b)** sAP comparison against DHLP evaluating line detection. Best results are highlighted in bold.

(a) LC-MAE ↓

| Method | LC-MAE |
|---|---|
| GPT4.0 [23] | 0.80 |
| GPT5.2 [23] | 0.61 |
| Gemini2.5-flash [29] | 0.71 |
| Gemini3.0-flash [29] | 0.68 |
| DHLP [14] | 0.47 |
| PaCoNet$_{Peak}$ | 0.38 |
| PaCoNet$_{DBScan}$ | **0.37** |

(b) sAP ↑

| Method | No Masked Filtering | | Masked Filtering | |
|---|---|---|---|---|
| | sAP$^5$ | sAP$^{10}$ | sAP$^5$ | sAP$^{10}$ |
| DHLP [14] | 28.00 | 35.89 | 40.56 | 46.55 |
| PaCoNet$_{Peak}$ | 40.48 | 44.24 | 61.66 | **68.39** |
| PaCoNet$_{DBScan}$ | 39.82 | 43.38 | 56.10 | 61.80 |

*Evaluation Metrics.* We use two complementary metrics. We use Mean Absolute Error [32] to quantify the difference between the number of predicted and ground-truth lines. As we similarly use MAE to quantify multiple types of error, we refer to this metric as "LC-MAE". We additionally report structural Average Precision (sAP) [14], which measures detection accuracy under geometric tolerance thresholds. We report sAP at AP$^5$ and AP$^{10}$.

*Category Separation Metrics.* To evaluate category separation design choices in ablation studies, we additionally report the Silhouette Coefficient (SC) [17] to assess color cluster separability, category-count MAE (CC-MAE) [32] to measure errors in predicted category counts, and Intersection-over-Union (IoU) [25] to quantify spatial agreement between predicted and ground-truth category masks when pixel-level assignments are available. Additional details are provided in the supplementary material.

### 5.2 Main Results

Our results in Table 1 demonstrate better performance compared to the baseline of VLMs and DHLP respectively. We compare PaCoNet against both general-purpose and task-specific baselines. As general-purpose baselines, we evaluate VLMs, including OpenAI models [23] and Gemini [29], which reflect current user practice for chart understanding. Although these models do not natively output explicit line-level predictions, we prompt them to regress line coordinates on cropped regions enabling a quantitative comparison using LC-MAE. As a task-specific baseline, we compare against DHLP [14], which is applied independently to each cropped region and evaluated using sAP.

### 5.3 Ablations

To assess the contribution of each component, we conduct ablation studies evaluating category separation, UNet-based restoration, and post-processing on line detection performance.

Table 2: Ablation studies on category separation: (a) Comparison of color spaces for category separation using SC. (b) Influence of input image resolution on cluster-based separation accuracy. (c) Comparison of density-based clustering algorithms. (d) Influence of input image resolution on peak-based category separation.

**(a)**

| Color Space | SC ↑ |
|---|---|
| RGB | 0.58 |
| LAB | 0.61 |
| $HSV_H$ | **0.84** |
| $HSV_{Full}$ | 0.60 |

**(b)**

| Resolution | CC-MAE ↓ | IoU ↑ |
|---|---|---|
| Full | **0.077** | 0.33 |
| Downscaled | 3.14 | **0.34** |

**(c)**

| Resolution | CC-MAE ↓ | IoU ↑ |
|---|---|---|
| Full | 4.27 | 0.35 |
| Downscaled | **1.65** | **0.36** |

**(d)**

| Method | CC-MAE ↓ | IoU ↑ |
|---|---|---|
| DBSCAN [5] | **0.81** | **0.35** |
| HDBSCAN [21] | 0.89 | **0.35** |

*Category Separation.* We analyze pixel color distributions within each cropped region and compare RGB, LAB, and HSV color spaces for their ability to separate categorical color modes. Table 2(a) shows the benefits of separating the clusters using the hue component of the HSV color space. We further analyze the effect of input resolution and observe that clustering-based separation benefits from downscaled images, while peak-based separation performs best at full resolution ( Table 2(b,c)). We additionally compare DBSCAN [5] and HDBSCAN [21], finding that DBSCAN consistently yields better performance in our setting. This aligns with the one-dimensional hue feature space, where category colors form relatively uniform, well-separated clusters ( Table 2(d)).

*Image Restoration.* In this ablation study, we examine the importance of image restoration. Specifically, we compare DHLP [14] trained on noisy images (see Figure 4, first column), due to the category separation, to DHLP [14] trained on the restored (see Figure 4, second column) images. Note that, we train separate UNet models for each category separation method: one for **peak-based separation** and one for **clustering-based separation**. In each case, the input $I_s$ is the image resulting from the respective separation method, and the target $I_c$ is its corresponding clean counterpart generated from synthetic ground truth images. The results, presented in Table 3, demonstrate that training on denoised images enhances performance. The structured patterns in the images become more distinct after the denoising step, leading to more accurate line detections. For qualitative results please see Figure 4.

*Masked Filtering.* We also assess the effectiveness of our masked filtering approach, as outlined in Subsection 4.5. By applying this post-processing step, our evaluation yielded the best results, as demonstrated in Table 3. These findings highlight the significance and efficiency of our masked filtering strategy.

These ablation studies confirm that our proposed approach, which explicitly reduces clutter and overdraw prior to analysis by combining category-wise

Table 3: Ablation study assessing the impact of key components in our pipeline, including category separation, line restoration, and post-processing. $sAP^5$ and $sAP^{10}$ are reported.

| Category Separation | Restoration | No Masked Filtering | | Masked Filtering | |
|---|---|---|---|---|---|
| | | $sAP^5 \uparrow$ | $sAP^{10} \uparrow$ | $sAP^5 \uparrow$ | $sAP^{10} \uparrow$ |
| Peaks | ✗ | 33.49 | 38.51 | 36.85 | 41.10 |
| Cluster | ✗ | 32.44 | 37.48 | 37.41 | 42.81 |
| Peaks | ✓ | 36.54 | 41.17 | 42.23 | **48.67** |
| Cluster | ✓ | 36.98 | 41.10 | **42.89** | 47.58 |

separation with UNet-based restoration, significantly improves line detection performance in parallel coordinate plots.

## 6 Use Cases

Beyond quantitative evaluation, PaCoNet enables post hoc visualization operations that are infeasible when working solely with raster images by exposing explicit datapoints. *Chart reconstruction:* using extracted line coordinates and category assignments, parallel coordinates plots are re-rendered from predicted datapoints, producing editable visualizations that preserve consistent polyline identity across all axes. *Category recoloring:* recovered category memberships allow reconstructed plots to be recolored independently of the original encoding, enabling flexible palette selection and improved visual accessibility. *Axis reordering:* explicit datapoints further enable axis reordering by re-rendering polylines under alternative axis arrangements, facilitating the exploration of different variable relationships ( Figure 5).

Together, these examples demonstrate how PaCoNet transforms static parallel coordinates images into editable, data-driven representations. Further details are provided in the supplementary material.

## 7 Limitations

PaCoNet is designed for standard parallel coordinates plots composed of straight polylines. More exotic variants, such as curved, surface-based, or heavily augmented parallel coordinates [6,20,35], are outside the scope of this work and are not explicitly supported. While we demonstrate qualitative generalization on real-world plots, the absence of quantitative evaluation on annotated real images remains a limitation, and constructing small-scale or weakly supervised benchmarks for real-world parallel coordinates is an important direction for future work. The current axis detection pipeline assumes approximately linear, vertically aligned axes with regular spacing; plots with non-linear, inverted, or highly irregular layouts are not explicitly handled and may degrade performance.

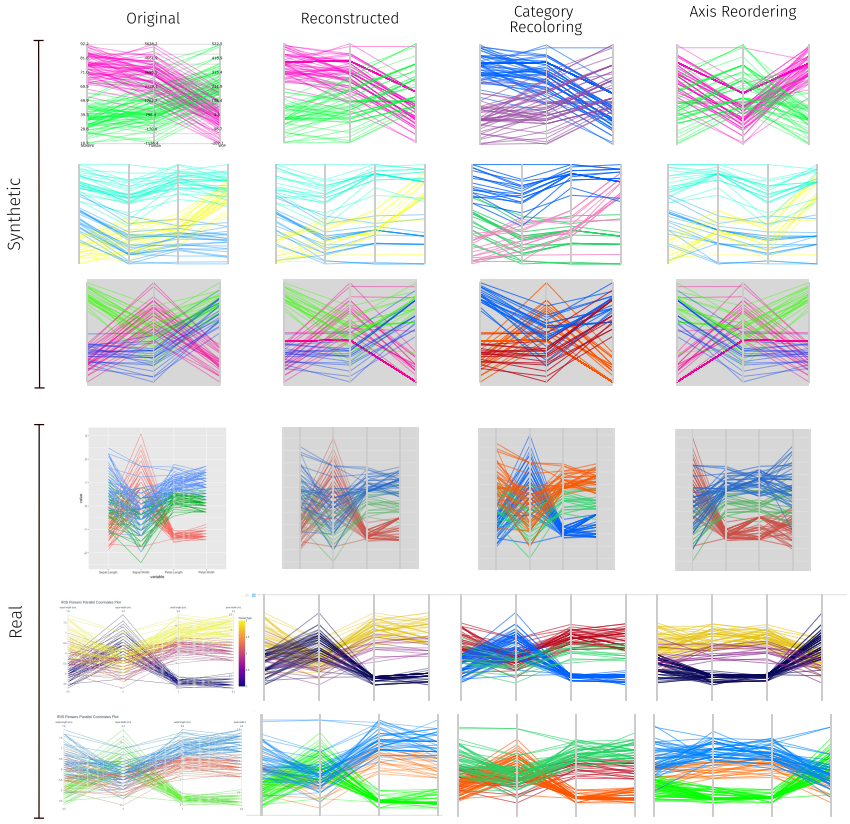


Fig. 5: Applications enabled by PaCoNet using extracted datapoints. From left to right: original plots, reconstructions from predicted data, category recoloring, and axis reordering. Top three rows show synthetic data from our test dataset, bottom three rows show real parallel coordinate plots.

PaCoNet does not recover semantic numeric values such as axis scales, tick labels, or textual annotations; instead, it serves as a structural preprocessing stage complementary to existing chart-mining pipelines, with OCR-based scale parsing left for future work.

## 8 Conclusion

In this paper, we introduced PaCoNet, the first deep learning approach specifically designed for automated data extraction from parallel coordinates. We identify line extraction task grounded in literature, providing clear metrics for quantitative evaluation. Our newly created large-scale dataset, composed of synthetic and real-

world parallel coordinate plots, enabled comprehensive training and evaluation, resulting in PaCoNet significantly outperforming baseline methods. This work not only advances automated data extraction from complex visualizations but also bridges the gap between computer vision and visualization research, paving the way for future investigations into advanced visualization analytics.

## References


1. Alsakran, J., Zhao, Y., Zhao, X.: Tile-based parallel coordinates and its application in financial visualization. In: Visualization and Data Analysis 2010
2. Boogaerts, T., Tranchevent, L.C., Pavlopoulos, G.A., Aerts, J., Vandewalle, J.: Visualizing high dimensional datasets using parallel coordinates: Application to gene prioritization. In: 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE)
3. Cui, Z., Chen, L., Wang, Y., Haehn, D., Wang, Y., Pfister, H.: Generalization of cnns on relational reasoning with bar charts. IEEE Transactions on Visualization and Computer Graphics (2024)
4. Davila, K., Lazarus, R., Xu, F., Rodríguez Alcántara, N., Setlur, S., Govindaraju, V., Mondal, A., Jawahar, C.: Chart-info 2024: A dataset for chart analysis and recognition. In: International Conference on Pattern Recognition (2024)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of KDD-96. AAAI Press (1996)
6. Graham, M., Kennedy, J.: Using curves to enhance parallel coordinate visualisations. In: Proceedings on Seventh International Conference on Information Visualization, 2003
7. Han, Q., Zhao, K., Xu, J., Cheng, M.M.: Deep hough transform for semantic line detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020), https://api.semanticscholar.org/CorpusID:212644678
8. Heinrich, J., Weiskopf, D.: State of the art of parallel coordinates. In: 34th Annual Conference of the European Association for Computer Graphics, Eurographics 2013 - State of the Art Reports. Eurographics Association, https://doi.org/10.2312/conf/EG2013/stars/095-116
9. Hough, P.V.: Method and means for recognizing complex patterns (Dec 18 1962), uS Patent 3,069,654
10. Inselberg, A.: The plane with parallel coordinates. The Visual Computer (1985)
11. Kato, H., Nakazawa, M., Yang, H.K., Chen, M., Stenger, B.: Parsing line chart images using linear programming. In: Proceedings of WACV (2022)
12. Kipouros, T., Inselberg, A., Parks, G., Savill, A.M.: Parallel coordinates in computational engineering design. In: 54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference (2013)
13. Lal, J., Mitkari, A., Bhosale, M., Doermann, D.S.: LineFormer: Rethinking line chart data extraction as instance segmentation. arXiv:2305.01837 (2023)
14. Lin, Y., Pintea, S.L., van Gemert, J.C.: Deep hough-transform line priors. In: Proceedings ECCV (2020)
15. Lind, M., Johansson, J., Cooper, M.: Many-to-many relational parallel coordinates displays. In: 2009 13th International Conference Information Visualisation
16. Liu, X., Klabjan, D., Bless, P.N.: Data extraction from charts via single deep neural network. arXiv:1906.11906 (2019)

17. Lovmar, L., Ahlford, A., Jonsson, M., Syvänen, A.C.: Silhouette scores for assessment of snp genotype clusters. BMC genomics (1) (2005)
18. Luo, J., Li, Z., Wang, J., Lin, C.Y.: ChartOCR: Data extraction from chart images via a deep hybrid framework. In: Proceedings of WACV (2021)
19. Masry, A., Do, X.L., Tan, J.Q., Joty, S., Hoque, E.: Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In: Findings of the association for computational linguistics: ACL 2022
20. McDonnell, K.T., Mueller, K.: Illustrative parallel coordinates. In: Computer Graphics Forum. Wiley Online Library (2008)
21. McInnes, L., Healy, J., Astels, S., et al.: hdbscan: Hierarchical density based clustering. J. Open Source Softw. (11) (2017)
22. Mustafa, O., Ali, M.K., Moetesum, M., Siddiqi, I.: Charteye: A deep learning framework for chart information extraction. In: 2023 International Conference on Digital Image Computing: Techniques and Applications (DICTA)
23. OpenAI: Chatgpt (2025), march 7 version, retrieved from `https://openai.com`
24. P., S.V., Hassan, M.Y., Singh, M.: LineEX: Data extraction from scientific line charts. In: Proceedings of WACV (2023)
25. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S.: Generalized intersection over union: A metric and a loss for bounding box regression. In: Proceedings of CVPR (June 2019)
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. ArXiv (2015), `https://api.semanticscholar.org/CorpusID:3719281`
27. Satyanarayan, A., Moritz, D., Wongsuphasawat, K., Heer, J.: Vega-lite: A grammar of interactive graphics. IEEE TVCG (2016)
28. Savva, M., Kong, N., Chhajta, A., Fei-Fei, L., Agrawala, M., Heer, J.: Revision: Automated classification, analysis and redesign of chart images. In: Proceedings of the ACM Symposium on User Interface Software and Technology (UIST) (2011)
29. Team, G., Anil, R., Borgeaud, S., Alayrac, J.B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A.M., Hauth, A., Millican, K., et al.: Gemini: a family of highly capable multimodal models. arXiv preprint arXiv:2312.11805 (2023)
30. Tyagi, A.K., Estro, T., Kuenning, G., Zadok, E., Mueller, K.: PC-Expo: A metrics-based interactive axes reordering method for parallel coordinate displays. IEEE TVCG (2023)
31. Wilks, A.R.: The new S language: a programming environment for data analysis and graphics. Chapman and Hall/CRC (2018)
32. Willmott, C.J., Matsuura, K.: Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. Climate research (2005)
33. Xu, Y., Xu, W., Cheung, D., Tu, Z.: Line segment detection using transformers without edges. In: Proceedings of CVPR (2021)
34. Xue, N., Wu, T., Bai, S., Wang, F.D., Xia, G.S., Zhang, L., Torr, P.H.S.: Holistically-attracted wireframe parsing. In: Proceedings of CVPR (2020)
35. Yuan, X., Guo, P., Xiao, H., Zhou, H., Qu, H.: Scattering points in parallel coordinates. IEEE Transactions on Visualization and Computer Graphics (2009)
36. Zhou, F., Zhao, Y., Chen, W., Tan, Y., Xu, Y., Chen, Y., Liu, C., Zhao, Y.: Reverse-engineering bar charts using neural networks. Journal of Visualization (2021)
37. Zhou, H., Yuan, X., Qu, H., Cui, W., Chen, B.: Visual clustering in parallel coordinates. In: Computer Graphics Forum (Proc. EuroVis). No. 3 (2008)